



Cyber Security (CYBER); CRA; Cybersecurity requirements for Virtualisation Execution Stack (VES) and Container Execution Stack (CES), including hypervisors and container runtime systems

Exercising one of the Principles of international standardization - Openness - and taking it to a different level, ETSI is piloting informal public consultations of the vertical standards in support of the Cyber Resilience Act at a much earlier stage than it is usual in the standardization world. In this context, please keep in mind that the standard draft herein is an INTERIM DRAFT, which expectably will be subject to substantial changes before its target publication date in the second semester of 2026.

Disclaimer

This **INTERIM DRAFT** document is provided for information and is for future development work within the ETSI Technical Committee CYBER EUSR Working Group (CYBER-EUSR) only. ETSI and its Members accept no liability for any further use/implementation of this Specification. Approved and published specifications and reports shall be obtained exclusively via the ETSI Documentation Service at <http://www.etsi.org/standards-search>

Commenting guidelines

Your comments to improve this early draft are very welcomed. To ensure the effectiveness of your contribution, please make sure to include the following elements in your comment:

1. Clear identification of the section of the draft your comment is referring to.
2. Objective proposal to delete content, add content or substitute content.
3. Concrete contribution (exact content you suggest including in the draft as an addition to, or in substitution of, existing content).
4. Brief rationale to justify the proposal (e.g. why the suggested content should be added an/or the existing content should be deleted or substituted). Please note that comments without concrete contributions will be noted but not acted upon by ETSI CYBER-EUSR. We trust and value your expertise and therefore expect you to take this opportunity to proactively contribute to solving the issues you find while analysing this early draft.

How to comment

To comment or provide feedback on the present interim draft please visit: <https://labs.etsi.org/rep/stan4cra/en-304-626> and submit your comments as “issues” following the guidelines provided on this site.

Alternatively, you may also send your comments to: cybersupport@etsi.org using the “[ETSI Commenting Format for Open Consultation.xlsx](https://docbox.etsi.org/CYBER/EUSR/OpenConsultation.xlsx)” available in <https://docbox.etsi.org/CYBER/EUSR/Open>

2

Reference

DEN/CYBER-EUS-0016

Keywords

CRA; Cybersecurity; Virtualization; Container

3

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

[Reproduction is only permitted for the purpose of standardization work undertaken within ETSI.](#)
[The copyright and the foregoing restrictions extend to reproduction in all media.](#)

© ETSI 2025.
All rights reserved.

Contents

4	Intellectual Property Rights	9
5	Foreword.....	9
6	Modal verbs terminology	10
7	Introduction	10
8	1 Scope	11
9	2 References	12
10	2.1 Normative references	12
11	2.2 Informative references	12
12	3 Definition of terms, symbols and abbreviations.....	13
13	3.1 Terms and Definitions	13
14	3.2 Symbols	14
15	3.3 Abbreviations.....	14
16	4 Product Context.....	16
17	4.1 Product architecture and components	16
18	4.1.1 General distinction of components.....	16
19	4.1.2 VES	16
20	4.1.2.1 Definition.....	16
21	4.1.2.2 Architectural types of VES.....	16
22	4.1.2.3 In-scope components	17
23	4.1.2.3.1 Hypervisor	17
24	4.1.2.3.2 Management and Orchestration System.....	17
25	4.1.2.4 Security-relevant environmental dependencies (out of scope)	19
26	4.1.3 CES	19
27	4.1.3.1 In-scope components	19
28	4.1.3.2 Security-relevant environmental dependencies (out of scope)	20
29	4.2 Product variants	21
30	4.2.0 Definition of conforming products.....	21
31	4.2.1 General product variants	21
32	4.2.2 Specific product variants.....	22
33	4.2.2.0 Applicability of the present document.....	22
34	4.2.2.1 Hypervisor with bundled Boot Manager and kernel/Operating System	22
35	4.2.2.2 Embedded Operating System for containerization or Virtualization	22
36	4.3 Use Cases.....	23
37	4.3.1 Purpose and Rationale.....	23
38	4.3.2 Use Cases for VES	23
39	4.3.3 Use Cases for CES	25
40	4.3.4 Use Case to Risk Mapping Table	27
41	4.4 Threats	28
42	4.4.1 Overview	28
43	4.4.2 VES	28
44	4.4.2.1 Hyper Type I.....	28
45	4.4.2.2 Hyper Type II	30
46	4.4.2.3 Orchestration and Management system	31
47	4.4.3 CES	32
48	4.4.3.1 Container Orchestrator	32
49	4.4.3.2 Container Engine	33
50	4.4.3.3 Container Runtime System	34
51	4.4.4 Hardware-level common threat	35
52	4.5 Security Objectives	35
53	4.5.1 Overview	35
54	4.5.2 Isolation.....	36
55	4.5.3 Integrity Protection.....	36
56	4.5.4 Authentication	36
57	4.5.5 Authorization.....	36

59	4.5.6	Confidentiality Protection	36
60	4.5.7	Availability and Resilience	36
61	4.5.8	Logging	36
62	4.5.9	Patches and Updates	36
63	4.5.10	Secure Configuration and Default	36
64	4.5.11	Data Minimization	37
65	4.5.12	Time Synchronization	37
66	4.6	Operational Environment Objectives	37
67	4.6.1	General Principles	37
68	4.6.2	Security Objectives for the Operational Environment with Requirement Traceability	37
69	4.6.3	Reference to Relevant CRA Harmonised Standards	39
70	4.7	Risk-Based Requirement Classes/Categorization and Application	40
71	4.7.1	Requirement Classes, Categorization and Security Category Levels	40
72	4.7.1.1	Definition of Requirement Classes: Basic, Elevated, Advanced	40
73	4.7.1.2	Security Category Level Definitions	40
74	4.7.1.3	Requirement Categorization	41
75	4.7.1.4	Guidance for Manufacturers	42
76	4.7.2	Requirement Application	42
77	4.7.2.1	Approaches for determining security requirements	42
78	4.7.2.2	SCL-Based Requirement Application	43
79	4.7.2.3	Requirement Application through Custom Security Profile Development	43
80	4.8	Roles and Responsibilities	44
81	5	Requirements	45
82	5.1	VES Security Requirements	45
83	5.1.0	General	45
84	5.1.1	Hypervisor Requirements	45
85	5.1.1.1	Isolation	45
86	5.1.1.1.0	General	45
87	5.1.1.1.1	VM Isolation	45
88	5.1.1.1.2	Control Plane Isolation	45
89	5.1.1.1.3	Network Plane Separation	46
90	5.1.1.2	Integrity Protection	46
91	5.1.1.2.1	Boot chain integrity verification	46
92	5.1.1.2.2	Guest VM image integrity verification	47
93	5.1.1.2.3	Runtime integrity protection	47
94	5.1.1.2.4	Remote attestation	48
95	5.1.1.3	Authentication	48
96	5.1.1.3.1	General	48
97	5.1.1.3.2	Administrative Authentication	48
98	5.1.1.3.3	Service Authentication	49
99	5.1.1.4	Authorization	50
100	5.1.1.4.0	General	50
101	5.1.1.4.1	Administrative Authorization	50
102	5.1.1.4.2	Service Authorization	50
103	5.1.1.5	Confidentiality Protection	50
104	5.1.1.6	Availability and Resilience	51
105	5.1.1.7	Logging	51
106	5.1.1.8	Patches and Updates	52
107	5.1.1.9	Secure Configuration and Default	52
108	5.1.1.10	Data Minimization	53
109	5.1.2	Applicability of Hypervisor Requirements	53
110	5.1.3	Management and Orchestration System Requirements	58
111	5.1.3.0	General applicability	58
112	5.1.3.1	Authentication	58
113	5.1.3.2	Authorization	58
114	5.1.3.3	Secure Configuration	58
115	5.1.3.4	Communication Security	59
116	5.1.3.5	Integrity Security	59
117	5.1.3.6	Communication Security	59
118	5.1.3.7	Patches and Updates	60
119	5.2	CES Security Requirements	60

120	5.2.0	General	60
121	5.2.1	CRS Requirements	60
122	5.2.1.1	Isolation	60
123	5.2.1.1.1	Container Isolation	60
124	5.2.1.1.2	Control Plane Isolation	61
125	5.2.1.1.3	Network Plane Isolation	62
126	5.2.1.2	Integrity Protection	62
127	5.2.1.2.1	Boot chain integrity verification	62
128	5.2.1.2.2	Container image integrity verification	63
129	5.2.1.2.3	Runtime integrity protection	64
130	5.2.1.2.4	Remote attestation	65
131	5.2.1.3	Authentication	65
132	5.2.1.3.0	General	65
133	5.2.1.3.1	Administrative Authentication	65
134	5.2.1.3.2	Service Authentication	66
135	5.2.1.4	Authorization	67
136	5.2.1.4.0	General	67
137	5.2.1.4.1	Administrative Authorization	67
138	5.2.1.4.2	Service Authorization	67
139	5.2.1.5	Confidentiality Protection	67
140	5.2.1.6	Availability and Resilience	68
141	5.2.1.7	Logging	69
142	5.2.1.8	Patches and Updates	70
143	5.2.1.9	Secure Configuration and Default	70
144	5.2.1.10	Data Minimization	71
145	5.2.2	CE Requirements	71
146	5.2.2.1	Isolation	71
147	5.2.2.2	Integrity protection	72
148	5.2.2.3	Authentication	72
149	5.2.2.4	Authorization	72
150	5.2.2.5	Confidentiality	73
151	5.2.3	CO Requirements	73
152	5.2.3.1	Isolation	73
153	5.2.3.2	Integrity protection	73
154	5.2.3.3	Authentication	74
155	5.2.3.4	Authorization	74
156	5.2.3.5	Confidentiality	74
157	5.3	Assurance Security Requirements	75
158	5.3.1	Vulnerability Management	75
159	5.3.2	Software Bill of Materials (SBOM)	75
160	5.4	Threat-to-Requirement Traceability Matrix	77
161	5.4.1	VES	77
162	5.4.1.1	Hypervisor	77
163	5.4.1.2	M&O System	77
164	5.4.2	CES	78
165	5.4.2.1	CRS	78
166	5.4.2.2	CE	78
167	5.4.2.3	CO	78
168	5.4.3	Hardware-level common threat	79
169	6	Conformance Assessment / Tests	79
170	6.1	Assessment Methodology	79
171	6.1.1	General Assessment Procedure	79
172	6.1.2	Advanced-class Equivalence and Technology-neutral Assessment	81
173	6.1.2.0	General	81
174	6.1.2.1	Equivalence of Hardware-based and Software-based Approaches	82
175	6.1.2.2	Technology-neutral Assessment Principles	82
176	6.1.2.3	Assessment of Diverse Architectural Approaches	83
177	6.2	Assessment Report Requirements	83
178	6.2.1	Mandatory Report Contents	83
179	6.2.2	Traceability Requirements	85
180	6.2.3	Evidence Package	85

181	6.3	VES.....	85
182	6.3.1	Hypervisor.....	85
183	6.3.1.1	Assessment for VM Isolation	85
184	6.3.1.1.1	Assessment Case AC-H-VM-ISO-001	85
185	6.3.1.1.2	Assessment Case AC-H-VM-ISO-003	87
186	6.3.1.2	Assessment for Control Plane Isolation.....	88
187	6.3.1.2.1	Assessment Case AC-H-CP-ISO-001	88
188	6.3.1.2.2	Assessment Case AC-H-CP-ISO-002	89
189	6.3.1.2.3	Assessment Case AC-H-CP-ISO-003	90
190	6.3.1.3	Assessment for Network Plane Separation	91
191	6.3.1.3.1	Assessment Case AC-H-NP-ISO-001.....	91
192	6.3.1.3.2	Assessment Case AC-H-NP-ISO-002.....	92
193	6.3.1.3.3	Assessment Case AC-H-NP-ISO-003.....	93
194	6.3.1.4	Assessment for Boot Chain Integrity Verification.....	95
195	6.3.1.4.1	Assessment Case AC-H-B-INT-001	95
196	6.3.1.4.2	Assessment Case AC-H-B-INT-002	96
197	6.3.1.4.3	Assessment Case AC-H-B-INT-003	97
198	6.3.1.4.4	Assessment Case AC-H-B-INT-004	99
199	6.3.1.5	Assessment for Guest VM Image Integrity Verification	100
200	6.3.1.5.1	Assessment Case AC-H-IMG-INT-001	100
201	6.3.1.5.2	Assessment Case AC-H-IMG-INT-002	101
202	6.3.1.5.3	Assessment Case AC-H-IMG-INT-003	102
203	6.3.1.5.4	6.3.1.5.3Assessment Case AC-H-IMG-INT-004	104
204	6.3.1.6	Assessment for Runtime Integrity Protection.....	105
205	6.3.1.6.1	Assessment Case AC-H-RP-INT-002	105
206	6.3.1.6.2	Assessment Case AC-H-RP-INT-003	106
207	6.3.1.6.3	Assessment Case AC-H-RP-INT-004	108
208	6.3.1.7	Assessment for Remote Attestation	109
209	6.3.1.7.1	Assessment Case AC-H-RA-INT-003	109
210	6.3.1.7.2	Assessment Case AC-H-RA-INT-004	111
211	6.3.1.8	Assessment for Administrative Authentication	113
212	6.3.1.8.1	Assessment Case AC-H-ADMIN-AUTH-001.....	113
213	6.3.1.8.2	Assessment Case AC-H-ADMIN-AUTH-002.....	114
214	6.3.1.8.3	Assessment Case AC-H-ADMIN-AUTH-003.....	115
215	6.3.1.9	Assessment for Service Authentication	116
216	6.3.1.9.1	Assessment Case AC-H-SERV-AUTH-001	116
217	6.3.1.9.2	Assessment Case AC-H-SERV-AUTH-002	117
218	6.3.1.9.3	Assessment Case AC-H-SERV-AUTH-003	118
219	6.3.1.10	Assessment for Administrative Authorization.....	119
220	6.3.1.10.1	Assessment Case AC-H-ADMIN-AUTHZ-001	119
221	6.3.1.10.2	Assessment Case AC-H-ADMIN-AUTHZ-002	120
222	6.3.1.10.3	Assessment Case AC-H-ADMIN-AUTHZ-003	121
223	6.3.1.11	Assessment for Service Authorization.....	122
224	6.3.1.11.1	Assessment Case AC-H-SERV-AUTHZ-001.....	122
225	6.3.1.11.2	Assessment Case AC-H-SERV-AUTHZ-002.....	123
226	6.3.1.11.3	Assessment Case AC-H-SERV-AUTHZ-003.....	123
227	6.3.1.12	Assessment for Confidentiality Protection	125
228	6.3.1.12.1	Assessment Case AC-H-CONF-001	125
229	6.3.1.12.2	Assessment Case AC-H-CONF-002	126
230	6.3.1.12.3	Assessment Case AC-H-CONF-003	128
231	6.3.1.12.4	Assessment Case AC-H-CONF-004	129
232	6.3.1.12.5	Assessment Case AC-H-CONF-005.....	131
233	6.3.1.13	Assessment for Availability and Resilience	132
234	6.3.1.13.1	Assessment Case AC-H-AVAIL-001	132
235	6.3.1.13.2	Assessment Case AC-H-AVAIL-002	133
236	6.3.1.13.3	Assessment Case AC-H-AVAIL-003	134
237	6.3.1.13.4	Assessment Case AC-H-AVAIL-004	135
238	6.3.1.14	Assessment for Logging	136
239	6.3.1.14.1	Assessment Case AC-H-LOG-001	136
240	6.3.1.14.2	Assessment Case AC-H-LOG-002	137
241	6.3.1.14.3	Assessment Case AC-H-LOG-003	138
242	6.3.1.15	Assessment for Patches and Updates.....	139

243	6.3.1.15.1	Assessment Case AC-H-UPD-001.....	139
244	6.3.1.15.2	Assessment Case AC-H-UPD-002.....	140
245	6.3.1.15.3	Assessment Case AC-H-UPD-003.....	141
246	6.3.1.15.4	Assessment Case AC-H-UPD-004.....	142
247	6.3.1.16	Assessment for Secure Configuration and Default.....	143
248	6.3.1.16.1	Assessment Case AC-H-CFG-001.....	143
249	6.3.1.16.2	Assessment Case AC-H-CFG-002.....	144
250	6.3.1.16.3	Assessment Case AC-H-CFG-003.....	145
251	6.3.1.17	Assessment for Data Minimization	146
252	6.3.1.17.1	Assessment Case AC-H-DM-001	146
253	6.3.1.17.2	Assessment Case AC-H-DM-002	147
254	6.3.2	M&O System	148
255	6.4	CES.....	149
256	6.4.1	CRS	149
257	6.4.1.1	Assessment for Container Isolation	149
258	6.4.1.1.1	Assessment Case AC-CRS-CN-ISO-001.....	149
259	6.4.1.1.2	Assessment Case AC-CRS-CN-ISO-002.....	150
260	6.4.1.1.3	Assessment Case AC-CRS-CN-ISO-003.....	151
261	6.4.1.2	Assessment for Control Plane Isolation.....	153
262	6.4.1.2.1	Assessment Case AC-CRS-CP-ISO-001	153
263	6.4.1.2.2	Assessment Case AC-CRS-CP-ISO-002	154
264	6.4.1.2.3	Assessment Case AC-CRS-CP-ISO-003	155
265	6.4.1.3	Assessment for Network Plane Isolation.....	156
266	6.4.1.3.1	Assessment Case AC-CRS-NP-ISO-001	156
267	6.4.1.3.2	Assessment Case AC-CRS-NP-ISO-002	158
268	6.4.1.3.3	Assessment Case AC-CRS-NP-ISO-003	159
269	6.4.1.4	Assessment for Boot Chain Integrity Verification.....	160
270	6.4.1.4.1	Assessment Case AC-CRS-B-INT-001	160
271	6.4.1.4.2	Assessment Case AC-CRS-B-INT-002	162
272	6.4.1.4.3	Assessment Case AC-CRS-B-INT-003	163
273	6.4.1.4.4	Assessment Case AC-CRS-B-INT-004	164
274	6.4.1.5	Assessment for Container Image Integrity Verification	165
275	6.4.1.5.1	Assessment Case AC-CRS-IMG-INT-001	165
276	6.4.1.5.2	Assessment Case AC-CRS-IMG-INT-002	167
277	6.4.1.5.3	Assessment Case AC-CRS-IMG-INT-003	168
278	6.4.1.5.4	Assessment Case AC-CRS-IMG-INT-004	170
279	6.4.1.6	Assessment for Runtime Integrity Protection.....	171
280	6.4.1.6.1	Assessment Case AC-CRS-RP-INT-002	171
281	6.4.1.6.2	Assessment Case AC-CRS-RP-INT-003	172
282	6.4.1.6.3	Assessment Case AC-CRS-RP-INT-004	174
283	6.4.1.7	Assessment for Remote Attestation.....	176
284	6.4.1.7.1	Assessment Case AC-CRS-RA-INT-003	176
285	6.4.1.7.2	Assessment Case AC-CRS-RA-INT-004	178
286	6.4.1.8	Assessment for Administrative Authentication	180
287	6.4.1.8.1	Assessment Case AC-CRS-ADMIN-AUTH-001	180
288	6.4.1.8.2	Assessment Case AC-CRS-ADMIN-AUTH-002	181
289	6.4.1.8.3	Assessment Case AC-CRS-ADMIN-AUTH-003	182
290	6.4.1.9	Assessment for Service Authentication	183
291	6.4.1.9.1	Assessment Case AC-CRS-SERV-AUTH-001	183
292	6.4.1.9.2	Assessment Case AC-CRS-SERV-AUTH-002	184
293	6.4.1.9.3	Assessment Case AC-CRS-SERV-AUTH-003	185
294	6.4.1.10	Assessment for Administrative Authorization.....	187
295	6.4.1.10.1	Assessment Case AC-CRS-ADMIN-AUTHZ-001.....	187
296	6.4.1.10.2	Assessment Case AC-CRS-ADMIN-AUTHZ-002.....	187
297	6.4.1.10.3	Assessment Case AC-CRS-ADMIN-AUTHZ-003.....	188
298	6.4.1.11	Assessment for Service Authorization.....	189
299	6.4.1.11.1	Assessment Case AC-CRS-SERV-AUTHZ-001	189
300	6.4.1.11.2	Assessment Case AC-CRS-SERV-AUTHZ-002	190
301	6.4.1.11.3	Assessment Case AC-CRS-SERV-AUTHZ-003	191
302	6.4.1.12	Assessment for Confidentiality Protection	192
303	6.4.1.12.1	Assessment Case AC-CRS-CONF-001	192
304	6.4.1.12.2	Assessment Case AC-CRS-CONF-002	194

305	6.4.1.12.3	Assessment Case AC-CRS-CONF-003	195
306	6.4.1.12.4	Assessment Case AC-CRS-CONF-004	197
307	6.4.1.12.5	Assessment Case AC-CRS-CONF-005	199
308	6.4.1.13	Assessment for Availability and Resilience	200
309	6.4.1.13.1	Assessment Case AC-CRS-AVAIL-001.....	200
310	6.4.1.13.2	Assessment Case AC-CRS-AVAIL-002.....	201
311	6.4.1.13.3	Assessment Case AC-CRS-AVAIL-003.....	202
312	6.4.1.13.4	Assessment Case AC-CRS-AVAIL-004.....	203
313	6.4.1.14	Assessment for Logging	204
314	6.4.1.14.1	Assessment Case AC-CRS-LOG-001.....	204
315	6.4.1.14.2	Assessment Case AC-CRS-LOG-002.....	205
316	6.4.1.14.3	Assessment Case AC-CRS-LOG-003.....	206
317	6.4.1.15	Assessment for Patches and Updates.....	207
318	6.4.1.15.1	Assessment Case AC-CRS-UPD-001	207
319	6.4.1.15.2	Assessment Case AC-CRS-UPD-002	208
320	6.4.1.15.3	Assessment Case AC-CRS-UPD-003	210
321	6.4.1.16	Assessment for Secure Configuration and Default.....	211
322	6.4.1.16.1	Assessment Case AC-CRS-CFG-001	211
323	6.4.1.16.2	Assessment Case AC-CRS-CFG-002	212
324	6.4.1.16.3	Assessment Case AC-CRS-CFG-003	213
325	6.4.1.17	Assessment for Data Minimization	214
326	6.4.1.17.1	Assessment Case AC-CRS-DM-001.....	214
327	6.4.1.17.2	Assessment Case AC-CRS-DM-002.....	215
328	6.4.2	CE.....	216
329	6.4.3	CO	216
330	Annex A (informative): Relationship between the present document and the requirements of		
331	Regulation (EU) 2024/2847		217
332	Annex B (informative): Cybersecurity Risk Assessment Methodology		221
333	B.0	Overview	221
334	B.1	Risk Assessment Process	221
335	B.2	Risk Factor Scoring.....	222
336	B.3	Calculation and Mapping to Likelihood/Impact Levels.....	225
337	B.3.0	General	225
338	B.3.1	Likelihood Score	225
339	B.3.2	Impact Score.....	225
340	B.4	Risk Matrix	226
341	B.5	Risk Methodology Applied to Use Cases	226
342	B.5.0	General	226
343	B.5.1	Risk Evaluation of VES use cases.....	226
344	B.5.1.1	Hypervisor Type I.....	226
345	B.5.1.2	Hypervisor Type II	232
346	B.5.1.3	M&O System.....	236
347	B.5.2	Risk Evaluation of CES use cases	237
348	B.5.2.1	CRS	237
349	B.5.2.2	CE.....	242
350	B.5.2.3	CO	242
351	B.5.3	Risk Evaluation of common hardware-level threat T-ALL-HW-ACCESS	242
352	Annex (informative): Bibliography.....		245
353	Annex (informative): Change history		246
354	History		249
355			
356			

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTETM** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This draft Harmonised European Standard (EN) has been produced by ETSI Technical Committee Cyber Security (CYBER) and is now submitted for the combined Public Enquiry and Vote phase of the ETSI EN Approval Procedure.

The present document has been prepared under the Commission's standardisation request C(2025)618 [i.3] to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) [i.1].

Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance with the normative clauses of the present document given in table A.1 confers, within the limits of the scope of the present document, a presumption of conformity with the corresponding requirements of that Regulation and associated EFTA regulations.

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	18 months after doa

391

Modal verbs terminology

392

393

394

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

395

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

396

Introduction

397

398

399

The present document defines cybersecurity requirements applicable to Virtualization Execution Stack (VES) and Container Execution Stack (CES), including hypervisors and container runtime systems. It applies to products with digital elements designed to host, manage, and isolate virtualized workloads and execution environments, using:

400

- Virtual machines (VMs): full system emulation with independent operating systems; or

401

- Containers: process isolation using OS-level mechanisms.

402

403

The present document supports compliance with Regulation (EU) 2024/2847, the Cyber Resilience Act, specifically addressing the essential cybersecurity requirements defined in Annex I, Parts I and II.

1 Scope

The present document is based on the definitions provided in the Cyber Resilience Act (CRA) [i.2] for hypervisors and container runtime systems as core components:

- Hypervisors are software products with digital elements that abstract and/or allocate computing resources and enable the execution and management of virtual machines that are logically separated from each other and/or from the physical hardware. Hypervisors may run directly on hardware (bare metal), on top of an operating system, or within another virtual machine (nested Virtualization). In the context of this category of products, a virtual machine is a software-defined logical separation of a computing environment, which includes a virtualized set of hardware resources (e.g. CPU, memory, storage, network interfaces) and typically hosts its own operating system. This category includes but is not limited to Type 1 hypervisors, Type 2 hypervisors, and hybrid hypervisors.
- Container runtime systems (CRS) are software products with digital elements that manage the execution and lifecycle of containers running on a single host operating system as isolated processes, allocating resources and allowing the management of individual containers. In the context of this category of products, a container is a software-based execution environment that encapsulates one or more software components and their dependencies in a single package, enabling it to run independently and consistently.

While the CRA definition identifies the core execution components, actual market products typically include additional elements beyond the hypervisor kernel or container runtime binary. These additional components provide essential management, orchestration, and operational capabilities that are necessary for real-world deployment and are therefore included within the scope of the present document.

The present document focuses on:

- Virtualization Execution Stack (VES): set of software components responsible for deploying, running, and managing virtual machines (VMs) on shared physical infrastructure. It includes:
 - A hypervisor (Type I or Type II) that abstracts hardware and ensures isolation between VMs:
 - Type I hypervisors run directly on bare-metal hardware.
 - Type II hypervisors run on top of a host operating system.
 - An orchestration and management system that automates the deployment, scaling, and coordination of VMs across the infrastructure, and provides configuration, monitoring, and lifecycle control of the VMs.
- Container Execution Stack (CES): A layered set of software components enabling the deployment, management, and execution of containerized applications. It includes:
 - Container Runtime System (CRS): The Container Runtime System is the low-level software component responsible for creating, running, and managing containers on a host OS. It interfaces directly with the underlying system kernel to provide process isolation, resource control, and namespace management, and is responsible for executing container processes (e.g. using namespaces and control groups on Linux, or equivalent mechanisms on other operating systems).
 - Container Engine (CE): A Container Engine manages the full lifecycle of containers including image pulling, creation, execution, logging, and storage. It typically interacts with the container runtime system to perform low-level execution tasks and provides APIs or CLIs to manage containers.
 - Container Orchestrator (CO): A Container Orchestrator automates the deployment, scaling, networking, availability, and lifecycle management of containerized applications across clusters of machines. It coordinates how and where containers are scheduled to run and manages communication between them.

Accordingly, the present document defines security requirements not only for the core execution systems identified in the CRA but also for the broader product context in which these systems are deployed, ensuring alignment with market reality and comprehensive coverage of security risks.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the [ETSI docbox](#).

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] [prEN 40000-1-3](#): "Cybersecurity requirements for products with digital elements - Vulnerability Handling".

[2] [ENISA-ECCG](#): "Agreed Cryptographic Mechanisms", Version 2.0, April 2025.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding but are not required for conformance to the present document.

[i.1] [Regulation \(EU\) 2024/2847](#) of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).

[i.2] Commission Implementing [Regulation \(EU\) 2025/2392](#) of 28 November 2025 on the technical description of the categories of important and critical products with digital elements pursuant to Regulation (EU) 2024/2847 of the European Parliament and of the Council.

[i.3] [C\(2025\)618 - Standardisation request M/606](#): Commission Implementing decision of 3.2.2025 on a standardisation request to the European Committee for Standardisation (CEN), the European Committee for Electrotechnical Standardisation (Cenelec) and the European Telecommunications Standards Institute (ETSI) as regards products with digital elements in support of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) .

[i.4] ETSI EN 304 623: "CYBER; CRA; Cybersecurity requirements for boot managers".

[i.5] ETSI EN 304 626: "CYBER; CRA; Cybersecurity requirements for operating systems".

[i.6] ETSI EN 304 625: "CYBER; CRA; Cybersecurity requirements for physical and virtual network interfaces".

[i.7] ETSI EN 304 624: "CYBER; CRA; Cybersecurity requirements for Public key infrastructure and digital certificate issuance software".

[i.8] ETSI EN 304 622: "CYBER; CRA; Cybersecurity requirements for Security information and event management (SIEM) systems".

491	[i.9]	ETSI EN 304 636: "CYBER; CRA; Cybersecurity requirements for firewalls, intrusion detection
492		and/or prevention systems".
493	[i.10]	prEN 40000-1-1 : "Cybersecurity requirements for products with digital elements - Vocabulary".
494	[i.11]	prEN 40000-1-2 : "Cybersecurity requirements for products with digital elements - Part 1-2:
495		Principles for cyber resilience".

496 3 Definition of terms, symbols and abbreviations

497 3.1 Terms and Definitions

498 For the purposes of the present document, the terms given in CRA Regulation [i.1], prEN 40000-1-1 [i.10] and the
499 following apply.

500 **confidential computing**: security capability that provides data-in-use protection through hardware-based isolated
501 execution

502 **container**: software-based isolated execution environment that packages an application with its dependencies and runs
503 using the host operating system kernel

504 **Container Execution Stack (CES)**: layered set of software components enabling the deployment, management, and
505 execution of containerized applications

506 **dedicated management network / interface**: A network interface or network segment that is exclusively used for
507 management plane traffic and is not shared with guest workload traffic, storage traffic, data-plane traffic, VM/container
508 traffic, or tenant-accessible networks.

509 NOTE: A dedicated management interface may be implemented as:

- 510 • a physically separate network interface assigned solely to management traffic; or
- 511 • a logically isolated network (e.g. VLAN, VXLAN, VRF) where isolation is enforced by the hypervisor /
512 CRS / host networking layer and cannot be modified or bypassed by tenant workloads.

513 **hypervisor**: software that enables the creation and management of virtual machines by abstracting, allocating and
514 controlling access to a host's hardware resources, providing isolation between virtual machines

515 **impact**: severity of harm if a threat event occurs

516 **likelihood**: probability of a threat event occurring

517 **microVM**: lightweight virtual machine designed for minimal resource overhead and rapid startup, providing an isolated
518 execution environment with a reduced attack surface compared to traditional virtual machines

519 **multi-tenancy**: architectural property of a virtualized or containerized environment in which multiple tenants share the
520 same underlying infrastructure while maintaining isolation of their workloads and data

521 **strong authentication**: authentication that provides high-assurance identity verification for access to administrative
522 interfaces, control functions or sensitive data

523 **tenant**: isolated unit of use and management within a virtualized or containerized environment, representing a single
524 user or organization sharing the underlying infrastructure

525 **type I hypervisor (Bare-Metal / Native)**: hypervisor that runs directly on physical hardware to create and manage
526 virtual machines

527 **type II hypervisor (Hosted)**: hypervisor that runs as an application on top of a general-purpose host operating system
528 to create and manage virtual machines

529 **Virtualization Execution Stack (VES)**: set of software components responsible for deploying, running, and managing
530 Virtual Machines (VMs) on shared physical infrastructure

531 **Virtual Machine (VM):** software-based virtualized computing environment that runs a guest operating system on
 532 abstracted hardware resources provided by a hypervisor

533 **Trusted Execution Environment (TEE):** isolated execution environment that protects the confidentiality and integrity
 534 of code and data against access or modification by untrusted software

535 3.2 Symbols

536 Void.

537 3.3 Abbreviations

538 For the purposes of the present document, the following abbreviations apply:

539	3DES	Triple Data Encryption Standard
540	ABAC	Attribute-Based Access Control
541	ACL	Access Control List
542	ACM	Access Control Mechanism
543	AES	Advanced Encryption Standard
544	API	Application Programming Interface
545	ASLR	Address Space Layout Redundancy
546	ASSUR	Assurance
547	BIOS	Basic Input/Output System
548	CE	Container Engine
549	CES	Container Execution Stack
550	CFG	Configuration
551	CI/CD	Continuous Integration / Continuous Delivery
552	CIS	Centre for Internet Security
553	CLI	Command Line Interface
554	CO	Container Orchestrator
555	CONF	Confidentiality
556	CRA	Cyber Resilience Act
557	CRI-O	Container Runtime Interface - Orchestrator (runtime)
558	CRS	Container Runtime System
559	CPU	Central Processing Unit
560	CTL	Control
561	DEF	Default
562	DMA	Direct Memory Access
563	DoS	Denial-of-Service
564	DRTM	Dynamic Root of Trust Measurement
565	ECDSA	Elliptic Curve Digital Signature Algorithm
566	ENISA	European Union Agency for Cybersecurity
567	ESXi	Elastic Sky X Integrated (VMware hypervisor name)
568	EUCC	EU Cybersecurity Certification Scheme on Common Criteria
569	FaaS	Function as a Service
570	GCM	Galois/Counter Mode (for encryption)
571	gRPC	gRPC Remote Procedure Call
572	HRoT	Hardware Root of Trust
573	HSM	Hardware Security Module
574	IaaS	Infrastructure as a Service
575	I/O	Input/Output
576	ID	Identification
577	IMA	Integrity Measurement Architecture
578	IMG	Image
579	IOMMU	Input/Output Memory Management Unit
580	ISO	International Organization for Standardization
581	IT	Information Technology
582	JIT	Just-In-Time
583	JSON	JavaScript Object Notation
584	JWT	JSON Web Token

585	KVM	Kernel-based Virtual Machine
586	LAN	Local Area Network
587	LOG	Logging
588	M&O	Management and Orchestration
589	MFA	Multi-Factor Authentication
590	mTLS	mutual Transport Layer Security
591	NIC	Network Interface Card
592	NTP	Network Time Protocol
593	OE	Operational Environment
594	OPA	Open Policy Agent
595	OS	Operating System
596	OSS	Open Source Software
597	PC	Personal Computer
598	PCR	Platform Configuration Register (related to TPM)
599	PDP	Policy Decision Point
600	PKI	Public Key Infrastructure
601	PTP	Precision Time Protocol
602	QEMU	Quick Emulator
603	RAM	Random Access Memory
604	RBAC	Role-Based Access Control
605	RC4	Rivest Cipher 4
606	REQ	Requirement
607	RPO	Recovery Point Objective
608	RSA	Rivest-Shamir-Adleman (Cryptographic algorithm)
609	SBOM	Software Bill of Materials
610	SCL	Security Category Level
611	SDN	Software-Defined Networking
612	SEV	Secure Encrypted Virtualization
613	SEV-ES	Secure Encrypted Virtualization-Encrypted State
614	SHA	Secure Hash Algorithm
615	SIEM	Security Information and Event Management
616	SPIFFE	Secure Production Identity Framework for Everyone
617	SR-IOV	Single Root Input/Output Virtualization
618	SSO	Single Sign-On
619	SUP	Support
620	TEE	Trusted Execution Environment
621	TLS	Transport Layer Security
622	TPM	Trusted Platform Module
623	TS	Technical Specification
624	U/C	Use Case
625	UEFI	Unified Extensible Firmware Interface
626	UI	User Interface
627	UPD	Update
628	VES	Virtualized Execution Stack
629	VLAN	Virtual Local Area Network
630	VM	Virtual Machine
631	VMM	Virtual Machine Monitor (Hypervisor Core)
632	VPN	Virtual Private Network
633	VRF	Virtual Routing and Forwarding
634	vTPM	Virtual Trusted Platform Module
635	VT	Virtualization Technology
636	VULM	Vulnerability Management
637	VXLAN	Virtual Extensible LAN
638	XACML	eXtensible Access Control Markup Language

639 4 Product Context

640 4.1 Product architecture and components

641 4.1.1 General distinction of components

642 The present document distinguishes between product components (within the scope of the present document) and
643 environmental dependencies (outside the scope of the present document but potentially security-relevant).

- 644 • In-scope components represent the functional building blocks that together define the product boundary as
645 considered in the present document. The requirements specified in the present document apply directly to these
646 components.
- 647 • Environmental dependencies, by contrast, are not part of the product itself but belong to its operational
648 environment (e.g. host operating system, firmware, or external services). These dependencies shall be secured
649 in accordance with other standards (see clause 4.6).

650 In some cases, a product placed on the market may be delivered together with one or more out-of-scope components
651 (for example, a VES bundled with a host OS). When this occurs, the manufacturer shall ensure that these bundled
652 components also conform to the CRA, by applying the relevant harmonised standards where available.

653 4.1.2 VES

654 4.1.2.1 Definition

655 The Virtualization Execution Stack (VES) represents the collection of software components that enable the creation,
656 management, and orchestration of virtualized workloads on a physical host.

657 A VES typically comprises two main components:

- 658 • The Hypervisor, which provides workload isolation, resource partitioning, and execution control.
- 659 • The Management and Orchestration (M&O) system, which enables provisioning, configuration, and lifecycle
660 management of virtual resources and hosts.

661 4.1.2.2 Architectural types of VES

662 Depending on the deployment model and integration with the host operating environment, the VES can be implemented
663 in one of the following forms:

- 664 • Type I (Bare-metal hypervisor):
665 The hypervisor runs directly on the physical hardware without a host operating system. It manages hardware
666 resources and runs guest virtual machines (VMs) directly.
- 667 • Type II (Hosted hypervisor):
668 The hypervisor executes as an application on top of a general-purpose host operating system, relying on the
669 host OS for device drivers and resource management.
- 670 • Hybrid (Mixed architecture):
671 Combines characteristics of both Type I and Type II architectures. The hypervisor runs close to the hardware
672 but leverages a general-purpose OS or service domain for device drivers, management, or orchestration
673 functions.

674 4.1.2.3 In-scope components

675 4.1.2.3.1 Hypervisor

- 676 • Hypervisor core kernel or microkernel: This is the heart of the hypervisor, responsible for core functions like
677 memory and process management.
- 678 • Virtual Machine Monitor (VMM) and guest VM lifecycle controller: The VMM is the component that
679 executes the guest operating system's instructions and manages the VM's operational state (creation, start, stop,
680 deletion).
- 681 • Resource scheduling components (CPU, memory, I/O): These modules are responsible for deciding which VM
682 gets access to which physical resource at any given time.
- 683 • Device Virtualization modules (e.g. disk controllers, SR-IOV support, paravirtualized device drivers, including
684 any driver components running inside guest operating systems that directly interact with the host hypervisor):
685 These components abstract the physical hardware and present virtual devices to the guest VMs.
- 686 • Paravirtualization drivers (when applicable): When the hypervisor employs Paravirtualization mechanisms, the
687 following components are considered in-scope regardless of execution context:
 - 688 - Backend Paravirtualization drivers and services running in the host or privileged domain.
 - 689 - Frontend Paravirtualization drivers provided by the hypervisor manufacturer for installation in guest
690 domains.
 - 691 - Inter-domain communication mechanisms used for paravirtual device operations (e.g. shared memory
692 channels, ring buffers, grant tables, virtio queues).
 - 693 - The privileged domain (e.g. Xen Dom0) only to the extent that it hosts hypervisor-critical management
694 or device backend services.

695 NOTE 1: Guest-OS-native or third-party Paravirtualization drivers not supplied by the hypervisor manufacturer
696 remain out of scope. The security boundaries and domain isolation over all inter-domain communication
697 channels are expected to be enforced by the hypervisor, irrespective of the driver's origin hypervisor-

- 698 • Virtual Trusted Platform Module (vTPM) components (when applicable): When the hypervisor provides
699 virtual TPM functionality to guest virtual machines, the following components are considered in-scope:
 - 700 - The vTPM backend service, resource manager, and associated storage for vTPM state.
 - 701 - Management and lifecycle interfaces used to provision, migrate, back up, or delete vTPM instances.
 - 702 - Isolation mechanisms ensuring that each vTPM instance is cryptographically and logically separated
703 from other guests and from the host.
 - 704 - Attestation flows relying on vTPM measurements, including any hypervisor-provided attestation agent or
705 reporting mechanism.

706 NOTE 2: Physical TPM hardware, firmware, and host platform Root of Trust remain out-of-scope.

- 707 • Logging and auditing subsystems native to the hypervisor: The systems that record events and actions directly
708 within the hypervisor layer for monitoring and security.
- 709 • Security update enforcement mechanisms: Tools or processes integrated into the hypervisor to ensure and
710 enforce security updates.

711 4.1.2.3.2 Management and Orchestration System

712 It is the administrative layer that sits above the hypervisor. It is used to deploy, configure, and manage the underlying
713 hypervisor and the entire virtualized infrastructure, often coordinating multiple hypervisors.

- 714 • Management and Orchestration components: This is the system itself, including the main logic for
715 infrastructure automation and management.

- Policy enforcement, scheduling, and automation engines used for orchestration
- Hypervisor management interface or CLI/API (local or remote): This is the external interface through which administrators or the Orchestration system interact with the Hypervisor. While it provides access to the hypervisor, it is generally considered part of the overarching M&O or management layer that enables control.

Deployment models of M&O system:

The Management and Orchestration (M&O) system may be deployed in different forms depending on the architecture of the virtualized infrastructure. Common deployment models include:

- Dedicated management host: The M&O system runs on separate physical hardware used exclusively for management and administration functions.
- Management cluster: The M&O system runs as one or more virtual machines on a dedicated cluster reserved for management and platform services.
- Virtualized M&O node: The M&O system runs as a virtual machine on the same virtualized infrastructure that it manages, placed on a host reserved for platform and management services.

Figure 4.1.1.2.2-1 and Figure 4.1.1.2.2-2 highlight in blue the components that are within the scope of the present document. Components not highlighted in blue are considered out of scope.

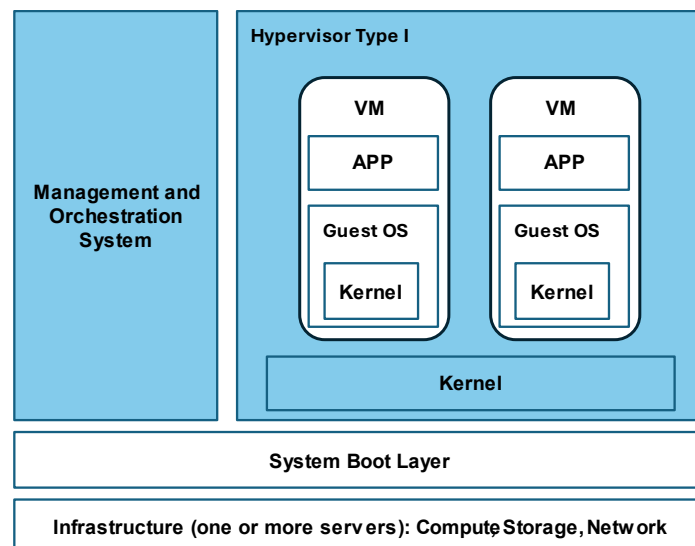


Figure 4.1.1.2.2-1: VES - Type I Hypervisor components

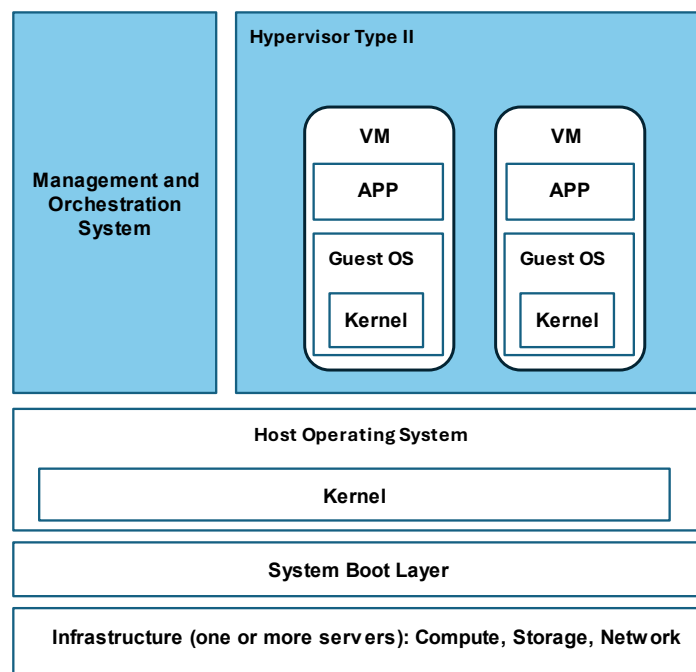


Figure 4.1.1.2.2-2: VES - Type I Hypervisor components

733

734

735 4.1.2.4 Security-relevant environmental dependencies (out of scope)

736 The following are considered outside the scope of the present document but may be security-relevant environmental
737 components:

- 738 • The Host Firmware and BIOS/UEFI.
- 739 • The Host Operating System, when it is not bundled as part of the hypervisor.
- 740 • Guest Operating Systems and virtualized application layers, with the exception of paravirtualized drivers that
741 directly interface with the hypervisor.
- 742 • Virtual network infrastructure beyond the hypervisor switch (e.g. SDN controllers).
- 743 • External logging platforms (e.g. SIEMs).
- 744 • External attestation systems.
- 745 • Hardware-level security controls (TPM, HSM) not embedded in the hypervisor or used to create a virtual
746 equivalent (vTPM).
- 747 • Identity management systems, secrets vaults, and PKI.
- 748 • Processes, tools, environments, or development/packaging formats whose primary purpose is to develop,
749 build, or package guest operating system artifacts, including but not limited to OS installation media, virtual
750 disk images, VM templates, build pipelines, compilers, and related image-building utilities.

751 4.1.3 CES

752 4.1.3.1 In-scope components

753 The following components are in the scope of the present document:

- 754 • CRS: it is the lowest-level layer responsible for the execution and isolation of containers:
 - 755 - The core binary (e.g. runc, crun).

- Enforcement modules that manage container isolation, resource control, and sandboxing through OS or application-level mechanisms (e.g. namespaces, microVMs, seccomp, cgroups).
- The API and services that manage the runtime and its containers on a host (e.g. containerd, CRI-O).
- Container Engine: it is the mid-level layer that manage the container lifecycle:
 - The engine and its interfaces for building, pulling, pushing, and running container images (e.g. Docker, Podman, containerd).
 - Management of container images including the verification a signature at the host level.
 - Management of container networking at the host level (e.g. creation of network interfaces, bridge networks, and network namespace assignment).
 - Management of volumes for persistent or shared storage.
 - Management of plugins as supported by the engine.
- Container Orchestrator: it is the top-level layer that manages the entire lifecycle of applications and services across a cluster of machines (e.g. Kubernetes®):
 - Features that ensure containers are deployed securely.
 - Image signature verification at the cluster level, secure scheduling to trusted nodes, and secrets management.
 - Policy enforcement that enforces security and isolation policies across the entire cluster (e.g. pod-level security policies and network isolation rules).
 - Lifecycle management the handles the scaling, networking, load balancing, and self-healing of containerized applications.

Figure 4.1.2.1-1 highlights in blue the components that are within the scope of the present document. Components not highlighted in blue are considered out of scope.

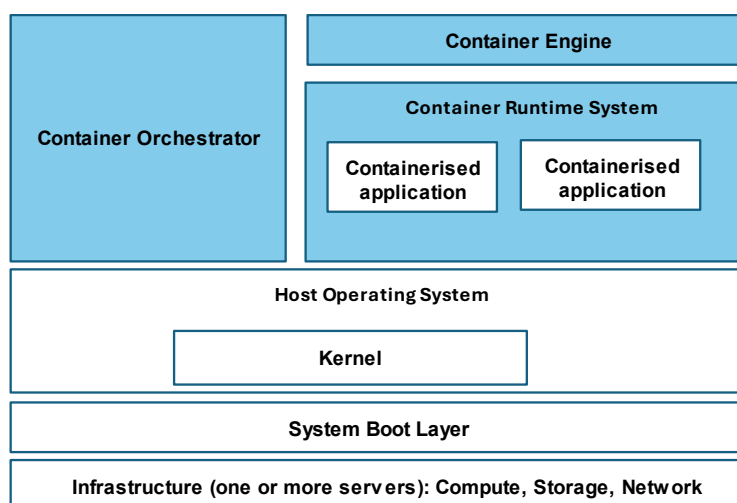


Figure 4.1.2.1-1: CES components

4.1.3.2 Security-relevant environmental dependencies (out of scope)

The following are considered outside the scope of the present document but may be security-relevant environmental components:

- The Host Operating System on which the CRS executes (OS kernel, drivers).
- Container registries (public/private).

- 785 • Container image build pipelines (CI/CD Pipelines).
- 786 • Third-party monitoring or security agents installed at runtime.
- 787 • Identity and Access Management systems used for authentication/authorization.
- 788 • Underlying hardware platform (TPM, HSM, hardware root-of-trust).
- 789 • Processes, tools, environments, or development/packaging formats whose primary purpose is to develop,
- 790 build, or package container artifacts, including but not limited to build pipelines, CI/CD systems, compilers,
- 791 language-specific package managers, container image builders, image layers, and image manifests.

792 4.2 Product variants

793 4.2.0 Definition of conforming products

794 The present document covers products implementing Virtualization or container execution functionalities as defined in
795 clauses 4.1.1 and 4.1.2.

796 A conforming product shall include at least one of the following core components:

- 797 • the Hypervisor, for products implementing a Virtualization Execution Stack (VES); or
- 798 • the Container Runtime System (CRS), for products implementing a Container Execution Stack (CES).

799 The Hypervisor and CRS represent the essential execution environments that enable the isolation, management, and
800 execution of virtualized or containerized workloads. They constitute the core mandatory components covered by the
801 present document.

802 Other components defined in clauses 4.1.1 and 4.1.2 such as the Management and Orchestration (M&O) System,
803 Container Engine (CE), or Container Orchestrator (CO) are considered optional extensions. These components are
804 included within the scope of the present document only when they are integrated as part of the product as placed on the
805 market.

806 4.2.1 General product variants

807 Products conforming to the present document may therefore be implemented in one or more of the following variants:

- 808 • Hypervisor only: A product implementing solely the hypervisor component.
- 809 • Hypervisor + M&O System: A product integrating both the hypervisor and its management and orchestration
810 components.
- 811 • Container Runtime System (CRS) only: A product implementing the container execution environment without
812 including the container engine or orchestrator.
- 813 • CRS + Container Engine (CE): A product combining the low-level runtime with the mid-level engine
814 responsible for image handling and lifecycle management.
- 815 • CRS + Container Engine + Container Orchestrator (CO): A product integrating all three layers of the container
816 execution stack.
- 817 • CRS + Container Orchestrator (CO): a product integrating the runtime directly with an orchestrator that
818 manages container scheduling, scaling and policy enforcement.
- 819 • CRS + integrated CO with CE: a product integrating the runtime with an orchestrator that natively provides
820 container-lifecycle and image-management functions.

821 For each component identified in clauses 4.1.1 and 4.1.2, the corresponding threats, security requirements, and
 822 assessment activities are defined separately in subsequent clauses of the present document. Only the elements relevant
 823 to the components implemented in the product shall apply. If a given component is not included in the product, the
 824 related threats, requirements, and assessment activities are considered not applicable for the purpose of conformity
 825 assessment.

826 The manufacturer shall declare, within the conformity assessment documentation, the composition and functional
 827 boundary of the product, identifying which components are included and assessed under the present document.

828 Products that do not include at least a Hypervisor or a Container Runtime System (CRS) are not considered
 829 Virtualization or Container Execution Stacks within the meaning of the present document and therefore fall outside its
 830 scope.

831 4.2.2 Specific product variants

832 4.2.2.0 Applicability of the present document

833 In addition to the general variants described above, certain Hypervisor or CRS may integrate auxiliary software
 834 components such as a boot manager, kernel or embedded operating system directly within its boundary.

835 This clause defines the specific applicability conditions under which the present document alone is sufficient to ensure
 836 conformity, and when additional CRA harmonised standards are required.

837 4.2.2.1 Hypervisor with bundled Boot Manager and kernel/Operating System

838 This variant denotes a product in which the Boot Manager (firmware/bootloader) and the (micro)kernel or host
 839 operating system required for the hypervisor are bundled within the product boundary.

840 The requirements defined in clause 5.1 are sufficient to ensure conformity for this variant when all of the following
 841 conditions are met:

- 842 a) The bundled Boot Manager and kernel/OS are single-purpose, used exclusively to initialize and run the
 843 hypervisor and its management functions.
- 844 b) The Boot Manager and kernel/OS are updated only through the product's authenticated update mechanism,
 845 with digital-signature verification and rollback protection consistent with REQ-H-UPD-002 and
 846 REQ-H-UPD-003.
- 847 c) No general-purpose services are exposed to users or external processes (e.g. public package managers,
 848 unrestricted shells or unrelated daemons).
- 849 d) Secure-boot and measured-boot mechanisms are implemented and enforced by the product in accordance with
 850 REQ-H-B-INT-001 to REQ-H-B-INT-003 and REQ-H-RA-INT-003; any trust anchors are under product
 851 control.

852 If one or more of these conditions are not fulfilled, the Boot Manager and/or kernel/OS shall be treated as Operational
 853 Environment (OE) components, and conformity should be demonstrated by applying the relevant requirements for the
 854 corresponding CRA harmonised standards as referenced in clause 4.6.3.

855 4.2.2.2 Embedded Operating System for containerization or Virtualization

856 This variant denotes a product in which a minimal embedded operating system is included solely to host the Container
 857 Runtime System (CRS) and, where applicable, integrated container-management components such as the Container
 858 Engine (CE) or Container Orchestrator (CO).

859 The requirements defined in clause 5.2 are sufficient to ensure conformity for this variant when all of the following
 860 conditions are met:

- 861 a) The embedded operating system is dedicated exclusively to container execution and does not host unrelated
 862 applications or general-purpose services.
- 863 b) Operating-system updates are delivered only through the product's signed update pipeline in accordance with
 864 REQ-CRS-UPD-002 and REQ-CRS-UPD-003.

- c) Kernel interfaces, namespaces, and capabilities are restricted in line with REQ-CRS-CN-ISO-002 and REQ-CRS-CN-ISO-003, and configuration-control requirements (REQ-CRS-CFG-002 and REQ-CRS-CFG-003).
- d) Image integrity, authenticity, and trust-anchor protection are enforced in accordance with REQ-CRS-IMG-INT-001 to REQ-CRS-IMG-INT-003, and runtime integrity protection per REQ-CRS-RP-INT-002 and REQ-CRS-RP-INT-003.

If any of the above conditions are not met, the embedded operating system shall be treated as an Operational Environment (OE) component, and conformity should be demonstrated using the CRA Operating System harmonised standard ETSI EN 304 626 [i.5], as referenced in clause 4.6.3.

4.3 Use Cases

4.3.1 Purpose and Rationale

In accordance with the proportionality principle mandated by the CRA [i.1], this clause defines representative use cases for VESs and CESs. These use cases illustrate typical deployment contexts that influence the applicability of security requirements.

Each use case is associated with a risk level, derived directly from the application of the risk assessment methodology detailed in clause B. While these use cases are representative, a more granular and detailed risk assessment will be necessary in scenarios where:

- The product combines characteristics of multiple use cases.
- The product does not align clearly with any single representative use case.
- The manufacturer identifies, through its own analysis, a threat exposure that is higher, lower, or different from that assumed in the representative use case.

To comply with CRA Annex I, Part I, (1), the product manufacturers shall apply a documented and effective risk assessment process to identify, evaluate, and address cybersecurity risks. The risk assessment methodology provided in clause B is included for information purposes only and is not normative. The risk assessment methodology in clause B is intended to illustrate one possible approach that may be adopted or replaced by an equivalent method already in use by the manufacturer.

4.3.2 Use Cases for VES

The risk characteristics for each VES use case are identified based on the risk assessment methodology defined in clause B.

UC-V1: VES for isolated, non-critical functions

- Description: A VES deployed on a dedicated host in an isolated environment. The host hosts a very small number of static VMs dedicated to non-critical functions. In this context, non-critical functions refer to operations whose compromise would not result in harm to human safety, disruption of critical infrastructure services, or exposure of sensitive data belonging to multiple users.
- Example environments:
 - Standalone servers or industrial PCs deployed in physically isolated networks.
 - Edge devices or gateways with highly restricted links.
 - Personal computers or laptops running a local hypervisor in a non-production environment.
- Illustrative end user business applications:
 - Isolated diagnostic data collection points in a factory.
 - Environmental monitoring systems for non-critical building zones.

- 906 - Local telemetry units for non-essential parameters in a grid.
- 907 - Read-only public information displays in industrial facilities (e.g. current temperature, non-operational
- 908 metrics).
- 909 - Consumer devices such as personal computers or laptops running local virtual machines for hobbyist
- 910 testing, educational purposes, or personal experimentation.
- 911 • Risk level: Low (see clause B.5.1).
- 912 • Main identified risks: These include malware exploiting unpatched systems in isolated industrial networks;
- 913 Unauthorized physical access to devices resulting in data corruption; and misconfigured hypervisors with
- 914 default administrative credentials discovered in industrial telemetry units.
- 915 **UC-V2: VES for general business workloads**
- 916 • Description: A VES deployed within a shared multi-tenant environment (private, public, or hybrid cloud),
- 917 hosting a heterogeneous mix of virtualized workloads (VMs) for various customers or independent internal
- 918 units. The platform's primary goal is achieving resource efficiency and providing standard logical separation.
- 919 This use case also includes deployments where the VES runs on connected edge hardware hosting multiple
- 920 VMs (including legacy or specialized workloads) that require stronger isolation.
- 921 • Example environments:
- 922 - Enterprise private cloud platforms shared across multiple departments.
- 923 - Public Cloud multi-tenant IaaS/PaaS used for non-critical services.
- 924 - On-premises Virtualization clusters for internal workloads.
- 925 - Internal development and test environments.
- 926 - Industrial PCs or edge gateways hosting multiple VM workloads (e.g. legacy HMI + modern data
- 927 gateway).
- 928 - uCPE (universal Customer Premises Equipment) devices running virtualized network appliances (SD-
- 929 WAN, firewall, router).
- 930 - Edge servers in transportation or building systems.
- 931 • Illustrative end user business applications:
- 932 - Corporate IT service platforms.
- 933 - Enterprise middleware supporting internal operations (e.g. ERP, HR, finance systems).
- 934 - Internal microservices for business applications.
- 935 - Educational and research platforms.
- 936 - Private banking or insurance back-end systems running on controlled virtualized infrastructures.
- 937 - Virtualized network functions at branch/edge sites (firewall, router, SD-WAN).
- 938 • Risk level: Medium (see clause B.5.1).
- 939 • Main identified risks: These include VM escapes exploiting vulnerabilities in Virtualization platforms (e.g.
- 940 VMware ESXi, KVM/QEMU); privilege escalation attacks enabling host-level access; Unauthorized access to
- 941 sensitive corporate data through misconfiguration or credential compromise; and exploitation of known
- 942 vulnerabilities in virtualized management interfaces causing service outages or data breaches. Edge
- 943 environments face higher exposure and less continuous supervision.

944 UC-V3: VES for critical workloads

- 945 • Description: A high-assurance VES forming the foundation of infrastructures that support critical or regulated
946 workloads requiring strong isolation, resource separation, and continuous security assurance. Such
947 environments may host sensitive applications, essential services, or externally sourced workloads whose
948 compromise could lead to major operational disruption, data breaches, or safety impacts. This scenario
949 demands maximum assurance isolation and strong separation of concerns, typically involving untrusted,
950 heterogeneous tenants (distinct organizations or independent units). In certain deployments, these platforms
951 may also operate at connected edge locations, where strong VM isolation is required to separate safety-
952 relevant and non-safety workloads on the same hardware.
- 953 • Example environments:
 - 954 - Public IaaS/PaaS clouds where tenants are considered untrusted relative to one another.
 - 955 - Private clouds supporting multiple departments or projects with strong isolation needs.
 - 956 - Regulated private clouds (e.g. dedicated to a critical sector like energy).
 - 957 - High-density Virtualization or micro-VM platforms supporting critical or externally validated workloads.
 - 958 - Connected edge systems requiring high-assurance separation between operational and support
959 workloads.
- 960 • Illustrative end user business applications:
 - 961 - Telecom core clouds hosting critical network functions.
 - 962 - Government or defence virtualized infrastructures processing sensitive or classified data.
 - 963 - Financial services platforms hosted in public multi-tenant clouds.
 - 964 - eHealth Platforms handling patient records (PHI).
 - 965 - Safety-related edge processing workloads where VM isolation protects operational integrity.
- 966 • Risk level: High (see clause B.5.1).
- 967 • Main identified risks: These include exploitation of hypervisor vulnerabilities enabling cross-workload or
968 cross-domain compromise; hardware-level side-channel attacks (e.g. Meltdown, Spectre, Foreshadow) leaking
969 sensitive data; denial-of-service attacks targeting shared compute, storage, or network planes; and
970 Unauthorized access through compromised management or orchestration components leading to system-wide
971 outages or data exposure. When deployed at the edge, additional risks include semi-trusted network exposure
972 and limited continuous monitoring.

973 4.3.3 Use Cases for CES

974 The risk characteristics for each CES use case are identified based on the risk assessment methodology defined in
975 clause B.

976 UC-C1: CES for isolated, non-critical functions

- 977 • Description: A CES deployed on a device with limited connectivity. This device typically performs non-
978 critical functions, often in an isolated environment. These applications perform non-critical functions, meaning
979 their compromise would not result in harm to human safety, disruption of critical infrastructure services, or
980 exposure of sensitive data belonging to multiple users.
- 981 • Example environments:
 - 982 - Isolated test/dev environments.
 - 983 - Basic controllers for non-critical functions.

- 984 - Consumer devices such as personal smartphones running isolated app containers for non-critical
 985 applications or laptops using local container engines (e.g. Docker Desktop, Podman) for personal
 986 projects.
- 987 • Illustrative end user business applications:
- 988 - Educational or hobbyist containerized workloads.
- 989 - Local telemetry or monitoring for non-critical parameters.
- 990 - Entertainment or personal productivity apps packaged in containers.
- 991 • Risk level: Low (see clause B.5.2).
- 992 • Main identified risks: These include container escape vulnerabilities in container runtimes (e.g. runc);
 993 incidents of exposed container management daemons without TLS protection (e.g. Docker daemon) in isolated
 994 environments; and misconfigurations or Unauthorized physical access leading to minor service disruption or
 995 data corruption.
- 996 **UC-C2: CES for general business workloads**
- 997 • Description: A CES deployed within a shared multi-tenant environment (private, public, or hybrid cloud),
 998 hosting a heterogeneous mix of containerized workloads (e.g. microservices, CI/CD pipelines). The tenants
 999 may belong to different organizations (e.g. a standard multi-client SaaS platform) or independent units within
 1000 the same organization. The platform's primary goal is resource efficiency, agility, and standard logical
 1001 separation. This use case also includes CES deployments at the network edge where the system runs
 1002 manufacturer-defined or business-relevant workloads, may store or process sensitive data, and is reachable
 1003 over external or semi-trusted networks, but where the consequence of compromise is operational or business
 1004 disruption rather than critical safety failure.
- 1005 • Example environments:
- 1006 - Enterprise private cloud infrastructures shared across multiple departments/teams (multi-tenancy at the
 1007 organizational level).
- 1008 - Public Cloud containerized clusters used for non-critical, general business applications.
- 1009 - On-premises container platforms.
- 1010 - Internal microservices architectures.
- 1011 - Connected edge devices and gateways operating in semi-trusted networks (e.g. industrial IoT gateways,
 1012 smart building controllers, retail analytics devices, agricultural monitoring and control systems).
- 1013 • Illustrative end user business applications:
- 1014 - Corporate IT shared services.
- 1015 - Educational and research platforms.
- 1016 - Retail and e-commerce frontends.
- 1017 - Enterprise middleware platforms.
- 1018 - Back-office systems (HR, general accounting).
- 1019 - Local AI inference and analytics (e.g. video analytics, quality monitoring, sensor fusion) performed at
 1020 the edge before forwarding to cloud services.
- 1021 • Risk level: Medium (see clause B.5.2).

- Main identified risks: These include configuration and access-control weaknesses in container orchestration platforms (e.g. Kubernetes); container breakouts through insecure host mount privileges or misconfigured namespaces; privilege escalation within containers or across pods; Unauthorized lateral movement in internal networks; and exploitation of exposed container registries, management APIs, or CI/CD integration points. Edge deployments add risks from semi-trusted networks, weak monitoring, and securing local credentials/updates.

1028 UC-C3: CES for critical workloads

- Description: A CES forming the foundation of platforms that support critical or regulated workloads with a consequence of failure classified as high impact. This platform demands high assurance isolation, typically involving dynamic, untrusted containers from heterogeneous tenants (distinct organizations or independent units). This also includes deployments at the edge where the containerized workloads perform safety-relevant, regulatory, or critical operational functions.
- Example environments:
 - Public container platforms hosting high-risk, multi-tenant applications.
 - Regulated private container platforms supporting essential services (e.g. energy, telecommunications).
 - Federated or cross-organization infrastructures hosting workloads from distinct trust domains.
 - Zero-trust environments where container-level isolation is required to be the strongest security boundary.
 - Edge deployments supporting critical services (e.g. industrial control gateways, medical data processing appliances, public infrastructure monitoring nodes).
- Illustrative end user business applications:
 - Government or telecom workloads with elevated assurance or compliance requirements.
 - Financial and e-identity services operated on high-assurance container infrastructures.
 - eHealth services handling patient records (PHI).
 - Critical data-analytics or control applications running in shared but tightly isolated environments.
 - Real-time safety, monitoring, or control workloads deployed on edge.
- Risk level: High (see clause B.5.2).
- Main identified risks: These include API server exposures in container orchestration platforms (e.g. Kubernetes®) enabling attackers to take control of clusters; container escape exploits abused in public cloud environments; supply chain attacks targeting public container images; and Unauthorized data access between tenant workloads caused by misconfigured networking or weak isolation. Edge deployments add risks from semi-trusted networks, limited monitoring, and ensuring update/image integrity.

1053 4.3.4 Use Case to Risk Mapping Table

1054 **Table 4.3.4-1: Use case to risk mapping**

Product	Use Case	Description	Risk Level	Clause
VES	UC-V1	VES for isolated, non-critical functions	Low	B.5.1
VES	UC-V2	VES for single-tenant cloud	Medium	B.5.1
VES	UC-V3	VES for multi-tenant cloud	High	B.5.1
CES	UC-C1	CES for isolated, non-critical functions	Low	B.5.2
CES	UC-C2	CES for single-tenant cloud	Medium	B.5.2
CES	UC-C3	CES for multi-tenant cloud	High	B.5.2

1055

4.4 Threats

4.4.1 Overview

Table 4.4.1-1 maps identified cybersecurity threats to the specific components they affect within the scope of the present document: Hypervisor Type I, Hypervisor Type II, Orchestration & Management System, Container Orchestrator, Container Engine, and Container Runtime System.

The first column lists the threat categories considered in the present document, grouped by their primary security impact, such as interference or escape, integrity compromise, Unauthorized access, data disclosure, denial of service, logging issues, malicious updates, insecure configurations, and supply chain compromise.

Each threat is referenced by its unique Threat ID, ensuring traceability to its detailed description in the threat catalogue in the following clauses. This mapping provides a clear overview of which threats are applicable to each component type, serving as the foundation for the risk assessment and for defining the corresponding security requirements and controls.

Table 4.4.1-1: Threats to components mapping

Threat	Hypervisor Type I	Hypervisor Type II	Orchestration & Management	Container Orchestrator	Container Engine	Container Runtime System
Interference / Escape	T-HYP1-INTF-ESC	T-HYP2-HOST-ABUSE	T-ORCH-CTRL-PLANE	T-CO-INTF-ESC	T-CE-INTF-ESC	T-CRS-INTF-ESC
Integrity Compromise	T-HYP1-INTEG-COMP	T-HYP2-INTEG-COMP	T-ORCH-INTEG-TRUST	T-CO-INTEG-TRUST	T-CE-INTEG-COMP	T-CRS-INTEG-COMP
Unauthorized Admin Access	T-HYP1-UNAUTH-ADMIN	T-HYP2-UNAUTH-ADMIN	T-ORCH-UNAUTH-ADMIN	T-CO-UNAUTH-ADMIN	T-CE-UNAUTH-ADMIN	T-CRS-UNAUTH-ADMIN
Unauthorized Privileged Actions	T-HYP1-UNAUTH-FUNC	T-HYP2-UNAUTH-FUNC	T-ORCH-UNAUTH-FUNC	T-CO-UNAUTH-FUNC	T-CE-UNAUTH-FUNC	T-CRS-UNAUTH-FUNC
Sensitive Data Disclosure	T-HYP1-DATA-DISC	T-HYP2-DATA-DISC	T-ORCH-DATA-DISC	T-CO-DATA-DISC	T-CE-DATA-DISC	T-CRS-DATA-DISC
DoS via Resource Exhaustion	T-HYP1-AVAIL-DOS	T-HYP2-AVAIL-DOS	T-ORCH-AVAIL-DOS	T-CO-AVAIL-DOS	T-CE-AVAIL-DOS	T-CRS-AVAIL-DOS
Insufficient / Tampered Logging	T-HYP1-LOG-INT	T-HYP2-LOG-INT	T-ORCH-LOG-INT	T-CO-LOG-INT	T-CE-LOG-INT	T-CRS-LOG-INT
Malicious / Compromised Update	T-HYP1-UPD-COMP	T-HYP2-UPD-COMP	T-ORCH-UPD-COMP	T-CO-UPD-COMP	T-CE-UPD-COMP	T-CRS-UPD-COMP
Insecure Default and Changed Configurations	T-HYP1-CFG-COMP	T-HYP2-CFG-COMP	T-ORCH-CFG-COMP	T-CO-CFG-COMP	T-CE-CFG-COMP	T-CRS-CFG-COMP
Supply Chain Compromise	T-HYP1-SC-COMP	T-HYP2-SC-COMP	T-ORCH-SC-COMP	T-CO-SC-COMP	T-CE-SC-COMP	T-CRS-SC-COMP
Compromise via hardware-level or platform trust components	T-ALL-HW-ACCESS					

4.4.2 VES

4.4.2.1 Hyper Type I

T-HYP1-INTF-ESC: Inter-VM interference and escape via hypervisor/host interfaces

- 1073 A malicious VM exploits flaws in the hypervisor, device emulation, vSwitch, or shared hardware to read/alter another
1074 VM's memory/state or escape to the hypervisor/root.
- 1075 **T-HYP1-INTEG-COMP:** Integrity compromise of workloads or platform components
- 1076 A malicious actor tampers with hypervisor binaries, kernel modules, configuration files, or VM images on the host
1077 storage, injecting malicious code or disabling isolation/security controls. The compromised hypervisor continues to
1078 operate without detection by higher-level systems, allowing the attacker to manipulate or monitor workloads.
- 1079 **T-HYP1-UNAUTH-ADMIN:** Unauthorized access to Type-1 hypervisor management interfaces
- 1080 An attacker exploits missing or weak authentication on the hypervisor's direct management interfaces (e.g. vSphere
1081 Client, ESXi Shell, SSH, vendor-specific APIs). This may be due to default credentials, weak password policies,
1082 outdated authentication protocols, or disabled MFA.
- 1083 **T-HYP1-UNAUTH-FUNC:** Unauthorized privileged actions or data access on a Type-1 hypervisor
- 1084 An attacker with valid but limited credentials (e.g. VM operator role) or a compromised low-privileged API token
1085 exploits insufficient authorization enforcement on the hypervisor management plane to perform high-impact actions or
1086 access restricted data.
- 1087 **T-HYP1-DATA-DISC:** Unauthorized disclosure of sensitive data in a Type-1 hypervisor environment
- 1088 An attacker exploits vulnerabilities, weak encryption, or misconfigurations in the hypervisor or its management plane to
1089 gain Unauthorized access to sensitive data from virtual machines or platform resources.
- 1090 This may include data in use (e.g. reading contents directly from RAM or CPU registers of running workloads), data at
1091 rest (e.g. accessing virtual disk images or configuration files), and data in transit (e.g. intercepting network traffic
1092 between VMs).
- 1093 **T-HYP1-AVAIL-DOS:** DoS via resource exhaustion in a Type-1 hypervisor environment
- 1094 An attacker either from a malicious or compromised guest VM, or via external network/API access, consumes
1095 excessive CPU, memory, I/O bandwidth, or network capacity on the hypervisor host, leading to degraded performance
1096 or full unavailability of hosted VMs.
- 1097 This also includes scenarios involving resource overcommitment, where CPU steal time, memory contention, or shared
1098 I/O bottlenecks impact the performance of co-located VMs, potentially leading to degraded availability even when
1099 individual VM resource limits are not exceeded.
- 1100 **T-HYP1-LOG-INT:** Insufficient logging or log tampering in a Type-1 hypervisor environment
- 1101 An attacker with access to a guest VM or hypervisor management interface exploits insufficient logging, disabled audit
1102 trails, or insecure log storage to hide malicious actions (e.g. VM configuration changes, hypervisor setting
1103 modifications). They may also tamper with existing logs to erase evidence of compromise.
- 1104 **T-HYP1-UPD-COMP:** Malicious or compromised update in a Type-1 hypervisor environment
- 1105 An attacker injects, replaces, or replays a hypervisor firmware or software update to gain control over the Virtualization
1106 layer. Because Type-1 hypervisors run directly on hardware, compromise here provides full access to all guest VMs and
1107 host resources.
- 1108 **T-HYP1-CFG-COMP:** Insecure or Unauthorized configuration in a Type-1 Hypervisor
- 1109 An attacker exploits insecure default settings or Unauthorized/insecure configuration changes to the hypervisor (via
1110 CLI, web UI, API, or configuration files). This can weaken isolation, expose management services, or disable
1111 logging/security controls, leading to Unauthorized access, privilege escalation, or compromise of workloads.
- 1112 **T-HYP1-SC-COMP:** Supply chain compromise in a Type-1 Hypervisor
- 1113 An attacker compromises a dependency used in the hypervisor's build process, such as a low-level driver, cryptographic
1114 library, or firmware, to inject a backdoor or exploitable vulnerability.
- 1115 Because the malicious code is baked into the hypervisor's core, every VM running on the host inherits the exposure,
1116 giving the attacker a privileged foothold.

1117 4.4.2.2 Hyper Type II

1118 **T-HYP2-HOST-ABUSE:** Guest-to-host escalation via host OS dependency and shared services

1119 A malicious VM abuses the host OS attack surface to access host resources or other VMs; weaker isolation than Type-1
1120 increases risk.

1121 **T-HYP2-INTEG-COMP:** Integrity compromise in hosted Virtualization

1122 An attacker compromises the host OS or hypervisor application binaries/configuration, or tampers with VM images in
1123 the host filesystem. Because Type-2 relies on the host OS, compromise of the host environment compromises the
1124 Virtualization layer and workloads.

1125 **T-HYP2-UNAUTH-ADMIN:** Unauthorized access to Type-2 hypervisor or host OS admin functions

1126 An attacker exploits weak authentication on the hypervisor application or on the host OS's administrative accounts.
1127 Since Type-2 hypervisors run on a general-purpose OS, compromise of the host admin account grants full control over
1128 the hypervisor and all guest VMs.

1129 **T-HYP2-UNAUTH-FUNC:** Unauthorized privileged actions or data access on a Type-2 hypervisor

1130 An attacker with authenticated but restricted access to the hypervisor application or underlying OS exploits inadequate
1131 authorization to perform actions reserved for administrators. Since Type-2 hypervisors rely on the host OS for resource
1132 access, weak separation can also allow escalation to host-level privileges.

1133 **T-HYP2-DATA-DISC:** Unauthorized disclosure of sensitive data in a Type-2 hypervisor environment

1134 An attacker leverages weaknesses in the hosted hypervisor or host OS to read sensitive data belonging to VMs or the
1135 platform.

1136 This may include data in use (e.g. extracting secrets from process memory), data at rest (e.g. accessing VM image files
1137 stored on the host), and data in transit (e.g. capturing inter-VM or host-VM communications).

1138 **T-HYP2-AVAIL-DOS:** DoS via resource exhaustion in a Type-2 hypervisor environment

1139 An attacker running on a guest VM or the host OS consumes excessive system resources (CPU, memory, storage,
1140 network), causing other VMs or the hypervisor process to stall or crash.

1141 **T-HYP2-LOG-INT:** Insufficient logging or log tampering in a Type-2 hypervisor environment

1142 An attacker running on the host OS or inside a VM exploits weak logging practices to erase or manipulate event records
1143 (e.g. VM lifecycle events, admin actions) to avoid detection. In a Type-2 setup, logs are often stored in user-accessible
1144 locations on the host, making them easier to modify.

1145 **T-HYP2-UPD-COMP:** Malicious or compromised update in a Type-2 hypervisor environment

1146 An attacker injects or manipulates updates to the hypervisor application running on top of the host OS. Because these
1147 updates often come via user-initiated downloads, attackers may deliver installers that have been modified with
1148 malicious code or downgrade the software to a vulnerable version.

1149 **T-HYP2-CFG-COMP:** Unauthorized or insecure configuration in a Type-2 hypervisor

1150 A Type-2 hypervisor may be exposed to insecure configuration either through Unauthorized or insecure modifications
1151 of its parameters via the host operating system environment, local malware, or its application interface, or through
1152 insecure defaults inherited from the host operating system or shipped by the hypervisor itself. Examples include
1153 permissive VM networking modes, host-guest shared folders enabled by default, or control channels without
1154 authentication. Such insecure or Unauthorized configurations can weaken VM isolation, expose the host system, or
1155 allow privilege escalation between guest VMs and the host.

1156 **T-HYP2-SC-COMP:** Supply chain compromise in a Type-2 Hypervisor

1157 An attacker compromises a dependency used by the Type-2 hypervisor, such as a driver, shared library, system
1158 component, or packaged module, during development, upstream distribution, or packaging.

1159 When the hypervisor is installed on a user's OS, the malicious dependency executes with host-level privileges, allowing
1160 the attacker to compromise both the hypervisor and the host OS.

1161 4.4.2.3 Orchestration and Management system

1162 **T-ORCH-CTRL-PLANE:** Control-plane abuse leading to cross-workload interference and privilege escalation

1163 An attacker with (or who obtains) control-plane/API access manipulates scheduling, networking, storage or policies to
1164 read/impact other tenants' workloads or to break isolation at scale.

1165 **T-ORCH-INTEG-TRUST:** Misplaced trust in compromised platform

1166 An orchestrator and management system treats a hypervisor host as trustworthy without verifying its integrity (no
1167 attestation, or attestation ignored). A compromised host can feed falsified state/configuration data back to the
1168 orchestrator and run tampered workloads while appearing legitimate.

1169 **T-ORCH-UNAUTH-ADMIN:** Unauthorized access to orchestration/management plane

1170 An attacker gains Unauthorized access to the orchestration platform's administrative APIs, dashboards, or CLI tools
1171 through weak authentication, missing identity federation controls, or compromised admin accounts. This provides
1172 cluster-wide control of all connected hypervisors and workloads.

1173 **T-ORCH-UNAUTH-FUNC:** Unauthorized privileged actions or data access in orchestration/management systems

1174 An attacker with authenticated but restricted access to the orchestration platform exploits insufficient authorization on
1175 APIs or dashboards to invoke privileged actions cluster-wide or access sensitive data.

1176 **T-ORCH-DATA-DISC:** Unauthorized disclosure of sensitive data via Virtualization orchestration/management system

1177 An attacker abuses vulnerabilities, excessive privileges, or misconfigured access controls in the orchestration platform
1178 to obtain sensitive data from managed VMs or platform resources.

1179 This may include data in use (e.g. retrieving memory dumps from running workloads), data at rest (e.g. accessing stored
1180 VM snapshots or configuration files), and data in transit (e.g. intercepting orchestrator-mediated traffic).

1181 **T-ORCH-AVAIL-DOS:** DoS via resource exhaustion in Virtualization orchestration/management systems

1182 An attacker overwhelms the orchestration control plane, APIs, or schedulers with excessive requests, malformed
1183 workload definitions, or frequent provisioning/de-provisioning cycles. This can lead to unresponsiveness of
1184 management tools and inability to operate workloads.

1185 **T-ORCH-LOG-INT:** Insufficient logging or log tampering in Virtualization orchestration/management systems

1186 An attacker targets the orchestration system's audit and logging functions to hide or falsify records of administrative
1187 actions, API calls, or workload lifecycle events. These systems often serve as the central source of truth, so tampering
1188 can mislead both human operators and automated security processes.

1189 **T-ORCH-UPD-COMP:** Malicious or compromised update in Virtualization orchestration/management systems

1190 An attacker tampers with updates for the orchestration platform or its plugins/extensions. A compromised orchestration
1191 system can push malicious configurations or updates to all managed hypervisors and workloads, multiplying the impact.

1192 **T-ORCH-CFG-COMP:** Unauthorized or insecure configuration in Virtualization orchestration/management systems

1193 An attacker exploits insecure default settings (such as default administrator accounts, weak API tokens, unencrypted
1194 management channels, or overly permissive role assignments) or performs Unauthorized modifications through the
1195 orchestration system's management interfaces or APIs. Because the orchestration system controls multiple hypervisors
1196 and cluster-wide policies, insecure defaults or misconfigurations can propagate across the entire environment,
1197 weakening isolation, exposing management services, or granting attackers elevated privileges.

1198 **T-ORCH-SC-COMP:** Supply chain compromise in Virtualization orchestration/management systems

1199 An attacker injects malicious code into a dependency (for example, a module, API library, or plugin) used in the
1200 orchestrator's control plane. Because orchestrators have administrative access to all managed hosts, the compromise
1201 cascades to the entire infrastructure once deployed.

1202 4.4.3 CES

1203 4.4.3.1 Container Orchestrator

1204 **T-CO-INTF-ESC:** Orchestrator-level cross-workload interference

1205 An attacker with access to orchestration APIs or permissions in one namespace/project abuses orchestration
1206 misconfigurations, policy gaps, or network segmentation flaws to interfere with workloads managed by the orchestrator.

1207 **T-CO-INTEG-TRUST:** Orchestrator trusts compromised nodes or components

1208 The orchestrator schedules workloads onto a container host (node) or engine whose binaries, configuration, or runtime
1209 state have been tampered with, without verifying its integrity. This may be due to the absence of attestation, or
1210 attestation being ignored or spoofed.

1211 **T-CO-UNAUTH-ADMIN:** Unauthorized access to orchestrator management plane

1212 An attacker gains access to the orchestrator's administrative interfaces (web dashboards, CLI tools, APIs) due to weak
1213 or missing authentication controls, default credentials, or compromised admin accounts. Since the orchestrator manages
1214 scheduling, scaling, and policy enforcement for all workloads, compromise can result in complete cluster control.

1215 **T-CO-UNAUTH-FUNC:** Unauthorized privileged operations or data access in the orchestrator

1216 An attacker with valid but low-privileged credentials or a compromised service account exploits insufficient
1217 authorization enforcement in the orchestrator's API server, CLI tools, or dashboards to perform privileged actions or
1218 read restricted data.

1219 **T-CO-DATA-DISC:** Unauthorized disclosure of sensitive data via the container orchestrator

1220 An attacker exploits vulnerabilities, excessive privileges, or misconfigured access controls in the orchestrator's API
1221 server, dashboard, or CLI to obtain sensitive data from managed workloads or control-plane resources.

1222 This may include data in use (e.g. reading container memory or runtime secrets), data at rest (e.g. accessing container
1223 images, volumes, or configuration files), and data in transit (e.g. capturing pod-to-pod or service communications).

1224 **T-CO-AVAIL-DOS:** DoS via resource exhaustion in the container orchestrator

1225 An attacker overwhelms the orchestrator's control plane, APIs, or schedulers with excessive requests, malformed
1226 deployment manifests, or frequent scaling operations. This can lead to delays or complete unavailability in workload
1227 scheduling, management, and monitoring.

1228 **T-CO-LOG-INT:** Insufficient logging or log tampering in the Container Orchestrator

1229 An attacker targets the orchestrator's event and audit logging systems to suppress or alter records of critical cluster
1230 events, administrative actions, or API calls. This can allow malicious changes—such as Unauthorized deployments,
1231 RBAC policy modifications, or workload scheduling—to go undetected.

1232 **T-CO-UPD-COMP:** Malicious or compromised update in the Container Orchestrator

1233 An attacker injects, tampers with, or replays updates to the orchestrator's control plane components (API server,
1234 scheduler, controller manager) or its plugins/extensions.

1235 A compromised orchestrator update could allow the attacker to modify workload scheduling, bypass RBAC rules, or
1236 deploy malicious workloads across all managed nodes.

1237 **T-CO-CFG-COMP:** Unauthorized or insecure configuration in the Container Orchestrator

1238 An attacker exploits insecure default settings (such as anonymous API access, default administrator accounts,
1239 unrestricted inter-pod communication, or permissive admission controller policies) or performs Unauthorized
1240 modifications of orchestrator configurations through its management interfaces (API server, dashboard, CLI) or
1241 configuration stores. Such weaknesses can compromise workload isolation, weaken network segmentation, bypass
1242 RBAC enforcement, or allow unverified workloads to run. Because the orchestrator governs multiple containers and
1243 services, insecure or Unauthorized configurations can rapidly escalate into cluster-wide compromise or lateral
1244 movement between workloads. When orchestrator-level configuration is propagated to container engines or container
1245 runtimes, the configuration source and propagation path shall be authenticated and authorized to prevent Unauthorized
1246 configuration changes from cascading across the environment.

1247 **T-CO-SC-COMP:** Supply chain compromise in a Container Orchestrator

1248 An attacker injects malicious code into a dependency (for example, a module, plugin, or API library) used in the
1249 orchestrator's control plane.

1250 Because the orchestrator controls all cluster workloads and nodes, the compromise propagates to the entire
1251 environment.

1252 4.4.3.2 Container Engine

1253 **T-CE-INTF-ESC:** Engine-level cross-container interference

1254 A malicious container exploits weaknesses in engine-level isolation enforcement (configuration, API exposure, or
1255 plugin vulnerabilities) to interfere with other containers running under the same engine.

1256 **T-CE-INTEG-COMP:** Integrity compromise of container engine and managed images

1257 An attacker tampers with the container engine's binaries, configuration files, or locally cached container images. The
1258 compromised engine can then execute manipulated workloads or disable controls, while still appearing healthy to the
1259 orchestrator.

1260 **T-CE-UNAUTH-ADMIN:** Unauthorized access to container engine control endpoints

1261 An attacker gains access to the container engine's local or remote control interfaces (e.g. Docker API socket, containerd
1262 gRPC API) due to weak authentication, unprotected sockets, or lack of TLS. This gives direct control over container
1263 lifecycle operations and engine configuration.

1264 **T-CE-UNAUTH-FUNC:** Unauthorized privileged operations or data access in the container engine

1265 An attacker with authenticated but restricted engine API access (local socket or remote endpoint) exploits weak
1266 authorization to invoke high-risk operations beyond their intended role.

1267 **T-CE-DATA-DISC:** Unauthorized disclosure of sensitive data via the container engine

1268 An attacker gains Unauthorized access to container image layers, configuration files, or secrets stored locally on the
1269 engine host due to weak encryption, insufficient file permissions, or misconfigured access to control sockets.

1270 This may include data in use (e.g. reading secrets from running container processes), data at rest (e.g. accessing local
1271 container image storage), and data in transit (e.g. intercepting traffic between containers using the engine's default
1272 network).

1273 **T-CE-AVAIL-DOS:** DoS via resource exhaustion in the container engine

1274 An attacker exploits the container engine's local API or socket to trigger excessive image pulls, container starts/stops, or
1275 logging operations, consuming CPU, memory, disk I/O, or network bandwidth on the host.

1276 **T-CE-LOG-INT:** Insufficient logging or log tampering in the Container Engine

1277 An attacker with access to the container engine API or control socket manipulates lifecycle logs (start/stop events,
1278 image pulls, error messages) to remove or alter records of their actions. Since the engine directly manages container
1279 execution, these logs are often key for local forensics.

1280 **T-CE-UPD-COMP:** Malicious or compromised update in the Container Engine

1281 An attacker injects or tampers with updates to the container engine binaries or API services. Since the engine directly
1282 interfaces with the host OS and launches containers, a compromised update can be used to execute arbitrary code with
1283 elevated privileges.

1284 **T-CE-CFG-COMP:** Unauthorized or insecure configuration in the Container Engine

1285 An attacker exploits insecure default settings (such as an exposed API socket without TLS/authentication, unrestricted
1286 default bridge networking, or permission to run privileged containers) or performs Unauthorized modifications to
1287 container engine daemon or runtime configurations (e.g. Docker daemon JSON, container runtime parameters). Such
1288 weaknesses can bypass container isolation, allow unsafe container operations, or provide a direct path to escalate from
1289 container workloads to the host operating system. Because the container engine mediates between containers, the
1290 container runtime stack (CRS), and the host OS, insecure or Unauthorized configurations can undermine both workload
1291 security and host-level protections. If configuration changes originate from the orchestrator or external automation, the
1292 container engine shall only accept configuration updates from authenticated and authorized sources to prevent
1293 unintended cluster-wide propagation of insecure settings.

1294 **T-CE-SC-COMP:** Supply chain compromise in a Container Engine

1295 A dependency used by the CE, such as a registry client library, image build tool, or network driver, is compromised
1296 during development or distribution.

1297 This malicious code can be triggered when the CE pulls or builds images, or when it sets up container networking.

1298 4.4.3.3 Container Runtime System

1299 **T-CRS-INTF-ESC:** Runtime-level container breakout

1300 A malicious container exploits flaws in OS-level isolation mechanisms or runtime implementation to escape its sandbox
1301 and interfere directly with the host or peer containers.

1302 **T-CRS-INTEG-COMP:** Runtime-level tampering and unverified trust

1303 An attacker modifies runtime binaries, configuration, or container state data (namespaces, cgroups, mounts) to disable
1304 or bypass security controls. Because the orchestrator and engine may not verify the runtime's state, a compromised
1305 runtime can operate undetected.

1306 **T-CRS-UNAUTH-ADMIN:** Unauthorized access to container runtime control mechanisms

1307 An attacker directly interacts with the container runtime binary or its control interfaces, bypassing higher-level engine
1308 or orchestrator authentication. This may occur if the runtime is invoked manually or through a compromised local
1309 account.

1310 **T-CRS-UNAUTH-FUNC:** Unauthorized privileged operations or data access in the container runtime system

1311 An attacker with authenticated local or delegated runtime access exploits insufficient authorization to manipulate
1312 running containers or modify runtime configurations directly.

1313 **T-CRS-DATA-DISC:** Unauthorized disclosure of sensitive data via the container runtime system

1314 An attacker exploits runtime-level weaknesses to read sensitive data from running containers or host resources that
1315 should be isolated.

1316 This may include data in use (e.g. reading contents from container or host memory), data at rest (e.g. accessing mounted
1317 volumes or container filesystem layers), and data in transit (e.g. intercepting inter-container or container-host
1318 communications).

1319 **T-CRS-AVAIL-DOS:** DoS via resource exhaustion in the container runtime system

1320 An attacker running inside a container exploits runtime isolation weaknesses to monopolize CPU, memory, or I/O
1321 resources, or floods inter-container or container-to-host communication channels to degrade performance.

1322 **T-CRS-LOG-INT:** Insufficient logging or log tampering in the Container Runtime System

1323 An attacker operating inside a container exploits insufficient runtime logging or insecure log handling to conceal
1324 actions such as privilege escalation attempts, restricted syscalls, or file system access violations. The runtime's own log,
1325 if tampered with, may give a false picture of container activity.

1326 **T-CRS-UPD-COMP:** Malicious or compromised update in the Container Runtime System

1327 An attacker injects, tampers with, or replays updates to the low-level runtime responsible for creating and managing
1328 containers. Since the runtime enforces isolation (namespaces, cgroups, seccomp), a compromised update could remove
1329 or weaken these controls.

1330 **T-CRS-CFG-COMP:** Unauthorized or insecure configuration in the Container Runtime System

1331 An attacker exploits insecure defaults (such as lack of enforced seccomp/AppArmor profiles, missing cgroup limits for
1332 CPU, memory, or I/O, or permissive filesystem mounts that expose sensitive host paths) or introduces Unauthorized
1333 changes to runtime-level parameters and configuration files. Such weaknesses undermine container isolation, disable
1334 syscall filtering, weaken resource constraints, or allow containers to access sensitive host resources. Because the
1335 container runtime system directly enforces workload isolation and resource governance, insecure or Unauthorized
1336 configurations at this layer can enable container escapes, denial-of-service attacks, or privilege escalation against the
1337 host. Where runtime configuration is applied automatically from a container engine or orchestrator, the runtime shall
1338 ensure that such configuration inputs are authenticated and authorized before application, to prevent malicious or
1339 unintended configuration propagation.

1340 **T-CRS-SC-COMP:** Supply chain compromise in a Container Runtime System

1341 An attacker compromises a low-level runtime dependency, such as a system call handling library, sandboxing module,
1342 or container image unpacker, used by the CRS.

1343 The malicious code executes whenever the CRS starts or runs containers, potentially enabling container escape.

1344 4.4.4 Hardware-level common threat

1345 **T-ALL-HW-ACCESS:** Compromise via hardware-level or platform trust components

1346 An attacker with access to hardware-level capabilities (for example compromised CPU, firmware, hardware security
1347 modules, or physical access to deployed systems) may compromise the hypervisor, container runtime system, or
1348 associated platform components below the software control plane. Such attacks may exploit manufacturer backdoors in
1349 processors or firmware, supply chain implants in platform components, or physical tampering vectors.

1350 Because hardware-level compromise can operate at privilege levels that exceed the visibility and control of software-
1351 based protection mechanisms, these attacks may bypass or undermine boot integrity verification, runtime integrity
1352 checks, attestation, and isolation mechanisms defined in the present document.

1353 The present document does not require products to fully prevent all forms of hardware-level compromise. Instead, for
1354 requirements that rely on platform-anchored, hardware-assisted, or externally protected trust mechanisms,
1355 manufacturers are expected to:

- 1356 • use available platform trust anchors and protection mechanisms where appropriate;
- 1357 • provide software-based or hybrid protections where feasible to achieve equivalent security outcomes; and
- 1358 • document trust assumptions and residual risks associated with hardware components, including the potential
1359 impact of compromise consistent with T-ALL-HW-ACCESS.

1360 4.5 Security Objectives

1361 4.5.1 Overview

1362 This clause sets out the common security objectives for all product categories. These objectives provide the foundation
1363 and are later refined into specific requirements for VES and CES in the subsequent clauses.

1364 4.5.2 Isolation

1365 The product shall enforce isolation between workloads, administrative functions, and network domains to prevent
1366 Unauthorized access, interference, or leakage across logical boundaries. Isolation shall be enforced by default, with
1367 configuration options available to adjust isolation levels in line with assessed risk and operational requirements.

1368 4.5.3 Integrity Protection

1369 The product shall ensure the integrity of platform components, configuration, and workloads by preventing
1370 Unauthorized modification and by detecting tampering at rest and at runtime.

1371 4.5.4 Authentication

1372 The product shall ensure that all access to administrative interfaces, control functions, or sensitive operations is subject
1373 to strong authentication of users, services, or components.

1374 4.5.5 Authorization

1375 The product shall enforce access control to ensure that only authorized users, services, or components can invoke
1376 functions or access data based on their assigned privileges.

1377 4.5.6 Confidentiality Protection

1378 The product shall ensure the confidentiality of sensitive data in transit, preventing Unauthorized disclosure.

1379 The product shall protect sensitive data at rest under its direct control, preventing its Unauthorized disclosure.

1380 The product shall support the use of host-level or external data-at-rest confidentiality controls for sensitive data stored
1381 on underlying operating systems or external storage accessed by the product.

1382 The product shall support the use of host-level or external secure deletion mechanisms for sensitive data stored on
1383 underlying operating systems or external storage accessed by the product, to ensure the data cannot be recovered after
1384 its intended use is complete.

1385 The product shall protect sensitive data in use under its direct control, preventing its Unauthorized disclosure.

1386 4.5.7 Availability and Resilience

1387 The product shall ensure its own availability and resilience against denial-of-service attacks, resource exhaustion, and
1388 other attempts to degrade or disrupt its essential functions.

1389 4.5.8 Logging

1390 The product shall generate security-relevant logs, including access attempts, configuration changes, and workload
1391 actions, and ensure those logs are protected from tampering.

1392 4.5.9 Patches and Updates

1393 The product shall support secure update mechanisms, including cryptographic validation of update packages and
1394 protection against rollback. It shall also allow for reversion to a previous verified secure state.

1395 4.5.10 Secure Configuration and Default

1396 The product shall provide mechanisms for secure configuration, allowing administrators to implement and maintain a
1397 secure operating posture, and shall protect these configurations from Unauthorized modification.

1398 The product shall be delivered with secure default settings. Only services essential to its core functionality shall be
1399 enabled by default. Basic protection mechanisms shall be automatically active upon installation, requiring no additional
1400 action from the user.

1401 4.5.11 Data Minimization

1402 The product shall ensure that only data strictly necessary for its secure and intended operation is collected, processed,
1403 and stored. This includes limiting logs, telemetry, and diagnostic information to essential elements, applying
1404 proportionate retention, and avoiding unnecessary exposure of sensitive information.

1405 4.5.12 Time Synchronization

1406 The product shall ensure accurate and reliable time synchronization for both its internal operations and security-related
1407 functions.

1408 4.6 Operational Environment Objectives

1409 4.6.1 General Principles

1410 Based on the out-of-scope elements identified for both the VES and the CES in clause 4.1, the Operational Environment
1411 (OE) comprises components that are not part of the VES/CES but are essential to maintain the security of the in-scope
1412 components.

1413 The manufacturer's responsibility under the present document is limited to implementing the security requirements
1414 applicable to the in-scope components defined in clause 4.1.

1415 However, for the VES/CES to operate securely and as intended, the OE components should function in a manner that
1416 supports and preserves the security of the product.

1417 The following security objectives for the Operational Environment specify the expected security properties of each OE
1418 component.

1419 Ensuring that these objectives are met is the responsibility of the product or system integrator.

1420 Compliance of the OE components with the CRA for example using the relevant harmonised standards, is considered a
1421 precondition for secure operation.

1422 4.6.2 Security Objectives for the Operational Environment with 1423 Requirement Traceability

1424 1. Boot Manager

1425 • The OE should verify the integrity and authenticity of all firmware and boot components before transferring
1426 control to the hypervisor or CRS (REQ-H-B-INT-001/002/003/004, REQ-CRS-B-INT-001/002/003/004).

1427 • The OE should enforce a chain of trust (REQ-H-B-INT-003/004, REQ-CRS-B-INT-003/004, REQ-H-RA-
1428 INT-003/004, REQ-CRS-RA-INT-003/004).

1429 • The OE should block boot if signature or hash verification fails (REQ-H-B-INT-001/002, REQ-CRS-B-INT-
1430 001/002).

1431 2. Host Operating System

1432 • The OE should enforce strict process, memory, and I/O isolation between all processes (REQ-H-VM-ISO-001,
1433 REQ-CRS-CN-ISO-001/002).

1434 • The OE should support software-assisted isolation (REQ-H-VM-ISO-003, REQ-CRS-CN-ISO-003).

1435 • The OE should provide syscall filtering, namespace isolation, and capability bounding to reduce kernel attack
1436 surface (REQ-CRS-CN-ISO-002, REQ-H-VM-ISO-001).

1437 • The OE should expose a secure entropy source to the CRS and Hypervisor (REQ-H-CONF-001/002/003,
1438 REQ-CRS-CONF-001/002/003).

1439 3. Virtual or Physical Network Infrastructure (including SDN)

- 1440 • The OE should segregate management, control, and data traffic using logical or physical separation (REQ-H-
1441 NP-ISO-001/002/003, REQ-CRS-NP-ISO-001/002/003).
- 1442 • The OE should protect management channels with encryption and mutual authentication (REQ-H-NP-ISO-
1443 003, REQ-CRS-NP-ISO-003).
- 1444 • The OE should enforce ACLs and rate limits to block Unauthorized cross-plane traffic and DoS attempts
1445 (REQ-H-NP-ISO-002, REQ-CRS-NP-ISO-002).
- 1446 • The OE should prevent MAC/IP spoofing and enforce integrity of network identities (REQ-H-NP-ISO-002,
1447 REQ-CRS-NP-ISO-002).
- 1448 • The OE should provide accurate, authenticated time synchronization for distributed components.

1449 4. Identity and Access Management / PKI Services

- 1450 • The OE should issue, rotate, and revoke credentials for administrators and services (REQ-H-ADMIN-AUTH-
1451 003, REQ-CRS-ADMIN-AUTH-003).
- 1452 • The OE should validate certificates against a trusted chain and enforce revocation checks (REQ-H-SERV-
1453 AUTH-003, REQ-CRS-SERV-AUTH-003).
- 1454 • The OE should store private keys in protected hardware or secure software vaults (REQ-H-CONF-
1455 001/002/003, REQ-CRS-CONF-001/002/003).
- 1456 • The OE should record certificate lifecycle events for auditing (REQ-H-LOG-003, REQ-CRS-LOG-003).

1457 5. Image Registry or Artifact Repository

- 1458 • The OE should verify digital signatures of all stored or pulled images (REQ-H-IMG-INT-001/002/003/004,
1459 REQ-CRS-IMG-INT-001/002/003/004).
- 1460 • The OE should restrict registry access to authenticated and authorized users (REQ-H-SERV-AUTHZ-001/002,
1461 REQ-CRS-SERV-AUTHZ-001/002).

1462 6. External Logging and SIEM Systems

- 1463 • The OE should collect and store logs through authenticated and encrypted channels (REQ-H-LOG-002,
1464 REQ-CRS-LOG-002).
- 1465 • The OE should preserve log integrity using signing or hashing mechanisms (REQ-H-LOG-003, REQ-CRS-
1466 LOG-003).
- 1467 • The OE should restrict log access to authorized entities (REQ-H-CONF-001, REQ-CRS-CONF-001).
- 1468 • The OE should correlate and alert on critical security events (REQ-H-LOG-001, REQ-CRS-LOG-001).
- 1469 • The OE should synchronize event timestamps with a trusted time source.

1470 7. External Attestation System

- 1471 • The OE should validate attestation evidence and verify trust measurements (REQ-H-RA-INT-003/004,
1472 REQ-CRS-RA-INT-003/004).
- 1473 • The OE should protect verifier credentials and trust anchors from compromise (REQ-H-CONF-001/002/003,
1474 REQ-CRS-CONF-001/002/003).
- 1475 • The OE should detect replayed or stale attestation data (REQ-H-RA-INT-003/004, REQ-CRS-RA-INT-
1476 003/004).
- 1477 • The OE should maintain logs of attestation requests and results (REQ-H-LOG-002, REQ-CRS-LOG-002).

1478 8. Trust Anchors

- 1479 • The OE should generate and protect cryptographic keys (REQ-H-B-INT-003/004, REQ-CRS-B-INT-003/004,
1480 REQ-H-RA-INT-003/004, REQ-CRS-RA-INT-003/004).
- 1481 • The OE should expose secure APIs for key usage and measurement retrieval (REQ-H-CONF-004/005,
1482 REQ-CRS-CONF-004/005).
- 1483 • The OE should provide measurement and reporting for trusted boot sequences (REQ-H-B-INT-003/004,
1484 REQ-CRS-B-INT-003/004).

1485 9. Firewall / Intrusion Detection or Prevention System

- 1486 • The OE should restrict external access to management and control interfaces (REQ-H-CP-ISO-001/002,
1487 REQ-CRS-CP-ISO-001/002).
- 1488 • The OE should enforce segmentation between management, control, and data planes (REQ-H-NP-ISO-
1489 001/002, REQ-CRS-NP-ISO-001/002).
- 1490 • The OE should detect and block anomalous or malicious network traffic (REQ-H-NP-ISO-002, REQ-CRS-
1491 NP-ISO-002).
- 1492 • The OE should log all blocked events (REQ-H-LOG-001, REQ-CRS-LOG-001).

1493 10. Storage Systems

- 1494 • The OE should encrypt data at rest including VM or container images, configuration files, and logs (REQ-H-
1495 CONF-002/004/005, REQ-CRS-CONF-002/004/005).
- 1496 • The OE should protect encryption keys in secure key management systems (REQ-H-CONF-
1497 001/002/003/004/005, REQ-CRS-CONF-001/002/003/004/005).
- 1498 • The OE should securely erase deleted data to prevent recovery (REQ-H-DM-001, REQ-CRS-DM-001).

1499 4.6.3 Reference to Relevant CRA Harmonised Standards

1500 Compliance of the Operational Environment components with the CRA essential requirements may be presumed when
1501 the corresponding harmonised standards listed below are applied.

1502 Where no harmonised standard yet exists, the system or product integrator should ensure equivalent security measures.

1503 **Table 4.6.3-1: Relevant CRA harmonised standards for OE components**

OE Component	Relevant CRA Reference
Boot Manager	ETSI EN 304 623 [i.4] (CRA for Boot Manager)
Host Operating System	ETSI EN 304 626 [i.5] (CRA for Operating System)
Network Infrastructure (Virtual / Physical / SDN)	ETSI EN 304 625 [i.6] (CRA Network Interfaces)
Identity & Access Management / PKI	ETSI EN 304 624 [i.7] (CRA PKI)
Image Registry / Artifact Repository	NA
External Logging / SIEM	ETSI EN 304 622 [i.8] (CRA SIEM)
External Attestation System	NA
Trust Anchors	CEN xxx xxx (CRA Security Modules)
Firewall / Intrusion Prevention System	ETSI EN 304 636 [i.9] (CRA Network Security Appliances)
Storage Systems	NA

1504

1505 4.7 Risk-Based Requirement Classes/Categorization and 1506 Application

1507 4.7.1 Requirement Classes, Categorization and Security Category Levels

1508 4.7.1.1 Definition of Requirement Classes: Basic, Elevated, Advanced

1509 To enable proportional application of security requirements under the CRA, requirements in the present document are
1510 expressed with requirement Classes: Basic, Elevated and Advanced. These levels indicate the depth or strength of the
1511 requirement.

1512 **Basic (B)**

- 1513 • Definition: The minimum class of requirement, providing fundamental protection.
- 1514 • Characteristics:
 - 1515 - Typically software-based or configuration safeguards.
 - 1516 - Establishes a baseline protection expected in all products.
 - 1517 - Suitable for low-risk environments.

1518 **Elevated (E)**

- 1519 • Definition: Intermediate class of protection, addressing increased exposure.
- 1520 • Characteristics:
 - 1521 - Builds upon or replaces Basic with stronger mechanisms.
 - 1522 - Leverages hardware-assisted features or stronger cryptographic protections.
 - 1523 - Suitable for medium-risk environments.

1524 **Advanced (A)**

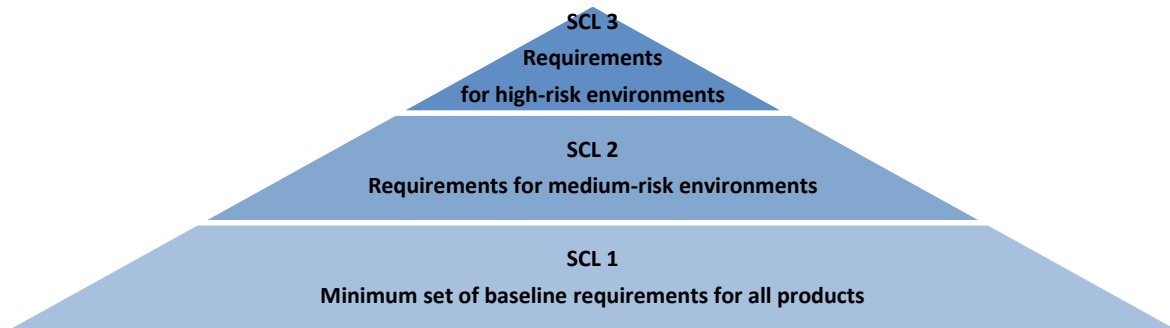
- 1525 • Definition: The highest class of protection, incorporating the strongest available safeguards.
- 1526 • Characteristics:
 - 1527 - Provides resilience against advanced threats.
 - 1528 - Incorporates hardware roots of trust and runtime protections.
 - 1529 - Suitable for high-risk environments.

1530 4.7.1.2 Security Category Level Definitions

1531 To support risk-proportional application of security requirements under the CRA, the present document defines three
1532 Security Category Levels (SCL) for security requirements. The defined SCLs support manufacturers in applying a risk-
1533 based approach, helping to determine applicable requirements aligned with the intended use case and proportionality
1534 principles under the CRA.

Table 4.7.1.2-1: Security category level definitions

Level	Description
SCL 1	A minimum set of baseline cybersecurity requirements applicable to all products. These requirements are essential for fundamental security posture, particularly for use cases where compromise results in low risk (e.g. UC-V1, UC-C1). This level corresponds to products assessed as Low Risk according to the methodology in clause B.
SCL 2	Requirements for medium-risk environments. Depending on the requirement categories defined in clause 4.7.1.3: i) cumulative requirements include all Basic + Elevated elements; ii) exclusive requirements apply at the Elevated level only; iii) single-class requirements at Elevated are tied to this SCL. This corresponds to products assessed as Medium Risk in clause B.
SCL 3	Requirements for high-risk environments. Depending on the requirement categories defined in clause 4.7.1.3: i) cumulative requirements include Basic + Elevated + Advanced; ii) exclusive requirements apply at the Advanced level only; iii) single-class requirements at Advanced are tied to this SCL. This corresponds to products assessed as High Risk in clause B.

**Figure 4.7.1.2-1: Security category level definitions**

4.7.1.3 Requirement Categorization

The requirements in the present document are defined using three categories: Cumulative, Exclusive, and Single-Class. Each category specifies how a requirement applies across the different classes (Basic, Elevated, Advanced). The following subclauses describe these categories, providing guidance to implementers on how to apply them.

(1) Cumulative Requirements (Additive)

- Definition: Higher classes include all lower classes. Each class adds more rigor on top of the previous one.
- Rationale: This model applies when protections build in layers and depend on the foundation provided by lower classes.
- Application: If a manufacturer is assigned SCL 3 (Advanced), then all requirements at Basic, Elevated, and Advanced shall be implemented.
- Example - VM Isolation:
 - Basic: CPU and memory isolation using software-based controls.
 - Elevated: Basic + hardware-assisted Virtualization and IOMMU separation.
 - Advanced: Elevated + mitigations for side-channel attacks.

(2) Exclusive Requirements (Non-Additive)

- Definition: Only one class applies. Classes represent alternative approaches, not cumulative layers.
- Rationale: This model applies when requirements define mutually exclusive mechanisms, where implementing one makes the others unnecessary.

- 1557 • Application: The manufacturer implements only the requirement at the class corresponding to the product's
1558 assigned SCL.
- 1559 • Example - Image Integrity Verification:
- 1560 - Basic: Hash verification of VM/container images before execution.
- 1561 - Elevated: Signature verification of images against a trusted public key.
- 1562 - Advanced: Signature verification with keys stored in a hardware root of trust.

1563 (3) Single-Class Requirements

- 1564 • Definition: A requirement is defined at only one class (Basic, Elevated, or Advanced). No variations exist at
1565 the other classes.
- 1566 • Rationale: Some controls cannot be meaningfully scaled across classes and are best expressed at a single
1567 strength level.
- 1568 • Application: A Single-Class requirement is tied to the SCL corresponding to its class:
- 1569 - Basic: SCL 1.
- 1570 - Elevated: SCL 2.
- 1571 - Advanced: SCL 3.
- 1572 The manufacturer implements the requirement only when the product is assigned to the SCL that matches the
1573 class.
- 1574 • Example:
- 1575 - Advanced-only (SCL3): Remote attestation of the hypervisor's runtime state.

1576 4.7.1.4 Guidance for Manufacturers

1577 To identify applicable requirements:

- 1578 1) Determine product use case and risk level (see clauses 4.3 and B).
- 1579 2) Identify requirement categorization:
- 1580 - If cumulative, implement all lower classes up to the assigned SCL.
- 1581 - If exclusive, implement only the class that matches the assigned SCL.
- 1582 - If single-class, implement the one requirement defined at that class.

1583 4.7.2 Requirement Application

1584 4.7.2.1 Approaches for determining security requirements

1585 The present document defines two approaches to identify the applicable security requirements. Both are grounded in
1586 risk assessment and aligned with the CRA [i.1] proportionality principle.

- 1587 • The SCL-based approach shall be applied where the product aligns with one of the representative use cases
1588 defined in clause 4.3.
- 1589 • The Custom Security Profile Development approach shall be used when representative use cases or clause B
1590 risk levels do not apply, with requirements determined and justified through a documented risk analysis.

1591 4.7.2.2 SCL-Based Requirement Application

1592 This approach provides manufacturers with predefined use cases mapped to Security Category Levels (SCLs) (see
1593 clause 4.7.1.2). Each SCL bundles the appropriate requirements for the assessed risk level (Low, Medium, High).

1594 The representative use cases defined in clause 4.3 are reference cases to support risk-based alignment of requirements.
1595 They do not exhaustively cover all deployment scenarios.

1596 Manufacturers may define alternative or product-specific use cases, provided that:

- 1597 • A risk assessment is conducted (see guidance in clause B) to clearly document the specific risk characteristics,
1598 threat exposure, and potential impacts of the product's environment.
- 1599 • The assigned SCL is justified based on the outcome of the risk assessment, considering the product's exposure,
1600 complexity, and intended purpose.
- 1601 • Documentation justifying the risk assessment and assigned SCL is maintained for conformity assessment
1602 purposes.

1603 Step 1 - Determine product use case and associated risk level

1604 Manufacturers shall begin by identifying the use case for their product (see reference cases in clause 4.3). The risk level
1605 for these use cases is pre-determined based on the application of the methodology in clause B:

- 1606 • If the product aligns with UC-V1 or UC-C1 → corresponds to a Low-risk use case.
- 1607 • If the product aligns with UC-V2 or UC-C2 → corresponds to a Medium-risk use case.
- 1608 • If the product aligns with UC-V3 or UC-C3 → corresponds to a High-risk use case.

1609 Manufacturers may also define their own use case. In such instances, the cybersecurity risk assessment methodology
1610 from clause B could be applied to determine the product's specific risk level (Low, Medium, or High), with the risk
1611 characteristics and threat exposure fully documented.

1612 Step 2 - Map use case and determined risk level to SCL and apply requirements

1613 **Table 4.7.2.1-1: Security category level definitions**

Use Case	Risk Level	SCL 1	SCL 2	SCL 3	Applicable requirements
UC-V1 / UC-C1	Low	x			Implement all requirements marked Basic, including cumulative, exclusive, and single-class Basic.
UC-V2 / UC-C2	Medium		x		Implement all requirements marked Basic and Elevated, including cumulative and exclusive Elevated, and single-class Elevated.
UC-V3 / UC-C3	High			x	Implement all requirements marked Basic, Elevated, and Advanced, including cumulative and exclusive Advanced, and single-class Advanced.

1614

1615 4.7.2.3 Requirement Application through Custom Security Profile Development

1616 Manufacturers shall apply the Custom Security Profile Development approach when their product use case does not
1617 clearly align with any representative use case, when the deployment context results in different risk levels than those
1618 assumed in clause B, or when the manufacturer applies its own documented risk assessment methodology. In this
1619 approach, the application of requirements shall be fully justified through a documented risk analysis.

1620 This approach requires manufacturers to derive a tailored security profile directly from the threats and risks identified
1621 for the product. Clause 5.4 provides a normative threat-to-requirement mapping, with requirements categorized as Basic
1622 (B), Elevated (E), and Advanced (A), and further classified as Cumulative, Exclusive, or Single-Class. Manufacturers
1623 shall apply this mapping when tailoring requirements to their specific product risk assessment.

1624 Step 1 - Assessment of an individual risk level for each identified threat

- 1625 • The manufacturer shall identify all applicable threats to the product using the threat catalogue in clause 4.4.

1626 • For each identified threat, the manufacturer shall determine likelihood and impact to calculate the associated
 1627 risk level, either by applying the methodology in clause B or by using an equivalent documented risk
 1628 methodology.

1629 • The outcome of this step shall be a complete threat list, each associated with a risk rating.

1630 **Step 2 - Application of requirements based on the assessed risk level of each individual threat**

1631 • For each identified threat, the manufacturer shall determine the corresponding requirement(s) from the
 1632 normative mapping in clause 5.4.

1633 • The categorization rules shall apply:

1634 - Cumulative: all requirements up to the level corresponding to the assessed risk (e.g. Basic + Elevated for
 1635 Medium risk).

1636 - Exclusive: only the requirement at the level corresponding to the assessed risk (e.g. Elevated requirement
 1637 for Medium risk, without inheriting Basic).

1638 - Single-Class: the requirement applies only at the class where it is defined (e.g. an Elevated-only
 1639 requirement applies exclusively for Medium risk).

1640 **Step 3 - Documentation and conformity evidence**

1641 • The manufacturer shall document, for each identified threat:

1642 a) the assessed risk level;

1643 b) the applicable requirement(s) from clause 5.4, including their class and categorization; and

1644 c) a justification that the applicable requirement(s) reduce the risk to an acceptable level.

1645 • The resulting custom security profile and supporting documentation shall form part of the manufacturer's
 1646 technical file and shall be made available for conformity assessment under the CRA.

1647 This process shall produce a custom security profile that is fully risk-based, tailored to the product's context, and
 1648 compliant with the normative threat-to-requirement mapping defined in clause 5.4.

1649 **4.8 Roles and Responsibilities**

1650 **Manufacturer**

1651 The organization responsible for design, development, and delivery of the product, including the implementation of its
 1652 security functions. The manufacturer provides the product documentation, security claims, and evidence required for
 1653 assessment.

1654 **Product Integrator**

1655 The organization responsible for deploying, configuring, or combining the product into a larger system or operational
 1656 environment. The product integrator ensures that:

1657 • Deployment meets the manufacturer's documented security configuration requirements.

1658 • Required platform capabilities, hardware features, and external trust anchors are present.

1659 • The product is operated in accordance with its secure deployment guidance.

1660

5 Requirements

1661

5.1 VES Security Requirements

1662

5.1.0 General

1663

This clause defines specific security requirements for VESs, instantiating the security objectives outlined in clause 4.5.

1664

5.1.1 Hypervisor Requirements

1665

5.1.1.1 Isolation

1666

5.1.1.1.0 General

1667

1668

This clause defines isolation requirements across three dimensions: (a) isolation between virtual machines, (b) isolation of hypervisor control/administrative planes, and (c) separation of network planes (management, guest, host).

1669

5.1.1.1.1 VM Isolation

1670

Classes are cumulative

1671

Basic & Elevated

1672

1673

1674

REQ-H-VM-ISO-001: The Hypervisor shall enforce isolation between virtual machines (workloads) at CPU, memory, I/O, and interrupt-handling levels, ensuring that one virtual machine cannot read, write, or otherwise interfere with another virtual machine's execution, data, or configuration state.

1675

Advanced

1676

1677

REQ-H-VM-ISO-003: The Hypervisor shall strengthen isolation with mitigations against known side-channel attacks to ensure robust separation between workloads sharing hardware resources.

1678

5.1.1.1.2 Control Plane Isolation

1679

Classes are cumulative

1680

Basic

1681

1682

1683

1684

REQ-H-CP-ISO-001: The Hypervisor shall provide access-control mechanisms for its administrative functions and interfaces. These mechanisms shall support configurations in which guest virtual machines are prevented from directly accessing or invoking administrative functions and interfaces, and in which only authenticated and authorized entities can perform management actions.

1685

Elevated

1686

1687

1688

1689

REQ-H-CP-ISO-002: The Hypervisor shall support configuration of dedicated logical separation between administrative traffic and guest or data-plane traffic. The Hypervisor shall support the use of distinct communication channels, ports, or virtual networks for administrative interfaces, enabling isolation of administrative traffic from guest workloads.

1690

Advanced

1691

1692

1693

REQ-H-CP-ISO-003: The Hypervisor shall support the use of cryptographically protected and mutually authenticated communication channels for all administrative interfaces, in order to reduce the risk of Unauthorized access or lateral movement across tenants or system domains.

1694 5.1.1.1.3 Network Plane Separation

1695 **Classes are cumulative**

1696 **Basic**

1697 REQ-H-NP-ISO-001: The Hypervisor shall maintain logical segregation between the management, guest, and host
1698 network planes. Virtual switching, routing, or equivalent network mechanisms shall ensure that traffic from one plane
1699 cannot traverse or address another plane by default. These mechanisms shall support explicit configuration of separate
1700 network contexts for each plane.

1701 **Elevated**

1702 REQ-H-NP-ISO-002: The Hypervisor shall support configuration of explicit traffic filtering and access-control rules
1703 between network planes to prevent Unauthorized or unintended cross-plane communication. These rules shall include
1704 filtering of packets at the Hypervisor's virtual network layer or at other enforcement points under the Hypervisor's
1705 control, ensuring that only management or orchestration traffic explicitly permitted by the configured policy is allowed
1706 to cross plane boundaries.

1707 **Advanced**

1708 REQ-H-NP-ISO-003: The Hypervisor shall support isolation of management traffic using dedicated physical interfaces
1709 or cryptographically protected network channels, preventing interception or modification by guest or host workloads.
1710 Where dedicated physical interfaces are not used, the cryptographic protection shall provide confidentiality, integrity,
1711 and authentication equivalent to a dedicated management network.

1712 5.1.1.2 Integrity Protection

1713 5.1.1.2.1 Boot chain integrity verification

1714 **Classes are exclusive**

1715 **Basic**

1716 REQ-H-B-INT-001: The Hypervisor shall implement an integrity verification mechanism for its executable core (kernel
1717 or microkernel) based on a trusted reference value.

1718 The Hypervisor shall invoke this integrity verification before initializing any guest workloads or exposing management
1719 interfaces. If the integrity verification of the executable core fails, the Hypervisor shall prevent initialization of guest
1720 workloads and activation of management interfaces.

1721 **Elevated**

1722 REQ-H-B-INT-002: The Hypervisor shall implement mechanisms to participate in a verifiable chain of trust for the
1723 early boot components that load the Hypervisor. Each boot stage delivered as part of the Hypervisor product, including
1724 at least any bootloader delivered with the Hypervisor and the Hypervisor executable core, shall validate the integrity
1725 and authenticity of the next stage before transferring control.

1726 If this validation fails, the Hypervisor shall prevent initialization of guest workloads and activation of management
1727 interfaces.

1728 NOTE 1: For the purposes of REQ-H-B-INT-002, "participate in a verifiable chain of trust" means that the
1729 Hypervisor components involved in the boot process are capable of integrating into a platform-provided
1730 secure boot chain and enforce integrity and authenticity verification of subsequent stages.

1731 **Advanced**

1732 REQ-H-B-INT-003: The Hypervisor shall rely on integrity and authenticity information for the boot chain on which it
1733 depends, as provided by verified boot mechanisms using hardware or software roots of trust. This information shall
1734 cover components used to load the Hypervisor, including the bootloader, the Hypervisor kernel or microkernel, and
1735 management services. The Hypervisor shall prevent initialization of guest workloads and management services if the
1736 verification status of any component in this boot chain cannot be established or is reported as failed.

1737 NOTE 2: Hardware- or software-based roots of trust, such as verified boot mechanisms or dedicated trust anchors,
1738 may be used to implement the validation required by REQ-H-B-INT-003.

1739 REQ-H-B-INT-004: The Hypervisor may support the use of hardware-based roots of trust to perform boot-chain
1740 integrity and authenticity verification in accordance with REQ-H-B-INT-003. Such hardware-based roots of trust
1741 include secure elements, dedicated trust anchors, or equivalent components providing protection against Unauthorized
1742 modification of the boot chain.

1743 5.1.1.2.2 Guest VM image integrity verification

1744 **Classes are exclusive**

1745 **Basic**

1746 REQ-H-IMG-INT-001: The Hypervisor shall implement an integrity verification mechanism for guest VM images
1747 based on a trusted reference value.

1748 The Hypervisor shall invoke this integrity verification automatically before executing or provisioning any guest VM. If
1749 the integrity verification of a guest VM image fails, the Hypervisor shall prevent execution or instantiation of that guest
1750 VM.

1751 **Elevated**

1752 REQ-H-IMG-INT-002: The Hypervisor shall implement mechanisms to verify both the integrity and the authenticity of
1753 guest VM images before execution or provisioning. This verification shall establish that the image originates from a
1754 trusted source and has not been altered since it was approved for deployment.

1755 The Hypervisor shall prevent execution or instantiation of any guest VM image whose integrity or authenticity cannot
1756 be successfully verified.

1757 NOTE 1: Authenticity verification may be implemented using public-key digital signatures validated through
1758 certificate chains rooted in trusted authorities, or other cryptographic schemes that provide a verifiable
1759 binding between the guest VM image and a trusted source and that prevent undetected modification of the
1760 image, provided that equivalent assurance is achieved.

1761 **Advanced**

1762 REQ-H-IMG-INT-003: The Hypervisor shall use integrity and authenticity verification for all guest VM images before
1763 execution or provisioning, based on trust anchors stored as protected trust material. The protected trust material shall
1764 not be modifiable during normal operational state.

1765 If verification of the integrity or authenticity of a guest VM image using the protected trust material fails, or if the
1766 verification status of the image cannot be established, the Hypervisor shall prevent execution or instantiation of that
1767 guest VM.

1768 NOTE 2: Protected trust material refers to keys, certificates or other trust anchors that are stored and managed so
1769 that they cannot be altered through Unauthorized administrative or guest-facing interfaces and can only
1770 be updated through a controlled, explicitly authorized security management process.

1771 REQ-H-IMG-INT-004: The Hypervisor may support anchoring the integrity and authenticity verification of guest VM
1772 images in hardware-based roots of trust. Hardware-based roots of trust include dedicated secure hardware modules or
1773 trust anchors that provide protected storage for trust material and resistance against Unauthorized modification of
1774 verification keys.

1775 5.1.1.2.3 Runtime integrity protection

1776 **Classes are exclusive**

1777 **Elevated**

1778 REQ-H-RP-INT-002: The Hypervisor shall enforce integrity protection for its configuration and control plane by
1779 ensuring that only authenticated and authorized modifications can be applied. The Hypervisor shall generate audit
1780 events for any security-relevant changes to configuration or internal control logic.

1781 **Advanced**

1782 REQ-H-RP-INT-003: The Hypervisor shall maintain the integrity of its security-critical runtime components and detect
 1783 Unauthorized modification of configuration state or control logic used to enforce security policies or isolation.

1784 The Hypervisor shall maintain a trusted baseline of the integrity state of these security-critical runtime components and
 1785 shall perform periodic or continuous integrity validation against this baseline.

1786 Upon detecting a deviation from the trusted baseline, the Hypervisor shall automatically trigger protective actions, such
 1787 as isolating the affected component, disabling affected functions, or restoring the component from a trusted state, and
 1788 shall generate a security alert that cannot be suppressed by the affected component.

1789 NOTE: Runtime integrity validation may be implemented using hardware-assisted mechanisms, software-based
 1790 mechanisms, or a combination of both, provided that the mechanisms are able to detect Unauthorized
 1791 modification of the security-critical runtime components defined by this requirement.

1792 REQ-H-RP-INT-004: The Hypervisor may support the use of hardware-based platform mechanisms to strengthen
 1793 runtime integrity validation and detection of Unauthorized modification of security-critical runtime components. Such
 1794 mechanisms include dedicated secure hardware modules, hardware-supported memory protection, or equivalent
 1795 platform features that provide additional assurance of runtime integrity.

1796 **5.1.1.2.4 Remote attestation**1797 **Single Class**1798 **Advanced**

1799 REQ-H-RA-INT-003: The Hypervisor may support measured boot and generation of attestation evidence describing its
 1800 bootloader, kernel or microkernel, security-critical configuration, and essential runtime state.

1801 This attestation evidence shall be cryptographically verifiable by an authorized verifier and bound to the Hypervisor
 1802 instance for which the measurements were produced.

1803 The Hypervisor shall protect attestation keys and measurement data against Unauthorized access, disclosure, or replay.

1804 The attestation evidence should be limited to the minimum information necessary to verify the integrity and
 1805 configuration state of the Hypervisor.

1806 NOTE 1: This requirement covers the Hypervisor capability to collect measurements, bind them cryptographically,
 1807 and expose them as attestation evidence. The design and operation of local or remote verifiers, attestation
 1808 protocols, and relying parties is outside the scope of the present document.

1809 REQ-H-RA-INT-004: Where attestation evidence produced by the Hypervisor is intended to be used in remote
 1810 verification scenarios, the Hypervisor should support configuration of privacy-preserving attestation options that reduce
 1811 linkability between attestation events while preserving the ability of authorized verifiers to assess the integrity and
 1812 configuration state of the Hypervisor.

1813 NOTE 2: Privacy-preserving options may include, for example, use of pseudonymous identifiers, separation of
 1814 identity and integrity information, or mechanisms for controlling the frequency and granularity of
 1815 attestation.

1816 **5.1.1.3 Authentication**1817 **5.1.1.3.1 General**

1818 This clause distinguishes between authentication of administrative users and authentication of external services that
 1819 access Hypervisor management functions or data.

1820 **5.1.1.3.2 Administrative Authentication**1821 **Classes are exclusive**1822 **Basic**

1823 REQ-H-ADMIN-AUTH-001: The Hypervisor shall require authentication for access to all administrative interfaces,
 1824 including local command-line interfaces, remote application programming interfaces, and virtual machine management
 1825 consoles.

1826 The Hypervisor shall enforce the use of non-default passwords for administrative accounts and shall support
 1827 configurable password complexity requirements for administrative authentication credentials.

1828 **Elevated**

1829 REQ-H-ADMIN-AUTH-002: The Hypervisor shall require authenticated access to all administrative interfaces and
 1830 shall support cryptographic key-based authentication for administrative accounts. The Hypervisor shall allow
 1831 administrative password-based login to be disabled for administrative interfaces.

- 1832 • Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].
- 1833 • Private keys used for administrative authentication shall be stored and managed in a manner that prevents
 1834 Unauthorized extraction or reuse by guest workloads or other unprivileged components.

1835 **Advanced**

1836 REQ-H-ADMIN-AUTH-003: The Hypervisor shall require strong authentication for all administrative interfaces and
 1837 shall support certificate-based authentication for administrative accounts across command-line interfaces, application
 1838 programming interfaces, and management consoles.

1839 Certificates used for administrative authentication shall be validated against a trusted public key infrastructure (PKI),
 1840 including verification of certificate validity, expiration, and revocation status.

1841 5.1.1.3.3 Service Authentication

1842 Classes are exclusive

1843 **Basic**

1844 The Hypervisor shall require authentication for all external services, including orchestration systems, monitoring tools,
 1845 and backup agents, before granting access to management functions or data.

1846 The Hypervisor shall support the use of distinct credentials for each service account and shall not rely on built-in shared
 1847 or hard-coded service credentials.

1848 The Hypervisor shall implement an account lockout mechanism for service accounts after a configurable number of
 1849 consecutive failed authentication attempts.

1850 **Elevated**

1851 REQ-H-SERV-AUTH-002: The Hypervisor shall require authenticated access for all external services and shall support
 1852 certificate-based mutual authentication for services accessing management functions or data.

- 1853 • Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].
- 1854 • Private keys used for service authentication shall be stored and managed in a manner that prevents
 1855 Unauthorized extraction or reuse by guest workloads or other unprivileged components.

1856 **Advanced**

1857 REQ-H-SERV-AUTH-003: The Hypervisor shall enforce certificate-based mutual authentication for all service
 1858 interactions that access management functions or data, with certificate paths validated against a trusted public key
 1859 infrastructure.

1860 Certificates used for service authentication shall be bound to specific service identities and shall be verified for validity,
 1861 expiration, and revocation status before access is granted.

1862 5.1.1.4 Authorization

1863 5.1.1.4.0 General

1864 This clause distinguishes between authorization of administrative users and authorization of external services that
1865 access Hypervisor management functions or data.

1866 5.1.1.4.1 Administrative Authorization

1867 **Classes are exclusive**

1868 **Basic**

1869 REQ-H-ADMIN-AUTHZ-001: The hypervisor shall restrict all administrative actions to authenticated accounts and
1870 enforce a default-deny policy for access to management functions.

1871 **Elevated**

1872 REQ-H-ADMIN-AUTHZ-002: The hypervisor shall enforce granular, role-based access control (RBAC) or attribute-
1873 based access control (ABAC) for administrative accounts, ensuring that accounts are granted only the minimum
1874 privileges required for their role.

1875 **Advanced**

1876 REQ-H-ADMIN-AUTHZ-003: The hypervisor shall enforce fine-grained authorization policies for administrative
1877 accounts, including separation of duties and support for time-bound or just-in-time privilege elevation, with the
1878 capability to integrate with external identity and policy decision systems where available.

1879 5.1.1.4.2 Service Authorization

1880 **Classes are exclusive**

1881 **Basic**

1882 REQ-H-SERV-AUTHZ-001: The hypervisor shall enforce a default-deny policy for all service accounts, ensuring that
1883 service accounts are only permitted to perform actions explicitly authorize.

1884 **Elevated**

1885 REQ-H-SERV-AUTHZ-002: The hypervisor shall enforce granular authorization for service accounts, binding allowed
1886 actions to their authenticated identity. Each service account shall be configured with least-privilege permissions.

1887 **Advanced**

1888 REQ-H-SERV-AUTHZ-003: The hypervisor shall enforce fine-grained, policy-driven authorization for service
1889 accounts, binding permissions to cryptographic service identities, supporting context-aware enforcement, with the
1890 capability to integrate with external authorization systems where available.

1891 5.1.1.5 Confidentiality Protection

1892 **Classes are cumulative**

1893 **Basic**

1894 REQ-H-CONF-001 (Access control for sensitive data): The Hypervisor shall provide access-control mechanisms to
1895 restrict access to sensitive data under its control, including guest VM images, snapshot data, configuration files,
1896 credentials, and security-relevant logs. Access to such sensitive data shall be permitted only to authenticated and
1897 authorized entities through controlled interfaces.

1898 **Elevated**

1899 REQ-H-CONF-002 (Data at rest): The Hypervisor shall support encryption of sensitive data at rest, including VM
1900 images, snapshot data, configuration files, credentials, and security-relevant logs.

1901 Encryption keys used for data-at-rest protection shall be stored and managed in a manner that prevents access by guest
1902 workloads and other unprivileged components.

1903 REQ-H-CONF-003 (Data in transit): The Hypervisor shall support the use of encrypted communication for
1904 management interfaces, for VM migration traffic, and for inter-VM traffic where such traffic is mediated by
1905 Hypervisor-controlled virtual networking components.

1906 When these communications are configured to use encrypted transport, the Hypervisor shall prevent fallback to
1907 unencrypted or weakly protected channels.

1908 **Advanced**

1909 REQ-H-CONF-004 (Data in use - confidential execution): The Hypervisor shall support provisioning and management
1910 of execution environments for guest workloads in which confidential data associated with those workloads is protected,
1911 during execution, from observation, inspection, or access through the Hypervisor, the host operating system, and
1912 administrative interfaces.

1913 On platforms that provide mechanisms for confidential execution, the Hypervisor shall be able to use these mechanisms
1914 to enforce the confidentiality of guest memory and associated processor state.

1915 REQ-H-CONF-005 (Hardware-backed confidentiality): The Hypervisor may support the use of hardware-based security
1916 mechanisms to strengthen protection of cryptographic keys and confidential execution state used for data-at-rest, data-
1917 in-transit, and data-in-use protection.

1918 NOTE: Such mechanisms include dedicated secure hardware modules, memory encryption features, or equivalent
1919 platform components that provide protected storage for keys and resistance against Unauthorized access
1920 to confidential workload data.

1921 **5.1.1.6 Availability and Resilience**

1922 Classes are cumulative

1923 **Basic**

1924 REQ-H-AVAIL-001: The Hypervisor shall detect failures of internal management and control-plane components and
1925 shall support recovery of these components without requiring a full host reboot, in order to maintain availability of
1926 running guest virtual machines.

1927 **Elevated**

1928 REQ-H-AVAIL-002: The Hypervisor shall provide mechanisms to support high availability of guest virtual machines,
1929 including automatic detection of host or Hypervisor service failure and automatic restart or migration of affected guest
1930 virtual machines to another available host, using native clustering capabilities or integration with external clustering or
1931 orchestration systems.

1932 REQ-H-AVAIL-003: The Hypervisor shall implement and enforce resource scheduling mechanisms that support
1933 configuration of minimum resource allocations for its security and management functions, covering at least CPU,
1934 memory, and I/O resources.

1935 These mechanisms shall protect the Hypervisor's control plane from resource exhaustion caused by guest workloads or
1936 external resource contention, ensuring that security and management functions remain able to operate under load.

1937 **Advanced**

1938 REQ-H-AVAIL-004: The Hypervisor shall support replication of the runtime state of selected guest virtual machines,
1939 including CPU and memory state and associated I/O buffers, to standby instances in order to enable failover with
1940 minimal interruption to workload execution in the event of host or Hypervisor service failure.

1941 **5.1.1.7 Logging**

1942 Classes are cumulative

1943 **Basic**

1944 REQ-H-LOG-001: The Hypervisor shall generate audit logs for security-relevant administrative actions and security
1945 events, including at least authentication attempts, configuration changes, and virtual machine lifecycle operations.

1946 **Elevated**

1947 REQ-H-LOG-002: The Hypervisor shall protect audit logs from Unauthorized access, modification, or deletion. Access
1948 to audit logs shall be restricted to authorized administrative roles or processes.

1949 The Hypervisor shall support secure export of audit logs to external log management or security information and event
1950 management systems using authenticated and encrypted communication channels.

1951 **Advanced**

1952 REQ-H-LOG-003: The Hypervisor shall support cryptographic protection and trusted time-stamping of audit logs to
1953 enable verification of the origin and integrity of audit records during forensic analysis.

1954 5.1.1.8 Patches and Updates

1955 **Classes are cumulative**

1956 **Applicability:**

1957 These requirements apply whenever the Hypervisor component is updated as part of the product lifecycle. The update
1958 mechanisms and components used to deliver and install patches and updates for the Hypervisor, including host
1959 operating system update services, management appliances, or orchestration-based update pipelines that are part of the
1960 evaluated solution, shall collectively satisfy the requirements defined in this clause.

1961 Where the Hypervisor is delivered only as part of a larger product image or appliance, the product update mechanism
1962 shall ensure that Hypervisor binaries and related components are updated in accordance with REQ-H-UPD-001 to REQ-
1963 H-UPD-004, as applicable.

1964 **Basic**

1965 REQ-H-UPD-001: The Hypervisor shall provide a mechanism to apply security patches and updates to the Hypervisor
1966 without requiring a full reinstallation of the Hypervisor or host system.

1967 **Elevated**

1968 REQ-H-UPD-002: The Hypervisor shall verify the authenticity and integrity of all patches and updates before
1969 installation, using digital signatures validated against trusted keys or certificates configured for the Hypervisor.

1970 If authenticity or integrity verification fails, the Hypervisor shall prevent installation of the patch or update.

1971 REQ-H-UPD-003: The Hypervisor shall implement rollback protection to prevent Unauthorized installation of
1972 outdated, revoked, or replayed updates.

1973 Authorized rollback to a previously trusted version shall only be permitted when all of the following conditions are met:

- 1974 • The rollback image is verified for integrity and authenticity against a trusted key or certificate.
- 1975 • The rollback action is explicitly authorized by an administrator with elevated privileges.
- 1976 • The rollback event is logged.

1977 **Advanced**

1978 REQ-H-UPD-004: The Hypervisor shall support live patching of critical Hypervisor components, including the kernel
1979 or microkernel, device drivers, and security modules, in order to remediate vulnerabilities without requiring shutdown
1980 of guest virtual machines and with only minimal interruption to workload execution.

1981 5.1.1.9 Secure Configuration and Default

1982 **Classes are cumulative**

1983 **Basic**

1984 REQ-H-CFG-001: The Hypervisor shall, by default, disable remote administrative access pathways and insecure
 1985 management interfaces that are not required for initial provisioning of the Hypervisor product. This includes remote
 1986 shell access without strong authentication, legacy or unencrypted management APIs, and vendor-specific diagnostic
 1987 ports that are not required for secure operation.

1988 The Hypervisor shall provide configuration mechanisms that allow administrators to explicitly enable only those
 1989 interfaces and services that are necessary for the intended Virtualization, orchestration, and core host management
 1990 operations.

1991 **Elevated**

1992 REQ-H-CFG-002: The Hypervisor shall validate configuration parameters before applying them and shall implement
 1993 configuration validation logic that detects and prevents configurations which would disable or bypass security
 1994 mechanisms or violate guarantees for virtual machine isolation, control-plane protection, or minimum resource
 1995 allocations for security and management functions.

1996 When a configuration change is rejected due to a security-related validation failure, the Hypervisor shall provide clear
 1997 feedback to the administrator indicating the reason for rejection.

1998 **Advanced**

1999 REQ-H-CFG-003: The Hypervisor shall support the definition and use of security configuration baselines that specify
 2000 secure settings for critical Hypervisor components, including at least I/O memory management unit configuration,
 2001 virtual switch policies, and host firewall rules, as implemented within the Hypervisor product.

2002 The Hypervisor shall provide capabilities to automatically validate current configuration settings against a selected
 2003 security baseline and shall generate alerts or audit events when non-compliant settings are detected.

2004 **5.1.1.10 Data Minimization**

2005 **Single Class**

2006 **Basic**

2007 REQ-H-DM-001: The Hypervisor shall limit log content, telemetry, and diagnostic data to information required for
 2008 operation, troubleshooting, and security monitoring. Logs, telemetry, and diagnostic outputs produced by the
 2009 Hypervisor shall not include sensitive security material such as plaintext authentication credentials, full cryptographic
 2010 keys, or complete virtual machine memory contents.

2011 Payload or application data originating from guest workloads shall not be included in Hypervisor logs, telemetry, or
 2012 diagnostic outputs by default. Where the Hypervisor supports inclusion of such payload content for debugging
 2013 purposes, this behaviour shall only be enabled through explicit administrative configuration and shall be clearly
 2014 indicated to the administrator.

2015 REQ-H-DM-002: The Hypervisor shall provide administrative controls to restrict and, where necessary, disable the
 2016 collection of specific categories of log, telemetry, and diagnostic data, including the ability to disable collection of
 2017 particular log types, telemetry streams, metrics, or debug payload capture.

2018 **5.1.2 Applicability of Hypervisor Requirements**

2019 To support proportional implementation of security requirements across different architectures, it is important to note
 2020 that there are two ways to implement security requirements. Either they are implemented as part of the hypervisor itself,
 2021 or they are delegated to the underlying hardware and software on which the hypervisor is running.

2022 • Implemented: If the hypervisor implements the security functionality, then it is the responsibility of the
 2023 hypervisor to fully meet the requirements. The required functionality needs to be implemented and configured
 2024 as prescribed.

2025 • Delegated: If the hypervisor delegates the security functionality, then it is the responsibility of the hypervisor
 2026 to correctly interface with the underlying software and hardware, use the underlying features correctly, and
 2027 provide clear documentation to users and/or integrators about the expected functionality of the underlying
 2028 software and hardware, and any additional configuration changes required to ensure the combination of
 2029 hypervisor, underlying software, and underlying hardware meets the requirement.

- 2030 The architecture of the hypervisor and the product boundary define whether a requirement should be implemented or
2031 delegated.
- 2032 • Implemented: If the requirement concerns functionality that is part of the hypervisor itself or part of a
2033 component that is distributed as part of the hypervisor product, then the requirement needs to be implemented
2034 by the hypervisor itself.
- 2035 • Delegated: If the requirement concerns functionality that is not part of the hypervisor and not part of a
2036 component that is distributed as part of the hypervisor product, then the requirement can be delegated to the
2037 underlying hardware or software.
- 2038 For example, as Type I hypervisors are typically running directly on hardware without an underlying operating system,
2039 they need to Implement all requirements. Whereas Type II hypervisors running on top of an underlying operating
2040 system can delegate common operating system functionalities such as secure updates and filesystem access control to
2041 that operating system.
- 2042 As an illustrative example, Table 5.1.2-1 provides a mapping of each requirement to its common implementation
2043 category for Type I (Bare-Metal), Hybrid (Minimal Host Kernel), and Type II (Hosted) hypervisors:
- 2044 1) A: Applicable - The Hypervisor fully implements and enforces the requirement within its architectural scope.
- 2045 2) C: Conditionally Applicable - The Hypervisor implements the requirement, but full enforcement is delegated
2046 on the underlying platform, minimal host kernel (Hybrid), or Host OS (Type II) providing supporting security
2047 functions.
- 2048 3) N/A: Not Applicable - The requirement is outside the architectural responsibility of the Hypervisor and is fully
2049 delegated to external components (e.g. Host OS, orchestrator, platform firmware).

2050

Table 5.1.2-1: Applicability of hypervisor requirements

Requirement ID	Type I	Hybrid	Type II	Rationale
Isolation				
REQ-H-VM-ISO-001	A	A	A	In all three architectures, the Virtualization layer provides CPU, memory, I/O, and interrupt separation between virtual machines; this isolation behaviour is intrinsic to the hypervisor or VMM regardless of the host design.
REQ-H-VM-ISO-003	A	A	C	Side-channel mitigations such as cache-flushing strategies, scheduling controls, or core pinning are implemented at the hypervisor/VMM layer. For Type II, the effectiveness of these mitigations also depends on how the Host OS exposes CPU features and scheduling controls, so enforcement depends partly on the underlying platform.
REQ-H-CP-ISO-001	A	A	A	In all architectures, the management plane of the hypervisor is logically distinct from guest workloads, and access control for administrative interfaces is implemented within the hypervisor's management stack.
REQ-H-CP-ISO-002	A	A	C	Type I and Hybrid hypervisors control the Virtualization layer and associated virtual switching, so they can directly define separate logical pathways for administrative and guest/data traffic. In Type II designs, the hypervisor defines its management endpoints, while separation of traffic often relies on Host OS networking configuration, which leads to conditional applicability.
REQ-H-CP-ISO-003	A	A	A	Cryptographically protected and mutually authenticated communication channels for administrative interfaces are part of the hypervisor's management interface design in Type I, Hybrid, and Type II deployments, even if cryptographic libraries come from the Host OS or platform.
REQ-H-NP-ISO-001	A	A	A	Logical segregation between management, guest, and host network planes is implemented through virtual switches, routing constructs, or equivalent mechanisms in the hypervisor solution across all three architectures.
REQ-H-NP-ISO-002	A	A	A	Filtering and access-control rules between network planes are realized at the virtual switch or equivalent enforcement points that are part of the hypervisor solution in Type I, Hybrid, and Type II designs.
REQ-H-NP-ISO-003	A	A	A	Protection of management traffic using dedicated physical interfaces or cryptographically protected channels is part of the hypervisor's networking and configuration model in all architectures; Type II implementations typically rely on Host OS interfaces but still configure their management traffic accordingly.
Integrity Protection				
REQ-H-B-INT-001	A	C	N/A	In Type I designs, the hypervisor product can perform integrity verification of its executable core before management interfaces and guest workloads

				become available. In Hybrid designs, the minimal host kernel or equivalent component performs verification anchored in platform firmware, so enforcement depends partly on firmware trust. In Type II architectures, integrity of the executable core is handled by the Host OS boot and code-integrity mechanisms, and the hypervisor process inherits this trust.
REQ-H-B-INT-002	A	C	N/A	For Type I hypervisors, the early boot stages delivered with the product participate in a chain of trust from bootloader to hypervisor core and services. Hybrid designs participate from the bootloader/host-kernel handoff upward while relying on platform firmware for the initial stage. In Type II architectures, the chain of trust from firmware to Host OS kernel is outside the hypervisor product; the hypervisor runs as an application on top of the already established OS trust chain.
REQ-H-B-INT-003	A	C	C	Type I and Hybrid hypervisors can consume verified-boot or measured-boot status from platform roots of trust and make use of this information when deciding whether to start management services and guests. In Type II setups, the hypervisor can rely on verified boot information exposed by the Host OS or platform and, in advanced deployments, can gate VM operations on that status, so this behaviour depends partly on OS/platform capabilities.
REQ-H-B-INT-004	A	C	N/A	Hardware-based roots of trust for boot-chain verification, such as secure elements or dedicated trust anchors, are typically integrated into the platform firmware and early-boot environment for Type I and Hybrid solutions; the hypervisor in these designs interacts more directly with such mechanisms. In Type II architectures, boot-chain verification is managed by the Host OS and firmware, so support for hardware-based roots of trust for the boot chain sits outside the hypervisor's scope.
REQ-H-IMG-INT-001	A	A	A	Integrity checking of guest VM images (for example, hash comparison against a manifest) before provisioning or execution is implemented within the hypervisor or its management components in all architectures.
REQ-H-IMG-INT-002	A	A	A	Verification of both integrity and authenticity of guest VM images using digital signatures or equivalent cryptographic schemes is performed by the hypervisor layer prior to VM launch, independently of the host type.
REQ-H-IMG-INT-003	A	A	C	For Type I and Hybrid hypervisors, trust anchors for image verification can be stored and protected within the hypervisor or management domain using platform trust services. In Type II environments, the hypervisor typically relies on Host OS keystores, hardware security modules, or OS abstractions to protect trust material, so support depends on the Host OS and platform.
REQ-H-IMG-INT-004	A	A	C	Anchoring integrity and authenticity verification of guest VM images in hardware-based roots of trust is common for Type I and Hybrid products that integrate with TPMs or secure elements. In Type II deployments, the hypervisor usually accesses such hardware through Host OS APIs, so use of these mechanisms is influenced by Host OS and platform capabilities.
REQ-H-RP-INT-002	A	A	A	Protection of configuration and control-plane state through authenticated, authorized modifications and the generation of audit events for security-relevant changes is handled within the hypervisor management plane and is not tied to a specific hypervisor architecture.
REQ-H-RP-INT-003	A	A	C	Type I and Hybrid hypervisors can maintain a baseline of security-critical runtime components and perform periodic or continuous integrity validation, with direct control over the execution environment. Type II hypervisors can apply similar mechanisms to their own processes and configuration but rely on Host OS facilities for deeper runtime integrity monitoring and some enforcement actions.
REQ-H-RP-INT-004	A	A	C	Hardware-based platform mechanisms for runtime integrity, such as dedicated security modules or hardware-supported memory protection, are typically integrated in the platform for Type I and Hybrid systems, and the hypervisor can interact with them directly. In Type II environments, access to such mechanisms is usually mediated by the Host OS, so support in the hypervisor depends on the OS integration layer.
REQ-H-RA-INT-003	A	A	C	Type I and Hybrid hypervisors can observe their bootloader, kernel or microkernel, security-critical configuration, and key runtime elements and can generate attestation evidence that reflects this state. In Type II designs, the hypervisor can expose evidence about its own configuration and some runtime state but depends on Host OS and platform attestation frameworks to cover deeper layers, so attestation capability is influenced by the underlying environment.
REQ-H-RA-INT-004	A	A	C	Privacy-preserving choices for attestation (for example, pseudonymous identifiers or separation of identity and integrity information) can be implemented directly in Type I and Hybrid hypervisor attestation components. In Type II environments, the hypervisor influences the privacy properties of its own evidence but often builds on Host OS or platform attestation and identity mechanisms, so support depends partly on external components.
Authentication				

REQ-H-ADMIN-AUTH-001	A	A	A	In all architectures, administrative access to hypervisor management interfaces (local CLI, remote APIs, VM consoles) is subject to authentication mechanisms implemented in the hypervisor's management layer.
REQ-H-ADMIN-AUTH-002	A	A	A	Cryptographic key-based authentication for administrative accounts and the ability to disable password-based login are provided through the same management interfaces in Type I, Hybrid, and Type II deployments.
REQ-H-ADMIN-AUTH-003	A	A	A	Certificate-based administrative authentication and certificate validation against a trusted PKI are functions of the hypervisor management plane in all three architectural models, even when the PKI integration uses Host OS services.
REQ-H-SERV-AUTH-001	A	A	A	External services such as orchestrators, monitoring tools, and backup agents interact with hypervisor management functions in all architectures, and service account authentication is handled within the hypervisor's access control model.
REQ-H-SERV-AUTH-002	A	A	A	Certificate-based mutual authentication for external services that interact with management functions or data is implemented at the hypervisor's service endpoints for Type I, Hybrid, and Type II, while cryptographic building blocks may come from the platform.
REQ-H-SERV-AUTH-003	A	A	A	Enforcement of mutual certificate-based authentication for service interactions, including validation of certificate status, is part of the hypervisor's service access model independently of the deployment architecture.
Authorization				
REQ-H-ADMIN-AUTHZ-001	A	A	A	Restriction of administrative actions to authenticated accounts and the use of a default-deny approach for management functions are realized within the hypervisor management-plane logic across all architectures.
REQ-H-ADMIN-AUTHZ-002	A	A	A	Role-based or attribute-based access control for administrative accounts, with privileges aligned to defined roles, is part of the hypervisor's authorization subsystem in Type I, Hybrid, and Type II designs.
REQ-H-ADMIN-AUTHZ-003	A	A	A	Fine-grained authorization policies, separation of duties, and time-bound or just-in-time privilege elevation are implemented at the hypervisor management plane, with the possibility to integrate external identity and policy decision systems in all architectures.
REQ-H-SERV-AUTHZ-001	A	A	A	Service accounts that access hypervisor functions follow a default-deny approach, with allowed actions explicitly configured, for all hypervisor types.
REQ-H-SERV-AUTHZ-002	A	A	A	Least-privilege authorization for service accounts, with permissions bound to authenticated service identities, is handled internally by the hypervisor solution and applies across Type I, Hybrid, and Type II.
REQ-H-SERV-AUTHZ-003	A	A	A	Policy-driven, fine-grained authorization based on cryptographic service identities and context-aware enforcement is part of the hypervisor's integration with external authorization systems in any of the three architectures.
Confidentiality Protection				
REQ-H-CONF-001	A	A	A	Access control for sensitive data under the hypervisor's control (such as VM images, snapshots, configuration, credentials, and security logs) is implemented within the management and storage logic of the hypervisor solution, regardless of whether underlying filesystems or storage ACLs are provided by a Host OS.
REQ-H-CONF-002	A	A	A	Protection of sensitive data at rest through encryption is part of the hypervisor solution in all architectures; Type II deployments often rely on Host OS or storage-layer capabilities, but the hypervisor still orchestrates or integrates those mechanisms for its managed data.
REQ-H-CONF-003	A	A	A	Encryption of management, migration, and hypervisor-mediated inter-VM traffic is handled by the hypervisor's networking and session-management design for Type I, Hybrid, and Type II, while the specific transport mechanisms may be provided by the Host OS or bundled components.
REQ-H-CONF-004	A	A	C	Execution environments that protect guest memory and processor state from observation by the hypervisor, host OS, and administrative interfaces rely on platform features such as confidential-computing extensions. Type I and Hybrid hypervisors interact more directly with these platform mechanisms, while Type II hypervisors depend on how the Host OS exposes and manages such capabilities.
REQ-H-CONF-005	A	A	C	Hardware-based mechanisms for protecting cryptographic keys and confidential execution state, such as secure elements or memory-encryption features, are typically integrated into the platform in Type I and Hybrid deployments and are used directly by the hypervisor stack. In Type II environments, the hypervisor generally accesses these mechanisms through Host OS abstractions, so adoption depends on OS and platform integration.
Availability and Resilience				

REQ-H-AVAIL-001	A	A	C	Detection of failures in internal management and control-plane components and recovery of these components without a full host reboot are implemented within the hypervisor and its management processes in all architectures, so that running virtual machines can continue to operate.
REQ-H-AVAIL-002	A	A	C	In Type I and Hybrid environments, clustering and automatic restart or migration of virtual machines are typically built into the hypervisor platform or closely integrated management nodes. In Type II environments, similar behaviour is often realized through external orchestration or Host OS clustering features, using interfaces exposed by the hypervisor.
REQ-H-AVAIL-003	A	A	C	Resource scheduling and reservation for security and management functions sit in the privileged scheduling layer of Type I and Hybrid hypervisors. Type II hypervisors use Host OS schedulers and resource controls (for example, priorities or cgroups), so effective reservation for their own control-plane functions depends on the Host OS configuration.
REQ-H-AVAIL-004	A	A	C	Replication of virtual machine runtime state to standby instances and rapid failover behaviour is implemented within the hypervisor and its clustering mechanisms in Type I and many Hybrid setups. In Type II designs, similar capabilities can exist but often rely on Host OS and external orchestrators for state capture and transfer, which makes practical support dependent on the surrounding environment.
Logging				
REQ-H-LOG-001	A	A	A	Generation of audit logs for security-relevant administrative actions and VM lifecycle operations is implemented by the hypervisor's logging subsystem across all of the architectures.
REQ-H-LOG-002	A	A	A	Protection of audit logs against Unauthorized access, modification, or deletion, and secure export to external log or SIEM systems over authenticated and encrypted channels, is implemented in the hypervisor solution; underlying transport mechanisms may come from the Host OS or platform.
REQ-H-LOG-003	A	A	A	Cryptographic protection and trusted time-stamping of audit logs, enabling verification of origin and integrity during forensic analysis, are capabilities of the hypervisor logging stack in Type I, Hybrid, and Type II deployments.
Patches and Updates				
REQ-H-UPD-001	A	A	A	Mechanisms for applying security patches and updates to the hypervisor without re-installing the entire host environment form part of the product's lifecycle management for Type I, Hybrid, and Type II solutions; Type II often uses Host OS package managers or application update frameworks.
REQ-H-UPD-002	A	A	A	Authenticated and integrity-checked patch and update workflows are used across all hypervisor architectures, even when the underlying signature verification routines are provided by the Host OS or external tooling.
REQ-H-UPD-003	A	A	A	Protections against Unauthorized rollback to outdated, revoked, or replayed updates are implemented in the update logic of the hypervisor solution in Type I, Hybrid, and Type II deployments, sometimes in combination with Host OS package-management features.
REQ-H-UPD-004	A	A	C	Live patching of critical hypervisor components (such as kernels, drivers, or security modules) is widely available in Type I and many Hybrid environments. In Type II scenarios, live patching of the Virtualization stack depends on how the hypervisor integrates with the Host OS and on OS patching capabilities, so support in practice is dependent on the surrounding platform.
Secure Configuration and Default				
REQ-H-CFG-001	A	A	A	Default configuration that leaves non-essential remote access paths and insecure management interfaces disabled is part of the initial setup profile for hypervisor products in all architectures.
REQ-H-CFG-002	A	A	A	Validation of configuration parameters, with detection of settings that would weaken isolation, control-plane protection, or resource allocations for management and security functions, is performed within the hypervisor configuration logic across Type I, Hybrid, and Type II.
REQ-H-CFG-003	A	A	A	Definition of security configuration baselines and comparison of current settings against those baselines (for example, IOMMU configuration, virtual switch policies, host firewall rules as exposed by the solution) are part of the hypervisor's configuration management capabilities in all three architectures.
Data Minimization				
REQ-H-DM-001	A	A	A	Limiting log content, telemetry, and diagnostic data to what is needed for operation, troubleshooting, and security monitoring, and avoiding inclusion of highly sensitive material, is controlled by the hypervisor's logging and telemetry implementation for Type I, Hybrid, and Type II.
REQ-H-DM-002	A	A	A	Administrative controls for narrowing or disabling specific categories of log, telemetry, and diagnostic collection, including debug payload capture, are implemented as part of the hypervisor's management and observability features for all architectures.

2051

2052

5.1.3 Management and Orchestration System Requirements

2053

5.1.3.0 General applicability

2054 These requirements apply to the Management and Orchestration (M&O) system regardless of its deployment model.
 2055 The M&O system may be deployed on dedicated physical hardware, on a dedicated management cluster, or as a virtual
 2056 machine running on the same virtualized infrastructure that it manages. When deployed as a virtual machine, it is hosted
 2057 on a management-designated host or cluster segment, and separation from tenant or workload domains is maintained
 2058 through logical isolation mechanisms such as network segmentation, access control policies, and the CPU, memory,
 2059 I/O, and integrity isolation mechanisms provided by the hypervisor.

2060

5.1.3.1 Authentication

2061 **Classes are exclusive**2062 **Basic**

2063 REQ-M&O-AUTH-001: The M&O system shall ensure that all administrative interfaces require authentication using
 2064 unique, non-default credentials.

2065 **Elevated**

2066 REQ-M&O-AUTH-002: The M&O system shall support multi-factor authentication (MFA) for administrative access
 2067 and shall enforce the use of revocable authentication mechanism for administrative and service API access. The M&O
 2068 system shall also ensure that internal component communication is authenticated.

2069 **Advanced**

2070 REQ-M&O-AUTH-003: The M&O system shall support integration with external identity and trust services to validate
 2071 user and service identities and shall support automated credential rotation and revocation where available.

2072

5.1.3.2 Authorization

2073 **Classes are exclusive**2074 **Basic**

2075 REQ-M&O-AUTHZ-001: The M&O system shall restrict all administrative actions to authenticated accounts and
 2076 enforce a default-deny policy for all resource access and control functions.

2077 **Elevated**

2078 REQ-M&O-AUTHZ-002: The M&O system shall enforce Role-Based Access Control (RBAC), ensuring that each
 2079 authenticated user or service account is assigned only the minimum privileges necessary for its defined role.

2080 **Advanced**

2081 REQ-M&O-AUTHZ-003: The M&O system shall support integration with an external identity and authorization
 2082 service to enforce centralized authorization policies and shall generate audit events for privilege changes and
 2083 authorization decisions.

2084

5.1.3.3 Secure Configuration

2085 **Classes are cumulative**2086 **Basic**

2087 REQ-M&O-CFG-001: The M&O system shall be delivered with secure default settings, including least-privilege access
 2088 roles, disabled unused services or ports, and enforced secure communication protocols.

2089 **Elevated**

2090 REQ-M&O-CFG-002: The M&O system shall require explicit authorization before applying any configuration changes
2091 to the hypervisor or managed workloads. It shall also require explicit authorization before applying any changes to its
2092 own internal configuration.

2093 **Advanced**

2094 REQ-M&O-CFG-003: The M&O system shall enforce security configuration baselines for all managed hypervisors and
2095 its own components and shall automatically detect and alert on any deviation from these baselines.

2096 5.1.3.4 Communication Security

2097 **Classes are cumulative**

2098 **Basic**

2099 REQ-M&O-COM-001: The M&O system shall encrypt all management and orchestration communications using
2100 industry-standard protocols (e.g. TLS 1.2 or 1.3), including traffic between the M&O system and managed components.

2101 **Elevated**

2102 REQ-M&O-COM-002: The M&O system shall enforce certificate-based mutual authentication for communication
2103 between its distributed M&O components and between the M&O system and all managed hypervisors.

2104 **Advanced**

2105 REQ-M&O-COM-003: The M&O system shall employ cryptographic identity verification of all managed hypervisors
2106 and workloads before initiating control operations, ensuring that commands are only sent to authenticated and attested
2107 endpoints.

2108 5.1.3.5 Integrity Security

2109 **Classes are exclusive**

2110 **Basic**

2111 REQ-M&O-INT-001: The M&O system shall verify the integrity of orchestration artifacts (including templates,
2112 playbooks, and scripts) at import or before execution by validating a cryptographic hash against a trusted reference
2113 value.

2114 **Elevated**

2115 REQ-M&O-INT-002: The M&O system shall verify the integrity and authenticity of all orchestration artifacts using
2116 digital signatures validated against a trusted keyring before execution.

2117 **Advanced**

2118 REQ-M&O-INT-003: The M&O system shall validate the integrity and authenticity of all third-party plugins,
2119 extensions, or integration modules before activation, and shall support administrative functions to register and revoke
2120 trusted signing keys.

2121 5.1.3.6 Communication Security

2122 **Classes are cumulative**

2123 **Basic**

2124 REQ-M&O-LOG-001: The M&O system shall log all administrative actions, including user logins, configuration
2125 changes, VM lifecycle events, orchestration workflows, and security-related alerts.

2126 **Elevated**

2127 REQ-M&O-LOG-002: The M&O system shall protect stored logs against tampering and support secure export to
2128 external logging or SIEM systems using encrypted, authenticated protocols.

2129 **Advanced**

2130 REQ-M&O-LOG-003: The M&O system shall support cryptographic signing and time-stamping of audit logs to ensure
2131 non-repudiation and integrity verification during forensic analysis.

2132 5.1.3.7 Patches and Updates

2133 **Classes are cumulative**

2134 **Basic**

2135 REQ-M&O-PM-001: The M&O system shall provide a mechanism to securely obtain and apply updates to itself and
2136 the managed hypervisors without requiring a full system reinstallation.

2137 **Elevated**

2138 REQ-M&O-PM-002: The M&O system shall verify the authenticity and integrity of all obtained updates (for both itself
2139 and managed hypervisors) using digital signatures prior to installation.

2140 **Advanced**

2141 REQ-M&O-PM-003: The M&O system shall support rollback protection to prevent the installation of Unauthorized or
2142 compromised update versions when the M&O system manages or distributes its own update packages.

2143 If update distribution is delegated to an external system, the M&O system shall verify and record update provenance
2144 and installation state but is not required to implement rollback protection itself.

2145 Applicability (for REQ-M&O-PM-003):

2146 This requirement applies only when the M&O system directly provides, packages, or distributes update artifacts for
2147 itself or managed hypervisors.

2148 When update installation is handled by an external OS or vendor-managed update system, rollback protection may be
2149 satisfied by that underlying system. The M&O system remains responsible for verifying update provenance and
2150 recording update state.

2151 5.2 CES Security Requirements

2152 5.2.0 General

2153 This clause defines specific security requirements for CESs, instantiating the security objectives outlined in clause 4.5.

2154 5.2.1 CRS Requirements

2155 5.2.1.1 Isolation

2156 5.2.1.1.1 Container Isolation

2157 **Classes are cumulative**

2158 **Basic**

2159 REQ-CRS-CN-ISO-001: The CRS shall enforce isolation between containers (workloads) using operating-system
2160 isolation mechanisms under its control, including at least process, filesystem, network, and resource management
2161 layers.

2162 The CRS shall support configurations in which one container cannot read, write, or otherwise interfere with another
2163 container's processes, filesystem contents, network endpoints, or allocated resources.

2164 **Elevated**

2165 REQ-CRS-CN-ISO-002: The CRS shall support configuration of strong separation between containers and the host
2166 operating system by restricting container access to kernel interfaces, privileged operations, and system resources outside
2167 their assigned execution context.

2168 This shall include the capability to:

- 2169 • Limit container privileges through reduction of kernel capabilities and prevention of direct access to host-level
2170 device nodes and sensitive filesystems, and
- 2171 • Restrict container access to kernel interfaces and system calls that are not required for the declared workload,
2172 using mechanisms such as syscall filtering, mandatory access control policies, or equivalent controls.

2173 **Advanced**

2174 REQ-CRS-CN-ISO-003: The CRS shall support execution of containers within hardened isolation environments that
2175 strengthen separation between container workloads and the host kernel, so that compromise of a container in such an
2176 environment does not directly provide an attacker with equivalent access to the host kernel or to other containers.

2177 NOTE 1: Hardened isolation environments may be implemented using, for example, micro-VM based container
2178 runtimes, user-space kernel sandboxes, or additional Virtualization or sandboxing layers.

2179 NOTE 2: REQ-CRS-CN-ISO-003 does not require all containers to use hardened isolation by default. It requires
2180 that the CRS be capable of providing such isolation for workloads where it is needed.

2181 **5.2.1.1.2 Control Plane Isolation**

2182 **Classes are cumulative**

2183 **Basic**

2184 REQ-CRS-CP-ISO-001: The CRS shall provide access-control mechanisms for its management and control interfaces.
2185 These mechanisms shall support configurations in which containerized workloads are prevented from directly accessing
2186 or invoking CRS management or control functions, and in which only authenticated and authorized entities can perform
2187 control or configuration actions.

2188 CRS management sockets, control APIs, and management daemons shall be deployable in a manner where they are not
2189 reachable from within container namespaces.

2190 **Elevated**

2191 REQ-CRS-CP-ISO-002: The CRS shall support configuration of dedicated logical separation between CRS
2192 management and control traffic and container data-plane traffic. The CRS shall support the use of host-only
2193 communication endpoints, distinct network namespaces, or dedicated management networks for its management and
2194 control interfaces, enabling isolation of CRS control traffic from container workloads.

2195 **Advanced**

2196 REQ-CRS-CP-ISO-003: The CRS shall support the use of cryptographically protected and mutually authenticated
2197 communication channels for all remote or network-exposed CRS management and control interfaces, in order to reduce
2198 the risk of Unauthorized access, interception, tampering, or lateral movement through the control plane.

2199 Where such channels are used, the CRS shall ensure that management and control traffic is transmitted only over
2200 communication paths that provide confidentiality, integrity protection, and mutual authentication between the CRS and
2201 authorized management entities.

2202 NOTE: For CRS management interfaces that are restricted to local host-only communication mechanisms with
2203 appropriate operating-system access controls, cryptographic protection at the transport layer may not be
2204 required, provided that equivalent protection of confidentiality, integrity, and authenticity is achieved
2205 through the underlying platform mechanisms.

2206 5.2.1.1.3 Network Plane Isolation

2207 **Classes are cumulative**

2208 **Basic**

2209 REQ-CRS-NP-ISO-001: The CRS shall maintain logical separation between container network traffic and host or CRS
2210 management network traffic, using virtual networking mechanisms under its control.

2211 The CRS shall support configurations in which containers cannot directly access host or CRS management interfaces or
2212 services by default, and in which separate network contexts can be defined for management traffic and container
2213 workload traffic.

2214 **Elevated**

2215 REQ-CRS-NP-ISO-002: The CRS shall support configuration of explicit traffic filtering and access-control rules for
2216 network flows between containers, between container networks, and between container networks and host or
2217 management networks.

2218 These rules shall be enforced at the CRS virtual networking layer or at other enforcement points under the control of the
2219 CRS, ensuring that only network flows explicitly permitted by the configured policy are allowed to cross these
2220 boundaries.

2221 **Advanced**

2222 REQ-CRS-NP-ISO-003: The CRS shall support the use of cryptographically protected and mutually authenticated
2223 communication channels for management and control traffic that traverses shared or untrusted networks, in order to
2224 maintain separation between management traffic and container data-plane traffic.

2225 Where such channels are used, the CRS shall ensure that management and control traffic is transmitted only over
2226 communication paths that provide confidentiality, integrity protection, and mutual authentication between the CRS and
2227 authorized management entities.

2228 NOTE: For management interfaces that are restricted to local host-only communication mechanisms with
2229 appropriate operating-system access controls, cryptographic protection at the transport layer may not be
2230 required, provided that equivalent protection of confidentiality, integrity, and authenticity is achieved
2231 through the underlying platform mechanisms.

2232 5.2.1.2 Integrity Protection

2233 5.2.1.2.1 Boot chain integrity verification

2234 **Classes are exclusive**

2235 **Basic**

2236 REQ-CRS-B-INT-001: The CRS shall implement an integrity verification mechanism for its core runtime engine and
2237 other CRS executables that are required to start container execution services.

2238 The CRS shall be able to invoke this integrity verification before enabling container scheduling, container lifecycle
2239 operations, or other container execution interfaces.

2240 If the integrity verification of any CRS executable covered by this mechanism fails, the CRS shall prevent startup of
2241 container execution services and shall generate a security-relevant event.

2242 **Elevated**

2243 REQ-CRS-B-INT-002: The CRS shall implement mechanisms to participate in a verifiable chain of trust for CRS
2244 startup components that are delivered as part of the CRS product and executed before container execution services are
2245 enabled.

2246 Each CRS startup stage delivered with the product and executed prior to exposing container execution interfaces shall
2247 validate the integrity and authenticity of the subsequent CRS stage before transferring control.

2248 If this validation fails for any CRS startup stage, the CRS shall prevent startup of container execution services and shall
2249 generate a security-relevant event.

2250 NOTE 1: For the purposes of REQ-CRS-B-INT-002, "participate in a verifiable chain of trust" means that CRS
2251 components involved in the startup sequence are capable of integrating into a platform-provided or CRS-
2252 defined chain of trust and enforcing integrity and authenticity verification of subsequent CRS stages.

2253 **Advanced**

2254 REQ-CRS-B-INT-003: Where the underlying platform exposes verified boot or integrity measurement status through a
2255 documented interface, the CRS shall support consuming this integrity and authenticity information for the operating
2256 system components and CRS components that shall be loaded and initialized before container execution services are
2257 enabled.

2258 The CRS shall support policies that prevent initialization of container execution services when the reported verification
2259 status for any required component in this chain indicates failure or when such status cannot be obtained.

2260 NOTE 2: REQ-CRS-B-INT-003 assumes that verified boot or integrity measurement mechanisms are provided and
2261 managed by the underlying platform or environment. The CRS responsibility is limited to consuming the
2262 exposed integrity and authenticity information and enforcing policies for container startup based on that
2263 information.

2264 REQ-CRS-B-INT-004: The CRS may support integration with hardware-based roots of trust to strengthen boot chain
2265 integrity and authenticity verification for CRS-critical components in accordance with REQ-CRS-B-INT-003.

2266 Such hardware-based roots of trust include secure elements, dedicated trust anchors, or equivalent components that
2267 provide protection against Unauthorized modification of CRS boot and startup components.

2268 NOTE 3: REQ-CRS-B-INT-004 does not require the CRS to provide or manage hardware-based roots of trust. It
2269 requires that, where such components are available in the platform, the CRS can use them as a source of
2270 integrity and authenticity assurance for CRS-critical components.

2271 5.2.1.2.2 Container image integrity verification

2272 **Classes are exclusive**

2273 **Basic**

2274 REQ-CRS-IMG-INT-001: The CRS shall implement an integrity verification mechanism for container images by
2275 validating that the image manifest and all referenced layers match their declared cryptographic digests before
2276 instantiation.

2277 The CRS shall invoke this integrity verification automatically before creating or starting any container from an image.
2278 If any mismatch, corruption, or removal of a referenced layer or manifest entry is detected, the CRS shall reject the
2279 image, shall prevent container instantiation from that image, and shall record a security-relevant event.

2280 NOTE 1: REQ-CRS-IMG-INT-001 focuses on integrity verification of container images using trusted reference
2281 values, such as cryptographic digests declared in image manifests or registries.

2282 **Elevated**

2283 REQ-CRS-IMG-INT-002: The CRS shall implement mechanisms to verify both the integrity and the authenticity of
2284 container images before execution. This verification shall establish that each image originates from a trusted publisher
2285 or approval authority and has not been altered after it was approved for deployment.

2286 Authenticity verification shall rely on cryptographic mechanisms that provide a verifiable binding between the image
2287 and the trusted publisher or approval authority and that prevent undetected modification of the image.

2288 The CRS shall prevent execution or instantiation of any container image whose integrity or authenticity cannot be
2289 successfully verified according to the configured trust policy.

2290 NOTE 2: Authenticity verification may be implemented using public-key digital signatures validated via certificate
2291 chains rooted in trusted authorities, keyless signing mechanisms bound to identified publishers, or
2292 equivalent cryptographic schemes that provide verifiable image provenance and tamper detection.

2293 **Advanced**

2294 REQ-CRS-IMG-INT-003: The CRS shall perform integrity and authenticity verification of container images using trust
 2295 anchors stored as protected trust material that is not modifiable during normal operational state.

2296 The protected trust material shall not be modifiable by container workloads, unprivileged users, or routine runtime
 2297 operations, and shall only be updated or replaced through a controlled, authenticated administrative process.

2298 If verification of the integrity or authenticity of a container image using this protected trust material fails, or if the
 2299 verification status of the image cannot be established, the CRS shall prevent execution or instantiation of that container
 2300 image and shall record a security relevant event.

2301 NOTE 3: Protected trust material refers to cryptographic keys, certificates, or other trust anchors used for container
 2302 image integrity and authenticity verification, stored and managed so that they cannot be altered through
 2303 Unauthorized administrative or workload facing interfaces and are updated only through explicitly
 2304 authorized security management procedures.

2305 REQ-CRS-IMG-INT-004: The CRS may support anchoring the integrity and authenticity verification of container
 2306 images in hardware assisted or platform provided trust services. Such mechanisms may include the use of protected key
 2307 storage, secure elements, or other platform trust services that provide resistance against Unauthorized modification of
 2308 verification keys and trust anchors.

2309 NOTE 4: REQ-CRS-IMG-INT-004 does not require the CRS to provide or manage hardware roots of trust. It
 2310 requires that, where such mechanisms are available in the underlying platform, the CRS can integrate
 2311 with them to store or validate trust anchors used for container image verification.

2312 **5.2.1.2.3 Runtime integrity protection**2313 **Classes are exclusive**2314 **Elevated**

2315 REQ-CRS-RP-INT-002: The CRS shall enforce integrity protection for its configuration and control plane by ensuring
 2316 that only authenticated and authorized modifications can be applied. The CRS shall generate audit events for any
 2317 security relevant changes to configuration or internal control logic.

2318 **Advanced**

2319 REQ-CRS-RP-INT-003: The CRS shall maintain the integrity of its security critical runtime components and detect
 2320 Unauthorized modification of configuration state or control logic used to enforce security policies or isolation.

2321 The CRS shall maintain a trusted baseline of the integrity state of these security critical runtime components and shall
 2322 perform periodic or continuous integrity validation against this baseline.

2323 Upon detecting a deviation from the trusted baseline, the CRS shall automatically trigger protective actions, such as
 2324 isolating the affected component, disabling affected functions, or restoring the component from a trusted state, and shall
 2325 generate a security alert that cannot be suppressed by the affected component.

2326 NOTE 1: Security critical runtime components include, for example, processes or modules that enforce container
 2327 isolation, access control, security policies, or integrity checks, and configuration data that directly
 2328 influences these security functions.

2329 NOTE 2: Runtime integrity validation may be implemented using hardware assisted mechanisms, software based
 2330 mechanisms, or a combination of both, provided that the mechanisms are able to detect Unauthorized
 2331 modification of the security critical runtime components defined by this requirement.

2332 REQ-CRS-RP-INT-004: The CRS may support the use of hardware based platform mechanisms or externally protected
 2333 trust material to strengthen runtime integrity validation and detection of Unauthorized modification of security critical
 2334 runtime components. Such mechanisms include dedicated secure hardware modules, hardware supported memory
 2335 protection, or equivalent platform features that provide additional assurance of runtime integrity.

2336 NOTE 3: REQ-CRS-RP-INT-004 does not require the CRS to provide or manage hardware roots of trust or
 2337 external trust services. It requires that, where such mechanisms are available in the underlying platform or
 2338 deployment environment, the CRS can integrate with them to strengthen its runtime integrity verification.

2339 5.2.1.2.4 Remote attestation

2340 **Single Class**

2341 **Advanced**

2342 REQ-CRS-RA-INT-003: The CRS may support measured launch and generation of attestation evidence describing its
2343 core runtime engine, security-critical configuration, and other runtime state that is relevant to the integrity of container
2344 execution.

2345 This attestation evidence shall be cryptographically verifiable by an authorized verifier and bound to the CRS instance
2346 for which the measurements were produced.

2347 The CRS shall protect attestation keys and measurement data against Unauthorized access, disclosure, or replay.

2348 The attestation evidence should be limited to the minimum information necessary to verify the integrity and
2349 configuration state of the CRS.

2350 NOTE 1: This requirement covers the CRS capability to collect measurements, bind them cryptographically to the
2351 CRS instance, and expose them as attestation evidence. The design and operation of local or remote
2352 verifiers, attestation protocols, and relying parties is outside the scope of the present document.

2353 NOTE 2: Where supported by the underlying platform or environment, the CRS may leverage platform trust
2354 services or externally protected trust material as trust anchors for attestation keys and measurements.

2355 REQ-CRS-RA-INT-004: Where attestation evidence produced by the CRS is intended to be used in remote verification
2356 scenarios, the CRS should support configuration of privacy-preserving attestation options that reduce linkability and
2357 correlation between attestation events while preserving the ability of authorized verifiers to assess the integrity and
2358 configuration state of the CRS.

2359 Such options should include, as appropriate for the deployment model:

- 2360 • Limiting attested information to what is necessary for integrity and configuration verification (data
2361 minimization);
- 2362 • Configuration of the conditions and intervals under which attestation can be triggered or repeated, in order to
2363 prevent excessive or Unauthorized monitoring;
- 2364 • Support for privacy-preserving techniques, such as the use of pseudonymous identifiers or separation of
2365 identity and integrity information, where technically feasible.

2366 NOTE 3: REQ-CRS-RA-INT-004 does not require specific attestation protocols or identity systems. It requires
2367 that, where remote attestation is used, the CRS provides configuration options that allow the operator to
2368 balance integrity assurance and privacy considerations.

2369 5.2.1.3 Authentication

2370 5.2.1.3.0 General

2371 This clause distinguishes between authentication of administrative users and authentication of external services that
2372 access CRS management functions or data.

2373 5.2.1.3.1 Administrative Authentication

2374 **Classes are exclusive**

2375 **Basic**

2376 REQ-CRS-ADMIN-AUTH-001: The CRS shall require authentication for access to all administrative interfaces,
2377 including local command-line interfaces, remote application programming interfaces, and administrative dashboards or
2378 consoles.

2379 The CRS shall enforce the use of unique, non-default credentials for administrative accounts and shall support
2380 configurable password complexity and lockout mechanisms to mitigate brute-force attempts.

2381 Where password-based authentication is used, the CRS shall store administrative passwords only using non-reversible,
 2382 salted hashing functions. Default credentials, if present, shall be disabled or shall be changed before administrative
 2383 access is granted.

2384 **Elevated**

2385 REQ-CRS-ADMIN-AUTH-002: The CRS shall require authenticated access to administrative interfaces and shall
 2386 support cryptographic key-based authentication for administrative accounts. The CRS shall allow administrative
 2387 password-based login to be disabled for administrative interfaces.

- 2388 • The CRS shall ensure that private keys used for administrative authentication are stored and managed in a
 2389 manner that prevents Unauthorized extraction or reuse by container workloads or other unprivileged
 2390 components.

- 2391 • Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].

2392 **Advanced**

2393 REQ-CRS-ADMIN-AUTH-003: The CRS shall require strong, cryptographically verifiable authentication for all
 2394 administrative interfaces and shall support binding administrative identities to a trusted and managed credential source,
 2395 such as a public key infrastructure or an external identity provider.

2396 The CRS shall support integration with external identity or credential management systems to facilitate controlled
 2397 issuance, rotation, and revocation of administrative credentials, and shall verify the validity and revocation status of
 2398 such credentials before granting administrative access.

2399 **5.2.1.3.2 Service Authentication**

2400	Classes are exclusive
------	------------------------------

2401 **Basic**

2402 REQ-CRS-SERV-AUTH-001: The CRS shall require authentication for external services accessing CRS application
 2403 programming interfaces or control interfaces, including orchestration systems, monitoring tools, and backup agents.

2404 The CRS shall support the use of distinct credentials for each service account and shall not rely on built-in shared or
 2405 hard-coded service credentials.

2406 The CRS shall provide configurable lockout or rate-limit mechanisms for service authentication attempts and shall store
 2407 service credentials in a manner that prevents exposure through configuration files, container images, or logs.

2408 **Elevated**

2409 REQ-CRS-SERV-AUTH-002: The CRS shall support mutual cryptographic authentication for service-to-CRS
 2410 interactions that access management functions or data, allowing both the CRS and the external service to verify each
 2411 other's identity before exchanging management or control information.

2412 Cryptographic materials used for service authentication, including keys, certificates, or tokens, shall be protected from
 2413 Unauthorized extraction or reuse by container workloads or other unprivileged components.

2414 Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].

2415 **Advanced**

2416 REQ-CRS-SERV-AUTH-003: The CRS shall enforce verifiable binding of service identities to cryptographic
 2417 credentials for all service interactions that access management functions or data. Service credentials shall be issued,
 2418 rotated, and revoked through trusted credential or identity management systems, and the CRS shall automatically reject
 2419 revoked or expired credentials during authentication.

2420 The CRS shall support integration with external identity or credential management systems in order to provide
 2421 traceability of service identities across deployments and to ensure that only services with valid, managed identities can
 2422 invoke CRS management or control functions.

2423 5.2.1.4 Authorization

2424 5.2.1.4.0 General

2425 This clause distinguishes between authorization of administrative users and authorization of external services that
2426 access CRS management functions or data.

2427 5.2.1.4.1 Administrative Authorization

2428 **Classes are exclusive**

2429 **Basic**

2430 REQ-CRS-ADMIN-AUTHZ-001: The CRS shall restrict all administrative actions to authenticated accounts and shall
2431 enforce a default-deny policy for access to management functions, permitting only those actions that are explicitly
2432 authorized.

2433 **Elevated**

2434 REQ-CRS-ADMIN-AUTHZ-002: The CRS shall enforce granular role-based access control or attribute-based access
2435 control for administrative accounts, ensuring that each administrative account is granted only the minimum privileges
2436 required for its operational role.

2437 **Advanced**

2438 REQ-CRS-ADMIN-AUTHZ-003: The CRS shall enforce fine-grained authorization policies for administrative
2439 accounts, including support for separation of duties and, where applicable, time-bound or just-in-time privilege
2440 elevation.

2441 The CRS shall support integration with external identity and policy decision systems to enable centralized definition
2442 and enforcement of administrative authorization policies.

2443 5.2.1.4.2 Service Authorization

2444 **Classes are exclusive**

2445 **Basic**

2446 REQ-CRS-SERV-AUTHZ-001: The CRS shall enforce a default-deny policy for all service accounts, ensuring that
2447 service accounts are only permitted to perform actions that are explicitly authorized by CRS configuration or policy.

2448 **Elevated**

2449 REQ-CRS-SERV-AUTHZ-002: The CRS shall enforce granular authorization for service accounts, binding permitted
2450 actions to the authenticated service identity and enforcing least-privilege permissions for each service account.

2451 **Advanced**

2452 REQ-CRS-SERV-AUTHZ-003: The CRS shall enforce fine-grained, policy-driven authorization for service accounts,
2453 binding permissions to cryptographic service identities and supporting context-aware enforcement.

2454 The CRS shall support integration with external authorization systems, where available, to enable centralized
2455 management and evaluation of authorization policies for service accounts.

2456 5.2.1.5 Confidentiality Protection

2457 **Classes are cumulative**

2458 **Basic**

2459 REQ-CRS-CONF-001 (Access control for sensitive data): The CRS shall provide access-control mechanisms to restrict
2460 access to sensitive data under its control, including container images, configuration data, stored secrets, and security-
2461 relevant logs.

2462 Access to such sensitive data shall be permitted only to authenticated and authorized entities through CRS-controlled
2463 interfaces and APIs.

2464 NOTE 1: Sensitive data includes, for example, container images stored in registries under CRS control,
2465 configuration objects that affect workload behaviour or security, secrets such as credentials and keys
2466 managed via the CRS, and logs that may contain security-relevant information.

2467 **Elevated**

2468 REQ-CRS-CONF-002 (Data at rest): The CRS shall support encryption of sensitive data at rest, including container
2469 images, configuration data, stored secrets, and security-relevant logs, either through native capabilities or by integrating
2470 with encrypted storage, registries, or secret-management services.

2471 Encryption keys used for data-at-rest protection shall be stored and managed in a manner that prevents access by
2472 container workloads and other unprivileged components.

2473 REQ-CRS-CONF-003 (Data in transit): The CRS shall support the use of encrypted communication for CRS
2474 management and control interfaces, for communication with image registries, and for other CRS APIs that handle
2475 sensitive data or control operations.

2476 When these communications are configured to use encrypted transport, the CRS shall prevent fallback or downgrade to
2477 unencrypted or weakly protected channels.

2478 NOTE 2: Encrypted communication may be implemented using TLS or equivalent cryptographic protocols,
2479 including integration with service meshes or platform-provided secure communication mechanisms,
2480 provided that confidentiality, integrity, and peer authentication are ensured.

2481 **Advanced**

2482 REQ-CRS-CONF-004 (Data in use - confidential execution): Where the underlying platform provides mechanisms for
2483 isolated or protected execution environments, the CRS shall support provisioning and management of execution
2484 environments for containerized workloads in which confidential data associated with those workloads is protected,
2485 during execution, from observation, inspection, or access through the CRS, the host operating system, and other
2486 workloads.

2487 The CRS shall provide configuration and API interfaces to deploy workloads into such protected execution
2488 environments and shall support secure provisioning of secrets and key material to those workloads such that the secrets
2489 and key material are not accessible to the CRS itself, the host operating system, or other workloads.

2490 NOTE 3: Platform mechanisms for confidential execution may include trusted execution environments, confidential
2491 containers, or other hardware or software mechanisms that provide protected memory and processor state
2492 isolation beyond standard container separation.

2493 REQ-CRS-CONF-005 (Hardware-backed confidentiality): The CRS may support the use of hardware-based or
2494 platform-provided security mechanisms to strengthen protection of cryptographic keys and confidential execution state
2495 used for data-at-rest, data-in-transit, and data-in-use protection.

2496 NOTE 4: Such mechanisms include dedicated secure hardware modules, memory encryption features, or equivalent
2497 platform components that provide protected storage for keys and resistance against Unauthorized access
2498 to confidential workload data. REQ-CRS-CONF-005 does not require the CRS to provide these
2499 mechanisms; it requires that, where they are available, the CRS can integrate with them to enhance
2500 confidentiality protection.

2501 **5.2.1.6 Availability and Resilience**

2502	Classes are cumulative
------	-------------------------------

2503 **Basic**

2504 REQ-CRS-AVAIL-001: The CRS shall maintain resource isolation between container workloads and shall protect CRS
2505 control-plane processes from resource exhaustion caused by container workloads. Resource isolation shall cover at least
2506 CPU and memory resources.

2507 The CRS shall provide mechanisms such that failure or abnormal termination of an individual container can be
 2508 contained within its execution context and does not, under normal operating conditions and within configured resource
 2509 limits, compromise the operation of the CRS control plane or cause other containers to terminate unexpectedly.

2510 **Elevated**

2511 REQ-CRS-AVAIL-002: The CRS shall provide automatic recovery mechanisms such that failed containers are restarted
 2512 based on defined policies without requiring manual intervention. Where supported by the deployment environment,
 2513 these mechanisms may be used in conjunction with orchestration or clustering systems to restore workloads on the same
 2514 or an alternative host.

2515 REQ-CRS-AVAIL-003: The CRS shall implement resource scheduling or reservation mechanisms that support
 2516 configuration of minimum resource allocations for CRS security and management functions, covering at least CPU,
 2517 memory, and I/O resources.

2518 These mechanisms shall protect the CRS control plane from resource starvation caused by container workloads or
 2519 external resource contention, ensuring that security and management functions remain able to operate under load.

2520 **Advanced**

2521 REQ-CRS-AVAIL-004: Where the deployment environment supports workload relocation, the CRS shall support
 2522 checkpoint and restore of the runtime state of selected container workloads, including process state and relevant in-
 2523 memory execution context, in order to mitigate control-plane or host-level failures.

2524 The CRS shall support the use of this checkpoint and restore capability to enable failover with minimal disruption to
 2525 workload execution when combined with orchestration or clustering mechanisms that coordinate relocation to a standby
 2526 instance or alternative host.

2527 **5.2.1.7 Logging**

2528 Classes are cumulative

2529 **Basic**

2530 REQ-CRS-LOG-001: The CRS shall generate audit logs for security-relevant administrative actions and security
 2531 events, including at least authentication attempts to CRS administrative or control interfaces, configuration changes
 2532 affecting CRS behaviour or security posture, and container lifecycle operations (creation, start, stop, deletion, and
 2533 restart) initiated through the CRS.

2534 The CRS shall ensure that audit logs are not writable or directly modifiable by container workloads or other
 2535 unprivileged processes running within container namespaces.

2536 **Elevated**

2537 REQ-CRS-LOG-002: The CRS shall protect audit logs from Unauthorized access, modification, or deletion. Access to
 2538 CRS audit logs shall be restricted to authorized administrative roles or trusted system processes under the control of the
 2539 CRS or host operating system.

2540 The CRS shall support secure export of audit logs to external log management or security information and event
 2541 management systems using authenticated communication channels.

2542 **Advanced**

2543 REQ-CRS-LOG-003: The CRS shall support cryptographic protection and trusted time-stamping of audit logs to enable
 2544 verification of the origin and integrity of audit records during forensic analysis. Where the CRS relies on external time
 2545 or signing services, audit records shall include sufficient metadata to allow verification of the trust source used for time-
 2546 stamping and integrity protection.

2547 5.2.1.8 Patches and Updates

2548 **Classes are cumulative**

2549 **Applicability:**

2550 These requirements apply whenever the CRS components are updated as part of the product lifecycle. The mechanisms
2551 and components used to deliver and install patches and updates for the CRS, including host operating system package
2552 managers, management appliances, or orchestration-based update pipelines that are part of the evaluated solution, shall
2553 collectively satisfy the requirements defined in this clause.

2554 Where the CRS is delivered only as part of a larger product image or appliance, the product update mechanism shall
2555 ensure that CRS binaries and related components are updated in accordance with REQ-CRS-UPD-001 to REQ-CRS-
2556 UPD-003, as applicable.

2557 **Basic**

2558 REQ-CRS-UPD-001: The CRS shall support applying security patches and updates to CRS components without
2559 requiring a full reinstallation of the CRS or the host system.

2560 **Elevated**

2561 REQ-CRS-UPD-002: The CRS, or the update mechanisms included in the evaluated solution for CRS components,
2562 shall verify the authenticity and integrity of all CRS patches and updates before installation, using digital signatures
2563 validated against trusted keys or certificates configured for the CRS.

2564 If authenticity or integrity verification of a CRS patch or update fails, the update mechanism shall prevent installation of
2565 that patch or update and shall record a security-relevant event.

2566 Trusted keys or certificates used for CRS update verification shall not be modifiable by container workloads or other
2567 unprivileged components during normal operation.

2568 **Advanced**

2569 REQ-CRS-UPD-003: Where the evaluated solution provides an update mechanism for CRS components that supports
2570 version rollback, the update mechanism shall implement rollback protection to prevent Unauthorized installation of
2571 outdated, revoked, or replayed CRS updates.

2572 Authorized rollback to a previously trusted CRS version shall only be permitted when all of the following conditions
2573 are met:

- 2574 • The rollback image or package is verified for integrity and authenticity against a trusted key or certificate.
- 2575 • The rollback action is explicitly authorized by an administrator with elevated privileges or an equivalent
2576 authorized control function.
- 2577 • The rollback operation is recorded as a security-relevant event in the audit logs, including the version reverted
2578 from and the version reverted to.

2579 5.2.1.9 Secure Configuration and Default

2580 **Classes are cumulative**

2581 **Basic**

2582 REQ-CRS-CFG-001: The CRS shall, by default, disable interfaces and features that are not required for container
2583 execution and CRS administration, including remote debugging endpoints, unauthenticated control sockets, and
2584 developer interfaces that expose management or control capabilities.

2585 Only administrators shall be able to explicitly enable such interfaces and features through CRS configuration.

2586 **Elevated**

2587 REQ-CRS-CFG-002: The CRS shall validate configuration parameters under its control before applying them and shall
 2588 reject settings that would disable or bypass security mechanisms or that would compromise container-to-container
 2589 isolation or controlled access to host resources.

2590 When a configuration change is rejected for security reasons, the CRS shall provide clear feedback to the administrator
 2591 indicating the reason for rejection.

2592 **Advanced**

2593 REQ-CRS-CFG-003: The CRS shall support the definition and use of security configuration baselines that define secure
 2594 settings for CRS daemon configuration and default container security profiles, including at least isolation-related
 2595 options and default access to host resources as implemented within the CRS product.

2596 The CRS shall provide capabilities to automatically validate current CRS configuration and default container profile
 2597 settings against a selected security baseline and shall generate alerts or audit events when non-compliant settings are
 2598 detected.

2599 **5.2.1.10 Data Minimization**

2600 **Single Class**

2601 **Basic**

2602 REQ-CRS-DM-001: The CRS shall limit logging, telemetry, and diagnostic data to information necessary for runtime
 2603 operation, troubleshooting, and security monitoring. Logs, telemetry, and diagnostic outputs produced by the CRS shall
 2604 not unintentionally expose sensitive security material such as plaintext credentials, full cryptographic keys, or complete
 2605 container memory contents.

2606 Payload content from container workloads shall not be included in CRS logs, telemetry, or diagnostic outputs by
 2607 default. Where the CRS supports inclusion of payload content for debugging purposes, this behaviour shall only be
 2608 enabled through explicit administrative configuration and shall be clearly indicated to the administrator.

2609 REQ-CRS-DM-002: The CRS shall provide administrative controls to disable or restrict the collection of specific
 2610 categories of log, telemetry, and runtime metric data. These controls shall allow administrators to disable or limit the
 2611 collection of particular log types, telemetry streams, metrics, or debug payload capture, in order to align data collection
 2612 with operational and regulatory requirements.

2613 **5.2.2 CE Requirements**

2614 **5.2.2.1 Isolation**

2615 **Classes are cumulative**

2616 **Basic**

2617 REQ-CE-ISO-001: The CE shall provide a secure interface to configure and invoke CRS operations for the creation,
 2618 execution, and termination of containers, ensuring that the underlying CRS enforces strong logical separation
 2619 consistently across all containers.

2620 **Elevated**

2621 REQ-CE-ISO-002: The CE shall prevent container workloads from accessing host administrative functions by
 2622 enforcing non-privileged execution, restricting host resource access, and disabling insecure defaults, including
 2623 privileged containers and host mounts.

2624 **Advanced**

2625 REQ-CE-ISO-003: The CE shall ensure its administrative interface is logically or physically isolated from container
 2626 network traffic and is accessible only to authorized host processes (e.g. the orchestrator agent), preventing Unauthorized
 2627 access or cross-plane interference.

2628 5.2.2.2 Integrity protection

2629 **Classes are exclusive**

2630 **Basic**

2631 REQ-CE-INTEG-001: The CE shall verify the integrity of container images and associated metadata, including
2632 manifests, before execution. Images that fail verification shall not be made available to the CRS.

2633 **Elevated**

2634 REQ-CE-INTEG-002: The CE shall verify the authenticity and integrity of container images using digital signatures
2635 validated against trusted keys or authorities. It shall also restrict interfaces and commands that allow post-instantiation
2636 modification of container runtime parameters.

2637 **Advanced**

2638 REQ-CE-INTEG-003: The CE shall prevent containers from accessing its administrative interfaces, including control
2639 APIs and UNIX sockets, by enforcing strict file permissions and namespace separation. It shall not expose its interfaces
2640 to container workloads by default.

2641 5.2.2.3 Authentication

2642 **Classes are exclusive**

2643 **Basic**

2644 REQ-CE-AUTH-001: The CE shall enforce strong authentication on all interfaces used by orchestration components or
2645 external agents to manage containers, ensuring that only authenticated actors can execute workload lifecycle or
2646 configuration operations.

2647 **Elevated**

2648 REQ-CE-AUTH-002: The CE shall support cryptographic key-based mutual authentication for communication with the
2649 orchestrator or authorized agents. Private keys shall be protected from Unauthorized extraction or reuse.

2650 **Advanced**

2651 REQ-CE-AUTH-003: The CE shall support certificate-based mutual authentication for all management interfaces and
2652 shall support integration with hardware-backed key storage for securing its identity credentials.

2653 5.2.2.4 Authorization

2654 **Classes are exclusive**

2655 **Basic**

2656 REQ-CE-AUTHZ-001: The CE shall enforce access control on all operations that affect container lifecycle or resource
2657 usage, using a default-deny policy and evaluating the authenticated identity of the caller.

2658 **Elevated**

2659 REQ-CE-AUTHZ-002: The CE shall enforce access control using a Role-Based Access Control (RBAC) model. It shall
2660 support the configuration of workload-level security policies that restrict access to sensitive host resources (devices,
2661 filesystem paths, syscalls) and translate these policies into runtime enforcement parameters for the CRS.

2662 **Advanced**

2663 REQ-CE-AUTHZ-003: The CE shall enforce fine-grained, Attribute-Based Access Control (ABAC) or context-aware
2664 authorization for control operations, supporting complex policy decision points and integration with external
2665 authorization systems.

2666 5.2.2.5 Confidentiality

2667 **Classes are cumulative**

2668 **Basic**

2669 REQ-CE-CONF-001 (Protection of stored data): The CE shall prevent Unauthorized access to container images,
2670 configuration data, and secrets stored or managed by the CE by enforcing CE-controlled access boundaries and ensuring
2671 that such data is not exposed to workloads or users unless explicitly authorized through CE configuration or APIs.

2672 **Elevated**

2673 REQ-CE-CONF-002 (Data at rest and in transit): The CE shall support encryption at rest for container images,
2674 configuration data, and secrets under its control, and shall ensure that CE administrative interfaces and communications
2675 with external services are protected using encrypted channels.

2676 **Advanced**

2677 REQ-CE-CONF-003 (Data in use): Where the underlying platform supports confidential execution, the CE shall
2678 provide configuration and API interfaces to enable workload execution in protected execution contexts and shall
2679 enforce placement and associated security policies such that declared confidential workloads run only in such protected
2680 contexts. The CE shall support secure provisioning of secrets and key material into the protected execution context such
2681 that they are not accessible to the CE, host operating system, or other workloads.

2682 5.2.3 CO Requirements

2683 5.2.3.1 Isolation

2684 **Classes are cumulative**

2685 **Basic**

2686 REQ-CO-ISO-001: The CO shall enforce strong logical network isolation between workloads scheduled in different
2687 namespaces or tenancy domains, using virtual network policies.

2688 **Elevated**

2689 REQ-CO-ISO-002: The CO shall ensure its management and orchestration interfaces (API server) are logically isolated
2690 from the container data plane to prevent Unauthorized pivot attempts from compromised workloads.

2691 **Advanced**

2692 REQ-CO-ISO-003: The CO shall enforce cryptographic protection on control plane communications and ensure that
2693 control plane interfaces cannot be accessed from workload or data plane networks.

2694 5.2.3.2 Integrity protection

2695 **Classes are exclusive**

2696 **Basic**

2697 REQ-CO-INTEG-001: The CO shall protect the integrity of its configuration and policy state, ensuring that only
2698 authorized components and roles can modify control plane configuration, workload specifications, or cluster policies.

2699 **Elevated**

2700 REQ-CO-INTEG-002: The CO shall verify the integrity of its control plane component binaries (including API server
2701 and scheduler) before operation, using cryptographic hashes or signatures validated against a trusted reference.

2702 **Advanced**

2703 REQ-CO-INTEG-003: The CO shall enforce integrity of workload deployments at admission time by applying
 2704 admission control policies and preventing deployment of workloads that violate integrity, authorization, or
 2705 configuration constraints.

2706 5.2.3.3 Authentication

2707 **Classes are exclusive**

2708 **Basic**

2709 REQ-CO-AUTH-001: The CO shall enforce authenticated access for all administrative and user interactions with the
 2710 control plane APIs and management interfaces. Password-based authentication is acceptable at this level, provided that
 2711 password strength and lockout policies are enforced.

2712 **Elevated**

2713 REQ-CO-AUTH-002: The CO shall enforce cryptographically verifiable authentication mechanisms for service
 2714 accounts and administrative identities and shall require mutual authentication for internal control plane communication.

2715 NOTE: This may be achieved through mechanisms such as mutual TLS (mTLS), token-based identity,
 2716 SPIFFE/SPIRE identities, or equivalent cryptographically authenticated identity systems.

2717 **Advanced**

2718 REQ-CO-AUTH-003: The CO shall enforce certificate-based authentication across all control plane and administrative
 2719 interfaces and shall support integration with a trusted Public Key Infrastructure (PKI) for issuing and managing these
 2720 certificates.

2721 5.2.3.4 Authorization

2722 **Classes are exclusive**

2723 **Basic**

2724 REQ-CO-AUTHZ-001: The CO shall enforce a default-deny authorization policy for all control plane actions and
 2725 resource access, restricting actions to authenticated entities.

2726 **Elevated**

2727 REQ-CO-AUTHZ-002: The CO shall enforce fine-grained access policies on workload and resource control operations
 2728 through Role-Based Access Control (RBAC), ensuring the principle of least privilege is applied to all users and service
 2729 accounts.

2730 **Advanced**

2731 REQ-CO-AUTHZ-003: The CO shall support Attribute-Based Access Control (ABAC) or context-aware authorization
 2732 for control plane actions, enabling policy enforcement based on runtime context.

2733 5.2.3.5 Confidentiality

2734 **Classes are cumulative**

2735 **Basic**

2736 REQ-CO-CONF-001 (Secrets Protection): The CO shall ensure that control plane secrets, including authentication
 2737 tokens, private keys, and service credentials, are not stored in plaintext and are accessible only through a controlled
 2738 interface. The CO shall support either internal secrets store or integration with a Key Management System (KMS).

2739 **Elevated**

2740 REQ-CO-CONF-002 (Data in transit): The CO shall ensure that the internal cluster communication, including control
 2741 plane traffic and orchestrator-mediated workload communication are encrypted. The CO shall enforce configuration
 2742 policies that prevent the use of unencrypted communication paths for control-plane traffic and orchestrator-managed
 2743 workload communication.

2744 **Advanced**

2745 REQ-CO-CONF-003 (Data in use): The CO shall provide support for scheduling and managing container workloads in
 2746 platform-supported confidential execution environments. The CO shall enforce workload placement, isolation
 2747 constraints, and associated security policies such that confidential workload data cannot be observed by the CO, host
 2748 operating system, or other workloads.

2749 **5.3 Assurance Security Requirements**

2750 **5.3.1 Vulnerability Management**

2751 REQ-ASSUR-VULM-001: Vulnerability handling requirements for VES and CES products shall comply with the
 2752 vulnerability handling requirements defined in prEN 40000-1-3 [1].

2753 REQ-ASSUR-VULM-002: For products incorporating open-source software (OSS), the manufacturer shall establish
 2754 processes to leverage OSS community vulnerability reporting and resolution mechanisms. This includes timely
 2755 monitoring of upstream advisories and security patches. The manufacturer shall ensure that applied updates are
 2756 regression tested, covering both Long-Term Support (LTS) releases and relevant interim point releases, to maintain
 2757 product stability and security.

2758 **NOTE:** Open-source components form a significant part of VES and CES implementations. Effective use of
 2759 community-driven security alerts and patches is essential to reduce exposure time. Regression testing
 2760 ensures that rapid security fixes do not introduce unintended disruptions in operational environments.

2761 **5.3.2 Software Bill of Materials (SBOM)**

2762 REQ-ASSUR-SBOM-001: SBOM requirements for VES and CES products shall comply with the SBOM requirements
 2763 defined in prEN 40000-1-3 [1].

2764 REQ-ASSUR-SBOM-002: The manufacturer shall provide a Software Bill of Materials (SBOM) for VES and CES
 2765 products (see in-scope components of the VES and CES as defined in clauses 4.1.1 and 4.1.2), identifying all software
 2766 components, top level dependencies, and their versions.

2767 REQ-ASSUR-SBOM-003: For VES, the SBOM shall include (at minimum):

- 2768 • Hypervisor kernel or microkernel.
- 2769 • Hypervisor management interfaces (CLI/API).
- 2770 • Virtual Machine Monitor (VMM) and lifecycle controllers.
- 2771 • Resource scheduling components (CPU, memory, I/O).
- 2772 • Device Virtualization modules.
- 2773 • Hardware-related dependencies bundled with the VES product.

2774 REQ-ASSUR-SBOM-004: For CES, the SBOM shall include (at minimum):

- 2775 • CRS: runtime binaries, enforcement modules, APIs/services.
- 2776 • Container Engine: engine binaries, image management tools, container networking modules (bridge,
 2777 namespace managers), storage/volume plugins, supported extensions.
- 2778 • Container Orchestrator: orchestrator binaries, admission controllers, scheduling modules, secrets management,
 2779 network policy enforcement, cluster lifecycle management components.
- 2780 • Hardware-related dependencies bundled with the CES product.

- 2781 NOTE: In this clause examples are provided to illustrate typical components and formats that may fall under the
2782 SBOM requirements. These examples are non-exhaustive and non-normative. For instance:
- 2783 ▪ Device Virtualization modules may include SR-IOV drivers, paravirtualized device drivers, or
2784 host-guest interface modules.
 - 2785 ▪ CRS runtime binaries may include runc or crun.
 - 2786 ▪ CRS enforcement modules may include namespaces, seccomp, cgroups, microVMs.
 - 2787 ▪ CRS APIs/services may include containerd, CRI-O.
 - 2788 ▪ Container engine binaries may include Docker or Podman.
 - 2789 ▪ Container orchestrator binaries may include Kubernetes.
 - 2790 ▪ Machine-readable SBOM formats may include SPDX or CycloneDX.
 - 2791 ▪ Hardware-related dependencies may include vTPM firmware, drivers for SR-IOV NICs, storage
2792 controllers, or other hardware-specific modules.

2793 5.4 Threat-to-Requirement Traceability Matrix

2794 5.4.1 VES

2795 5.4.1.1 Hypervisor

2796 **Table 5.4.1.1-1: Hypervisor - threats to requirements mapping**

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-HYP1-INTF-ESC/ T-HYP2-HOST-ABUSE	REQ-H-VM-ISO-001 REQ-H-CP-ISO-001 REQ-H-NP-ISO-001	REQ-H-VM-ISO-001 REQ-H-CP-ISO-002 REQ-H-NP-ISO-002	REQ-H-VM-ISO-003 REQ-H-CP-ISO-003 REQ-H-NP-ISO-003
T-HYP1-INTEG-COMP/T-HYP2-INTEG-CO	REQ-H-B-INT-001 REQ-H-IMG-INT-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002	REQ-H-B-INT-003 REQ-H-B-INT-004 REQ-H-IMG-INT-003 REQ-H-IMG-INT-004 REQ-H-RP-INT-003 REQ-H-RP-INT-004 REQ-H-RA-INT-003 REQ-H-RA-INT-004
T-HYP1-UNAUTH-ADMIN/ T-HYP2-UNAUTH-ADMIN	REQ-H-ADMIN-AUTH-001 REQ-H-SERV-AUTH-001	REQ-H-ADMIN-AUTH-002 REQ-H-SERV-AUTH-002	REQ-H-ADMIN-AUTH-003 REQ-H-SERV-AUTH-003
T-HYP1-UNAUTH-FUNC/ T-HYP2-UNAUTH-FUNC	REQ-H-ADMIN-AUTHZ-001 REQ-H-SERV-AUTHZ-001	REQ-H-ADMIN-AUTHZ-002 REQ-H-SERV-AUTHZ-002	REQ-H-ADMIN-AUTHZ-003 REQ-H-SERV-AUTHZ-003
T-HYP1-DATA-DISC/ T-HYP2-DATA-DISC	REQ-H-CONF-001 REQ-H-DM-001 REQ-H-DM-002	REQ-H-CONF-002 REQ-H-CONF-003	REQ-H-CONF-004 REQ-H-CONF-005
T-HYP1-AVAIL-DOS/ T-HYP2-AVAIL-DOS	REQ-H-AVAIL-001	REQ-H-AVAIL-002 REQ-H-AVAIL-003	REQ-H-AVAIL-004
T-HYP1-LOG-INT/ T-HYP2-LOG-INT	REQ-H-LOG-001	REQ-H-LOG-002	REQ-H-LOG-003
T-HYP1-UPD-COMP/ T-HYP2-UPD-COMP	REQ-H-UPD-001	REQ-H-UPD-002 REQ-H-UPD-003	REQ-H-UPD-004
T-HYP1-CFG-COMP/ T-HYP2-CFG-COMP	REQ-H-CFG-001	REQ-H-CFG-002	REQ-H-CFG-003
T-HYP1-SC-COMP/ T-HYP2-SC-COMP	REQ-H-B-INT-001 REQ-H-IMG-INT-001 REQ-H-UPD-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002 REQ-H-UPD-002 REQ-H-UPD-003	REQ-H-B-INT-003 REQ-H-B-INT-004 REQ-H-IMG-INT-003 REQ-H-IMG-INT-004 REQ-H-RP-INT-003 REQ-H-RP-INT-004 REQ-H-RA-INT-003 REQ-H-RA-INT-004 REQ-H-UPD-004

2797

2798 5.4.1.2 M&O System

2799 [To be completed in a future version]

2800 5.4.2 CES

2801 5.4.2.1 CRS

2802 Table 5.4.2.1-1: CRS - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-CRS-INTF-ESC	REQ-CRS-CN-ISO-001 REQ-CRS-CP-ISO-001 REQ-CRS-NP-ISO-001	REQ-CRS-CN-ISO-002 REQ-CRS-CP-ISO-002 REQ-CRS-NP-ISO-002	REQ-CRS-CN-ISO-003 REQ-CRS-CP-ISO-003 REQ-CRS-NP-ISO-003
T-CRS-INTEG-COMP	REQ-CRS-B-INT-001 REQ-CRS-IMG-INT-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002	REQ-CRS-B-INT-003 REQ-CRS-B-INT-004 REQ-CRS-IMG-INT-003 REQ-CRS-IMG-INT-004 REQ-CRS-RP-INT-003 REQ-CRS-RP-INT-004 REQ-CRS-RA-INT-003 REQ-CRS-RA-INT-004
T-CRS-UNAUTH-ADMIN	REQ-CRS-ADMIN-AUTH-001 REQ-CRS-SERV-AUTH-001	REQ-CRS-ADMIN-AUTH-002 REQ-CRS-SERV-AUTH-002	REQ-CRS-ADMIN-AUTH-003 REQ-CRS-SERV-AUTH-003
T-CRS-UNAUTH-FUNC	REQ-CRS-ADMIN-AUTHZ-001 REQ-CRS-SERV-AUTHZ-001	REQ-CRS-ADMIN-AUTHZ-002 REQ-CRS-SERV-AUTHZ-002	REQ-CRS-ADMIN-AUTHZ-003 REQ-CRS-SERV-AUTHZ-003
T-CRS-DATA-DISC	REQ-CRS-CONF-001 REQ-CRS-DM-001 REQ-CRS-DM-002	REQ-CRS-CONF-002 REQ-CRS-CONF-003	REQ-CRS-CONF-004 REQ-CRS-CONF-005
T-CRS-AVAIL-DOS	REQ-CRS-AVAIL-001	REQ-CRS-AVAIL-002 REQ-CRS-AVAIL-003	REQ-CRS-AVAIL-004
T-CRS-LOG-INT	REQ-CRS-LOG-001	REQ-CRS-LOG-002	REQ-CRS-LOG-003
T-CRS-UPD-COMP	REQ-CRS-UPD-001	REQ-CRS-UPD-002	REQ-CRS-UPD-003
T-CRS-CFG-COMP	REQ-CRS-CFG-001	REQ-CRS-CFG-002	REQ-CRS-CFG-003
T-CRS-SC-COMP	REQ-CRS-B-INT-001 REQ-CRS-IMG-INT-001 REQ-CRS-UPD-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002 REQ-CRS-UPD-002	REQ-CRS-B-INT-003 REQ-CRS-B-INT-004 REQ-CRS-IMG-INT-003 REQ-CRS-IMG-INT-004 REQ-CRS-RP-INT-003 REQ-CRS-RP-INT-004 REQ-CRS-RA-INT-003 REQ-CRS-RA-INT-004 REQ-CRS-UPD-003

2803

2804 5.4.2.2 CE

2805 [To be completed in a future version]

2806 5.4.2.3 CO

2807 [To be completed in a future version]

5.4.3 Hardware-level common threat

Table 5.4.3-1: Hypervisor - HW-level threat to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-ALL-HW-ACCESS	REQ-H-B-INT-001 REQ-H-IMG-INT-001 REQ-H-CONF-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002 REQ-H-CONF-002 REQ-H-CONF-003	REQ-H-B-INT-003, REQ-H-B-INT-004 REQ-H-IMG-INT-003, REQ-H-IMG-INT-004 REQ-H-RP-INT-003, REQ-H-RP-INT-004 REQ-H-RA-INT-003, REQ-H-RA-INT-004 REQ-H-CONF-004, REQ-H-CONF-005

Table 5.4.3-2: CRS - HW-level threat to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-ALL-HW-ACCESS	REQ-CRS-B-INT-001 REQ-CRS-IMG-INT-001 REQ-CRS-CONF-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002 REQ-CRS-CONF-002 REQ-CRS-CONF-003	REQ-H-CRS-INT-003, REQ-CRS-B-INT-004 REQ-CRS-IMG-INT-003, REQ-CRS-IMG-INT-004 REQ-CRS-RP-INT-003, REQ-CRS-RP-INT-004 REQ-CRS-RA-INT-003, REQ-CRS-RA-INT-004 REQ-CRS-CONF-004, REQ-CRS-CONF-005

6 Conformance Assessment / Tests

6.1 Assessment Methodology

6.1.1 General Assessment Procedure

This clause defines the methodology for assessing conformity with the VES and CES security requirements specified in the present document.

The assessment methodology shall be applied in a configuration-aware and risk-based manner. The depth, rigor, and specific assessment activities are determined by:

1. the product's declared configuration and included components as defined in clause 4.2;
2. the applicable SCL 1/2/3 corresponding to the assessed risk level (Low/Medium/High);
3. the threats applicable to the declared deployment and use case; and
4. whether the manufacturer applies the default SCL mapping or a custom security profile.

Assessment activities shall verify that the implemented requirements mitigate the threats associated with the declared deployment and use case.

Each requirement is associated with an assessment description that follows a structured format:

- 2827 1) **Assessment Reference:** Identifies the link to the exact requirement ID(s).
- 2828 2) **Assessment Objective:** Defines the security property or capability that shall be verified, ensuring that the
2829 assessment remains focused on the intent of the requirement.
- 2830 3) **Assessment Preparation:** Describes the environment, setup, and preconditions required before executing the
2831 test. It includes the following elements as applicable:
- 2832 - Test environment: Describe the hardware, software, and network setup used for the assessment,
2833 including versions, topology, and any relevant dependencies.
 - 2834 - Preconditions: Specify any configurations, credentials, or operational states that should be established
2835 before the test (e.g. product initialized, certificates loaded, user roles created).
 - 2836 - Required tools: Identify the tools or software necessary to perform the assessment (e.g. vulnerability
2837 scanners, protocol fuzzers, traffic Analyzers, static code Analyzers, cryptographic test suites).
 - 2838 - Reference any vendor-provided setup guides, configuration instructions, or operational manuals, as well
2839 as any relevant standards or technical notes, that define how the product shall be configured or operated
2840 for the assessment.
 - 2841 - Product configuration documentation: Confirm that the assessment environment reflects the product
2842 composition declared by the manufacturer as defined in clause 4.2, including which components are
2843 included or excluded. For example:
 - 2844 ▪ If the product includes only the Hypervisor, M&O components are not assessed.
 - 2845 ▪ If the product includes both the Hypervisor and the M&O system, each component is assessed
2846 against its respective requirements, and the assessment shall additionally verify the security-
2847 relevant interactions between them.
 - 2848 ▪ If optional components are excluded, the tester shall verify that their absence does not affect
2849 requirement fulfilment.
- 2850 4) **Assessment Activities:** Provides execution steps to be performed. Assessment activities shall be traceable to
2851 the threats, risk level, and requirement class they address. For each assessed requirement, the tester shall:
- 2852 - Identify the threat(s) from clause 4.4 that the requirement is intended to mitigate.
 - 2853 - Verify that the rigor of the assessment corresponds to the applicable SCL (SCL 1 / SCL 2 / SCL 3)
2854 associated with the assessed risk level (Low / Medium / High).
 - 2855 - Confirm that the requirement class (Basic / Elevated / Advanced) is applied consistently with the
2856 declared product configuration and assessed risk.
 - 2857 - Record evidence demonstrating that the implemented control mitigates the identified threat(s).
- 2858 Assessment activities should include, as applicable:
- 2859 - Review documentation to confirm that the described implementation matches the requirement (e.g.
2860 verify that the security architecture document specifies TLS 1.2 or higher for all external interfaces, or
2861 that the password policy aligns with the defined threshold).
 - 2862 - Perform security functional or penetration tests to verify that implemented controls prevent Unauthorized
2863 access or data modification (e.g. attempt to log in with invalid credentials to test lockout enforcement or
2864 try to modify protected configuration files without administrative privileges).
 - 2865 - Analyze code or binaries to identify potential security weaknesses or misconfigurations (e.g. perform
2866 static analysis to detect hardcoded credentials or use dynamic analysis tools to identify buffer overflow
2867 or injection vulnerabilities).
 - 2868 - Inspect configurations to ensure that required security parameters are correctly applied (e.g. check that
2869 weak cipher suites are disabled, two-factor authentication is enabled, and least-privilege access controls
2870 are configured in the system).

- 2871 - Observe runtime behaviour to confirm that protections such as encryption, authentication, and integrity
 2872 verification operate as intended (e.g. monitor network traffic to ensure data in transit is encrypted or
 2873 observe system logs to verify successful validation of digital signatures during startup).
- 2874 5) **Assessment Verdict:** Defines the pass/fail criteria.
- 2875 - **Pass:**
 2876 The assessment is considered passed if the product demonstrably fulfils the requirement and meets the
 2877 defined security thresholds.
 2878 Examples of such thresholds include:
- 2879 ▪ Minimum cryptographic strength (e.g. AES-128 or higher);
 - 2880 ▪ Password policy limits (e.g. minimum of 12 characters);
 - 2881 ▪ Login protection mechanisms (e.g. account lockout after five consecutive failed attempts);
 - 2882 ▪ Resistance to a specified attack potential (e.g. equivalent to CSA High/AVA_VAN.3 or higher).
- 2883 - **Fail:**
 2884 The assessment is considered failed if the requirement is not fulfilled, or if the defined security
 2885 thresholds are not achieved (e.g. insufficient key length, missing authentication enforcement, or
 2886 inadequate resistance to the required attack potential).
- 2887 6) **Assessment Evidence:** Defines the artefacts and documentation that shall be collected to demonstrate that the
 2888 requirement has been assessed and fulfilled. The evidence shall be sufficient to enable independent verification
 2889 of the assessment results and to demonstrate compliance with the relevant CRA essential requirements. The
 2890 Assessment Evidence includes, where applicable:
- 2891 - Test or assessment reports showing the steps performed and results obtained;
 - 2892 - Logs, configuration files, or audit traces demonstrating the implementation of the requirement;
 - 2893 - Screenshots, captures, or console outputs confirming the correct execution or protection behaviour;
 - 2894 - Relevant vendor or design documentation describing the applied security measures;
 - 2895 - Evidence of the declared product composition used to scope the assessment (as per clause 4.2)
 - 2896 - Use-case description and risk level justification applied to determine the applicable SCL
 - 2897 - Threat applicability analysis demonstrating which threats from clause 4.4 were considered for the
 2898 assessed configuration
 - 2899 - Configuration-specific test results where feature or deployment variations affect implementation of the
 2900 requirement
 - 2901 - Evidence of security-relevant interactions between in-scope components when multiple components (e.g.
 2902 Hypervisor and M&O system) are included in the product
- 2903 The assessment is considered incomplete if the Assessment Evidence is missing, inconsistent, or insufficient to
 2904 verify fulfilment of the requirement.

2905 6.1.2 Advanced-class Equivalence and Technology-neutral Assessment

2906 6.1.2.0 General

2907 This clause complements the general assessment procedure defined in clause 6.1.1. It provides additional guidance for
 2908 assessing Advanced-class requirements, in particular where such requirements refer to platform-anchored, hardware-
 2909 assisted, or externally protected trust mechanisms.

2910 6.1.2.1 Equivalence of Hardware-based and Software-based Approaches

2911 For Advanced-class requirements that refer to platform-anchored, hardware-assisted, or externally protected trust
2912 mechanisms, products may fulfil the requirement by implementing:

- 2913 • Hardware-based trust services; or
- 2914 • Software-based or hybrid mechanisms that provide demonstrably equivalent security outcomes and verifiable
2915 assurance.

2916 Equivalence shall be demonstrated by showing that the implementation achieves all of the following properties, as
2917 applicable:

- 2918 1) **Cryptographic chain of trust**
2919 A verifiable cryptographic chain of trust is established from boot through runtime, where each stage validates
2920 the integrity and authenticity of the next stage using digital signatures or equivalent integrity verification of
2921 authenticated components.
- 2922 2) **Protected key storage**
2923 Cryptographic keys and credentials used to enforce trust are managed and stored under strict access-control
2924 policies that prevent Unauthorized reading, modification, or substitution, including by privileged software
2925 components.
- 2926 3) **Runtime integrity verification**
2927 Continuous or periodic validation of security-critical components, such as configuration data, binaries, and
2928 privileged memory regions, is implemented, with automated detection and documented response to integrity
2929 violations.
- 2930 4) **Attestation capability**
2931 The product can generate cryptographically verifiable evidence of its configuration and integrity state that can
2932 be validated by an authorized verifier.
- 2933 5) **Transparency of trust anchors and residual assumptions**
2934 Where the implementation anchors trust in hardware components (for example CPU security features,
2935 firmware trust stores, physical TPM, TEE, hardware security modules) or in protected software-based trust
2936 material, the product documentation shall describe the trust anchors and protected trust material on which the
2937 implementation depends for boot-chain integrity, image verification, runtime integrity, attestation, or data-in-
2938 use protection. The documentation shall identify at least:
 - 2939 - The hardware and software components that act as trust anchors or provide protected trust material;
 - 2940 - The security properties that are assumed for these components; and
 - 2941 - Residual risks if those assumptions do not hold, including the limits of what software-based mechanisms
2942 can detect or prevent and potential impacts of hardware-level or supply-chain compromise consistent
2943 with the threat scenarios defined in clause 4.4 (such as T-ALL-HW-ACCESS).

2944 Testers shall verify achievement of the intended security outcomes of the relevant Advanced-class requirements, rather
2945 than requiring the use of specific hardware technologies. Hardware-based and software-based implementations shall be
2946 considered equivalent when they demonstrably provide the same level of protection against the applicable threats and
2947 offer verifiable assurance of their security properties, taking into account the documented trust assumptions and residual
2948 risks. Software-based implementations without hardware dependencies may be used to meet Advanced-class
2949 requirements where they demonstrably achieve comparable protection against the mapped threats.

2950 6.1.2.2 Technology-neutral Assessment Principles

2951 When applying the methodology in clause 6.1.1 to Advanced-class requirements, testers shall:

- 2952 • Evaluate the implementation against the applicable threats identified in clause 4.4 and the declared use case
2953 and risk level, rather than against a preferred reference architecture;
- 2954 • Focus on security outcomes and effectiveness of threat mitigation, not on the presence of particular hardware
2955 features or vendor technologies;

- 2956 • Not classify an implementation as deficient solely because it does not use specific hardware mechanisms such
2957 as TPM, TEE, or IOMMU, if the product demonstrates equivalent security properties through alternative
2958 architectural means;
- 2959 • Consider compensating controls such as transparency, auditability, rapid update capability, formal verification
2960 or extensive testing results when hardware immutability or hardware trust anchors are not used; and
- 2961 • Verify that documentation of trust anchors, trust assumptions, and residual risks is available and consistent
2962 with the implemented architecture and the applicable Advanced-class requirements.

2963 These principles apply in particular to Advanced-class requirements that reference platform-anchored trust mechanisms
2964 for boot integrity, image verification, runtime integrity, remote attestation, or data-in-use protection.

2965 6.1.2.3 Assessment of Diverse Architectural Approaches

2966 For products that rely on diverse architectural approaches, including Paravirtualization, hardware or software based
2967 isolation mechanisms, distributed trust models or formally verified components, testers should support the assessment
2968 with evidence that covers, as applicable:

- 2969 • Architectural documentation describing the security architecture, including security-relevant components,
2970 boundaries, trust models, and protection mechanisms.
- 2971 • Security analysis demonstrating how the architecture achieves the required security properties, including any
2972 formal verification results, security proofs, or extensive testing outcomes provided by the manufacturer.
- 2973 • Results of security testing or equivalent assessment activities demonstrating resistance to relevant attack
2974 vectors for the declared architecture.
- 2975 • Documentation of external dependencies, trust assumptions, and jurisdictional considerations associated with
2976 external trust services or infrastructure.

2977 6.2 Assessment Report Requirements

2978 6.2.1 Mandatory Report Contents

2979 The assessment report shall provide sufficient information to demonstrate that the assessment was performed in
2980 accordance with the present document, that the scope of the assessment corresponds to the declared product
2981 configuration, and that the assessed requirements mitigate the applicable threats.

2982 The report shall include the following clauses:

2983 (a) Product Identification and Configuration

- 2984 • Product name, version, manufacturer, and build or release identifier.
- 2985 • Declared product composition as defined in clause 4.2, clearly identifying:
 - 2986 - Components included in the assessed product.
 - 2987 - Components excluded from the product and therefore not assessed.
- 2988 • Architectural classification (e.g. Hypervisor Type 1 / Type 2 / Hybrid, Virtual Execution Stack configuration).
- 2989 • Any declared platform or hardware dependencies relevant to security enforcement.

2990 (b) Use Case, Risk, and Requirement Application

- 2991 • Declared use case(s) and deployment context.
- 2992 • Risk level (Low / Medium / High) determined in accordance with clause B, and corresponding Security
2993 Category Level (SCL 1 / 2 / 3).
- 2994 • Requirement application approach:

- 2995 - Default SCL mapping or
- 2996 - Custom security profile.
- 2997 • For custom profiles, the report shall include a complete mapping between applicable threats, risk levels, and
- 2998 selected requirements.

2999 **(c) Threat Applicability**

- 3000 • List of applicable threats from clause 4.4 relevant to the declared configuration and use case.
- 3001 • Threats not applicable, with justification (e.g. architectural exclusion, physically absent feature, or non-
- 3002 supported interface).

3003 **(d) Assessment Activities and Results**

3004 For each assessed requirement, the report shall provide:

- 3005 • Requirement ID and full requirement text.
- 3006 • Requirement class (Basic / Elevated / Advanced).
- 3007 • Linked threat(s) from clause 4.4 and rationale for applicability.
- 3008 • Assessment activities performed, including any class-specific or configuration-specific procedures.
- 3009 • Test environment and configuration details (e.g. hardware features, firmware level, cluster topology,
- 3010 VM/container layout).
- 3011 • Observations and results.
- 3012 • Verdict (Pass / Fail) with justification.
- 3013 • Assessment Evidence Reference (using categories defined in clause 6.2.3).

3014 Where alternative assessment paths are permitted (e.g. due to unavailable platform features), the report shall document:

- 3015 • The unavailable capability and reason
- 3016 • The alternative procedure used
- 3017 • Justification for equivalence of assurance

3018 **(e) Component-Level and System-Level Findings**

- 3019 • Results for each individual component in the product composition.
- 3020 • System-level interaction findings, including isolation, trust boundaries, and interface mediation, where
- 3021 multiple components are present.
- 3022 • Identification of any constraints, operational assumptions, or observed limitations.

3023 **(f) Overall Conformity Statement**

- 3024 • Statement of conformity or non-conformity with the present document.
- 3025 • Assigned SCL (1 / 2 / 3) when using the default SCL approach, or confirmation of correct application of a
- 3026 custom security profile.
- 3027 • Residual risks, limitations, known deviations, or conditional security dependencies.

3028 **(g) Operational Environment Dependencies for Advanced Class Assessments**

3029 When an assessment at the Advanced class (SCL 3) relies on Operational Environment (OE) hardware components to
 3030 achieve the required assurance level, such as Trusted Platform Modules (TPM), Trusted Execution Environments
 3031 (TEE), Hardware Security Modules (HSM), or other platform trust anchors the assessment report shall explicitly
 3032 document:

- 3033 • The specific hardware security components and their security properties required for conformity (e.g. secure
- 3034 key storage, measured boot, attestation capability).
- 3035 • The manner in which these components contribute to the enforcement of the assessed requirements.
- 3036 • Any platform prerequisites or firmware dependencies necessary for correct operation.
- 3037 • Where applicable, available alternative configurations or mitigations when such hardware components are
- 3038 absent.
- 3039 The report shall reference corresponding evidence under category [PLT-CAP] (Platform Capability Evidence).

3040 6.2.2 Traceability Requirements

- 3041 The assessment report shall provide complete traceability demonstrating:
- 3042 Threat → Risk Level → Requirement → Assessment Activity → Evidence → Verdict
- 3043 This traceability shall be sufficient to enable an independent reviewer to:
- 3044 • Confirm that all applicable threats were addressed.
 - 3045 • Verify that the rigor of assessment is consistent with the declared SCL.
 - 3046 • Reconstruct assessment decisions without direct access to the product.

3047 6.2.3 Evidence Package

- 3048 The assessment shall be supported by an evidence package referenced by the report. Evidence shall be organized using
- 3049 the following evidence categories:

Tag	Evidence Category	Purpose	Examples
[LOG]	Execution Logs & Test Outputs	Shows what was done and what happened during testing	Console logs, CLI output, packet captures, fuzz results
[CONFIG]	System & Configuration State	Captures how the system was configured at test time	Config files, ACLs, hypervisor settings, routing tables
[SCOPE]	Product Composition & Scope Evidence	Shows which components are included/excluded per clause 4.2	Product composition statement, SBOM, component list
[RA]	Risk & Threat Applicability Evidence	Shows link to clause B scoring & threat selection	Risk scoring tables, threat mapping justification
[DESIGN]	Architecture & Implementation Documentation	Explains how the security mechanism works	Security architecture docs, mitigation mechanism descriptions
[OBSERVATION]	Captured Traces & Measurement Evidence	Instrumentation proving enforcement or behaviour	Screenshots, Wireshark traces, leakage graphs, audit views
[PLT-CAP]	Platform Capability Evidence	Documents feature availability or dependency	CPU/IOMMU features, trust anchor presence, firmware versions

3050

3051 6.3 VES

3052 6.3.1 Hypervisor

3053 6.3.1.1 Assessment for VM Isolation

3054 6.3.1.1.1 Assessment Case AC-H-VM-ISO-001

3055 1. Assessment Reference

3056 Requirement: REQ-H-VM-ISO-001 (Basic & Elevated)

3057 **2. Assessment Objective**

3058 Verify that the Hypervisor enforces isolation between virtual machines at CPU, memory, I/O and interrupt-handling
3059 levels so that one VM cannot read, write or otherwise interfere with another VM's execution, data or configuration state.

3060 **3. Assessment Preparation**

3061 The assessment shall have access to:

- 3062 • Documentation of the Hypervisor's isolation model (CPU scheduling, memory Virtualization, IOMMU/device
3063 assignment, virtual switching, interrupt Virtualization).
- 3064 • Documentation identifying which resources are shared vs dedicated across VMs.
- 3065 • A test configuration with at least two VMs on the same host, under different administrative domains or with
3066 clearly distinct identities and data.
- 3067 • Ability to run test code inside each VM (e.g. to probe memory, devices and configuration interfaces).

3068 **4. Assessment Activities**

3069 The assessment shall at least:

- 3070 1) Memory isolation
 - 3071 - From VM A, attempt to access memory that belongs to VM B using out-of-range pointers, direct device
3072 mappings, or interfaces that would expose host/other-VM memory if not isolated.
 - 3073 - Verify that such attempts fail (e.g. access violation, I/O error) and do not disclose VM B data.
- 3074 2) I/O isolation
 - 3075 - Identify virtual and, where applicable, passed-through physical devices assigned to each VM.
 - 3076 - From VM A, attempt to access devices or storage explicitly assigned to VM B (e.g. by enumerating
3077 devices, probing block devices, or accessing device identifiers).
 - 3078 - Verify that VM A can only access its own assigned devices and cannot read or modify data belonging to
3079 VM B's devices.
- 3080 3) CPU and interrupt isolation
 - 3081 - Review documentation and configuration to confirm that CPU state and interrupts are virtualized per VM
3082 (e.g. no direct access to other VMs' registers or interrupt handlers).
 - 3083 - Under normal operation and during stress (high CPU load in VM A), verify that VM B continues to
3084 operate correctly and does not receive spurious interrupts or experience behaviour indicating direct
3085 interference beyond expected scheduling effects.
- 3086 4) Configuration/state isolation
 - 3087 - From VM A, attempt to access Hypervisor or management interfaces that would allow inspection or
3088 modification of VM B configuration without proper administrative context.
 - 3089 - Verify that such attempts are blocked and that VM A cannot change VM B's configuration or lifecycle
3090 state.

3091 **5. Assessment Verdict**

- 3092 • Pass: tests and documentation confirm that CPU, memory, I/O and interrupt-handling isolation is enforced;
3093 one VM cannot directly read, write or interfere with another VM's execution, data or configuration state.
- 3094 • Fail: a VM can access another VM's memory or devices, influence its configuration/lifecycle without proper
3095 authorization, or otherwise bypass the isolation guarantees.

3096 **6. Assessment Evidence**

- 3097 • [SCOPE] Description of the VMs, host and resources used in the isolation assessment.

- 3098 • [CONFIG] Hypervisor configuration for CPU scheduling, memory/IOMMU, device assignment and VM
- 3099 definitions.
- 3100 • [DESIGN] Documentation of the Hypervisor isolation mechanisms across CPU, memory, I/O and interrupts.
- 3101 • [OBSERVATION] / [LOG] Results of memory/I-O/interrupt/configuration tests showing blocked access to
- 3102 other VMs' resources and normal operation under load.

3103 6.3.1.1.2 Assessment Case AC-H-VM-ISO-003

3104 1. Assessment Reference

3105 Requirement: REQ-H-VM-ISO-003 (Advanced)

3106 2. Assessment Objective

3107 Verify that the Hypervisor strengthens isolation between workloads sharing hardware resources by implementing
3108 mitigations against known side-channel attacks, and that these mitigations are effective and configurable as
3109 documented.

3110 3. Assessment Preparation

3111 The assessment shall have access to:

- 3112 • Documentation describing the side-channel threat model considered (e.g. cache, branch prediction, speculative
3113 execution, shared functional units, timing channels) and the corresponding mitigations implemented by the
3114 Hypervisor.
- 3115 • Documentation of configuration options to enable/disable or tune side-channel mitigations (e.g. core/CPU
3116 pinning or scheduling options, disabling or constraining simultaneous multithreading, cache partitioning or
3117 flushing strategies, isolation profiles, time/granularity controls).
- 3118 • A test configuration with at least two co-resident VMs on the same host, with side-channel mitigations
3119 configurable according to the documentation.

3120 4. Assessment Activities

3121 The assessment shall at least:

- 3122 1) Mitigation identification and configuration
 - 3123 - Review documentation to identify the concrete mitigations provided (e.g. scheduling constraints,
3124 hardware feature controls, memory/cache isolation techniques, noise/timing controls).
 - 3125 - Verify in the configuration that these mitigations can be enabled/disabled or selected (e.g. via profiles,
3126 flags, or policy settings) as described.
- 3127 2) Baseline behaviour without strengthened mitigations
 - 3128 - Where safely possible and consistent with the product's guidance, configure a test mode with mitigations
3129 relaxed or set to the weakest documented profile (if such a profile exists for testing).
 - 3130 - Run simple side-channel-relevant measurement workloads (e.g. co-resident timing or cache-sensitivity
3131 micro-benchmarks) in two VMs and record observable cross-VM leakage characteristics in that
3132 configuration (e.g. variability or correlation metrics).
- 3133 3) Behaviour with mitigations enabled
 - 3134 - Enable the recommended side-channel mitigation configuration for multi-tenant or high-isolation
3135 deployments as described in the documentation.
 - 3136 - Repeat side-channel-relevant measurement workloads between co-resident VMs and verify that observed
3137 leakage indicators are reduced or constrained in line with the Hypervisor's stated security objectives (e.g.
3138 reduced correlation, increased noise, or enforced scheduling/isolation preventing the originally observed
3139 pattern).
 - 3140 - Verify that key mitigation controls are active in the system configuration as expected.
- 3141 4) Administrative visibility

- 3142 - Verify that administrators can see which side-channel mitigation mode/profile is in effect (e.g. via
3143 configuration views, status commands or logs) so that the strengthened isolation posture is observable
3144 and auditable.

3145 5. Assessment Verdict

- 3146 • Pass: the Hypervisor implements and exposes side-channel mitigations as documented; these can be
3147 enabled/configured for workloads sharing hardware resources; and tests/measurements show that enabling
3148 these mitigations changes the system behaviour in a way that is consistent with strengthened isolation against
3149 known side-channel attacks.
- 3150 • Fail: no meaningful mitigations are provided; mitigations cannot be configured or observed; or enabling them
3151 does not produce any observable strengthening of isolation consistent with the product's side-channel model.

3152 6. Assessment Evidence

- 3153 • [CONFIG] Hypervisor configuration showing selected side-channel mitigation options, profiles or policies
3154 (before and after enabling strengthened isolation).
- 3155 • [DESIGN] Documentation of the side-channel threat model and implemented mitigations, including
3156 configuration guidance.
- 3157 • [PLT-CAP] Platform capability evidence where relevant (e.g. CPU features used for isolation,
3158 cache/partitioning capabilities).
- 3159 • [OBSERVATION] Measurement results or traces from side-channel-relevant tests demonstrating the effect of
3160 enabled mitigations on cross-VM leakage indicators.

3161 6.3.1.2 Assessment for Control Plane Isolation

3162 6.3.1.2.1 Assessment Case AC-H-CP-ISO-001

3163 1. Assessment Reference

3164 Requirement: REQ-H-CP-ISO-001 (Basic)

3165 2. Assessment Objective

3166 Verify that:

- 3167 1) The Hypervisor provides access-control mechanisms for its administrative functions and interfaces.
- 3168 2) The Hypervisor can be configured so that guest virtual machines cannot directly access or invoke
3169 administrative functions/interfaces.
- 3170 3) Only authenticated and authorized entities can perform management actions.

3171 3. Assessment Preparation

3172 The assessment shall have access to:

- 3173 • Documentation describing Hypervisor administrative interfaces (CLI, GUI, APIs, management agents) and
3174 access-control mechanisms (authentication, authorization, roles).
- 3175 • Documentation describing how guest VMs connect to the Hypervisor (management vs non-management
3176 channels).
- 3177 • A test configuration with:
- 3178 - at least one administrative account;
- 3179 - at least one guest VM with network connectivity;
- 3180 - the ability to attempt management actions from both an admin context and from within a guest VM.

3181 4. Assessment Activities

3182 The assessment shall at least:

- 3183 1) Identify all exposed administrative interfaces and confirm from configuration that authentication and
3184 authorization controls are configured (e.g. user accounts, roles, access policies).
- 3185 2) From a legitimate administrative context, perform management operations (e.g. VM create/start/stop,
3186 configuration changes) and verify that successful access requires authentication and that authorization rules
3187 (roles/permissions) are enforced.
- 3188 3) From within a guest VM, attempt to:
- 3189 - reach administrative endpoints (management IP/ports, APIs, management agents);
- 3190 - perform management actions using unauthenticated or guest-level credentials.
- 3191 Verify that:
- 3192 - direct access to administrative interfaces is blocked by configuration (e.g. routing/firewall, ACLs) or
3193 requires proper administrative authentication; and
- 3194 - guest VM principals are not authorized to perform management actions.

3195 5. Assessment Verdict

- 3196 • Pass: access-control mechanisms exist; Hypervisor can be configured so guest VMs cannot directly invoke
3197 administrative functions; and only authenticated, authorized entities can perform management actions.
- 3198 • Fail: guest VMs can directly access administrative interfaces and perform management actions; or
3199 administrative interfaces are accessible without proper authentication/authorization.

3200 6. Assessment Evidence

- 3201 • [CONFIG] Access-control configuration for administrative interfaces (users, roles, policies, network ACLs).
- 3202 • [DESIGN] Documentation of administrative functions/interfaces and the access-control model.
- 3203 • [OBSERVATION] / [LOG] Evidence of successful authenticated/authorized admin actions and failed/blocked
3204 attempts from guest VMs.

3205 6.3.1.2.2 Assessment Case AC-H-CP-ISO-002

3206 1. Assessment Reference

3207 Requirement: REQ-H-CP-ISO-002 (Elevated)

3208 2. Assessment Objective

3209 Verify that:

- 3210 1) The Hypervisor supports configuration of dedicated logical separation between administrative traffic and
3211 guest/data-plane traffic.
- 3212 2) The Hypervisor supports the use of distinct communication channels, ports or virtual networks for
3213 administrative interfaces, enabling isolation of administrative traffic from guest workloads.

3214 3. Assessment Preparation

3215 The assessment shall have access to:

- 3216 • Documentation describing how administrative interfaces are exposed (management networks, dedicated
3217 interfaces, ports, VLANs, virtual switches).
- 3218 • Documentation describing recommended practices for separating management and data/guest traffic.
- 3219 • A test configuration with:
- 3220 - at least one management network/interface;
- 3221 - at least one guest network/interface used by VMs;

3222 - the ability to inspect routing/switching and firewalling rules.

3223 **4. Assessment Activities**

3224 The assessment shall at least:

- 3225 1) Configure the Hypervisor according to documentation so that administrative interfaces use dedicated logical
3226 constructs (e.g. separate VLAN/virtual network, dedicated IP/port set, separate interface).
- 3227 2) Verify via configuration and inspection that administrative endpoints (management IP/ports, APIs, consoles)
3228 are bound only to the designated management network/ports and not exposed on guest/data-plane networks.
- 3229 3) From a guest VM network, attempt to connect to management endpoints that should be reachable only via the
3230 management network and verify that such attempts are blocked.
- 3231 4) From an administrative host placed on the management network, verify that management interfaces are
3232 reachable via the dedicated channel and that guest/data traffic does not traverse this path.

3233 **5. Assessment Verdict**

- 3234 • Pass: the Hypervisor supports and can be configured with distinct communication channels/ports/virtual
3235 networks for administrative interfaces; administrative traffic is logically separated from guest/data-plane
3236 traffic; and guest networks cannot be used to reach management endpoints in the assessed configuration.
- 3237 • Fail: no effective mechanism to separate administrative and guest/data-plane traffic; or administrative
3238 interfaces are reachable over guest networks despite intended separation.

3239 **6. Assessment Evidence**

- 3240 • [CONFIG] Network/interface configuration showing distinct management vs guest/data networks, ports and
3241 bindings.
- 3242 • [DESIGN] Documentation of control-plane vs data-plane separation mechanisms and recommended
3243 topologies.
- 3244 • [OBSERVATION] / [LOG] Connection tests from guest and admin networks showing blocked access from
3245 guest side and permitted access from management side.

3246 **6.3.1.2.3 Assessment Case AC-H-CP-ISO-003**

3247 **1. Assessment Reference**

3248 Requirement: REQ-H-CP-ISO-003 (Advanced)

3249 **2. Assessment Objective**

3250 Verify that:

- 3251 1) The Hypervisor supports cryptographically protected communication channels for all administrative interfaces.
- 3252 2) Administrative channels can be configured for mutual authentication between the Hypervisor and
3253 administrative clients/systems.
- 3254 3) Use of such channels reduces the risk of Unauthorized access or lateral movement (e.g. no cleartext admin
3255 protocols, no unauthenticated management channels).

3256 **3. Assessment Preparation**

3257 The assessment shall have access to:

- 3258 • Documentation of supported cryptographic protocols for administrative interfaces, supported cipher suites and
3259 mutual authentication options (client certificates, key-based methods).
- 3260 • Documentation of procedures for configuring certificates/keys and enabling mutual authentication.
- 3261 • A test configuration where administrative interfaces can be configured to use cryptographically protected,
3262 mutually authenticated channels, and where protocol-level inspection (e.g. via test tools) is possible.

3263 4. Assessment Activities

3264 The assessment shall at least:

- 3265 1) For each administrative interface type (e.g. web console, API endpoint, management agent), configure
3266 cryptographic protection according to product documentation.
- 3267 2) Enable mutual authentication where supported and configure at least one administrative client with valid
3268 credentials and one with invalid or missing credentials.
- 3269 3) Verify, using connection attempts and protocol inspection, that:
 - 3270 - administrative traffic is encrypted and uses the configured cryptographic protocol;
 - 3271 - servers authenticate to clients (e.g. certificate validation); and
 - 3272 - clients present valid credentials (keys/certificates) to complete administrative sessions for interfaces
3273 configured with mutual authentication.
- 3274 4) Attempt connections without valid credentials or using obsolete/weak protocol modes (if configurable) and
3275 verify they are rejected.
- 3276 5) Confirm there are no remaining administrative interfaces exposed in cleartext or without authentication in the
3277 assessed configuration.

3278 5. Assessment Verdict

- 3279 • Pass: all administrative interfaces can use cryptographically protected channels; mutual authentication can be
3280 enabled and enforced; and Unauthorized/unauthenticated access attempts over these channels are rejected.
- 3281 • Fail: some administrative interfaces cannot be protected cryptographically; mutual authentication cannot be
3282 enabled where required; or cleartext/unauthenticated management channels remain exposed.

3283 6. Assessment Evidence

- 3284 • [CONFIG] Cryptographic and authentication configuration for administrative interfaces (protocol settings,
3285 certificates/keys, mutual-auth options).
- 3286 • [DESIGN] Documentation of supported protocols, mutual authentication mechanisms and security profiles for
3287 management channels.
- 3288 • [LOG] / [OBSERVATION] Protocol traces and connection logs showing encrypted, mutually authenticated
3289 sessions, and failed attempts without valid credentials or with disallowed protocol modes.

3290 6.3.1.3 Assessment for Network Plane Separation

3291 6.3.1.3.1 Assessment Case AC-H-NP-ISO-001

3292 1. Assessment Reference

3293 Requirement: REQ-H-NP-ISO-001 (Basic)

3294 2. Assessment Objective

3295 Verify that:

- 3296 1) The Hypervisor maintains logical segregation between management, guest and host network planes.
- 3297 2) Virtual switching, routing or equivalent mechanisms ensure that traffic from one plane cannot traverse or
3298 address another plane by default.
- 3299 3) Separate network contexts can be explicitly configured for each plane.

3300 3. Assessment Preparation

3301 The assessment shall have access to:

- 3302 • Documentation of the Hypervisor's network model (e.g. virtual switches, port groups, VLANs, routing, host
3303 network interfaces) and the definitions of "management", "guest" and "host" planes.
- 3304 • Documentation describing how to configure separate network contexts for each plane.
- 3305 • A test configuration with at least:
- 3306 - one management network/plane,
- 3307 - one guest/data network/plane used by VMs,
- 3308 - host-level services (e.g. backup/monitoring) that may represent a host network plane,
- 3309 - tools to generate and observe traffic between these networks.

3310 4. Assessment Activities

3311 The assessment shall at least:

- 3312 1) Configure the Hypervisor with distinct network contexts for management, guest and host traffic as per
3313 documentation (e.g. separate port groups, VLANs or virtual networks).
- 3314 2) Verify via configuration and inspection that management interfaces are bound only to the management plane,
3315 guest VM interfaces only to the guest plane, and host services only to host/appropriate plane.
- 3316 3) From each plane (management endpoint, guest VM, host service), attempt to send traffic to addresses in the
3317 other planes without adding any explicit cross-plane rules and verify that:
- 3318 - by default, traffic from one plane does not reach or address endpoints in another plane;
- 3319 - no routing or switching entries exist that would allow cross-plane traffic beyond what is documented as
3320 default.

3321 5. Assessment Verdict

- 3322 • Pass: management, guest and host planes are logically segregated; by default, traffic from one plane cannot
3323 traverse or address another; and separate network contexts for each plane can be configured and observed.
- 3324 • Fail: planes are not effectively segregated; or default configuration allows cross-plane traffic contrary to the
3325 requirement.

3326 6. Assessment Evidence

- 3327 • [CONFIG] Network configuration (e.g. virtual switches, port groups, VLAN settings, IP addressing and
3328 bindings of interfaces to planes).
- 3329 • [DESIGN] Documentation of network plane definitions and segregation mechanisms.
- 3330 • [OBSERVATION] / [LOG] Packet captures, routing/switching tables and connection tests showing lack of
3331 cross-plane reachability in the default/segregated configuration.

3332 6.3.1.3.2 Assessment Case AC-H-NP-ISO-002

3333 1. Assessment Reference

3334 Requirement: REQ-H-NP-ISO-002 (Elevated)

3335 2. Assessment Objective

3336 Verify that:

- 3337 1) The Hypervisor supports configuration of explicit traffic filtering and access-control rules between network
3338 planes.
- 3339 2) These rules can be enforced at the Hypervisor's virtual network layer or other enforcement points under its
3340 control.
- 3341 3) Only cross-plane traffic explicitly permitted by policy (e.g. management/orchestration flows) is allowed to
3342 cross plane boundaries.

3343 3. Assessment Preparation

3344 The assessment shall have access to:

- 3345 • Documentation of mechanisms for defining traffic filtering and access-control between planes (e.g. virtual
- 3346 switch ACLs, distributed firewalls, router filters, security groups).
- 3347 • Documentation of example policies permitting specific management/orchestration flows.
- 3348 • A test configuration with:
 - 3349 - management, guest and, where applicable, host planes as in AC-H-NP-ISO-001;
 - 3350 - at least one management/orchestration system that legitimately needs to communicate across planes.

3351 4. Assessment Activities

3352 The assessment shall at least:

- 3353 1) Define and apply explicit cross-plane filtering rules at the Hypervisor virtual network layer or equivalent
- 3354 enforcement point to:
 - 3355 - allow only documented management/orchestration flows (e.g. from management plane to guest VM
 - 3356 agents on specific ports); and
 - 3357 - deny all other cross-plane traffic by default.
- 3358 2) From each plane, generate test traffic to other planes:
 - 3359 - traffic matching permitted flows;
 - 3360 - traffic that should be blocked (e.g. arbitrary guest-to-management or guest-to-host connections,
 - 3361 unexpected ports/protocols).
- 3362 3) Verify that:
 - 3363 - permitted flows succeed and follow the configured policy;
 - 3364 - non-permitted cross-plane traffic is blocked at the configured enforcement point; and
 - 3365 - changes to the policy (e.g. adding/removing a rule) have the expected effect on cross-plane
 - 3366 communication.

3367 5. Assessment Verdict

- 3368 • Pass: the Hypervisor provides configurable filtering/access-control rules between planes; these rules are
- 3369 enforced at the virtual network or equivalent layer; and only explicitly permitted cross-plane traffic is allowed
- 3370 while other cross-plane traffic is blocked.
- 3371 • Fail: cross-plane filtering cannot be configured; or enforcement is ineffective; or cross-plane traffic bypasses
- 3372 the configured policy.

3373 6. Assessment Evidence

- 3374 • [CONFIG] Defined cross-plane filtering/ACL rules and their association with management/guest/host
- 3375 networks.
- 3376 • [DESIGN] Documentation of enforcement mechanisms and their role in plane separation.
- 3377 • [OBSERVATION] / [LOG] Connection attempts and packet traces showing success of permitted flows and
- 3378 blocking of non-permitted cross-plane traffic.

3379 6.3.1.3.3 Assessment Case AC-H-NP-ISO-003

3380 1. Assessment Reference

3381 Requirement: REQ-H-NP-ISO-003 (Advanced)

3382 2. Assessment Objective

3383 Verify that:

- 3384 1) The Hypervisor supports isolation of management traffic using either:
 - 3385 - dedicated physical interfaces; or
 - 3386 - cryptographically protected network channels.
- 3387 2) Management traffic is protected against interception or modification by guest or host workloads.
- 3388 3) Where dedicated physical interfaces are not used, cryptographic protection provides confidentiality, integrity

3389 and authentication equivalent to a dedicated management network.

3390 3. Assessment Preparation

3391 The assessment shall have access to:

- 3392 • Documentation describing supported approaches for isolating management traffic (e.g. dedicated NICs/ports,

3393 out-of-band networks, IPsec/TLS tunnels, overlay management networks).
- 3394 • Documentation of cryptographic protocols and authentication mechanisms used when dedicated physical

3395 interfaces are not available.
- 3396 • A test configuration with:
 - 3397 - either a dedicated physical management interface/network or a cryptographically protected management

3398 channel;
 - 3399 - guest workloads generating network traffic;
 - 3400 - the ability to capture and inspect traffic from guest-accessible paths.

3401 4. Assessment Activities

3402 Depending on the isolation method, the assessment shall at least:

3403 Dedicated physical interface case

- 3404 1) Configure the Hypervisor so that all management interfaces use a dedicated physical NIC or physical

3405 management network segment.
- 3406 2) Verify via configuration and inspection that:
 - 3407 - management traffic egresses/ingresses only via the dedicated physical interface/network;
 - 3408 - guest VMs and host non-management workloads do not have access to that interface/network (e.g. no

3409 virtual NICs or routes pointing to it).
- 3410 3) From guest networks and host non-management interfaces, capture traffic and verify that management traffic

3411 is not visible on those paths.

3412 Cryptographically protected channel case (when dedicated physical interfaces are not used)

- 3413 4) Configure management traffic over a cryptographically protected channel (e.g. IPsec, TLS VPN, mutually

3414 authenticated TLS overlay) between the Hypervisor and management endpoints.
- 3415 5) Verify via protocol inspection and logs that:
 - 3416 - management traffic is encrypted and integrity-protected;
 - 3417 - mutual authentication between Hypervisor and management entities is in place.
- 3418 6) From guest networks and host non-management workloads, capture traffic and verify that:
 - 3419 - management traffic is only observable in encrypted form (no cleartext management protocols); and
 - 3420 - attempts to inject or modify management traffic are rejected or fail to affect the management channel.

3421 5. Assessment Verdict

- 3422 • Pass: the Hypervisor supports either dedicated physical interfaces or cryptographically protected channels to
3423 isolate management traffic; management traffic is not exposed in cleartext or modifiable by guest/host
3424 workloads; and cryptographic protection, where used instead of dedicated interfaces, provides confidentiality,
3425 integrity and authentication equivalent to a dedicated management network.
- 3426 • Fail: management traffic is visible or modifiable from guest or host workloads; dedicated management
3427 interfaces or equivalent cryptographic protection cannot be configured; or cleartext management flows appear
3428 on shared paths without adequate isolation.

3429 6. Assessment Evidence

- 3430 • [CONFIG] Network and interface configuration showing use of dedicated management interfaces or
3431 cryptographically protected management channels.
- 3432 • [DESIGN] Documentation of management traffic isolation model and, where applicable, cryptographic
3433 protocols and authentication methods.
- 3434 • [PLT-CAP] Evidence of platform/network capabilities used (e.g. dedicated NICs, VPN/IPsec support).
- 3435 • [OBSERVATION] Packet captures and logs demonstrating that management traffic uses the isolated interface
3436 or encrypted channel and is not exposed to guest or non-management workloads.

3437 6.3.1.4 Assessment for Boot Chain Integrity Verification

3438 6.3.1.4.1 Assessment Case AC-H-B-INT-001

3439 1. Assessment Reference

3440 Requirement: REQ-H-B-INT-001 (Basic)

3441 2. Assessment Objective

3442 Verify that:

- 3443 1) The Hypervisor implements an integrity verification mechanism for its executable core (kernel or microkernel)
3444 based on a trusted reference value.
- 3445 2) Integrity verification of the executable core is invoked before any guest workloads are initialized or
3446 management interfaces are exposed.
- 3447 3) If integrity verification of the executable core fails, the Hypervisor prevents initialization of guest workloads
3448 and activation of management interfaces.

3449 3. Assessment Preparation

3450 The assessment shall have access to:

- 3451 • Documentation describing the executable core integrity verification mechanism, including how trusted
3452 reference values are established and stored.
- 3453 • Documentation of the boot sequence, indicating when integrity verification occurs relative to guest
3454 initialization and management interface activation.
- 3455 • A test configuration where the Hypervisor can be booted with:
 - 3456 - an intact, correctly signed/verified executable core; and
 - 3457 - a deliberately altered or invalid executable core (or equivalent method) to trigger verification failure,
3458 consistent with product guidance.

3459 4. Assessment Activities

3460 The assessment shall at least:

- 3461 1) Review documentation and configuration to identify the integrity verification mechanism, the trusted reference
3462 values, and the point in the boot sequence at which verification is invoked.

- 3463 2) Boot the Hypervisor with an intact executable core and verify that:
- 3464 - integrity verification is invoked as documented;
- 3465 - the Hypervisor completes initialization; and
- 3466 - guest workloads and management interfaces can be initialized/activated normally.
- 3467 3) Boot the Hypervisor with an executable core that fails integrity verification (e.g. corrupted or otherwise invalid
- 3468 according to the documented mechanism) and verify that:
- 3469 - the integrity verification detects the failure; and
- 3470 - guest workloads are not initialized and management interfaces are not activated.

3471 5. Assessment Verdict

- 3472 • Pass: executable-core integrity verification is based on a trusted reference value; it is invoked before guest and
- 3473 management initialization; and failure of verification prevents initialization of guest workloads and activation
- 3474 of management interfaces.
- 3475 • Fail: integrity verification is not implemented or not based on a trusted reference; it does not occur before
- 3476 guest/management initialization; or verification failure does not block guest and management initialization.

3477 6. Assessment Evidence

- 3478 • [DESIGN] Documentation of the executable core integrity verification mechanism and boot sequence.
- 3479 • [CONFIG] Boot and verification configuration, including references to trusted reference values.
- 3480 • [OBSERVATION] / [LOG] Boot logs or traces showing successful verification and normal startup, and
- 3481 logs/traces showing detection of an integrity failure and prevention of guest/management initialization.

3482 6.3.1.4.2 Assessment Case AC-H-B-INT-002

3483 1. Assessment Reference

3484 Requirement: REQ-H-B-INT-002 (Elevated)

3485 2. Assessment Objective

3486 Verify that:

- 3487 1) The Hypervisor implements mechanisms to participate in a verifiable chain of trust for early boot components
- 3488 that load the Hypervisor.
- 3489 2) Each boot stage delivered as part of the Hypervisor product, including at least any bootloader delivered with
- 3490 the Hypervisor and the Hypervisor executable core, validates the integrity and authenticity of the next stage
- 3491 before transferring control.
- 3492 3) If this validation fails at any such stage, the Hypervisor prevents initialization of guest workloads and
- 3493 activation of management interfaces.

3494 3. Assessment Preparation

3495 The assessment shall have access to:

- 3496 • Documentation of the boot chain architecture for components delivered as part of the Hypervisor product (e.g.
- 3497 bootloader(s), Hypervisor kernel/microkernel, early management services).
- 3498 • Documentation of how each boot stage validates integrity and authenticity of the next stage, including where
- 3499 the trust anchors or reference values reside and how they integrate into a platform secure-boot or equivalent
- 3500 verified boot chain.
- 3501 • A test configuration that allows:
- 3502 - a normal boot with all stages intact; and

- 3503 - a controlled failure in integrity/authenticity validation of one of the Hypervisor-delivered boot stages
 3504 (e.g. modified image, invalid signature), consistent with product guidance.

3505 **4. Assessment Activities**

3506 The assessment shall at least:

- 3507 1) Review documentation to identify all boot stages delivered as part of the Hypervisor product and the validation
 3508 performed at each stage (e.g. signature check, hash comparison).
- 3509 2) In a normal configuration, boot the system and verify from logs or diagnostic outputs that each Hypervisor-
 3510 delivered boot stage validates the integrity/authenticity of the next stage before passing control, and that the
 3511 overall boot completes with guest and management initialization available.
- 3512 3) Introduce a deliberate integrity/authenticity failure in one Hypervisor-delivered boot stage (e.g. corrupted or
 3513 unsigned image, or image signed with an untrusted key) and boot the system.
- 3514 4) Verify that:
- 3515 - the failing validation is detected at the appropriate stage; and
- 3516 - guest workloads are not initialized and management interfaces are not activated following the failed
 3517 validation.

3518 **5. Assessment Verdict**

- 3519 • Pass: all Hypervisor-delivered boot stages validate the integrity and authenticity of the next stage before
 3520 transferring control; validation failures at any such stage are detected; and, when such failures occur, the
 3521 Hypervisor prevents initialization of guest workloads and activation of management interfaces.
- 3522 • Fail: some Hypervisor-delivered boot stages do not validate the next stage; validation failures are not detected;
 3523 or validation failures do not prevent guest/management initialization.

3524 **6. Assessment Evidence**

- 3525 • [DESIGN] Documentation of the Hypervisor-related boot chain and per-stage validation mechanisms,
 3526 including integration into a platform secure/verified boot chain.
- 3527 • [CONFIG] Boot configuration and key/certificate or reference-value configuration for Hypervisor-delivered
 3528 stages.
- 3529 • [OBSERVATION] / [LOG] Boot logs or traces showing successful chained validation of stages and separate
 3530 logs/traces demonstrating detection of a validation failure and corresponding prevention of guest/management
 3531 initialization.

3532 **6.3.1.4.3 Assessment Case AC-H-B-INT-003**

3533 **1. Assessment Reference**

3534 Requirement: REQ-H-B-INT-003 (Advanced)

3535 **2. Assessment Objective**

3536 Verify that:

- 3537 1) The Hypervisor relies on integrity and authenticity information for the boot chain on which it depends, as
 3538 provided by verified boot mechanisms using hardware or software roots of trust.
- 3539 2) This information covers components used to load the Hypervisor, including the bootloader, the Hypervisor
 3540 kernel or microkernel, and management services.
- 3541 3) The Hypervisor prevents initialization of guest workloads and management services if the verification status of
 3542 any component in this boot chain cannot be established or is reported as failed.

3543 3. Assessment Preparation

3544 The assessment shall have access to:

- 3545 • Documentation of the verified boot mechanism and roots of trust (hardware or software) used to provide
3546 integrity and authenticity information for the Hypervisor boot chain components.
- 3547 • Documentation describing how the Hypervisor consumes or relies on verification status (e.g. secure boot APIs,
3548 measurement/attestation interfaces, verified boot status flags) for the bootloader, Hypervisor core and
3549 management services.
- 3550 • A test configuration where:
 - 3551 - the platform verified boot mechanism is enabled and correctly configured;
 - 3552 - the Hypervisor is deployed with all boot chain components covered by the verified boot mechanism;
 - 3553 - a failure or indeterminate verification status for at least one boot-chain component can be induced in a
3554 controlled manner, consistent with product guidance.

3555 4. Assessment Activities

3556 The assessment shall at least:

- 3557 1) Review documentation to confirm which boot-chain components (bootloader, Hypervisor kernel/microkernel,
3558 management services) are covered by verified boot and how verification status is made available to the
3559 Hypervisor.
- 3560 2) In a correctly configured environment, boot the system and verify that:
 - 3561 - verification status for each relevant boot-chain component is successfully established by the verified boot
3562 mechanism;
 - 3563 - the Hypervisor relies on this status when completing its initialization; and
 - 3564 - guest workloads and management services are initialized only when all required components are in a
3565 verified-good state.
- 3566 3) Introduce a controlled failure or indeterminate status for one boot-chain component (e.g. a component failing
3567 verification according to the verified boot mechanism) and reboot.
- 3568 4) Verify that:
 - 3569 - the Hypervisor detects that verification status for that component is failed or cannot be established; and
 - 3570 - the Hypervisor prevents initialization of guest workloads and management services in this condition.

3571 5. Assessment Verdict

- 3572 • Pass: the Hypervisor relies on verified boot information from hardware or software roots of trust for the
3573 bootloader, Hypervisor core and management services; and when verification status for any such component is
3574 missing or failed, guest workloads and management services are not initialized.
- 3575 • Fail: verified boot information is not used for all required components; the Hypervisor initializes
3576 guest/management services despite failed or indeterminate verification status; or the Hypervisor cannot obtain
3577 or act on verification status for the boot chain.

3578 6. Assessment Evidence

- 3579 • [DESIGN] Documentation of the verified boot mechanism, roots of trust and the way verification status for
3580 boot-chain components is exposed to and used by the Hypervisor.
- 3581 • [PLT-CAP] Evidence of platform capabilities providing verified boot / root-of-trust support used by the
3582 Hypervisor.
- 3583 • [CONFIG] Verified boot configuration and Hypervisor integration settings.

- 3584 • [OBSERVATION] / [LOG] Boot logs or status outputs showing verified-good operation and, separately,
3585 logs/status showing detection of failed/unknown verification and prevention of guest/management
3586 initialization.

3587 6.3.1.4.4 Assessment Case AC-H-B-INT-004

3588 1. Assessment Reference

3589 Requirement: REQ-H-B-INT-004 (Advanced, conditional "may")

3590 Applicability:

3591 This assessment case applies only when the manufacturer claims that the Hypervisor supports hardware-based roots of
3592 trust for boot-chain integrity and authenticity verification in accordance with REQ-H-B-INT-003.

3593 2. Assessment Objective

3594 Verify that, when claimed:

- 3595 1) The Hypervisor supports the use of hardware-based roots of trust (e.g. secure elements, dedicated trust
3596 anchors, or equivalent components) to perform boot-chain integrity and authenticity verification in accordance
3597 with REQ-H-B-INT-003.
- 3598 2) These hardware-based roots of trust provide protection against Unauthorized modification of the boot chain.

3599 3. Assessment Preparation

3600 The assessment shall have access to:

- 3601 • Documentation describing the hardware-based root(s) of trust used for boot-chain verification (e.g. platform
3602 security module, secure element, dedicated trust anchor).
- 3603 • Documentation explaining how the Hypervisor and its boot-chain components integrate with these hardware-
3604 based roots of trust to perform integrity and authenticity checks.
- 3605 • A test configuration on a platform that provides the claimed hardware-based root of trust and where boot-chain
3606 verification using this root can be exercised.

3607 4. Assessment Activities

3608 The assessment shall at least:

- 3609 1) Review documentation to confirm which hardware-based root(s) of trust are used, which boot-chain
3610 components are covered, and how the Hypervisor relies on verification results from these components in line
3611 with REQ-H-B-INT-003.
- 3612 2) In a normal configuration, boot the system and verify (e.g. via status interfaces or logs) that the hardware-
3613 based root of trust participates in verification of the bootloader, Hypervisor core and associated components as
3614 claimed.
- 3615 3) Where feasible and consistent with product guidance, attempt to introduce an Unauthorized modification to a
3616 boot-chain component protected by the hardware-based root of trust (e.g. tampered image not signed with the
3617 expected key) and reboot.
- 3618 4) Verify that:
 - 3619 - the hardware-based verification detects the modification; and
 - 3620 - consistent with REQ-H-B-INT-003, the Hypervisor does not initialize guest workloads or management
3621 services when the hardware-based verification reports failure or an untrusted state.

3622 5. Assessment Verdict

- 3623 • Pass: when support is claimed, hardware-based roots of trust are used for boot-chain integrity and authenticity
3624 verification in conjunction with the verified boot mechanisms required by REQ-H-B-INT-003; and
3625 Unauthorized modifications to protected boot-chain components are detected and prevent guest/management
3626 initialization.

- 3627 • Fail: claimed hardware-based support is not actually used for boot-chain verification; Unauthorized
- 3628 modifications to protected components are not detected; or the Hypervisor does not react as required when
- 3629 hardware-based verification fails.

3630 6. Assessment Evidence

- 3631 • [DESIGN] Documentation of integration between the Hypervisor boot chain and hardware-based roots of trust
- 3632 used for verification.
- 3633 • [PLT-CAP] Evidence of the presence and configuration of the hardware-based root(s) of trust (e.g. platform
- 3634 security module capabilities, secure element configuration).
- 3635 • [CONFIG] Boot and verification configuration showing use of hardware-based trust anchors for the boot
- 3636 chain.
- 3637 • [OBSERVATION] / [LOG] Boot logs or status outputs demonstrating normal verified operation using
- 3638 hardware roots of trust and, where tested, detection of tampered components and resulting prevention of
- 3639 guest/management initialization.

3640 6.3.1.5 Assessment for Guest VM Image Integrity Verification

3641 6.3.1.5.1 Assessment Case AC-H-IMG-INT-001

3642 1. Assessment Reference

3643 Requirement: REQ-H-IMG-INT-001 (Basic)

3644 2. Assessment Objective

3645 Verify that:

- 3646 1) The Hypervisor implements an integrity verification mechanism for guest VM images based on trusted
- 3647 reference values.
- 3648 2) Integrity verification is automatically invoked before executing or provisioning any guest VM.
- 3649 3) If integrity verification for a guest VM image fails, the Hypervisor prevents execution or instantiation of that
- 3650 guest VM.

3651 3. Assessment Preparation

3652 The assessment shall have access to:

- 3653 • Documentation describing the guest VM image integrity verification mechanism, including how trusted
- 3654 reference values (e.g. hashes, manifests) are established and stored.
- 3655 • Documentation of the VM provisioning/startup sequence, indicating when image integrity verification is
- 3656 invoked.
- 3657 • A test environment with:
 - 3658 - at least one repository or storage location for guest VM images;
 - 3659 - at least one valid guest VM image;
 - 3660 - the ability to create a deliberately modified (integrity-failing) copy of a guest VM image in a controlled
 - 3661 manner.

3662 4. Assessment Activities

3663 The assessment shall at least:

- 3664 1) Review documentation and configuration to identify:
 - 3665 - which guest VM images are subject to integrity verification;
 - 3666 - the trusted reference values used;

- 3667 - the point in the provisioning/startup process where verification occurs.
- 3668 2) With an unmodified guest VM image, request provisioning/startup of the VM and verify, using logs or status
3669 outputs, that:
- 3670 - integrity verification is invoked automatically;
- 3671 - the VM is instantiated or started successfully when verification succeeds.
- 3672 3) Replace or alter a guest VM image so that it fails integrity verification according to the documented
3673 mechanism, then request provisioning/startup of that VM. Verify that:
- 3674 - the Hypervisor reports an integrity verification failure;
- 3675 - execution or instantiation of that VM is prevented.

3676 5. Assessment Verdict

- 3677 • Pass: guest VM image integrity verification based on trusted reference values is automatically invoked before
3678 VM provisioning/execution; and a verification failure for an image prevents that VM from being instantiated
3679 or executed.
- 3680 • Fail: integrity verification is not implemented or not based on trusted reference values; it is not invoked
3681 automatically; or VMs can be instantiated despite integrity verification failures.

3682 6. Assessment Evidence

- 3683 • [DESIGN] Documentation of the image integrity verification mechanism and its place in the VM lifecycle.
- 3684 • [CONFIG] Configuration showing enabled integrity verification and reference values for guest VM images.
- 3685 • [LOG] / [OBSERVATION] Records of successful verification and VM startup for a valid image, and records
3686 showing verification failure and blocked startup for a modified image.

3687 6.3.1.5.2 Assessment Case AC-H-IMG-INT-002

3688 1. Assessment Reference

3689 Requirement: REQ-H-IMG-INT-002 (Elevated)

3690 2. Assessment Objective

3691 Verify that:

- 3692 1) The Hypervisor verifies both integrity and authenticity of guest VM images before execution or provisioning.
- 3693 2) Authenticity verification establishes that the image originates from a trusted source and has not been altered
3694 since approval.
- 3695 3) The Hypervisor prevents execution or instantiation of any guest VM image whose integrity or authenticity
3696 cannot be successfully verified.

3697 3. Assessment Preparation

3698 The assessment shall have access to:

- 3699 • Documentation describing integrity and authenticity verification for guest VM images, including trust model
3700 (e.g. signatures, certificate chains, trust stores, or equivalent cryptographic schemes).
- 3701 • Documentation of how trusted publishers/sources are represented and configured.
- 3702 • A test environment with:
- 3703 - one or more guest VM images created and "approved" using the trusted process (e.g. signed or otherwise
3704 bound to a trusted source);
- 3705 - images modified or created so that integrity or authenticity verification will fail (e.g. tampered content,
3706 invalid or missing signature, signature under an untrusted key), prepared in a controlled way.

3707 **4. Assessment Activities**

3708 The assessment shall at least:

- 3709 1) Review documentation and configuration to identify:
- 3710 - how integrity and authenticity checks are combined;
- 3711 - what constitutes a "trusted source" and how trust anchors or approval data are configured.
- 3712 2) For an image that is correctly produced and approved:
- 3713 - request provisioning/startup of a VM from that image;
- 3714 - verify that integrity and authenticity verification succeed;
- 3715 - confirm that the VM is instantiated/executed.
- 3716 3) For images that are invalid in different ways (e.g. modified after approval, missing or invalid signature, signed
- 3717 by an untrusted source):
- 3718 - request provisioning/startup;
- 3719 - verify that the Hypervisor reports integrity/authenticity verification failure or inability to verify;
- 3720 - confirm that execution/instantiation of such VMs is prevented.

3721 **5. Assessment Verdict**

- 3722 • Pass: integrity and authenticity of guest VM images are verified prior to provisioning/execution; only images
- 3723 from trusted sources that have not been altered since approval can be instantiated; and images whose
- 3724 integrity/authenticity cannot be verified are blocked.
- 3725 • Fail: authenticity is not verified; untrusted or modified images can be instantiated; or failure to verify
- 3726 integrity/authenticity does not prevent VM instantiation.

3727 **6. Assessment Evidence**

- 3728 • [DESIGN] Documentation of the guest VM image trust model, including integrity and authenticity verification
- 3729 and trusted sources.
- 3730 • [CONFIG] Trust store or equivalent configuration defining trusted publishers/sources and related policies.
- 3731 • [LOG] / [OBSERVATION] Records showing successful verification and startup for approved images and
- 3732 failure/blocked startup for images with invalid integrity or authenticity.

3733 **6.3.1.5.3 Assessment Case AC-H-IMG-INT-003**

3734 **1. Assessment Reference**

3735 Requirement: REQ-H-IMG-INT-003 (Advanced)

3736 **2. Assessment Objective**

3737 Verify that:

- 3738 1) The Hypervisor uses integrity and authenticity verification for all guest VM images before execution or
- 3739 provisioning, based on trust anchors stored as protected trust material.
- 3740 2) Protected trust material (keys, certificates or other trust anchors) is stored and managed so that it is not
- 3741 modifiable during normal operational state via standard administrative or guest-facing interfaces.
- 3742 3) If verification of a guest VM image using the protected trust material fails, or the verification status cannot be
- 3743 established, the Hypervisor prevents execution or instantiation of that guest VM.

3744 **3. Assessment Preparation**

3745 The assessment shall have access to:

- 3746
- Documentation describing:
 - 3747 - the nature and storage of protected trust material (e.g. trust stores, key vaults, secure configuration
 - 3748 objects);
 - 3749 - the protections that prevent modification of trust material during normal operation;
 - 3750 - the process for controlled update of trust material (e.g. dedicated security management procedure).
 - A test environment with:
 - 3752 - at least one valid guest VM image verified against the protected trust material;
 - 3753 - the ability to simulate conditions where verification fails or the trust anchors no longer match the image
 - 3754 (in a controlled manner);
 - 3755 - an administrative interface reflective of "normal operational state".

3756 **4. Assessment Activities**

3757 The assessment shall at least:

- 3758 1) Review documentation and configuration to confirm:
- 3759 - which trust anchors are used to verify guest VM images;
 - 3760 - where and how this trust material is stored;
 - 3761 - which operations are permitted in normal operational state versus dedicated security update procedures.
- 3762 2) In normal operation, attempt to modify or replace trust material through standard administrative channels (e.g. routine configuration interfaces) and verify that such modifications are prevented or require a controlled security management process.
- 3763
- 3764
- 3765 3) With valid trust material and a correctly approved guest VM image, request provisioning/startup and verify
- 3766 that integrity/authenticity verification succeeds and the VM is instantiated.
- 3767 4) Create a test condition where verification fails or becomes indeterminate for a guest VM image when checked
- 3768 against the protected trust material (e.g. image altered after approval, trust anchor no longer matches image),
- 3769 and request provisioning/startup. Verify that:
- 3770 - verification failure or inability to establish verification status is detected;
 - 3771 - the Hypervisor prevents execution or instantiation of that VM.

3772 **5. Assessment Verdict**

- 3773 • Pass: all guest VM images are verified for integrity and authenticity against protected trust material; trust
- 3774 anchors are not modifiable during normal operation; and images that fail or cannot be verified against this
- 3775 material are prevented from execution/instantiation.
- 3776 • Fail: trust material is modifiable in normal operation without a controlled process; images can be instantiated
- 3777 without verification against protected trust anchors; or failed/indeterminate verification does not prevent VM
- 3778 instantiation.

3779 **6. Assessment Evidence**

- 3780 • [DESIGN] Documentation of protected trust material, its storage and update procedures, and its use in image
- 3781 verification.
- 3782 • [CONFIG] Configuration of trust stores or equivalent structures, including permissions or controls that protect
- 3783 them in normal operation.
- 3784 • [LOG] / [OBSERVATION] Evidence of successful verification and startup for valid images, and evidence of
- 3785 blocked startup when verification against protected trust material fails or is not possible.

3786 6.3.1.5.4 6.3.1.5.3 Assessment Case AC-H-IMG-INT-004

3787 **Applicability:**

3788 This assessment case applies only when the manufacturer claims that the Hypervisor supports anchoring guest VM
3789 image integrity/authenticity verification in hardware-based roots of trust.

3790 **1. Assessment Reference**

3791 Requirement: REQ-H-IMG-INT-004 (Advanced, conditional "may")

3792 **2. Assessment Objective**

3793 Verify that, when claimed:

- 3794 1) The Hypervisor can anchor guest VM image integrity and authenticity verification in hardware-based roots of
3795 trust (e.g. dedicated secure hardware modules or trust anchors).
- 3796 2) These hardware-based roots of trust provide protected storage for trust material and resistance to Unauthorized
3797 modification of verification keys, in line with the trust anchors used for REQ-H-IMG-INT-003.

3798 **3. Assessment Preparation**

3799 The assessment shall have access to:

- 3800 • Documentation describing the hardware-based root(s) of trust used for guest VM image verification (e.g.
3801 hardware security module, platform trust anchor) and how trust material is stored and accessed.
- 3802 • Documentation explaining how the Hypervisor integrates these hardware-based roots into the guest VM image
3803 verification process.
- 3804 • A test environment on a platform that provides the claimed hardware-based roots of trust and where image
3805 verification using these roots can be exercised.

3806 **4. Assessment Activities**

3807 The assessment shall at least:

- 3808 1) Review documentation to identify:
 - 3809 - which hardware-based roots of trust are used;
 - 3810 - which keys/certificates or trust anchors relevant to guest VM image verification are stored there;
 - 3811 - how these roots interact with the image verification process described for REQ-H-IMG-INT-003.
- 3812 2) In a normal configuration, provision/start a VM from an approved guest VM image and verify, using logs or
3813 status outputs, that:
 - 3814 - verification uses trust material stored in the hardware-based root of trust;
 - 3815 - verification succeeds and the VM is instantiated.
- 3816 3) Where feasible and consistent with vendor guidance, attempt to simulate an attack or misconfiguration that
3817 would require modification or replacement of verification keys (e.g. attempting to change trust material
3818 through normal interfaces, or using an image signed with an unregistered key) and verify that:
 - 3819 - trust material within the hardware-based root cannot be altered via normal operational interfaces;
 - 3820 - images requiring untrusted or altered keys are rejected and the associated VMs are not instantiated.

3821 **5. Assessment Verdict**

- 3822 • Pass: when support is claimed, guest VM image verification can be anchored in hardware-based roots of trust
3823 that protect verification keys and trust material from Unauthorized modification; and images that do not verify
3824 against these hardware-protected trust anchors cannot be instantiated.
- 3825 • Fail: claimed hardware-backed support is not actually used for image verification; trust material in the
3826 hardware-based root can be altered through normal interfaces; or images that should fail verification against
3827 hardware-protected keys can still be instantiated.

3828 6. Assessment Evidence

- 3829 • [DESIGN] Documentation of integration between Hypervisor guest VM image verification and hardware-
3830 based roots of trust.
- 3831 • [PLT-CAP] Evidence of the hardware-based trust components present and configured on the platform.
- 3832 • [CONFIG] Configuration showing use of hardware-based trust anchors for image verification.
- 3833 • [LOG] / [OBSERVATION] Logs or status outputs demonstrating normal verification using hardware-
3834 protected trust material and rejection of images that cannot be verified against that material.

3835 6.3.1.6 Assessment for Runtime Integrity Protection

3836 6.3.1.6.1 Assessment Case AC-H-RP-INT-002

3837 1. Assessment Reference

3838 Requirement: REQ-H-RP-INT-002 (Elevated)

3839 2. Assessment Objective

3840 Verify that:

- 3841 1) The Hypervisor enforces integrity protection for its configuration and control plane so that only authenticated
3842 and authorized modifications can be applied.
- 3843 2) The Hypervisor generates audit events for any security-relevant changes to configuration or internal control
3844 logic.

3845 3. Assessment Preparation

3846 The assessment shall have access to:

- 3847 • Documentation describing:
 - 3848 - the Hypervisor configuration and control-plane model;
 - 3849 - which configuration items and control-plane elements are considered security-relevant;
 - 3850 - which roles or identities are permitted to modify them;
 - 3851 - the interfaces (local/remote) through which such modifications are performed.
- 3852 • Documentation of the auditing or logging mechanism for configuration and control-plane changes.
- 3853 • A test configuration with:
 - 3854 - at least one administrative account/role with privileges to change security-relevant configuration or
3855 control-plane logic;
 - 3856 - at least one non-privileged or Unauthorized account/role;
 - 3857 - access to configuration/control-plane interfaces under test;
 - 3858 - audit/logging enabled and accessible for inspection.

3859 4. Assessment Activities

3860 The assessment shall at least:

- 3861 1) Review documentation and Hypervisor settings to identify:
 - 3862 - the set of security-relevant configuration parameters and control-plane elements;
 - 3863 - the authentication and authorization mechanisms governing their modification;
 - 3864 - how audit events are generated and recorded for such changes.

- 3865 2) Using an authenticated and authorized administrative account:
- 3866 - perform representative security-relevant configuration changes and, where supported, changes to control-
- 3867 plane logic (for example enabling/disabling a security feature, changing access-control rules, modifying
- 3868 isolation-relevant parameters);
- 3869 - verify that the Hypervisor accepts these changes;
- 3870 - verify that corresponding audit events are generated, including at least information on the acting identity,
- 3871 the nature of the change, and time of the event.
- 3872 3) Using an unauthenticated context, a non-privileged account, or an interface that should not permit these
- 3873 changes:
- 3874 - attempt to perform the same or equivalent security-relevant modifications;
- 3875 - verify that these attempts are rejected and do not alter configuration or control-plane state;
- 3876 - verify that failed or denied modification attempts are logged as appropriate (if specified in the product
- 3877 behaviour).

3878 5. Assessment Verdict

- 3879 • Pass: only authenticated and authorized entities can apply security-relevant configuration or control-plane
- 3880 changes; attempts by unauthenticated or Unauthorized entities are rejected without altering state; and audit
- 3881 events are generated for security-relevant changes to configuration or internal control logic.
- 3882 • Fail: security-relevant changes can be made without proper authentication or authorization; or such changes do
- 3883 not result in audit events; or Unauthorized attempts are not prevented.

3884 6. Assessment Evidence

- 3885 • [DESIGN] Documentation of the configuration and control-plane model, including the definition of security-
- 3886 relevant changes and the access-control model.
- 3887 • [CONFIG] Hypervisor configuration showing role/permission assignments for configuration and control-plane
- 3888 interfaces, and logging/audit configuration.
- 3889 • [LOG] Audit or event logs capturing successful authorized changes and rejected Unauthorized attempts.
- 3890 • [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating
- 3891 acceptance of authorized changes and rejection of Unauthorized ones.

3892 6.3.1.6.2 Assessment Case AC-H-RP-INT-003

3893 1. Assessment Reference

3894 Requirement: REQ-H-RP-INT-003 (Advanced)

3895 2. Assessment Objective

3896 Verify that:

- 3897 1) The Hypervisor maintains the integrity of its security-critical runtime components and detects Unauthorized
- 3898 modification of configuration state or control logic used to enforce security policies or isolation.
- 3899 2) The Hypervisor maintains a trusted baseline of the integrity state of these security-critical runtime components
- 3900 and performs periodic or continuous integrity validation against this baseline.
- 3901 3) Upon detecting a deviation from the trusted baseline, the Hypervisor automatically triggers protective actions
- 3902 (such as isolating the affected component, disabling affected functions, or restoring from a trusted state) and
- 3903 generates a security alert that cannot be suppressed by the affected component.

3904 **3. Assessment Preparation**

3905 The assessment shall have access to:

- 3906
- Documentation that:
 - 3907 - identifies which runtime components are considered security-critical for policy enforcement and isolation
 - 3908 (for example security modules, policy engines, core control-plane daemons);
 - 3909 - describes how the trusted baseline for these components is established, stored, and updated;
 - 3910 - explains whether integrity validation is periodic, continuous, or event-driven, and how it is configured;
 - 3911 - describes the set of protective actions that can be triggered when deviations are detected;
 - 3912 - describes the alerting mechanism and how alerts are exposed to management/monitoring systems.
 - A Hypervisor test deployment where runtime integrity validation is enabled and its status can be observed (for
 - 3913 example through logs, status commands, or telemetry).
 - A controlled means, consistent with vendor guidance, to simulate or induce an integrity deviation affecting at
 - 3915 least one security-critical runtime component or its configuration state (for example a supported test hook,
 - 3916 debug mode, test image, or vendor-provided integrity-test procedure).
 - 3917

3918 **4. Assessment Activities**

3919 The assessment shall at least:

- 3920 1) Review documentation and configuration to confirm:
- 3921 - the list or classes of security-critical runtime components covered by integrity validation;
 - 3922 - the method used to create and protect the trusted baseline;
 - 3923 - the frequency or trigger conditions for integrity checks;
 - 3924 - the configured protective actions and alerting behaviour.
- 3925 2) In a normal (non-tampered) configuration:
- 3926 - operate the Hypervisor under representative load for a suitable period;
 - 3927 - verify from logs, status outputs, or telemetry that integrity validation is active (for example periodic
 - 3928 checks, continuous monitoring, or attestation events for the defined components);
 - 3929 - confirm that no integrity deviation alerts are present when the baseline is intact.
- 3930 3) Introduce a controlled integrity deviation affecting a security-critical runtime component or its configuration
- 3931 state, using a method sanctioned for testing:
- 3932 - allow the Hypervisor to perform its runtime validation;
 - 3933 - verify that the deviation from the trusted baseline is detected;
 - 3934 - verify that the documented protective actions are automatically triggered (for example isolation or
 - 3935 disablement of the affected component, restoration from a trusted baseline, or equivalent measures);
 - 3936 - verify that a security alert is generated and remains visible via Hypervisor or management/monitoring
 - 3937 interfaces, and that this alert cannot be cleared or suppressed solely by the affected component.

3938 **5. Assessment Verdict**

- 3939
- Pass: security-critical runtime components are covered by a trusted baseline; the Hypervisor performs periodic
 - 3940 or continuous validation against this baseline; controlled deviations are detected; documented protective
 - 3941 actions are automatically applied; and a security alert is generated that the affected component cannot
 - 3942 suppress.

- 3943 • Fail: runtime integrity validation does not cover security-critical components; is not active or observable;
- 3944 deviations are not detected; protective actions are not triggered; or alerts can be suppressed or removed by the
- 3945 affected component.

3946 6. Assessment Evidence

- 3947 • [DESIGN] Documentation of security-critical runtime components, trusted baseline creation and protection,
- 3948 runtime integrity mechanisms, and protective/alerting behaviour.
- 3949 • [CONFIG] Configuration enabling runtime integrity validation, including monitored components, check
- 3950 parameters, and alerting/protection settings.
- 3951 • [LOG] Integrity-check logs and security alerts from normal operation and from the controlled deviation
- 3952 scenario, including timestamps and component identifiers.
- 3953 • [OBSERVATION] Test records or monitoring outputs showing integrity checks in operation, detection of the
- 3954 induced deviation, execution of protective actions, and persistence of the generated alert.
- 3955 • [PLT-CAP] Where hardware- or platform-assisted runtime integrity mechanisms are used, evidence of the
- 3956 underlying platform capabilities and their configuration.

3957 6.3.1.6.3 Assessment Case AC-H-RP-INT-004

3958 **Applicability:**

3959 This assessment case applies only where the manufacturer claims that the Hypervisor supports hardware-based platform
 3960 mechanisms to strengthen runtime integrity validation and detection of Unauthorized modification of security-critical
 3961 runtime components.

3962 **1. Assessment Reference**

3963 Requirement: REQ-H-RP-INT-004 (Advanced, conditional "may")

3964 **2. Assessment Objective**

3965 Verify that, when claimed:

- 3966 1) The Hypervisor supports the use of hardware-based platform mechanisms (such as dedicated secure hardware
- 3967 modules, hardware-supported memory protection, or equivalent platform features) to strengthen runtime
- 3968 integrity validation and detection of Unauthorized modification of security-critical runtime components.
- 3969 2) These mechanisms provide additional assurance of runtime integrity beyond software-only mechanisms.

3970 **3. Assessment Preparation**

3971 The assessment shall have access to:

- 3972 • Documentation describing:
 - 3973 - the specific hardware-based mechanisms used (for example secure hardware modules, memory-
 - 3974 protection features, platform integrity services);
 - 3975 - which security-critical runtime components are protected or monitored using these mechanisms;
 - 3976 - how the Hypervisor integrates these mechanisms into its runtime integrity validation process.
- 3977 • Evidence that the test platform provides the claimed hardware capabilities and that they are enabled.
- 3978 • A Hypervisor deployment configured to use the hardware-based mechanisms for runtime integrity validation,
- 3979 consistent with product guidance.

3980 **4. Assessment Activities**

3981 The assessment shall at least:

- 3982 1) Review documentation and platform information to confirm:
 - 3983 - the availability and configuration of the relevant hardware-based mechanisms;

- 3984 - the mapping between these mechanisms and the security-critical runtime components identified for
3985 integrity protection;
- 3986 - how these mechanisms feed into the runtime integrity checks assessed under REQ-H-RP-INT-003.
- 3987 2) With hardware-based mechanisms enabled, operate the Hypervisor in a normal configuration and verify,
3988 through logs, status outputs, or diagnostics, that:
- 3989 - runtime integrity validation for the stated components explicitly uses the hardware-based mechanisms
3990 (for example hardware-backed measurements, protected memory regions, or equivalent);
- 3991 - no integrity violations are reported when the system is intact.
- 3992 3) Where feasible and consistent with vendor guidance, perform one of the following:
- 3993 - compare runtime integrity behaviour with and without the hardware-based mechanisms enabled; or
- 3994 - induce a controlled deviation targeting a component protected by the hardware-based mechanisms.
- 3995 In either case, verify that the use of hardware-based mechanisms contributes to detection or prevention of
3996 Unauthorized modification (for example by blocking the modification or by providing a hardware-backed
3997 indication of tampering that leads to detection and protective actions), and that any resulting alerts are
3998 generated as described.

3999 5. Assessment Verdict

- 4000 • Pass: when support is claimed, the Hypervisor integrates hardware-based platform mechanisms into runtime
4001 integrity validation for security-critical components; these mechanisms provide additional protection or
4002 detection capability beyond software-only mechanisms; and attempts to modify protected components are
4003 detected or prevented in line with the documented design.
- 4004 • Fail: the claimed hardware-based mechanisms are not effectively integrated with runtime integrity validation;
4005 provide no observable contribution to protection or detection; or protected components can be modified
4006 without detection or prevention despite the claimed support.

4007 6. Assessment Evidence

- 4008 • [DESIGN] Documentation of the integration between Hypervisor runtime integrity validation and hardware-
4009 based platform mechanisms, including the set of protected components and expected assurances.
- 4010 • [PLT-CAP] Evidence of platform capabilities (for example secure hardware modules, memory-protection
4011 features, integrity services) and their configuration.
- 4012 • [CONFIG] Hypervisor configuration showing that hardware-assisted runtime integrity options are enabled and
4013 associated with the relevant components.
- 4014 • [LOG] / [OBSERVATION] Logs, diagnostic outputs, or test results demonstrating use of hardware-based
4015 mechanisms during runtime integrity validation and, where tested, detection or prevention of Unauthorized
4016 modifications to protected components.

4017 6.3.1.7 Assessment for Remote Attestation

4018 6.3.1.7.1 Assessment Case AC-H-RA-INT-003

4019 1. Assessment Reference

4020 Requirement: REQ-H-RA-INT-003 (Advanced)

4021 2. Assessment Objective

4022 Verify that:

- 4023 1) The Hypervisor supports measured boot and generates attestation evidence describing its bootloader, kernel or
4024 microkernel, security-critical configuration, and essential runtime state.
- 4025 2) The attestation evidence is cryptographically verifiable by an authorized verifier and is bound to the specific
4026 Hypervisor instance for which the measurements were produced.

- 4027 3) Attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay.
- 4028 4) The attestation evidence is limited to the minimum information necessary to verify the integrity and
- 4029 configuration state of the Hypervisor.

4030 3. Assessment Preparation

4031 The assessment shall have access to:

- 4032 • Documentation describing:
 - 4033 - the measured-boot and attestation design, including which components and configuration items are
 - 4034 measured (bootloader, kernel or microkernel, security-critical configuration, essential runtime state);
 - 4035 - how measurements are collected, stored, and associated with a Hypervisor instance;
 - 4036 - the format and export mechanism of the attestation evidence (for example data fields, encoding, transport
 - 4037 or API);
 - 4038 - the cryptographic mechanisms used to protect the attestation evidence and bind it to a Hypervisor
 - 4039 instance (for example signature algorithms, keys, certificate chains);
 - 4040 - the protection model for attestation keys and measurement data, including which entities are permitted to
 - 4041 read, manage, or update them.
- 4042 • A test configuration with:
 - 4043 - a Hypervisor instance with measured boot and attestation enabled according to product documentation;
 - 4044 - an interface to request attestation evidence (for example CLI, API, or management interface);
 - 4045 - a verification tool or reference verifier configured with the relevant trust anchors to validate the
 - 4046 attestation evidence;
 - 4047 - at least one guest or non-privileged context that can be used to attempt Unauthorized access to attestation
 - 4048 keys or raw measurement data.

4049 4. Assessment Activities

4050 The assessment shall at least:

- 4051 1) Review documentation and Hypervisor settings to identify:
 - 4052 - the set of elements covered by measurements (bootloader, kernel or microkernel, security-critical
 - 4053 configuration, essential runtime state);
 - 4054 - how these measurements are represented in the attestation evidence;
 - 4055 - which cryptographic algorithms and keys are used to protect and sign the evidence;
 - 4056 - how the attestation evidence is bound to a specific Hypervisor instance (for example instance identifiers,
 - 4057 platform identity, or key binding);
 - 4058 - how attestation keys and measurement data are stored and access-controlled.
- 4059 2) After a normal boot of the Hypervisor:
 - 4060 - request attestation evidence using the documented interface;
 - 4061 - use the verification tool to confirm that the evidence verifies successfully under the configured trust
 - 4062 anchors and corresponds to the expected Hypervisor instance;
 - 4063 - compare fields in the evidence with the deployed software and configuration (for example bootloader
 - 4064 and kernel versions, key security-critical settings) to confirm consistency;
 - 4065 - review the content of the evidence to confirm that it is limited to the information required to verify
 - 4066 integrity and configuration state and does not unnecessarily expose unrelated sensitive operational data.

- 3) From a guest workload, non-privileged account, or other context that should not have direct access:
- attempt to read or export attestation private keys or raw measurement data using available interfaces;
 - verify that such attempts are rejected or blocked and that keys and measurements cannot be obtained through these paths.
- 4) Where anti-replay behaviour is documented:
- capture a valid attestation response;
 - attempt to reuse or replay this response in a context that expects fresh attestation;
 - verify, using the verifier or tool, that replayed evidence is detected or rejected in accordance with the documented behaviour.

5. Assessment Verdict

- Pass: the Hypervisor generates attestation evidence covering the bootloader, kernel or microkernel, security-critical configuration, and essential runtime state; the evidence is cryptographically verifiable and bound to the specific Hypervisor instance; attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay; and the evidence content is limited to the minimum necessary to verify the integrity and configuration state of the Hypervisor.
- Fail: attestation evidence is not generated as specified; cannot be cryptographically verified or is not clearly bound to a specific instance; Unauthorized entities can access or export attestation keys or measurement data or successfully replay stale evidence where freshness is required; or the evidence content includes unnecessary sensitive information beyond what is required to verify integrity and configuration state.

6. Assessment Evidence

- [DESIGN] Documentation of the measured-boot and attestation design, including measured components, evidence format, cryptographic protection, instance binding, and the protection model for attestation keys and measurements.
- [CONFIG] Hypervisor and, where applicable, platform configuration enabling measured boot and attestation and defining trust anchors used by verifiers.
- [LOG] Records of attestation requests and responses and any logged events related to attestation key access failures or replay handling.
- [OBSERVATION] Captured attestation evidence samples and verifier outputs showing successful validation for normal runs and rejection or detection of replayed or tampered evidence.
- [PLT-CAP] Evidence of platform capabilities used to support measurement and attestation, where the implementation relies on platform services.

6.3.1.7.2 Assessment Case AC-H-RA-INT-004

1. Assessment Reference

Requirement: REQ-H-RA-INT-004 (Advanced)

2. Assessment Objective

Verify that:

- 1) The Hypervisor supports configuration of privacy-preserving attestation options for remote attestation use cases that reduce linkability between attestation events.
- 2) These privacy-preserving options preserve the ability of authorized verifiers to assess the integrity and configuration state of the Hypervisor.

4107 3. Assessment Preparation

4108 The assessment shall have access to:

- 4109 • Documentation describing:
 - 4110 - the Hypervisor remote attestation model, including how Hypervisor identity and instance information are
 - 4111 represented in attestation evidence;
 - 4112 - the privacy-preserving options available (for example use of pseudonymous identifiers, separation of
 - 4113 identity and integrity information, control of attestation frequency or granularity);
 - 4114 - how these options are configured;
 - 4115 - the expected impact of each option on attestation evidence fields and verifier behaviour.
- 4116 • A test configuration with:
 - 4117 - a Hypervisor deployment where remote attestation to one or more authorized verifiers is enabled;
 - 4118 - the ability to configure and modify the documented privacy-preserving options;
 - 4119 - tools or procedures to trigger multiple attestation events and capture the resulting attestation evidence
 - 4120 under different privacy configurations;
 - 4121 - access to an authorized verifier or verification tool capable of validating integrity and configuration state
 - 4122 from the attestation evidence in both baseline and privacy-enhanced modes.

4123 4. Assessment Activities

4124 The assessment shall at least:

- 4125 1) Review documentation and Hypervisor settings to identify:
 - 4126 - the default behaviour of remote attestation when privacy-preserving options are not enabled;
 - 4127 - the specific privacy options available and their documented effects on identifiers, metadata, and
 - 4128 attestation frequency or granularity;
 - 4129 - any prerequisites or constraints for enabling these options.
- 4130 2) With privacy-preserving options disabled or in their default state:
 - 4131 - trigger multiple attestation events over time;
 - 4132 - collect the corresponding attestation evidence;
 - 4133 - Analyze the evidence to determine the stability and linkability of identifiers and other potentially
 - 4134 identifying fields across events;
 - 4135 - use the authorized verifier or tool to confirm that the integrity and configuration state of the Hypervisor
 - 4136 can be correctly assessed from this baseline evidence set.
- 4137 3) With one or more privacy-preserving options enabled as documented:
 - 4138 - trigger multiple attestation events under the privacy-enhanced configuration;
 - 4139 - collect the resulting attestation evidence;
 - 4140 - compare this evidence set with the baseline to verify that linkability between attestation events is reduced
 - 4141 in the documented manner (for example through pseudonymous or rotated identifiers, separation of
 - 4142 identity and integrity data, or reduced authorized identifying metadata);
 - 4143 - verify, using the authorized verifier or tool, that the integrity and configuration state of the Hypervisor
 - 4144 can still be reliably assessed using the privacy-enhanced evidence and any associated metadata or
 - 4145 policies.

4146 5. Assessment Verdict

- 4147 • Pass: when remote attestation is used, the Hypervisor provides configurable privacy-preserving attestation
4148 options that, when enabled, reduce linkability between attestation events as documented and still allow
4149 authorized verifiers to correctly assess the Hypervisor's integrity and configuration state.
- 4150 • Fail: privacy-preserving options do not meaningfully reduce linkability between attestation events; or enabling
4151 these options prevents authorized verifiers from correctly assessing the Hypervisor's integrity and
4152 configuration state.

4153 6. Assessment Evidence

- 4154 • [DESIGN] Documentation of the remote attestation model, identity representation, and privacy-preserving
4155 options, including their expected effects on attestation evidence and verifier behaviour.
- 4156 • [CONFIG] Hypervisor and verifier configuration showing remote attestation settings and selected privacy-
4157 preserving options.
- 4158 • [LOG] Records of attestation events and related configuration changes, including timestamps and any
4159 metadata relevant to privacy options.
- 4160 • [OBSERVATION] Sets of attestation evidence collected before and after enabling privacy-preserving options,
4161 with verifier outputs showing successful integrity and configuration assessment in both modes and reduced
4162 linkability when privacy options are enabled.

4163 6.3.1.8 Assessment for Administrative Authentication

4164 6.3.1.8.1 Assessment Case AC-H-ADMIN-AUTH-001

4165 1. Assessment Reference

4166 Requirement: REQ-H-ADMIN-AUTH-001 (Basic)

4167 2. Assessment Objective

4168 Verify that:

- 4169 1) All interfaces that provide administrative functionality require successful authentication before any
4170 administrative operation is performed.
- 4171 2) Default administrative passwords, if present, cannot be used as persistent credentials without enforced change
4172 or are disabled.
- 4173 3) The Hypervisor supports configurable password complexity requirements for administrative accounts and
4174 enforces the configured policy.

4175 3. Assessment Preparation

4176 The assessment shall have access to:

- 4177 • The declared product composition and list of components exposing administrative interfaces.
- 4178 • A list of all administrative interfaces (e.g. local CLI, remote management, APIs, VM management consoles).
- 4179 • Documentation on administrative accounts (including any factory default administrative credentials) and
4180 password policy options.
- 4181 • At least one initialized configuration with one or more administrative accounts and access to the identified
4182 interfaces.

4183 4. Assessment Activities

4184 The assessment shall at least:

- 4185 1) Confirm from documentation and configuration which interfaces provide administrative functionality and
4186 verify that, in the assessed configuration, each such interface requires authentication before administrative
4187 operations.

- 4188 2) Review installation and hardening documentation for default administrative credentials and verify that, in the
 4189 assessed configuration, any such credentials are either disabled or subject to enforced change before use for
 4190 administrative operations.
- 4191 3) Confirm that password complexity requirements for administrative accounts can be configured and verify that
 4192 non-compliant passwords are not accepted for administrative accounts in the assessed configuration.

4193 5. Assessment Verdict

- 4194 • Pass: all administrative interfaces require authentication; default administrative credentials, if present, cannot
 4195 be used as persistent credentials without enforced change or are disabled; and password complexity for
 4196 administrative accounts is configurable and enforced.
- 4197 • Fail: at least one administrative interface allows unauthenticated administrative operations; or default
 4198 administrative credentials can be used for ongoing administrative access; or password complexity for
 4199 administrative accounts cannot be configured or is not enforced.

4200 6. Assessment Evidence

- 4201 • [SCOPE] Declared product composition and list of administrative interfaces considered.
- 4202 • [CONFIG] Administrative authentication and password policy configuration.
- 4203 • [DESIGN] Description of administrative access model, default accounts and password policy mechanism.
- 4204 • [LOG] / [OBSERVATION] Logs, screenshots or console outputs showing authentication enforcement and
 4205 rejection of non-compliant passwords.

4206 6.3.1.8.2 Assessment Case AC-H-ADMIN-AUTH-002

4207 1. Assessment Reference

4208 Requirement: REQ-H-ADMIN-AUTH-002 (Elevated)

4209 2. Assessment Objective

4210 Verify that:

- 4211 1) Administrative interfaces require authenticated access and support cryptographic key-based authentication for
 4212 administrative accounts.
- 4213 2) Password-based login for administrative interfaces can be disabled.
- 4214 3) Algorithms and key lengths for key-based administrative authentication conform to the cryptographic profile
 4215 declared by the manufacturer, aligned with [2].
- 4216 4) Private keys used for administrative authentication are protected from access or reuse by guest workloads and
 4217 other unprivileged components.

4218 3. Assessment Preparation

4219 The assessment shall have access to:

- 4220 • Documentation on key-based administrative authentication for each administrative interface.
- 4221 • Documentation of applicable cryptographic requirements or profiles for administrative authentication.
- 4222 • Documentation of key storage locations and access-control mechanisms for administrative private keys.
- 4223 • At least one configuration where administrative accounts use key-based authentication on relevant interfaces
 4224 and password-based login can be disabled.

4225 4. Assessment Activities

4226 The assessment shall at least:

- 4227 1) Confirm from documentation and configuration that administrative accounts can use cryptographic key-based
 4228 authentication on relevant interfaces and verify that an administrative account can obtain administrative access
 4229 using the key-based mechanism.

4230 2) Confirm that the product provides configuration to disable password-based login for administrative interfaces
 4231 and verify that, where disabled, password-only administrative logins are rejected while key-based
 4232 authentication remains available.

4233 3) Review configuration or observable protocol characteristics to confirm that algorithms, key lengths and
 4234 protocol versions used for key-based administrative authentication comply with the declared cryptographic
 4235 profile and that weaker parameters identified as insufficient are not used.

4236 4) Review the design and configuration of key storage for administrative private keys and confirm that guest
 4237 workloads and other unprivileged components cannot read, export or otherwise misuse these keys.

4238 5. Assessment Verdict

4239 • Pass: administrative interfaces require authenticated access and support key-based administrative
 4240 authentication; password-based login for administrative interfaces can be disabled and is effectively blocked
 4241 where configured; cryptographic parameters comply with the declared profile; and administrative private keys
 4242 are protected from guest and unprivileged access.

4243 • Fail: key-based administrative authentication is not available or not functional; or password-based login cannot
 4244 be disabled or remains effective when disabled; or cryptographic parameters are weaker than the declared
 4245 profile; or administrative private keys are accessible to guest workloads or other unprivileged components.

4246 6. Assessment Evidence

4247 • [CONFIG] Configuration for key-based administrative authentication, password-login disablement and
 4248 cryptographic settings.

4249 • [DESIGN] Documentation of key management and key storage architecture and cryptographic profile for
 4250 administrative authentication.

4251 • [LOG] / [OBSERVATION] Evidence of successful key-based administrative authentication and rejection of
 4252 password-only attempts when disabled.

4253 • [PLT-CAP] Where applicable, evidence of platform capabilities used for key protection.

4254 6.3.1.8.3 Assessment Case AC-H-ADMIN-AUTH-003

4255 1. Assessment Reference

4256 Requirement: REQ-H-ADMIN-AUTH-003 (Advanced)

4257 2. Assessment Objective

4258 Verify that:

4259 1) Certificate-based strong authentication is supported and enforceable for administrative accounts on
 4260 administrative interfaces declared in scope.

4261 2) Certificates used for administrative authentication are validated against a trusted PKI, including chain of trust,
 4262 validity period and, where supported, revocation status.

4263 3) Administrative accounts configured for certificate-based strong authentication cannot use weaker
 4264 authentication methods, except for explicitly documented emergency mechanisms.

4265 3. Assessment Preparation

4266 The assessment shall have access to:

4267 • Documentation describing certificate-based administrative authentication for each relevant interface (e.g.
 4268 mutual TLS, client certificates).

4269 • Documentation of PKI integration: trusted CAs, account/certificate mapping, revocation mechanisms (CRL,
 4270 OCSP or equivalent).

4271 • At least one configuration with administrative accounts using certificates issued by a trusted CA, and
 4272 untrusted, expired and, where supported, revoked certificates (or equivalent) for negative checks.

4. Assessment Activities

The assessment shall at least:

- 1) Confirm from documentation and configuration that administrative accounts can be configured for certificate-based authentication on administrative interfaces declared in scope and verify that administrative access is granted only when a valid certificate is presented in line with the configured policy.
- 2) Confirm that the product validates administrative certificates against the configured trusted CAs and checks validity period, and verify that untrusted, expired or not-yet-valid certificates are rejected; where revocation is supported, verify that revoked certificates are rejected when revocation information is available.
- 3) Review configuration options for enforcing certificate-based strong authentication and verify that administrative accounts configured for such authentication cannot obtain administrative access using weaker methods alone (e.g. password-only), except where an explicitly documented emergency mechanism applies and its conditions are met.

5. Assessment Verdict

- Pass: certificate-based authentication is available and enforceable for administrative accounts on administrative interfaces declared in scope; PKI validation (trusted CAs, validity, and, where supported, revocation) behaves as documented; downgrade to weaker authentication for accounts configured for certificate-based strong authentication is not permitted except under documented emergency mechanisms.
- Fail: certificate-based administrative authentication cannot be configured or is not functioning; or PKI validation is incomplete or inconsistent with documentation; or downgrade to weaker methods is possible for accounts intended to use certificate-based strong authentication.

6. Assessment Evidence

- [CONFIG] Configuration for certificate-based administrative authentication, trusted CA configuration, revocation settings and account-to-certificate mappings.
- [DESIGN] Documentation of the architecture for certificate-based administrative authentication.
- [LOG] / [OBSERVATION] Evidence of successful authentication with valid administrative certificates and rejection of untrusted, expired or revoked certificates, and of rejected downgrade attempts.
- [PLT-CAP] Where relevant, evidence of platform capabilities used to support trust anchors or protect key and trust stores.

6.3.1.9 Assessment for Service Authentication

6.3.1.9.1 Assessment Case AC-H-SERV-AUTH-001

1. Assessment Reference

Requirement: REQ-H-SERV-AUTH-001 (Basic)

2. Assessment Objective

Verify that:

- 1) External services accessing Hypervisor management functions or data are authenticated.
- 2) Each service account uses distinct, non-default credentials and no shared or hard-coded service credentials are relied upon.
- 3) A configurable lockout mechanism exists and is enforced for service accounts after consecutive failed authentication attempts.

3. Assessment Preparation

The assessment shall have access to:

- Declared product composition and list of external services/interfaces accessing management functions or data (e.g. orchestration, monitoring, backup).

- 4316 • Documentation on service accounts, default/built-in service credentials, credential policy and lockout options.
- 4317 • At least one configuration with representative external services integrated via service accounts.

4318 **4. Assessment Activities**

4319 The assessment shall at least:

- 4320 1) Identify all external services accessing management functions or data and verify that each is required to
4321 authenticate before access.
- 4322 2) Review documentation/configuration to confirm distinct credentials per service account and absence of shared
4323 or hard-coded service credentials in normal operation.
- 4324 3) Confirm that lockout thresholds for service accounts can be configured and verify that repeated failed
4325 authentication attempts trigger lockout according to the configured parameters.

4326 **5. Assessment Verdict**

- 4327 • Pass: external services accessing management functions or data require authentication; each service account
4328 uses distinct, non-default credentials without reliance on shared/hard-coded credentials; and a configurable
4329 lockout mechanism for service accounts exists and is enforced.
- 4330 • Fail: unauthenticated service access is possible; or shared/default/hard-coded service credentials are required
4331 for normal use; or a lockout mechanism for service accounts is missing, not configurable, or not enforced.

4332 **6. Assessment Evidence**

- 4333 • [SCOPE] List of external services and interfaces accessing management functions or data.
- 4334 • [CONFIG] Service authentication, service account and lockout configuration.
- 4335 • [DESIGN] Description of service account model, handling of default/built-in service credentials and lockout
4336 mechanism.
- 4337 • [LOG] / [OBSERVATION] Evidence of enforced service authentication and lockout after failed attempts.

4338 **6.3.1.9.2 Assessment Case AC-H-SERV-AUTH-002**

4339 **1. Assessment Reference**

4340 Requirement: REQ-H-SERV-AUTH-002 (Elevated)

4341 **2. Assessment Objective**

4342 Verify that:

- 4343 1) External services accessing management functions or data require authenticated access and can use certificate-
4344 based mutual authentication.
- 4345 2) Algorithms and key lengths for service mutual authentication conform to the declared cryptographic profile,
4346 aligned with [2].
- 4347 3) Private keys used for service authentication are protected from extraction or reuse by guest workloads and
4348 other unprivileged components.

4349 **3. Assessment Preparation**

4350 The assessment shall have access to:

- 4351 • Documentation on certificate-based mutual authentication for external services.
- 4352 • Documentation on cryptographic profiles for service authentication (algorithms, key sizes, disallowed
4353 options).
- 4354 • Documentation on key management and storage for service authentication keys, including separation from
4355 guest workloads/unprivileged components.

- 4356 • At least one configuration where representative services access management functions or data using
4357 certificate-based mutual authentication.

4358 **4. Assessment Activities**

4359 The assessment shall at least:

- 4360 1) Confirm from documentation/configuration that services accessing management functions or data are
4361 authenticated and that certificate-based mutual authentication is supported; verify that a service configured for
4362 mutual authentication can establish an authenticated session and access management functions or data.
- 4363 2) Review configuration/protocol characteristics to confirm that algorithms, key lengths and protocol versions
4364 used for service mutual authentication comply with the declared cryptographic profile and that weaker/non-
4365 compliant options are not used.
- 4366 3) Review key storage design and configuration to confirm that private keys used for service authentication
4367 reside in protected key stores and cannot be read, exported or misused by guest workloads or other
4368 unprivileged components.

4369 **5. Assessment Verdict**

- 4370 • Pass: authenticated access and certificate-based mutual authentication are available for services accessing
4371 management functions or data; cryptographic parameters conform to the declared profile; and service
4372 authentication keys are protected from guest and unprivileged access.
- 4373 • Fail: service access does not consistently require authentication; or mutual authentication is not
4374 available/functional where claimed; or cryptographic parameters are weaker than the declared profile; or
4375 service authentication keys are accessible to guest workloads or other unprivileged components.

4376 **6. Assessment Evidence**

- 4377 • [CONFIG] Mutual authentication configuration for services, including certificates and crypto settings.
- 4378 • [DESIGN] Service authentication and key management architecture, including storage and protection of
4379 service keys and applied cryptographic profile.
- 4380 • [LOG] / [OBSERVATION] Evidence of successful certificate-based mutual authentication and rejection of
4381 non-compliant or unauthenticated connections.
- 4382 • [PLT-CAP] Where relevant, evidence of platform capabilities used to protect service authentication keys.

4383 **6.3.1.9.3 Assessment Case AC-H-SERV-AUTH-003**

4384 **1. Assessment Reference**

4385 Requirement: REQ-H-SERV-AUTH-003 (Advanced)

4386 **2. Assessment Objective**

4387 Verify that:

- 4388 1) Certificate-based mutual authentication is enforced for service interactions that access Hypervisor management
4389 functions or data.
- 4390 2) Certificate paths for service authentication are validated against a trusted PKI.
- 4391 3) Certificates used for service authentication are bound to specific service identities.
- 4392 4) Certificates are checked for validity, expiration and revocation status before access is granted.

4393 **3. Assessment Preparation**

4394 The assessment shall have access to:

- 4395 • Documentation describing enforcement of certificate-based mutual authentication for service interactions
4396 accessing management functions or data, including in-scope interfaces.

- 4397 • Documentation of the PKI trust model (trusted CAs, trust stores, path validation rules) used for service authentication.
- 4398
- 4399 • Documentation on how service identities are bound to certificates and how validity and revocation information (CRL, OCSP or equivalent) is obtained and processed.
- 4400
- 4401 • At least one configuration in which all such service interactions use certificate-based mutual authentication,
- 4402 with valid and invalid/expired/revoked service certificates (or equivalent conditions) available for testing.

4403 **4. Assessment Activities**

4404 The assessment shall at least:

- 4405 1) Confirm from documentation/configuration that certificate-based mutual authentication is enforced for service interactions accessing management functions or data and verify that such interactions cannot be established without successful mutual certificate authentication.
- 4406
- 4407
- 4408 2) Review identity-binding rules and verify that certificates used for service authentication are associated with specific service identities as documented.
- 4409
- 4410 3) Confirm that certificate paths are validated against a trusted PKI and verify that certificates from untrusted CAs or with invalid chains are rejected.
- 4411
- 4412 4) Verify that the Hypervisor checks certificate validity period and rejects expired or not-yet-valid certificates, and, where revocation mechanisms are supported, that revoked (or otherwise invalidated) service certificates are rejected before access is granted.
- 4413
- 4414

4415 **5. Assessment Verdict**

- 4416 • Pass: certificate-based mutual authentication is enforced for service interactions accessing management functions or data; certificate paths are validated against a trusted PKI; certificates are bound to specific service identities; and validity, expiration and revocation (or equivalent) are checked and enforced before access is granted.
- 4417
- 4418
- 4419
- 4420 • Fail: service interactions can occur without enforced certificate-based mutual authentication; or certificate paths are not properly validated; or certificates are not effectively bound to specific service identities; or validity/expiration/revocation checks are missing or ineffective.
- 4421
- 4422

4423 **6. Assessment Evidence**

- 4424 • [CONFIG] Configuration enforcing certificate-based mutual authentication for services, including trust stores, CA configuration and identity bindings.
- 4425
- 4426 • [DESIGN] Documentation of the PKI trust model for service authentication, identity binding and revocation handling.
- 4427
- 4428 • [LOG] / [OBSERVATION] Evidence of successful service interactions with valid certificates and rejection of unauthenticated, untrusted, expired or revoked certificates.
- 4429
- 4430 • [PLT-CAP] Where relevant, evidence of platform capabilities used to support certificate validation and PKI trust.
- 4431

4432 **6.3.1.10 Assessment for Administrative Authorization**

4433 **6.3.1.10.1 Assessment Case AC-H-ADMIN-AUTHZ-001**

4434 **1. Assessment Reference**

4435 Requirement: REQ-H-ADMIN-AUTHZ-001 (Basic)

4436 **2. Assessment Objective**

4437 Verify that:

- 4438 1) All administrative actions are restricted to authenticated administrative accounts.
- 4439 2) A default-deny policy is enforced for access to management functions.

4440 **3. Assessment Preparation**

4441 The assessment shall have access to:

- 4442 • Documentation of administrative roles, accounts and management interfaces.
- 4443 • Documentation of the authorization model for administrative actions (including default behaviour).
- 4444 • At least one configuration with multiple administrative accounts and access to management functions.

4445 **4. Assessment Activities**

4446 The assessment shall at least:

- 4447 1) Verify that unauthenticated users and non-administrative accounts cannot perform administrative actions on
4448 any management interface.
- 4449 2) Confirm from configuration and tests that unspecified administrative actions are denied by default, and that
4450 only explicitly authorized administrative actions are permitted for authenticated accounts.

4451 **5. Assessment Verdict**

- 4452 • Pass: administrative actions can only be performed by authenticated administrative accounts and any non-
4453 authorized administrative action is denied by default.
- 4454 • Fail: unauthenticated or non-administrative users can perform administrative actions; or unspecified
4455 administrative actions are allowed.

4456 **6. Assessment Evidence**

- 4457 • [CONFIG] Administrative authorization configuration and default policies.
- 4458 • [DESIGN] Description of administrative authorization model and default-deny behaviour.
- 4459 • [LOG] / [OBSERVATION] Evidence of denied attempts by unauthenticated/non-admin users and permitted
4460 actions only for authorized administrative accounts.

4461 **6.3.1.10.2 Assessment Case AC-H-ADMIN-AUTHZ-002**

4462 **1. Assessment Reference**

4463 Requirement: REQ-H-ADMIN-AUTHZ-002 (Elevated)

4464 **2. Assessment Objective**

4465 Verify that:

- 4466 1) RBAC or ABAC is enforced for administrative accounts.
- 4467 2) Administrative accounts are granted only the minimum privileges required (least privilege).

4468 **3. Assessment Preparation**

4469 The assessment shall have access to:

- 4470 • Documentation describing RBAC/ABAC for administrative accounts (roles, attributes, permissions).
- 4471 • At least one configuration with multiple administrative roles/attributes and clearly differentiated privilege
4472 levels.

4473 **4. Assessment Activities**

4474 The assessment shall at least:

- 4475 1) Verify that administrative permissions are defined and enforced via RBAC or ABAC (e.g. role-based or
4476 attribute-based rules) rather than a single "full admin" profile.
- 4477 2) For selected administrative accounts, confirm that they can perform only the actions associated with their
4478 defined role/attributes and that higher-privilege actions are denied, demonstrating least-privilege enforcement.

4479 **5. Assessment Verdict**

- 4480 • Pass: RBAC or ABAC is used to control administrative actions and selected administrative accounts are
4481 restricted to the minimum privileges defined for their role/attributes.
- 4482 • Fail: RBAC/ABAC is not effectively enforced; or administrative accounts have broader privileges than
4483 defined; or least-privilege cannot be demonstrated.

4484 **6. Assessment Evidence**

- 4485 • [CONFIG] RBAC/ABAC configuration for administrative accounts (roles/attributes and permissions).
- 4486 • [DESIGN] Documentation of the administrative authorization model and least-privilege approach.
- 4487 • [LOG] / [OBSERVATION] Evidence of allowed actions within a role's scope and denied actions outside that
4488 scope.

4489 **6.3.1.10.3 Assessment Case AC-H-ADMIN-AUTHZ-003**

4490 **1. Assessment Reference**

4491 Requirement: REQ-H-ADMIN-AUTHZ-003 (Advanced)

4492 **2. Assessment Objective**

4493 Verify that:

- 4494 1) Fine-grained authorization policies are enforced for administrative accounts.
- 4495 2) Separation of Duties (SoD) is implemented for administrative functions where required.
- 4496 3) Time-bound or just-in-time (JIT) privilege elevation is supported and enforced.
- 4497 4) Integration with external identity and/or policy decision systems is supported where available.

4498 **3. Assessment Preparation**

4499 The assessment shall have access to:

- 4500 • Documentation of policy-based administrative authorization (policy language/rules, SoD, JIT).
- 4501 • Documentation describing how external identity or policy decision systems (if supported) are integrated.
- 4502 • At least one configuration with:
- 4503 - separate administrative roles/functions demonstrating SoD; and
- 4504 - an example of time-bound or JIT privilege elevation.

4505 **4. Assessment Activities**

4506 The assessment shall at least:

- 4507 1) Verify that administrative permissions are expressed as fine-grained policies (not only static roles), and that
4508 different administrative functions can be assigned to different accounts to enforce SoD (e.g. configuration vs
4509 approval).
- 4510 2) Verify that JIT or time-bound elevation can be configured and that elevated privileges are granted only for a
4511 limited scope/time and then automatically revoked.
- 4512 3) Where integration with external identity/policy systems is supported, verify that the Hypervisor can use such
4513 systems to make administrative authorization decisions or to retrieve administrative identity/attributes used in
4514 policies.

4515 **5. Assessment Verdict**

- 4516 • Pass: administrative authorization is policy-driven and fine-grained; SoD is enforceable and demonstrable;
4517 JIT/time-bound elevation is supported and correctly revoked; and, where applicable, integration with external
4518 identity/policy systems works as documented.

- 4519 • Fail: authorization cannot be expressed as fine-grained policies; SoD cannot be enforced; JIT/time-bound
- 4520 elevation is not available or not effective; or supported external integration cannot be used for administrative
- 4521 authorization.

4522 6. Assessment Evidence

- 4523 • [CONFIG] Administrative policy configuration showing fine-grained rules, SoD and JIT/time-bound
- 4524 elevation.
- 4525 • [DESIGN] Documentation of policy-driven authorization, SoD model and external identity/policy integration.
- 4526 • [LOG] / [OBSERVATION] Evidence of policy decisions enforcing SoD, temporary elevation and subsequent
- 4527 revocation; and, where applicable, decisions influenced by external identity/policy systems.

4528 6.3.1.11 Assessment for Service Authorization

4529 6.3.1.11.1 Assessment Case AC-H-SERV-AUTHZ-001

4530 1. Assessment Reference

4531 Requirement: REQ-H-SERV-AUTHZ-001 (Basic)

4532 2. Assessment Objective

4533 Verify that:

- 4534 1) A default-deny policy is enforced for service accounts.
- 4535 2) Service accounts can only perform actions explicitly authorized.

4536 3. Assessment Preparation

4537 The assessment shall have access to:

- 4538 • List of Hypervisor service accounts and service-facing interfaces.
- 4539 • Documentation of the authorization model for service accounts (policies, defaults).
- 4540 • At least one configuration with several service accounts, including one with no explicit permissions.

4541 4. Assessment Activities

4542 The assessment shall at least:

- 4543 1) Confirm from documentation/configuration that the default behaviour for service accounts is to deny actions
- 4544 not explicitly authorized.
- 4545 2) Verify that a service account with no explicit permissions cannot perform management or data actions.
- 4546 3) For a service account with a defined set of permissions, verify that allowed actions succeed while non-
- 4547 authorized actions are denied.

4548 5. Assessment Verdict

- 4549 • Pass: default behaviour for service accounts is deny; an unprivileged service account cannot perform actions;
- 4550 and only explicitly authorized actions are permitted.
- 4551 • Fail: unspecified actions for service accounts are allowed; or an unprivileged service account can perform
- 4552 operations.

4553 6. Assessment Evidence

- 4554 • [SCOPE] List of service accounts and relevant interfaces.
- 4555 • [CONFIG] Service-account authorization and default policy configuration.
- 4556 • [DESIGN] Description of the service-account authorization model and default-deny behaviour.

- 4557 • [LOG] / [OBSERVATION] Evidence of denied operations for unprivileged service accounts and permitted
4558 operations only when explicitly authorized.

4559 6.3.1.11.2 Assessment Case AC-H-SERV-AUTHZ-002

4560 1. Assessment Reference

4561 Requirement: REQ-H-SERV-AUTHZ-002 (Elevated)

4562 2. Assessment Objective

4563 Verify that:

- 4564 1) Granular authorization is enforced for service accounts, binding allowed actions to their authenticated identity.
- 4565 2) Each service account is configured with least-privilege permissions.

4566 3. Assessment Preparation

4567 The assessment shall have access to:

- 4568 • Documentation on how service identities are represented and bound to permissions.
- 4569 • Documentation on the permission/role model for service accounts.
- 4570 • At least one configuration with multiple service accounts having different, scoped permission sets.

4571 4. Assessment Activities

4572 The assessment shall at least:

- 4573 1) Confirm that authorization decisions for service accounts are based on the authenticated service identity and
4574 not only on generic properties (e.g. IP or network zone).
- 4575 2) For at least two service accounts with different permission sets, verify that each can only perform the actions
4576 explicitly bound to its identity and that actions assigned to another service identity are denied.
- 4577 3) Verify that a service account configured with minimal permissions cannot perform operations outside that
4578 minimal set (least privilege).

4579 5. Assessment Verdict

- 4580 • Pass: authorization for service accounts is identity-bound and granular; and least-privilege permission sets can
4581 be configured and are enforced.
- 4582 • Fail: actions are not clearly bound to service identity; or service accounts can perform operations outside their
4583 defined permissions; or least-privilege cannot be demonstrated.

4584 6. Assessment Evidence

- 4585 • [CONFIG] Service-account authorization configuration showing bindings between specific identities and
4586 allowed actions, including minimal-permission accounts.
- 4587 • [DESIGN] Documentation of the service-account authorization model, identity binding and least-privilege
4588 approach.
- 4589 • [LOG] / [OBSERVATION] Evidence of allowed operations within each service account's permissions and
4590 denied operations outside that scope.

4591 6.3.1.11.3 Assessment Case AC-H-SERV-AUTHZ-003

4592 1. Assessment Reference

4593 Requirement: REQ-H-SERV-AUTHZ-003 (Advanced)

4594 2. Assessment Objective

4595 Verify that:

- 4596 1) Fine-grained, policy-driven authorization is enforced for service accounts.

- 4597 2) Permissions are bound to cryptographic service identities.
- 4598 3) Context-aware enforcement (e.g. namespace, cluster, labels or similar attributes) is supported for service-
4599 account authorization decisions.
- 4600 4) Integration with external authorization systems is supported where available.

4601 3. Assessment Preparation

4602 The assessment shall have access to:

- 4603 • Documentation of policy-driven authorization for service accounts (policy language or rules, conditions,
4604 effects).
- 4605 • Documentation on how cryptographic service identities (e.g. certificates, tokens) are bound to permissions.
- 4606 • Documentation on context attributes available for policy decisions.
- 4607 • Documentation on integration with external authorization systems (e.g. external Policy Decision Point (PDP)
4608 or policy engine), if supported.
- 4609 • At least one configuration where:
 - 4610 - service-account policies reference cryptographic identities and at least one context attribute; and
 - 4611 - where supported, an external authorization system is integrated.

4612 4. Assessment Activities

4613 The assessment shall at least:

- 4614 1) Verify that service-account authorization is expressed as policies (not just fixed roles) and that permissions for
4615 at least one service account are defined via such policies.
- 4616 2) Verify that policies can reference cryptographic identifiers of service accounts and that, in the assessed
4617 configuration, authorization decisions for at least one service depend on its cryptographic identity.
- 4618 3) Verify that at least one context attribute (e.g. namespace, cluster, workload label) can influence authorization
4619 decisions for service accounts and that access is allowed or denied based on that context.
- 4620 4) Where integration with external authorization is supported, verify that the Hypervisor can consult or delegate
4621 service-account authorization decisions to an external system and that policy changes in that system are
4622 reflected in authorization outcomes.

4623 5. Assessment Verdict

- 4624 • Pass: service-account authorization is fine-grained and policy-driven; permissions are bound to cryptographic
4625 service identities; context-aware enforcement is supported; and, where applicable, external authorization
4626 integration is effective.
- 4627 • Fail: policies cannot be used to express fine-grained authorization; or permissions are not bound to
4628 cryptographic identities; or context cannot meaningfully influence authorization decisions; or supported
4629 external authorization integration does not work as intended.

4630 6. Assessment Evidence

- 4631 • [CONFIG] Policy definitions and configuration for service-account authorization referencing cryptographic
4632 identities and context attributes, and, where applicable, external authorization integration settings.
- 4633 • [DESIGN] Documentation of the policy-driven service authorization architecture, including cryptographic
4634 identity binding, context handling and external authorization interfaces.
- 4635 • [LOG] / [OBSERVATION] Evidence of authorization decisions depending on cryptographic identity and
4636 context, and of changes resulting from external authorization policy updates where supported.

4637 6.3.1.12 Assessment for Confidentiality Protection

4638 6.3.1.12.1 Assessment Case AC-H-CONF-001

4639 1. Assessment Reference authorization 4640 Requirement: REQ-H-CONF-001 (Basic)

4641 2. Assessment Objective 4642 Verify that:

- 4643 1) The Hypervisor provides access-control mechanisms to restrict access to sensitive data under its control,
4644 including at least guest VM images, snapshot data, configuration files, credentials, and security-relevant logs.
- 4645 2) Access to such sensitive data is permitted only to authenticated and authorized entities through controlled
4646 interfaces, and cannot be obtained directly by unauthenticated or Unauthorized contexts.

4647 3. Assessment Preparation

4648 The assessment shall have access to:

- 4649 • Documentation describing:
 - 4650 - the categories of sensitive data managed by the Hypervisor (VM images, snapshots, configuration files,
4651 credentials, security-relevant logs);
 - 4652 - the storage locations and interfaces through which this data can be accessed or managed (for example
4653 management UI, CLI, APIs, hypervisor file stores);
 - 4654 - the access-control model and roles/identities allowed to read or modify each category of sensitive data.
- 4655 • A test configuration with:
 - 4656 - at least one administrative account or role with permissions to manage sensitive data;
 - 4657 - at least one non-privileged or operational account with limited permissions;
 - 4658 - where relevant, one or more guest VMs whose images, snapshots, and logs can be created and
4659 manipulated;
 - 4660 - access to the Hypervisor management interfaces and, where applicable, to the underlying storage paths
4661 for sensitive data.

4662 4. Assessment Activities

4663 The assessment shall at least:

- 4664 1) Review documentation and Hypervisor configuration to identify:
 - 4665 - which repositories or storage locations hold VM images, snapshots, configuration files, credentials, and
4666 security-relevant logs;
 - 4667 - which interfaces (UI, CLI, API) are intended for accessing or managing each category of sensitive data;
 - 4668 - which roles/accounts are authorized to use these interfaces.
- 4669 2) Using an authenticated administrative account:
 - 4670 - attempt to access and, where permitted, modify or export representative items of each sensitive data
4671 category via the documented controlled interfaces;
 - 4672 - verify that access is successful only through these interfaces and behaves as documented.
- 4673 3) Using a non-privileged account, guest VM context, or unauthenticated connection (for example direct
4674 filesystem or storage access if exposed, or calling management APIs without proper credentials):
 - 4675 - attempt to read or modify VM images, snapshots, configuration files, credentials, and security-relevant
4676 logs;

- 4677 - verify that such attempts are denied by the Hypervisor access-control mechanisms and do not result in
4678 disclosure or modification of sensitive data.

- 4679 4) Where storage paths are visible at the host level, attempt to access sensitive data directly on disk or in
4680 configuration directories from a context that is not granted the corresponding management permissions, and
4681 verify that operating-system or Hypervisor controls prevent Unauthorized access.

4682 5. Assessment Verdict

- 4683 • Pass: access to VM images, snapshots, configuration files, credentials, and security-relevant logs is only
4684 possible through controlled interfaces and only for authenticated and authorized entities; attempts by
4685 unauthenticated or Unauthorized contexts to access or modify such data are prevented.

- 4686 • Fail: sensitive data can be read or modified without using controlled interfaces, or by unauthenticated or
4687 Unauthorized entities; or the documented access-control model is not enforced in practice.

4688 6. Assessment Evidence

- 4689 • [DESIGN] Documentation of sensitive data types, storage locations, and the associated access-control model.
- 4690 • [CONFIG] Hypervisor configuration showing role/permission assignments and any access-control rules for
4691 management interfaces and storage locations.
- 4692 • [LOG] Security or audit logs capturing successful authorized accesses and rejected Unauthorized attempts,
4693 where such logging is implemented.
- 4694 • [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating
4695 authorized access succeeding and Unauthorized access failing.
- 4696 • [SCOPE] Evidence identifying which data stores and components were considered in-scope for sensitive data.

4697 6.3.1.12.2 Assessment Case AC-H-CONF-002

4698 1. Assessment Reference

4699 Requirement: REQ-H-CONF-002 (Elevated)

4700 2. Assessment Objective

4701 Verify that:

- 4702 1) The Hypervisor supports encryption of sensitive data at rest, including VM images, snapshot data,
4703 configuration files, credentials, and security-relevant logs, as specified by the product.
- 4704 2) When data-at-rest protection is enabled, encryption is effectively applied to the configured sensitive data
4705 repositories.
- 4706 3) Encryption keys used for data-at-rest protection are stored and managed in a manner that prevents access by
4707 guest workloads and other unprivileged components.

4708 3. Assessment Preparation

4709 The assessment shall have access to:

- 4710 • Documentation describing:
 - 4711 - the data-at-rest encryption design, including which types of data and storage locations can be encrypted
4712 (VM images, snapshots, configuration files, credentials, security logs);
 - 4713 - supported key-management approaches (for example internal key store, external KMS, hardware-backed
4714 keys);
 - 4715 - how encryption and key management are configured and monitored by the administrator;
 - 4716 - restrictions on access to encryption keys and key-management interfaces.

- 4717 • A test configuration with:
 - 4718 - one or more data stores configured to use the Hypervisor data-at-rest encryption mechanisms for the
 - 4719 relevant categories of sensitive data;
 - 4720 - at least one administrative account authorized to configure and monitor encryption and key management;
 - 4721 - at least one guest workload and one non-privileged host context that can be used to attempt access to
 - 4722 encrypted data and key material;
 - 4723 - where applicable, access to the underlying storage devices or files to inspect whether data is stored
 - 4724 encrypted.

4725 **4. Assessment Activities**

4726 The assessment shall at least:

- 4727 1) Review documentation and configuration to identify:
 - 4728 - which sensitive data categories and storage locations are covered by data-at-rest encryption;
 - 4729 - how encryption is enabled and managed (for example per datastore, per volume, per file);
 - 4730 - where encryption keys are stored and how they are protected from guest and non-privileged access.
- 4731 2) With the Hypervisor configured according to product guidance:
 - 4732 - create or update representative VM images, snapshots, configuration files, credentials, and security logs
 - 4733 under the encrypted configuration;
 - 4734 - inspect the underlying storage (for example by viewing raw blocks, file contents from outside the
 - 4735 Hypervisor context, or using vendor-provided diagnostic tools) to confirm that these items are stored in
 - 4736 an encrypted form and not as plaintext;
 - 4737 - verify that disabling or removing access to encryption keys (for example by unmounting the key store or
 - 4738 revoking KMS access) prevents normal decryption and access to the protected data.
- 4739 3) From guest workloads and other non-privileged contexts:
 - 4740 - attempt to access encryption keys or key stores using available interfaces, filesystems, or APIs;
 - 4741 - verify that such attempts are denied and that key material is not readable or exportable by these contexts.
- 4742 4) Where external or hardware-backed key management is used, verify that:
 - 4743 - keys used for data-at-rest encryption are stored in the external or hardware-backed mechanism as
 - 4744 documented;
 - 4745 - the Hypervisor only receives derived or wrapped keys as documented and does not expose raw key
 - 4746 material to guest or non-privileged components.

4747 **5. Assessment Verdict**

- 4748 • Pass: the Hypervisor supports and correctly applies encryption for the configured categories of sensitive data
- 4749 at rest; encrypted data is not stored in plaintext on underlying storage; and encryption keys are managed and
- 4750 protected so that guest workloads and other unprivileged components cannot access them.
- 4751 • Fail: sensitive data at rest configured to be protected is stored unencrypted or in recoverable plaintext;
- 4752 encryption keys are accessible to guest workloads or other unprivileged contexts; or the implemented
- 4753 behaviour differs from the documented encryption and key-management model.

4754 **6. Assessment Evidence**

- 4755 • [DESIGN] Documentation of the data-at-rest encryption model, including covered data types, key-
- 4756 management design, and protection measures for key material.
- 4757 • [CONFIG] Hypervisor configuration showing enabled encryption for relevant datastores and the configured
- 4758 key-management settings.

- 4759 • [LOG] Events or audit trails related to enabling/disabling encryption, key-management operations, and failed
4760 access attempts to protected data or keys.
- 4761 • [OBSERVATION] Test records and storage inspections (for example hex dumps, tool outputs) demonstrating
4762 that protected data is stored encrypted and that data becomes inaccessible when keys are revoked.
- 4763 • [PLT-CAP] Evidence of any platform or external key-management capabilities used to support data-at-rest
4764 encryption.

4765 6.3.1.12.3 Assessment Case AC-H-CONF-003

4766 1. Assessment Reference

4767 Requirement: REQ-H-CONF-003 (Elevated)

4768 2. Assessment Objective

4769 Verify that:

- 4770 1) The Hypervisor supports the use of encrypted communication for management interfaces, for VM migration
4771 traffic, and for inter-VM traffic where such traffic is mediated by Hypervisor-controlled virtual networking
4772 components.
- 4773 2) When encrypted transport is configured for these communications, traffic is actually protected with encryption
4774 consistent with the configuration.
- 4775 3) The Hypervisor prevents fallback from configured encrypted transport to unencrypted or weakly protected
4776 channels.

4777 3. Assessment Preparation

4778 The assessment shall have access to:

- 4779 • Documentation describing:
 - 4780 - the Hypervisor management interfaces, VM migration protocols, and inter-VM communication paths that
4781 can be protected by encrypted transport;
 - 4782 - supported encryption mechanisms and relevant configuration options (for example TLS profiles, SSH,
4783 IPsec, or other secure tunnelling mechanisms);
 - 4784 - how encryption requirements and anti-fallback behaviour are enforced and monitored.
- 4785 • A test configuration with:
 - 4786 - at least one management interface accessible over a network;
 - 4787 - at least one VM migration path between Hypervisor hosts;
 - 4788 - at least two VMs whose inter-VM traffic can be mediated by the Hypervisor virtual networking layer;
 - 4789 - tools to capture and inspect network traffic (for example packet capture on relevant interfaces).

4790 4. Assessment Activities

4791 The assessment shall at least:

- 4792 1) Review documentation and configuration to identify:
 - 4793 - which management, migration, and inter-VM traffic flows are in scope for encrypted transport;
 - 4794 - the expected encryption protocols, ciphers, and configuration settings;
 - 4795 - any configuration options that control whether unencrypted or weakly protected channels are allowed.
- 4796 2) Configure encrypted transport for:
 - 4797 - a representative management interface;
 - 4798 - a VM migration operation between two Hypervisor hosts;

- 4799 - an inter-VM communication path mediated by the Hypervisor virtual network.
- 4800 For each case, initiate the relevant communication while capturing network traffic on appropriate interfaces,
4801 and verify that:
- 4802 - payloads are encrypted and not readable as clear text;
- 4803 - negotiated protocols and cipher suites match the documented configuration.
- 4804 3) Attempt to downgrade or misconfigure these communications by:
- 4805 - disabling encryption in client tools or attempting to force use of plaintext protocols;
- 4806 - proposing weak or unsupported cipher suites;
- 4807 - modifying configuration to request unencrypted channels where encrypted transport has been mandated.
- 4808 Verify that, when encrypted transport is configured as required, the Hypervisor does not silently fall back to
4809 unencrypted or weakly protected channels, and instead either maintains encrypted communication or fails the
4810 connection as documented.

4811 5. Assessment Verdict

- 4812 • Pass: the Hypervisor supports encrypted transport for management interfaces, VM migration traffic, and
4813 Hypervisor-mediated inter-VM traffic; when configured to use encrypted transport, captured traffic is
4814 effectively encrypted; and attempts to fall back to unencrypted or weakly protected channels are prevented or
4815 cause the connection to fail as documented.
- 4816 • Fail: encrypted transport cannot be enabled as documented; traffic that is configured to be encrypted is
4817 observed in plaintext or with weaker protection than configured; or the Hypervisor silently falls back to
4818 unencrypted or weakly protected channels when downgrade or misconfiguration is attempted.

4819 6. Assessment Evidence

- 4820 • [DESIGN] Documentation of supported secure transport mechanisms for management, migration, and inter-
4821 VM traffic, including anti-fallback behaviour.
- 4822 • [CONFIG] Hypervisor configuration showing enabled encryption settings for tested interfaces and flows.
- 4823 • [LOG] Events or audit records related to secure channel establishment, failures, and attempted downgrades.
- 4824 • [OBSERVATION] Packet captures and analysis outputs demonstrating encrypted traffic when protection is
4825 configured and showing connection failure or refusal when downgrade is attempted.
- 4826 • [PLT-CAP] Evidence of any platform networking or cryptographic capabilities used to support secure
4827 transport.

4828 6.3.1.12.4 Assessment Case AC-H-CONF-004

4829 1. Assessment Reference

4830 Requirement: REQ-H-CONF-004 (Advanced)

4831 2. Assessment Objective

4832 Verify that:

- 4833 1) The Hypervisor supports provisioning and management of execution environments for guest workloads in
4834 which confidential data associated with those workloads is protected, during execution, from observation,
4835 inspection, or access through the Hypervisor, the host operating system, and administrative interfaces.
- 4836 2) On platforms that provide mechanisms for confidential execution, the Hypervisor can use these mechanisms to
4837 enforce confidentiality of guest memory and associated processor state for workloads designated as
4838 confidential.

3. Assessment Preparation

The assessment shall have access to:

- Documentation describing:
 - the confidential-execution model supported by the Hypervisor, including which guest types or configurations can run with confidential execution;
 - the platform mechanisms used (for example memory-encryption features, secure enclaves, or other confidential-computing capabilities);
 - management workflows for provisioning, monitoring, and decommissioning confidential workloads;
 - limitations on Hypervisor and host visibility into confidential guest memory and processor state.
- A test configuration with:
 - a platform that provides the documented confidential-execution features;
 - the Hypervisor configured to support confidential workloads according to product guidance;
 - at least one guest VM that can be instantiated both in normal mode and in confidential-execution mode;
 - tools or interfaces normally usable by Hypervisor administrators or host operators to inspect guest memory or processor state (for example debug APIs, snapshot tools, or memory-dump mechanisms).

4. Assessment Activities

The assessment shall at least:

- 1) Review documentation and configuration to identify:
 - which workloads or guest types can be run in confidential-execution mode;
 - which platform features are relied upon and how they are enabled;
 - which Hypervisor or host-level interfaces are restricted when a guest is running confidentially.
- 2) Instantiate a guest VM in normal (non-confidential) mode and, using available Hypervisor or host tools:
 - confirm that it is possible to inspect or dump guest memory or processor state in the manner described by the product;
 - confirm that this behaviour matches documentation for non-confidential workloads.
- 3) Instantiate an equivalent guest VM in confidential-execution mode and:
 - attempt to use the same inspection, debugging, or memory-dump tools from the Hypervisor and host;
 - verify that direct inspection of confidential guest memory and associated processor state is blocked or that retrieved data is unintelligible ciphertext, in accordance with the confidential-execution model;
 - verify that management interfaces clearly indicate the confidential status of the workload.
- 4) Where platform-level reporting is available (for example platform capabilities or status registers relating to memory encryption or secure execution):
 - confirm that the relevant features are enabled for the confidential workload;
 - verify that disabling or misconfiguring these features (where possible) prevents confidential mode from being established or results in a documented error state.

5. Assessment Verdict

- Pass: the Hypervisor can provision and manage confidential-execution environments for guest workloads; when such environments are used, confidential workload memory and processor state cannot be inspected or accessed via Hypervisor, host, or administrative interfaces beyond what is documented; and the implementation correctly relies on platform confidential-execution mechanisms where available.

- 4879 • Fail: workloads configured as confidential can still have their memory or processor state inspected in clear
4880 form via standard Hypervisor or host tools; platform confidential-execution features are not used or are
4881 misconfigured; or the behaviour diverges from the documented confidential-execution model.

4882 6. Assessment Evidence

- 4883 • [DESIGN] Documentation of the confidential-execution architecture, including reliance on platform features
4884 and restrictions on Hypervisor/host access.
- 4885 • [CONFIG] Hypervisor and platform configuration showing how confidential execution is enabled and which
4886 guests are configured to use it.
- 4887 • [LOG] Events and audit records relating to creation, modification, and failure of confidential workloads or
4888 platform confidential-execution features.
- 4889 • [OBSERVATION] Test records showing successful inspection of non-confidential guests and blocked or
4890 unintelligible access attempts for confidential guests, plus management UI/CLI outputs indicating confidential
4891 status.
- 4892 • [PLT-CAP] Evidence of platform confidential-computing capabilities used by the Hypervisor.

4893 6.3.1.12.5 Assessment Case AC-H-CONF-005

4894 1. Assessment Reference

4895 Requirement: REQ-H-CONF-005 (Advanced, optional)

4896 2. Assessment Objective

4897 Verify that, where claimed:

- 4898 1) The Hypervisor supports the use of hardware-based security mechanisms to strengthen protection of
4899 cryptographic keys and confidential execution state used for data-at-rest, data-in-transit, and data-in-use
4900 protection.
- 4901 2) Such mechanisms, where present (for example dedicated secure hardware modules, memory-encryption
4902 features, or equivalent platform components), are integrated in a way that provides protected storage for keys
4903 and resistance against Unauthorized access to confidential workload data, consistent with the product
4904 documentation.

4905 3. Assessment Preparation

4906 The assessment shall have access to:

- 4907 • Documentation describing:
 - 4908 - which hardware-based mechanisms are supported (for example TPMs, HSMs, secure enclaves, memory-
4909 encryption engines);
 - 4910 - how these mechanisms are used to protect cryptographic keys and confidential execution state for the
4911 Hypervisor and its workloads;
 - 4912 - configuration and operational procedures for enabling and monitoring hardware-backed confidentiality.
- 4913 • A test configuration with:
 - 4914 - a platform providing the claimed hardware-based mechanisms;
 - 4915 - the Hypervisor configured to use these mechanisms for key storage and/or confidential workload
4916 protection according to product guidance;
 - 4917 - tools or interfaces to query the hardware modules or platform features (for example vendor utilities, OS
4918 interfaces, or Hypervisor management commands);
 - 4919 - guest and non-privileged host contexts that can be used to attempt access to keys or confidential state.

4. Assessment Activities

The assessment shall at least:

- 1) Review documentation and configuration to identify:
 - which keys and confidential state are intended to be protected by hardware mechanisms (for example master keys for data-at-rest encryption, keys for secure transport, confidential-execution state);
 - which hardware modules or features are used and how they are bound to the Hypervisor deployment;
 - how fallback behaviour is handled if hardware support is unavailable.
- 2) On a platform with the relevant hardware present and enabled:
 - configure the Hypervisor to use these mechanisms as documented;
 - verify, using vendor or platform tools, that key material or confidential state is stored or managed by the hardware module (for example keys created in a TPM/HSM, memory-encryption status flags);
 - confirm that the Hypervisor does not expose raw key material corresponding to hardware-protected keys to guest or non-privileged contexts.
- 3) From guest workloads and other non-privileged contexts:
 - attempt to access hardware-protected keys or confidential execution state via normal system interfaces;
 - verify that such attempts are rejected or result only in opaque, non-extractable handles in accordance with the hardware security model.
- 4) Where possible, simulate removal or unavailability of the hardware mechanism and verify that the Hypervisor behaviour matches the documented fallback or error handling (for example refusal to start protected workloads, or disabling of specific features).

5. Assessment Verdict

- Pass: the Hypervisor, when hardware-backed confidentiality is claimed, correctly integrates the documented hardware mechanisms to protect cryptographic keys and confidential execution state; keys and confidential state bound to hardware protection are not accessible in raw form to guest or other unprivileged components; and platform and Hypervisor behaviour matches the documented model.
- Fail: hardware mechanisms claimed to be used are not actually invoked; keys or confidential state intended to be hardware-protected are exposed in software or to unprivileged contexts; or fallback and error-handling behaviours deviate from the documentation.

6. Assessment Evidence

- [DESIGN] Documentation of hardware-backed confidentiality mechanisms, including key-protection and confidential-state-protection models.
- [CONFIG] Hypervisor and platform configuration showing use of hardware modules or features for key and state protection.
- [LOG] Events related to initialization, use, and failure of hardware security mechanisms.
- [OBSERVATION] Test records and tool outputs demonstrating that keys and confidential state are managed by hardware mechanisms and not exposed to unprivileged contexts.
- [PLT-CAP] Evidence of hardware capabilities and their security properties as used by the Hypervisor.

6.3.1.13 Assessment for Availability and Resilience

6.3.1.13.1 Assessment Case AC-H-AVAIL-001

1. Assessment Reference

Requirement: REQ-H-AVAIL-001 (Basic)

4961 **2. Assessment Objective**

4962 Verify that:

- 4963 1) The Hypervisor detects failures of internal management and control-plane components.
- 4964 2) The Hypervisor supports recovery of those components without requiring a full host reboot.
- 4965 3) Availability of running guest virtual machines is maintained during such failures and recovery.

4966 **3. Assessment Preparation**

4967 The assessment shall have access to:

- 4968 • Documentation describing Hypervisor management and control-plane components and associated health
- 4969 monitoring or watchdog mechanisms.
- 4970 • Documentation of supported recovery methods for management/control components (e.g. process restart,
- 4971 service failover) and their impact on guest VMs.
- 4972 • At least one configuration with running guest VMs and access to management/control interfaces.

4973 **4. Assessment Activities**

4974 The assessment shall at least:

- 4975 1) Identify key management and control-plane components (e.g. management daemons, schedulers, control
- 4976 services) and confirm that mechanisms exist to monitor their health and detect failures.
- 4977 2) In the assessed configuration, induce or simulate a failure of at least one such component (e.g. stop a
- 4978 management service) and verify that the Hypervisor detects the failure.
- 4979 3) Verify that the affected component(s) can be recovered (e.g. automatically restarted or restored) without
- 4980 rebooting the host, and that running guest VMs continue operating during the failure and recovery.

4981 **5. Assessment Verdict**

- 4982 • Pass: failures of internal management/control-plane components are detected; recovery is possible without a
- 4983 full host reboot; and running guest VMs remain available during failure and recovery.
- 4984 • Fail: failures are not reliably detected; recovery requires a full host reboot; or guest VMs lose availability as a
- 4985 result of management/control-plane component failure and recovery.

4986 **6. Assessment Evidence**

- 4987 • [CONFIG] Configuration of management/control components and health monitoring/restart mechanisms.
- 4988 • [DESIGN] Documentation of management/control-plane architecture and failure/recovery behaviour.
- 4989 • [LOG] / [OBSERVATION] Evidence of failure detection, component recovery without host reboot, and
- 4990 continued availability of guest VMs during the procedure.

4991 **6.3.1.13.2 Assessment Case AC-H-AVAIL-002**

4992 **1. Assessment Reference**

4993 Requirement: REQ-H-AVAIL-002 (Elevated)

4994 **2. Assessment Objective**

4995 Verify that:

- 4996 1) The Hypervisor provides mechanisms to support high availability of guest virtual machines.
- 4997 2) Host or Hypervisor service failure is automatically detected.
- 4998 3) Affected guest VMs are automatically restarted or migrated to another available host using native clustering or
- 4999 integration with external clustering/orchestration systems.

5000 **3. Assessment Preparation**

5001 The assessment shall have access to:

- 5002 • Documentation of Hypervisor high-availability mechanisms for guest VMs, including clustering and
5003 integrations with external orchestrators.
- 5004 • Documentation describing detection of host/Hypervisor service failure and the associated response.
- 5005 • A test environment with at least two Hypervisor hosts participating in a cluster or managed by an orchestrator,
5006 hosting one or more test VMs.

5007 **4. Assessment Activities**

5008 The assessment shall at least:

- 5009 1) Confirm that high-availability or clustering features are enabled/configured for selected guest VMs.
- 5010 2) Induce or simulate failure of a Hypervisor host or critical Hypervisor service (e.g. stopping the Hypervisor
5011 service or isolating the host) and verify that the failure is automatically detected by the
5012 HA/cluster/orchestration mechanism.
- 5013 3) Verify that affected guest VMs are automatically restarted or migrated to another available host, without
5014 requiring manual intervention, and that the VMs become available again after the recovery action.

5015 **5. Assessment Verdict**

- 5016 • Pass: host/Hypervisor service failures are automatically detected; guest VMs configured for HA are
5017 automatically restarted or migrated to another host using native or integrated clustering/orchestration; and
5018 availability of such VMs is restored automatically.
- 5019 • Fail: failures are not automatically detected; or affected guest VMs are not automatically restarted/migrated; or
5020 manual intervention is required to restore availability in contradiction with the documented HA behaviour.

5021 **6. Assessment Evidence**

- 5022 • [CONFIG] High-availability/clustering configuration for guest VMs and host group/cluster settings.
- 5023 • [DESIGN] Documentation of HA/cluster/orchestration architecture and failure-handling behaviour.
- 5024 • [LOG] / [OBSERVATION] Evidence of failure detection and subsequent automatic restart or migration of
5025 guest VMs.

5026 **6.3.1.13.3 Assessment Case AC-H-AVAIL-003**

5027 **1. Assessment Reference**

5028 Requirement: REQ-H-AVAIL-003 (Elevated)

5029 **2. Assessment Objective**

5030 Verify that:

- 5031 1) The Hypervisor implements resource scheduling mechanisms that allow configuration of minimum resource
5032 allocations for security and management functions (CPU, memory and I/O).
- 5033 2) These mechanisms protect the Hypervisor control plane from resource exhaustion caused by guest workloads
5034 or external contention, ensuring that security and management functions remain able to operate under load.

5035 **3. Assessment Preparation**

5036 The assessment shall have access to:

- 5037 • Documentation describing Hypervisor resource scheduling mechanisms (e.g. reservations, priorities, quotas,
5038 cgroup-like mechanisms) and how they apply to security and management functions.
- 5039 • Documentation specifying how minimum CPU, memory and I/O resources can be configured for these
5040 functions.
- 5041 • A test configuration with several guest VMs and the ability to generate high load from guest workloads.

5042 **4. Assessment Activities**

5043 The assessment shall at least:

- 5044 1) Confirm that resource scheduling or reservation mechanisms exist and that minimum CPU, memory and I/O
5045 allocations can be configured for security and management components (e.g. management agents, control-
5046 plane services, logging, security monitoring).
- 5047 2) Configure such minimum allocations in the assessed environment.
- 5048 3) Generate high resource load from guest workloads and/or external contention (e.g. CPU- and I/O-intensive
5049 tasks) and verify that:
- 5050 - management and security interfaces remain responsive; and
- 5051 - critical control-plane functions (e.g. VM lifecycle operations, logging, access control) continue to
5052 operate despite guest load.

5053 **5. Assessment Verdict**

- 5054 • Pass: resource scheduling mechanisms support configuration of minimum CPU, memory and I/O allocations
5055 for security/management functions; and under high guest load or external contention, these functions remain
5056 able to operate as intended.
- 5057 • Fail: minimum allocations for security/management functions cannot be configured; or under load, the control
5058 plane becomes starved of resources, causing loss or severe degradation of security or management functions.

5059 **6. Assessment Evidence**

- 5060 • [CONFIG] Resource scheduling/reservation configuration for security and management components.
- 5061 • [DESIGN] Documentation of resource scheduling mechanisms and their application to control-plane
5062 protection.
- 5063 • [LOG] / [OBSERVATION] Evidence that management and security functions remain operational under load
5064 (e.g. successful management operations, logs, monitoring outputs during stress tests).

5065 **6.3.1.13.4 Assessment Case AC-H-AVAIL-004**

5066 **1. Assessment Reference**

5067 Requirement: REQ-H-AVAIL-004 (Advanced)

5068 **2. Assessment Objective**

5069 Verify that:

- 5070 1) The Hypervisor supports replication of the runtime state of selected guest VMs (including CPU, memory and
5071 relevant I/O buffers) to standby instances.
- 5072 2) In the event of host or Hypervisor service failure, failover to the standby instance can be performed with
5073 minimal interruption to workload execution.

5074 **3. Assessment Preparation**

5075 The assessment shall have access to:

- 5076 • Documentation describing live or near-live state replication/failover mechanisms for guest VMs (e.g.
5077 continuous or periodic state replication, checkpointing) and supported limitations.
- 5078 • Documentation on configuration of primary and standby instances for selected guest VMs and any
5079 requirements for networking/storage alignment.
- 5080 • A test environment with at least one pair of Hypervisor hosts supporting a primary/standby configuration for a
5081 test VM.

5082 **4. Assessment Activities**

5083 The assessment shall at least:

- 5084 1) Configure a test guest VM with runtime state replication to a standby instance as described in the
5085 documentation.
- 5086 2) Verify that the runtime state (CPU and memory, and associated I/O buffers as applicable) of the primary VM
5087 is replicated to the standby according to the Hypervisor's replication mechanism.
- 5088 3) Induce or simulate failure of the host or Hypervisor service hosting the primary VM and verify that the VM
5089 fails over to the standby instance.
- 5090 4) Observe the impact on workload execution and confirm that service interruption is limited to the minimal
5091 duration inherent to the replication/failover mechanism (for example, brief pause or reconnection), without
5092 requiring full VM reboot from scratch.

5093 **5. Assessment Verdict**

- 5094 • Pass: runtime state replication to a standby instance is supported and functional for selected guest VMs;
5095 failover in response to host/Hypervisor failure brings the standby instance into service using replicated state;
5096 and workload interruption is minimal compared with a full restart.
- 5097 • Fail: runtime state replication to a standby instance cannot be configured or does not function as documented;
5098 or failover requires full VM restart without using replicated state; or workload interruption is not consistent
5099 with "minimal interruption" as described in the documentation.

5100 **6. Assessment Evidence**

- 5101 • [CONFIG] Primary/standby and state-replication configuration for selected guest VMs.
- 5102 • [DESIGN] Documentation of VM state replication and failover architecture, including scope of replicated state
5103 and expected failover behaviour.
- 5104 • [LOG] / [OBSERVATION] Evidence of ongoing state replication, recorded failover events and observed
5105 workload behaviour during and after failover.

5106 **6.3.1.14 Assessment for Logging**

5107 **6.3.1.14.1 Assessment Case AC-H-LOG-001**

5108 **1. Assessment Reference**

5109 Requirement: REQ-H-LOG-001 (Basic)

5110 **2. Assessment Objective**

5111 Verify that the Hypervisor:

- 5112 1) Generates audit logs for security-relevant administrative actions and security events.
- 5113 2) Logs at least: authentication attempts, configuration changes, and virtual machine lifecycle operations.

5114 **3. Assessment Preparation**

5115 The assessment shall have access to:

- 5116 • Documentation of the Hypervisor logging/auditing capabilities and log formats.
- 5117 • Documentation describing which actions/events are considered security-relevant.
- 5118 • A configuration with administrative access enabled and at least one VM that can be created, modified,
5119 started/stopped and deleted.

5120 **4. Assessment Activities**

5121 The assessment shall at least:

- 5122 1) Review documentation/configuration to identify the audit log sources and where audit logs are stored.

- 5123 2) Perform, in the assessed configuration:
- 5124 - successful and failed administrative authentication attempts;
- 5125 - administrative configuration changes (e.g. policy or settings updates);
- 5126 - VM lifecycle operations (e.g. create, start, stop, delete).
- 5127 3) Verify that audit logs record these actions/events with sufficient detail (at least: type of action/event, affected
- 5128 object, timestamp, and responsible account or process where applicable).

5129 5. Assessment Verdict

- 5130 • Pass: the Hypervisor generates audit logs for security-relevant administrative actions and security events,
- 5131 including authentication attempts, configuration changes, and VM lifecycle operations, with sufficient detail.
- 5132 • Fail: any of the required actions/events are not logged, or log records lack sufficient information to support
- 5133 accountability and analysis.

5134 6. Assessment Evidence

- 5135 • [CONFIG] Logging/audit configuration for the Hypervisor.
- 5136 • [DESIGN] Documentation of audit logging behaviour and logged event categories.
- 5137 • [LOG] / [OBSERVATION] Sample audit log entries showing authentication attempts, configuration changes
- 5138 and VM lifecycle operations.

5139 6.3.1.14.2 Assessment Case AC-H-LOG-002

5140 1. Assessment Reference

5141 Requirement: REQ-H-LOG-002 (Elevated)

5142 2. Assessment Objective

5143 Verify that:

- 5144 1) Audit logs are protected from Unauthorized access, modification or deletion.
- 5145 2) Access to audit logs is restricted to authorized administrative roles or processes.
- 5146 3) The Hypervisor supports secure export of audit logs to external log management/SIEM systems using
- 5147 authenticated and encrypted channels.

5148 3. Assessment Preparation

5149 The assessment shall have access to:

- 5150 • Documentation on storage locations for audit logs and associated access controls (file permissions, roles,
- 5151 APIs).
- 5152 • Documentation on roles/processes allowed to access or manage audit logs.
- 5153 • Documentation of log export mechanisms and supported secure transport options (e.g. TLS-protected syslog,
- 5154 HTTPS, secure agents).
- 5155 • A configuration with audit logging enabled and at least one external log management or SIEM endpoint (test
- 5156 instance) reachable via a secure channel.

5157 4. Assessment Activities

5158 The assessment shall at least:

- 5159 1) Review configuration and underlying platform permissions to confirm that audit log storage is restricted to
- 5160 authorized administrative roles/processes and that modification/deletion by non-authorized users or workloads
- 5161 is prevented.

- 5162 2) Attempt to read, modify or delete audit logs using a non-privileged account or process and verify that these
5163 operations are denied.
- 5164 3) Configure secure export of audit logs to an external log management/SIEM endpoint using an authenticated,
5165 encrypted channel as documented (e.g. TLS with server authentication and, where applicable, client
5166 authentication).
- 5167 4) Verify, via configuration and protocol inspection, that the export occurs over authenticated and encrypted
5168 channels and that logs received by the external system correspond to those generated locally.

5169 5. Assessment Verdict

- 5170 • Pass: audit logs are protected against Unauthorized access/modification/deletion; only authorized
5171 administrative roles/processes can access them; and secure, authenticated and encrypted export to external
5172 log/SIEM systems is supported and functional.
- 5173 • Fail: non-authorized access or tampering with audit logs is possible; or access controls on audit logs are
5174 insufficient; or log export uses unauthenticated/unencrypted channels or cannot be securely configured.

5175 6. Assessment Evidence

- 5176 • [CONFIG] Audit log storage permissions, role mappings and export configuration (including security
5177 settings).
- 5178 • [DESIGN] Documentation of audit log protection model and secure export mechanisms.
- 5179 • [LOG] / [OBSERVATION] Evidence of denied access attempts by non-privileged accounts and successful
5180 secure export (e.g. protocol traces, external log records).

5181 6.3.1.14.3 Assessment Case AC-H-LOG-003

5182 1. Assessment Reference

5183 Requirement: REQ-H-LOG-003 (Advanced)

5184 2. Assessment Objective

5185 Verify that:

- 5186 1) The Hypervisor supports cryptographic protection of audit logs (e.g. signatures, MACs, chained hashes) to
5187 enable verification of the origin and integrity of audit records.
- 5188 2) Trusted time-stamping is applied to audit logs, enabling verification of the timing of events during forensic
5189 analysis.
- 5190 3) The combination of cryptographic protection and time-stamping allows later verification of origin and
5191 integrity of audit records.

5192 3. Assessment Preparation

5193 The assessment shall have access to:

- 5194 • Documentation of cryptographic mechanisms used to protect audit logs (e.g. per-record signatures, HMAC,
5195 hash chains, log sealing) and how verification is performed.
- 5196 • Documentation of the time source and time-stamping method used for audit records (e.g. synchronized system
5197 time, external time service, secure clock).
- 5198 • A configuration with cryptographic log protection and trusted time-stamping enabled, with the ability to export
5199 or access raw audit log files/streams for verification.

5200 4. Assessment Activities

5201 The assessment shall at least:

- 5202 1) Enable cryptographic protection and trusted time-stamping for audit logs as described in the documentation.
- 5203 2) Generate a set of audit events (e.g. logins, configuration changes, VM lifecycle operations) and capture the
5204 resulting audit logs.

- 5205 3) Using documented tools or procedures, verify that:
- 5206 - cryptographic checks (e.g. signatures/MACs/hash chains) over the audit records succeed, demonstrating
- 5207 integrity and origin as per design; and
- 5208 - each audit record includes a trusted timestamp consistent with the configured time source.
- 5209 4) Attempt to alter or remove part of the audit log (in a test copy) and confirm that integrity verification fails or
- 5210 clearly indicates tampering.

5211 5. Assessment Verdict

- 5212 • Pass: cryptographic mechanisms and trusted time-stamping are available and can be used to verify the origin
- 5213 and integrity of audit records; and tampering with audit logs is detectable through the provided verification
- 5214 process.
- 5215 • Fail: cryptographic protection or trusted time-stamping of audit logs cannot be enabled; or verification of
- 5216 origin and integrity is not possible or ineffective; or modifications to audit logs cannot be reliably detected.

5217 6. Assessment Evidence

- 5218 • [CONFIG] Configuration enabling cryptographic protection and trusted time-stamping of audit logs.
- 5219 • [DESIGN] Documentation of log protection and time-stamping architecture, including verification procedures
- 5220 and time sources.
- 5221 • [LOG] / [OBSERVATION] Audit logs and verification outputs showing successful integrity/origin checks for
- 5222 unmodified logs and detection of tampering in modified logs.

5223 6.3.1.15 Assessment for Patches and Updates

5224 6.3.1.15.1 Assessment Case AC-H-UPD-001

5225 1. Assessment Reference

5226 Requirement: REQ-H-UPD-001 (Basic)

5227 2. Assessment Objective

5228 Verify that:

- 5229 1) The Hypervisor supports applying security patches and updates without a full reinstallation of the Hypervisor
- 5230 or host system.
- 5231 2) Applicable update mechanisms for the evaluated solution (native Hypervisor updater, host OS update service,
- 5232 management appliance, orchestration pipeline) can update Hypervisor binaries and related components in-
- 5233 place.

5234 3. Assessment Preparation

5235 The assessment shall have access to:

- 5236 • Documentation of Hypervisor update mechanisms (direct Hypervisor updates, host OS updates, appliance
- 5237 images, orchestration pipelines) used in the evaluated solution.
- 5238 • Documentation describing which components constitute the Hypervisor and related components in scope
- 5239 (binaries, modules, drivers).
- 5240 • A test configuration where a newer Hypervisor version or security patch is available for application.

5241 4. Assessment Activities

5242 The assessment shall at least:

- 5243 1) Identify the update mechanism(s) used to deliver patches/updates to the Hypervisor in the evaluated solution.
- 5244 2) Apply a Hypervisor update or security patch using the documented mechanism(s) and verify that the update
- 5245 completes without requiring a full reinstallation of the Hypervisor or host system.

- 5246 3) Verify after update that the Hypervisor version/build reflects the applied patch and that existing configuration
5247 and guest VMs remain available according to the product's normal update behaviour.

5248 5. Assessment Verdict

- 5249 • Pass: the Hypervisor (as deployed in the evaluated solution) can be patched or updated using the documented
5250 mechanism(s) without full reinstallation of the Hypervisor or host system, and the updated Hypervisor is active
5251 afterwards.
- 5252 • Fail: patching or update of the Hypervisor requires full reinstallation of the Hypervisor or host system; or
5253 Hypervisor updates cannot be applied through the documented product update mechanism.

5254 6. Assessment Evidence

- 5255 • [CONFIG] Update mechanism configuration (repositories, management appliance/orchestrator settings, image
5256 update settings).
- 5257 • [DESIGN] Documentation of the Hypervisor update model and relation to host OS/appliance/orchestrator.
- 5258 • [LOG] / [OBSERVATION] Evidence of a successful in-place Hypervisor update (before/after version, update
5259 logs, continuity of configuration/VMs).

5260 6.3.1.15.2 Assessment Case AC-H-UPD-002

5261 1. Assessment Reference

5262 Requirement: REQ-H-UPD-002 (Elevated)

5263 2. Assessment Objective

5264 Verify that:

- 5265 1) The Hypervisor verifies authenticity and integrity of patches and updates using digital signatures validated
5266 against trusted keys or certificates configured for the Hypervisor.
- 5267 2) Updates are blocked if authenticity or integrity verification fails.

5268 3. Assessment Preparation

5269 The assessment shall have access to:

- 5270 • Documentation describing the update verification process (signature format, trust anchors, key/certificate
5271 management).
- 5272 • Documentation of where trusted keys/certificates are configured and how they are protected.
- 5273 • Ability to obtain:
- 5274 - a valid, properly signed update; and
- 5275 - a modified or incorrectly signed update (or equivalent test artefact) to simulate verification failure.

5276 4. Assessment Activities

5277 The assessment shall at least:

- 5278 1) Apply a valid signed update and verify that signature/integrity verification is performed as documented and
5279 that the update is accepted and installed.
- 5280 2) Attempt to apply an update with invalid or missing signatures, or an intentionally modified update package,
5281 and verify that:
- 5282 - authenticity/integrity verification fails; and
- 5283 - the Hypervisor prevents installation of the update and leaves the Hypervisor in a non-updated, consistent
5284 state.
- 5285 3) Confirm from documentation/configuration that the trusted keys/certificates used for verification are explicitly
5286 configured and not trivially modifiable by non-authorized parties.

5287 5. Assessment Verdict

- 5288 • Pass: all patches/updates are subject to authenticity and integrity verification using trusted keys/certificates,
5289 and updates with failed verification are not installed.
- 5290 • Fail: updates can be installed without signature/integrity verification; or verification can be trivially bypassed;
5291 or updates with failed verification can still be installed.

5292 6. Assessment Evidence

- 5293 • [CONFIG] Configuration of update verification, trust anchors (keys/certificates) and update sources.
- 5294 • [DESIGN] Documentation of the update verification process and trust model.
- 5295 • [LOG] / [OBSERVATION] Evidence showing successful installation of a valid signed update and prevention
5296 of installation for an invalid/tampered update.

5297 6.3.1.15.3 Assessment Case AC-H-UPD-003

5298 1. Assessment Reference

5299 Requirement: REQ-H-UPD-003 (Elevated)

5300 2. Assessment Objective

5301 Verify that:

- 5302 1) The Hypervisor implements rollback protection to prevent Unauthorized installation of outdated, revoked or
5303 replayed updates.
- 5304 2) Authorized rollback to a previously trusted version is only permitted when:
 - 5305 - the rollback image is verified for integrity and authenticity against a trusted key/certificate;
 - 5306 - the rollback action is explicitly authorized by an administrator with elevated privileges; and
 - 5307 - the rollback event is logged.

5308 3. Assessment Preparation

5309 The assessment shall have access to:

- 5310 • Documentation of rollback protection mechanisms (version tracking, anti-replay, revocation handling) and
5311 rollback procedures.
- 5312 • Documentation of administrative roles/privileges required to initiate rollback and the logging of
5313 update/rollback events.
- 5314 • A configuration that supports both forward updates and rollback to a previous version.

5315 4. Assessment Activities

5316 The assessment shall at least:

- 5317 1) Confirm from documentation and configuration that mechanisms exist to detect and prevent installation of
5318 outdated, revoked or replayed updates (e.g. version counters, manifests, revocation lists).
- 5319 2) Attempt to install an older, revoked or replayed update image without using the documented authorized
5320 rollback procedure and verify that the installation is blocked by rollback protection.
- 5321 3) Perform an authorized rollback using the documented procedure, ensuring that:
 - 5322 - the rollback image is verified for integrity and authenticity;
 - 5323 - the rollback action requires and is performed by an administrator with elevated privileges; and
 - 5324 - the rollback event is recorded in the audit/update logs.
- 5325 4) Verify that, after rollback, the Hypervisor runs the earlier trusted version and that the logs capture the rollback
5326 with sufficient detail (who, when, from/to which version).

5327 **5. Assessment Verdict**

5328 • Pass: Unauthorized installation of outdated/revoked/replayed updates is prevented; and authorized rollback is
5329 only possible following integrity/authenticity verification, elevated administrative approval, and logging of the
5330 rollback event.

5331 • Fail: outdated or revoked updates can be installed without appropriate controls; or rollback does not require
5332 elevated authorization; or rollback events are not logged.

5333 **6. Assessment Evidence**

5334 • [CONFIG] Update/rollback configuration, including version/rollback controls and admin permission settings.

5335 • [DESIGN] Documentation of rollback protection mechanisms and authorized rollback process.

5336 • [LOG] / [OBSERVATION] Evidence of blocked Unauthorized rollback attempts and logged authorized
5337 rollback operations.

5338 **6.3.1.15.4 Assessment Case AC-H-UPD-004**

5339 **1. Assessment Reference**

5340 Requirement: REQ-H-UPD-004 (Advanced)

5341 **2. Assessment Objective**

5342 Verify that:

5343 1) The Hypervisor supports live patching of critical Hypervisor components (including kernel/microkernel,
5344 device drivers and security modules) to remediate vulnerabilities without requiring shutdown of guest VMs.

5345 2) Live patching produces only minimal interruption to workload execution.

5346 **3. Assessment Preparation**

5347 The assessment shall have access to:

5348 • Documentation describing live patching capabilities (supported components, limitations, expected impact on
5349 workloads).

5350 • Documentation on how live patches are prepared, distributed and applied.

5351 • A test configuration with:

5352 - running guest VMs executing representative workloads; and

5353 - at least one applicable live patch for a critical component.

5354 **4. Assessment Activities**

5355 The assessment shall at least:

5356 1) Identify which critical components (kernel/microkernel, drivers, security modules) are supported for live
5357 patching.

5358 2) With guest VMs and workloads running, apply a live patch to at least one supported critical component
5359 according to the documented procedure.

5360 3) Observe guest VM behaviour and workload execution during and after live patching and verify that:

5361 - guest VMs are not shut down for the patch to be applied; and

5362 - any interruption to workload execution is limited to brief, documented pauses (if any), consistent with
5363 "minimal interruption" as described by the product.

5364 4) Verify that the Hypervisor reports the updated patch level/version for the patched component and that logs
5365 record the live patch operation.

5366 **5. Assessment Verdict**

5367 • Pass: live patching for critical Hypervisor components is supported and functional; guest VMs remain running
5368 during patch application; and workload interruption is minimal and consistent with product documentation.

5369 • Fail: critical components cannot be patched without shutting down guest VMs; or patching causes significant
5370 disruption beyond documented expectations; or live patching behaviour diverges from the requirement.

5371 6. Assessment Evidence

5372 • [CONFIG] Live patching configuration and applied patch information.

5373 • [DESIGN] Documentation of live patching architecture, supported components and expected impact.

5374 • [LOG] / [OBSERVATION] Evidence of live patch application, updated patch level and observed workload
5375 continuity with only minimal interruption.

5376 6.3.1.16 Assessment for Secure Configuration and Default

5377 6.3.1.16.1 Assessment Case AC-H-CFG-001

5378 1. Assessment Reference

5379 Requirement: REQ-H-CFG-001 (Basic)

5380 2. Assessment Objective

5381 Verify that:

5382 1) By default, the Hypervisor disables remote administrative access pathways and insecure management
5383 interfaces that are not required for initial provisioning (e.g. unauthenticated/weak remote shell, legacy or
5384 unencrypted management APIs, non-essential vendor diagnostic ports).

5385 2) The Hypervisor provides configuration mechanisms that allow administrators to explicitly enable only those
5386 interfaces and services needed for the intended Virtualization, orchestration and core host management
5387 operations.

5388 3. Assessment Preparation

5389 The assessment shall have access to:

5390 • Documentation listing Hypervisor administrative/management interfaces (remote shell, APIs, diagnostic ports,
5391 web consoles, etc.) and which are required for initial provisioning.

5392 • Documentation describing configuration options to enable/disable interfaces and services.

5393 • A fresh or default installation of the Hypervisor in its documented default configuration.

5394 4. Assessment Activities

5395 The assessment shall at least:

5396 1) In the default configuration, attempt to access all documented administrative and management interfaces
5397 (including remote shell, legacy APIs, diagnostic ports) and verify that only those required for initial
5398 provisioning are enabled; insecure or unnecessary interfaces are disabled by default.

5399 2) Review configuration mechanisms (e.g. CLI, API, UI) and verify that administrators can explicitly enable
5400 specific interfaces/services; then enable a subset of interfaces required for typical
5401 Virtualization/orchestration/host management operations and confirm:

5402 - chosen interfaces become accessible; and

5403 - interfaces/services not explicitly enabled remain disabled.

5404 5. Assessment Verdict

5405 • Pass: non-essential and insecure administrative/management interfaces are disabled by default; and
5406 administrators can enable only those interfaces/services required for intended operations via explicit
5407 configuration.

- 5408 • Fail: insecure or non-required administrative interfaces are enabled by default; or there is no practical way to
- 5409 selectively enable only necessary interfaces/services.

5410 6. Assessment Evidence

- 5411 • [CONFIG] Default and modified configuration showing interface/service enablement states.
- 5412 • [DESIGN] Documentation of management interfaces, default exposure, and configuration mechanisms.
- 5413 • [OBSERVATION] / [LOG] Connection attempts and screenshots/CLI output showing which interfaces are
- 5414 reachable in default and explicitly enabled states.

5415 6.3.1.16.2 Assessment Case AC-H-CFG-002

5416 1. Assessment Reference

5417 Requirement: REQ-H-CFG-002 (Elevated)

5418 2. Assessment Objective

5419 Verify that:

- 5420 1) The Hypervisor validates configuration parameters before applying them.
- 5421 2) Configuration validation logic detects and prevents configurations that would:
 - 5422 - disable or bypass security requirements defined in the present document; or
 - 5423 - violate guarantees for VM isolation, control-plane protection, or minimum resource allocations for
 - 5424 security/management functions.
- 5425 3) When a configuration change is rejected for security reasons, the Hypervisor provides clear feedback
- 5426 indicating the reason for rejection.

5427 3. Assessment Preparation

5428 The assessment shall have access to:

- 5429 • Documentation describing configuration validation, including examples of invalid or disallowed settings (e.g.
- 5430 disabling mandatory controls, removing isolation, reducing reserved resources below minimums).
- 5431 • Documentation listing mandatory security controls and key guarantees (VM isolation, control-plane
- 5432 protection, minimum security/management resources).
- 5433 • A test environment where security-relevant configuration parameters can be changed.

5434 4. Assessment Activities

5435 The assessment shall at least:

- 5436 1) Identify a set of configuration parameters whose insecure values would:
 - 5437 - disable/bypass mandatory security controls;
 - 5438 - break VM isolation;
 - 5439 - undermine control-plane protection; or
 - 5440 - reduce reserved resources for security/management below documented minima.
- 5441 2) Attempt to apply such insecure configurations and verify that the Hypervisor rejects or blocks them based on
- 5442 configuration validation.
- 5443 3) Observe the feedback presented to the administrator for each rejected change and verify that it clearly indicates
- 5444 the security-related reason (e.g. "would disable mandatory control X", "would violate minimum resource
- 5445 reservation for control plane", "would compromise VM isolation").
- 5446 4) Apply valid configuration changes and verify that they are accepted and correctly enforced, demonstrating that
- 5447 validation is specific to insecure cases, not a generic failure.

5448 5. Assessment Verdict

- 5449 • Pass: configuration parameters are validated before application; attempts to apply configurations that
- 5450 disable/bypass mandatory security controls or violate isolation/control-plane/minimum-resource guarantees are
- 5451 prevented; and clear feedback is provided for security-related rejections.
- 5452 • Fail: unsafe configurations can be applied; or configuration validation does not cover the specified guarantees;
- 5453 or security-related rejection feedback is missing or unclear.

5454 6. Assessment Evidence

- 5455 • [CONFIG] Examples of attempted insecure configurations and resulting states (accepted/rejected).
- 5456 • [DESIGN] Documentation of configuration validation logic and mappings to mandatory controls and key
- 5457 guarantees.
- 5458 • [LOG] / [OBSERVATION] CLI/UI output and error messages showing rejection of insecure configurations
- 5459 and the reasons given; evidence of accepted secure configuration changes.

5460 6.3.1.16.3 Assessment Case AC-H-CFG-003

5461 1. Assessment Reference

5462 Requirement: REQ-H-CFG-003 (Advanced)

5463 2. Assessment Objective

5464 Verify that:

- 5465 1) The Hypervisor supports defining and using security configuration baselines specifying secure settings for
- 5466 critical components, including at least:
 - 5467 - I/O memory management unit (IOMMU) configuration;
 - 5468 - virtual switch policies;
 - 5469 - host firewall rules, as implemented within the Hypervisor product.
- 5470 2) The Hypervisor can automatically validate current configuration against a selected security baseline.
- 5471 3) The Hypervisor generates alerts or audit events when non-compliant settings are detected.

5472 3. Assessment Preparation

5473 The assessment shall have access to:

- 5474 • Documentation describing security configuration baselines (how they are defined, stored, selected and
- 5475 applied).
- 5476 • Documentation showing how baseline rules cover IOMMU configuration, virtual switch policies and host
- 5477 firewall rules within the Hypervisor product.
- 5478 • Documentation of automatic baseline validation mechanisms and alert/audit behaviour on non-compliance.
- 5479 • A test configuration where at least one security baseline can be defined/selected and where deviations from
- 5480 that baseline can be introduced.

5481 4. Assessment Activities

5482 The assessment shall at least:

- 5483 1) Define or select a security configuration baseline that includes explicit requirements for IOMMU settings,
- 5484 virtual switch policies and host firewall rules, according to product capabilities.
- 5485 2) Ensure the Hypervisor configuration is aligned with the baseline and run the baseline validation function;
- 5486 verify that the Hypervisor reports compliance and does not generate non-compliance alerts/events.
- 5487 3) Introduce one or more deliberate deviations from the baseline (e.g. relax a firewall rule, disable or weaken a
- 5488 virtual switch policy, misconfigure IOMMU-related settings) and run baseline validation again.

- 5489 4) Verify that:
- 5490 - the Hypervisor detects the non-compliant settings; and
- 5491 - alerts or audit events are generated indicating which configuration elements are non-compliant with the
- 5492 selected baseline.

5493 5. Assessment Verdict

- 5494 • Pass: security baselines can be defined/used for critical components (IOMMU, virtual switch, host firewall);
- 5495 baseline validation can be run automatically; and non-compliant settings trigger alerts or audit events
- 5496 identifying the deviations.
- 5497 • Fail: security baselines cannot be defined or do not cover the required components; automatic validation
- 5498 against a baseline is not available; or non-compliance does not result in clear alerts/audit events.

5499 6. Assessment Evidence

- 5500 • [CONFIG] Example security baseline definition and corresponding Hypervisor configuration (before and after
- 5501 deviation).
- 5502 • [DESIGN] Documentation of security baseline management, validation logic and alert/audit integration.
- 5503 • [LOG] / [OBSERVATION] Outputs from baseline validation showing compliant and non-compliant states,
- 5504 including generated alerts or audit log entries identifying non-compliant settings.

5505 6.3.1.17 Assessment for Data Minimization

5506 6.3.1.17.1 Assessment Case AC-H-DM-001

5507 1. Assessment Reference

5508 Requirement: REQ-H-DM-001 (Basic)

5509 2. Assessment Objective

5510 Verify that:

- 5511 1) Hypervisor logs, telemetry and diagnostic outputs are limited to information required for operation,
- 5512 troubleshooting and security monitoring.
- 5513 2) Logs/telemetry/diagnostics do not contain sensitive security material such as plaintext authentication
- 5514 credentials, full cryptographic keys or complete VM memory contents.
- 5515 3) Payload or application data from guest workloads is not included in Hypervisor logs/telemetry/diagnostics by
- 5516 default.
- 5517 4) Where inclusion of payload content is supported for debugging, it is only enabled by explicit administrative
- 5518 configuration and this state is clearly indicated to the administrator.

5519 3. Assessment Preparation

5520 The assessment shall have access to:

- 5521 • Documentation of Hypervisor logging/telemetry/diagnostic mechanisms, including examples of fields and any
- 5522 "debug" or "deep inspection" modes.
- 5523 • Documentation describing which information is explicitly excluded from logs/telemetry (credentials, keys,
- 5524 VM memory, guest payload).
- 5525 • A test configuration with:
- 5526 - administrative access to configure logging/telemetry/diagnostics;
- 5527 - at least one VM/guest workload that processes identifiable test payload data;
- 5528 - ability to generate authentication events and crypto operations with known test values.

5529 4. Assessment Activities

5530 The assessment shall at least:

- 5531 1) Review documentation and configuration to identify:
- 5532 - what categories of events and data are logged/collected by default; and
- 5533 - any options for deep/verbose/debug logging that may include payload or detailed context.
- 5534 2) In default configuration, perform:
- 5535 - successful and failed administrative authentication attempts using known test usernames/password
- 5536 patterns;
- 5537 - crypto operations (e.g. key generation/usage) with identifiable test keys;
- 5538 - normal VM workload execution with test payload data (e.g. specific strings or patterns in application
- 5539 traffic).
- 5540 Then inspect logs, telemetry and diagnostics to verify that:
- 5541 - plaintext credentials are not present;
- 5542 - full cryptographic keys or complete VM memory dumps are not present; and
- 5543 - guest payload/application data is not present by default.
- 5544 3) Where the Hypervisor supports payload inclusion for debugging, enable the relevant option(s) using
- 5545 administrative configuration and verify that:
- 5546 - only an administrator can enable this behaviour;
- 5547 - configuration/status clearly indicates that payload capture/debug logging is enabled; and
- 5548 - after enabling, test payload data from guest workloads appears in logs/diagnostics as documented.

5549 5. Assessment Verdict

- 5550 • Pass: by default, logs/telemetry/diagnostics are limited to necessary operational and security data; they do not
- 5551 contain plaintext credentials, full keys or complete VM memory; guest payload data is excluded by default;
- 5552 and any payload inclusion requires explicit administrative configuration and is clearly indicated.
- 5553 • Fail: sensitive security material or guest payload data is present in default logs/telemetry/diagnostics; or
- 5554 payload/debug capture can be enabled without explicit administrative action or without clear indication to the
- 5555 administrator.

5556 6. Assessment Evidence

- 5557 • [CONFIG] Logging/telemetry/diagnostic configuration (default and with debug/payload options, if any).
- 5558 • [DESIGN] Documentation of data minimization behaviour, including exclusions for credentials, keys, VM
- 5559 memory and guest payload.
- 5560 • [LOG] / [OBSERVATION] Samples of logs/telemetry/diagnostics under default configuration and with
- 5561 debug/payload options enabled, demonstrating absence/presence of test secrets and payload as appropriate.

5562 6.3.1.17.2 Assessment Case AC-H-DM-002

5563 1. Assessment Reference

5564 Requirement: REQ-H-DM-002 (Basic)

5565 2. Assessment Objective

5566 Verify that:

- 5567 1) The Hypervisor provides administrative controls to restrict and, where necessary, disable collection of specific
- 5568 categories of log, telemetry and diagnostic data.

- 5569 2) These controls include the ability to disable collection of particular log types, telemetry streams, metrics, or
5570 debug payload capture.

5571 3. Assessment Preparation

5572 The assessment shall have access to:

- 5573 • Documentation describing configurable categories of logs, telemetry streams, metrics and any debug/payload
5574 capture options.
- 5575 • Documentation of administrative roles/permissions required to modify logging/telemetry configuration.
- 5576 • A test configuration where different log categories, telemetry streams and metrics can be enabled/disabled and
5577 where events can be generated to exercise them.

5578 4. Assessment Activities

5579 The assessment shall at least:

- 5580 3) Identify, from documentation and configuration, the configurable categories (e.g. audit logs vs debug logs,
5581 specific telemetry streams, metrics groups, payload capture options).
- 5582 4) Using administrative controls, configure a subset of categories to be disabled (e.g. disable a particular debug
5583 log type, a telemetry stream and, if supported, debug payload capture).
- 5584 5) Generate events and workload activity that would normally produce entries in the disabled categories and
5585 verify that:
 - 5586 - no new records are produced for the disabled log types/streams/metrics; and
 - 5587 - other, non-disabled categories continue to function as configured.
- 5588 6) Re-enable one or more previously disabled categories and verify that records are again produced for those
5589 categories when new events occur.

5590 5. Assessment Verdict

- 5591 • Pass: administrators can restrict/disable specific log types, telemetry streams, metrics and any debug payload
5592 capture; disabled categories no longer produce data while others continue to operate; and re-enabling restores
5593 data collection as configured.
- 5594 • Fail: logging/telemetry categories cannot be selectively restricted or disabled; or documented categories
5595 remain collected despite being configured as disabled.

5596 6. Assessment Evidence

- 5597 • [CONFIG] Logging/telemetry/metrics configuration before and after enabling/disabling categories, including
5598 debug/payload options.
- 5599 • [DESIGN] Documentation of administrative controls for data collection and category granularity (log types,
5600 streams, metrics, payload capture).
- 5601 • [LOG] / [OBSERVATION] Evidence showing presence/absence of data from specific categories when
5602 enabled vs disabled, demonstrating effective restriction and disabling.

5603 6.3.2 M&O System

5604 [To be completed in a future version]

5605 6.4 CES

5606 6.4.1 CRS

5607 6.4.1.1 Assessment for Container Isolation

5608 6.4.1.1.1 Assessment Case AC-CRS-CN-ISO-001

5609 1. Assessment Reference

5610 Requirement: REQ-CRS-CN-ISO-001 (Basic)

5611 2. Assessment Objective

5612 Verify that:

- 5613 1) The CRS enforces isolation between containers using OS isolation mechanisms under its control, including at
5614 least process, filesystem, network and resource management layers.
- 5615 2) The CRS supports configurations in which one container cannot read, write or otherwise interfere with another
5616 container's processes, filesystem contents, network endpoints or allocated resources.

5617 3. Assessment Preparation

5618 The assessment shall have access to:

- 5619 • Documentation of the CRS container isolation model (e.g. namespaces, cgroups/quotas, virtual networking,
5620 per-container filesystems).
- 5621 • Documentation describing how containers are configured to achieve strict isolation (e.g. non-shared PID
5622 (Process ID) /IPC (Inter-Process Communication) /network/mount namespaces, independent root filesystems).
- 5623 • A test environment with at least two containers running on the same CRS-managed host, under different
5624 identities/workloads, with the ability to execute commands inside each container.

5625 4. Assessment Activities

5626 The assessment shall at least:

- 5627 1) Process isolation
 - 5628 - From container A, list processes and attempt to inspect or signal processes belonging to container B.
 - 5629 - Verify that only processes of container A are visible/controllable.
- 5630 2) Filesystem isolation
 - 5631 - From container A, enumerate filesystem paths and attempt to access directories or files that are part of
5632 container B's filesystem (other than intentionally shared volumes, if any).
 - 5633 - Verify that container A cannot read or modify container B's filesystem contents.
- 5634 3) Network isolation
 - 5635 - Identify network endpoints assigned to each container.
 - 5636 - From container A, attempt to bind to IP addresses/ports assigned to container B or otherwise interfere
5637 with B's network endpoints beyond normal, configured connectivity (e.g. service-level communication).
 - 5638 - Verify that container A cannot hijack or directly interfere with container B's endpoints.
- 5639 4) Resource isolation
 - 5640 - Review configured resource controls (e.g. CPU and memory limits) and generate load in container A.
 - 5641 - Verify that container B continues to operate as expected and that its resource allocations behave
5642 according to the configured limits, without arbitrary starvation caused by A beyond normal scheduling
5643 behaviour.

5644 5. Assessment Verdict

- 5645 • Pass: documentation and tests confirm that process, filesystem, network and resource isolation are enforced,
5646 and that one container cannot read, write or otherwise interfere with another container's execution, data or
5647 configuration state in the assessed configuration.
- 5648 • Fail: a container can see or control another container's processes, access its filesystem unexpectedly, interfere
5649 with its network endpoints or consume resources in a way that bypasses configured isolation.

5650 6. Assessment Evidence

- 5651 • [SCOPE] Description of CRS deployment, host and containers used for the isolation assessment.
- 5652 • [CONFIG] CRS configuration for namespaces, networking, storage and resource limits for the test containers.
- 5653 • [DESIGN] Documentation of CRS container isolation mechanisms across process, filesystem, network and
5654 resource layers.
- 5655 • [OBSERVATION] / [LOG] Command outputs/logs showing isolation behaviour and blocked cross-container
5656 access attempts.

5657 6.4.1.1.2 Assessment Case AC-CRS-CN-ISO-002

5658 1. Assessment Reference

5659 Requirement: REQ-CRS-CN-ISO-002 (Elevated)

5660 2. Assessment Objective

5661 Verify that:

- 5662 1) The CRS supports configuration of strong separation between containers and the host OS by restricting
5663 container access to kernel interfaces, privileged operations and system resources outside their assigned
5664 execution context.
- 5665 2) The CRS can:
 - 5666 - limit container privileges (e.g. reduced kernel capabilities, prevention of access to host-level device
5667 nodes and sensitive filesystems); and
 - 5668 - restrict container access to kernel interfaces and system calls not required for the declared workload,
5669 using mechanisms such as syscall filtering, mandatory access control (MAC) policies or equivalent
5670 controls.

5671 3. Assessment Preparation

5672 The assessment shall have access to:

- 5673 • Documentation of CRS mechanisms for strong container-host separation (e.g. capabilities model, device-node
5674 access controls, mount policies, syscall filtering, MAC policies).
- 5675 • Documentation of typical secure configurations (profiles, templates or policies) for non-privileged containers.
- 5676 • A test environment where containers can be created with differing privilege profiles (e.g. reduced capabilities,
5677 syscall filters, MAC labels).

5678 4. Assessment Activities

5679 The assessment shall at least:

- 5680 1) Privilege reduction
 - 5681 - Configure a container with a restricted privilege profile as recommended (e.g. reduced capabilities, non-
5682 privileged user, no direct device-node access).
 - 5683 - Inside the container, attempt operations that require elevated privileges or kernel capabilities (e.g.
5684 loading kernel modules, raw socket creation, mounting host filesystems, accessing /dev nodes that should
5685 be restricted).

- 5686 - Verify that such operations are denied.
- 5687 2) Host resource protection
- 5688 - From the restricted container, attempt to access sensitive host filesystems or directories (e.g. host root,
5689 system configuration directories, host-level logs) that are not explicitly mounted into the container.
- 5690 - Verify that these locations are not accessible.
- 5691 3) Kernel interface and syscall restriction
- 5692 - Configure syscall filtering or equivalent kernel-interface restriction (e.g. seccomp profiles, MAC rules)
5693 for the container according to documentation for the declared workload.
- 5694 - From within the container, attempt to invoke known disallowed system calls or interact with kernel
5695 interfaces that the profile is intended to block.
- 5696 - Verify that such calls are blocked (e.g. denied, terminated) while permitted system calls required for
5697 normal workload operation continue to function.

5698 5. Assessment Verdict

- 5699 • Pass: the CRS can configure strong separation between containers and the host; privilege reduction
5700 mechanisms are effective; container access to device nodes and sensitive filesystems is restricted; and
5701 syscall/filtering/MAC-equivalent controls prevent access to non-required kernel interfaces while allowing
5702 necessary workload functions.
- 5703 • Fail: containers can retain broad privileges, directly access sensitive host resources, or bypass syscall/interface
5704 restrictions in the assessed secure configuration.

5705 6. Assessment Evidence

- 5706 • [CONFIG] Container profiles showing reduced capabilities, device/mount restrictions and syscall/MAC
5707 policies.
- 5708 • [DESIGN] Documentation of CRS mechanisms for strong container-host separation and recommended secure
5709 profiles.
- 5710 • [OBSERVATION] / [LOG] Outputs showing denied privileged operations, blocked access to host resources
5711 and enforcement of syscall/MAC restrictions.

5712 6.4.1.1.3 Assessment Case AC-CRS-CN-ISO-003

5713 1. Assessment Reference

5714 Requirement: REQ-CRS-CN-ISO-003 (Advanced)

5715 2. Assessment Objective

5716 Verify that:

- 5717 1) The CRS supports execution of containers within hardened isolation environments that strengthen separation
5718 between container workloads and the host kernel (e.g. micro-VM-based runtimes, user-space kernel
5719 sandboxes, additional Virtualization/sandbox layers or equivalent).
- 5720 2) In such hardened environments, compromise of a container does not directly grant an attacker equivalent
5721 access to the host kernel or to other containers.

5722 3. Assessment Preparation

5723 The assessment shall have access to:

- 5724 • Documentation describing hardened isolation environments supported by the CRS for containers (e.g.
5725 architecture diagrams and isolation properties of micro-VM runtimes, sandbox layers).
- 5726 • Documentation explaining configuration and operational conditions for enabling hardened isolation for
5727 selected workloads.

- 5728 • A test configuration where at least one container can be deployed in a hardened isolation mode and another
- 5729 container in a standard mode on the same host.

5730 4. Assessment Activities

5731 The assessment shall at least:

5732 1) Hardened environment configuration

- 5733 - Configure a container to run using the CRS hardened isolation mechanism (e.g. micro-VM-backed
- 5734 container, user-space kernel sandbox).
- 5735 - Verify from configuration/status that the container is indeed running with the hardened isolation mode
- 5736 enabled as documented.

5737 2) Containment of compromise within hardened environment

- 5738 - Within the hardened-isolation container, attempt operations that would normally be indicative of host
- 5739 compromise if successful (e.g. direct interaction with host kernel interfaces, access to host filesystems,
- 5740 access to other containers' namespaces or storage, escalation to host-level privileges).
- 5741 - Verify that such operations are confined or blocked by the hardened isolation layer and do not result in
- 5742 direct access to the host kernel or other containers.

5743 3) Separation from other containers

- 5744 - With at least one additional container running (standard or hardened), attempt from the hardened
- 5745 container to interact with or control the other container beyond normal, configured communication (e.g.
- 5746 attach to its processes, access its filesystems, inject code).
- 5747 - Verify that these attempts fail and that cross-container compromise via the hardened environment is
- 5748 prevented.

5749 4) Administrative visibility

- 5750 - Verify that administrators can identify containers running with hardened isolation (e.g. via configuration,
- 5751 labels, status commands or logs), so that use of this mode is auditable and can be targeted to high-risk
- 5752 workloads.

5753 5. Assessment Verdict

- 5754 • Pass: the CRS supports hardened isolation environments for containers; such environments can be configured
- 5755 for selected workloads; and in the assessed configuration, compromise of a container within a hardened
- 5756 environment does not directly provide equivalent access to the host kernel or other containers.
- 5757 • Fail: hardened isolation mode is not available or cannot be configured; or a compromised container in a
- 5758 hardened environment can directly access the host kernel or other containers in ways inconsistent with the
- 5759 strengthened isolation objective.

5760 6. Assessment Evidence

- 5761 • [CONFIG] CRS configuration showing containers deployed in hardened isolation mode and any associated
- 5762 policies.
- 5763 • [DESIGN] Documentation of hardened isolation architecture, including threat model and separation properties
- 5764 with respect to the host kernel and other containers.
- 5765 • [PLT-CAP] Where applicable, evidence of platform capabilities used to implement hardened environments
- 5766 (e.g. hardware or software Virtualization features).
- 5767 • [OBSERVATION] Logs/traces and test results demonstrating that host/kernel and cross-container access
- 5768 attempts from a hardened container are blocked or confined as designed.

5769 6.4.1.2 Assessment for Control Plane Isolation

5770 6.4.1.2.1 Assessment Case AC-CRS-CP-ISO-001

5771 1. Assessment Reference

5772 Requirement: REQ-CRS-CP-ISO-001 (Basic)

5773 2. Assessment Objective

5774 Verify that:

- 5775 1) The CRS provides access-control mechanisms for its management and control interfaces.
- 5776 2) The CRS can be configured such that containerized workloads cannot directly access or invoke CRS
5777 management or control functions.
- 5778 3) Only authenticated and authorized entities can perform control or configuration actions.
- 5779 4) CRS management sockets, control APIs and management daemons can be deployed so they are not reachable
5780 from within container namespaces.

5781 3. Assessment Preparation

5782 The assessment shall have access to:

- 5783 • Documentation of CRS management and control interfaces (local sockets, REST APIs, CLIs, management
5784 daemons) and their access-control mechanisms.
- 5785 • Documentation of how these interfaces are bound/deployed (e.g. host network namespace, Unix domain
5786 sockets, host-only endpoints).
- 5787 • A test environment with:
 - 5788 - at least one administrative account or management client;
 - 5789 - at least one container with its own namespaces;
 - 5790 - the ability to run commands inside containers and on the host.

5791 4. Assessment Activities

5792 The assessment shall at least:

- 5793 1) Identify all CRS management/control interfaces and confirm that authentication and authorization controls are
5794 configured (e.g. accounts, roles, policies).
 - 5795 2) From an authorized management context (host or management workstation), perform control actions (e.g.
5796 create/delete container, change policy) and verify that successful access requires authentication and that
5797 authorization (role/permission) checks are enforced.
 - 5798 3) From within a container:
 - 5799 - enumerate reachable network endpoints and local sockets;
 - 5800 - attempt to connect to CRS management sockets, control APIs or management daemons using no
5801 credentials and container-level credentials;
 - 5802 - attempt to perform control/configuration actions.
- 5803 Verify that:
- 5804 - CRS management endpoints are not reachable from within container namespaces (e.g. bound only to host
5805 network namespace or host-only sockets); and
 - 5806 - container principals cannot perform management or control actions.

5807 5. Assessment Verdict

- 5808 • Pass: access-control mechanisms exist; CRS can be deployed so that management/control interfaces are not
5809 reachable from container namespaces; and only authenticated, authorized entities can perform
5810 control/configuration actions.
- 5811 • Fail: containers can reach and invoke CRS management/control functions; or management interfaces do not
5812 enforce proper authentication/authorization.

5813 6. Assessment Evidence

- 5814 • [CONFIG] CRS management/control interface configuration, including bindings (host-only vs container-
5815 visible) and access-control settings.
- 5816 • [DESIGN] Documentation of CRS management/control model and access-control mechanisms.
- 5817 • [OBSERVATION] / [LOG] Evidence of successful authorized management actions and failed/blocked
5818 attempts from within containers to reach management endpoints or perform control actions.

5819 6.4.1.2.2 Assessment Case AC-CRS-CP-ISO-002

5820 1. Assessment Reference

5821 Requirement: REQ-CRS-CP-ISO-002 (Elevated)

5822 2. Assessment Objective

5823 Verify that:

- 5824 1) The CRS supports configuration of dedicated logical separation between CRS management and control traffic
5825 and container data-plane traffic.
- 5826 2) The CRS supports the use of host-only communication endpoints, distinct network namespaces or dedicated
5827 management networks for its management/control interfaces, enabling isolation of CRS control traffic from
5828 container workloads.

5829 3. Assessment Preparation

5830 The assessment shall have access to:

- 5831 • Documentation describing how CRS management/control interfaces can be exposed over host-only endpoints,
5832 distinct network namespaces or dedicated management networks.
- 5833 • Documentation of recommended topologies for separating management/control traffic from container data-
5834 plane traffic.
- 5835 • A test configuration with:
 - 5836 - at least one management/control endpoint exposed via host-only or dedicated management connectivity;
 - 5837 - at least one container data-plane network/namespace;
 - 5838 - tools to inspect network namespaces, interfaces, routes and firewall rules.

5839 4. Assessment Activities

5840 The assessment shall at least:

- 5841 1) Configure the CRS so that management/control interfaces use one of the supported separation mechanisms
5842 (e.g. host-only socket, host network namespace only, dedicated management network or VLAN).
- 5843 2) Verify via configuration and inspection that:
 - 5844 - CRS management/control endpoints are bound only to the designated management/host-only
5845 mechanisms; and
 - 5846 - they are not exposed on container data-plane networks or in container network namespaces.

- 5847 3) From a container attached only to the data-plane network/namespace, attempt to connect to CRS
5848 management/control endpoints; verify that such attempts are blocked or cannot reach the endpoints.
- 5849 4) From an administrative context on the host or on the dedicated management network, verify that
5850 management/control interfaces are reachable via the designated paths, and that container data traffic does not
5851 traverse those paths.

5852 5. Assessment Verdict

- 5853 • Pass: CRS management/control traffic can be logically separated from container data-plane traffic using host-
5854 only endpoints, distinct network namespaces or dedicated management networks; and container data-plane
5855 networks cannot be used to reach CRS management/control interfaces in the assessed configuration.
- 5856 • Fail: no effective mechanism to separate CRS management/control and container data-plane traffic; or
5857 management/control interfaces remain reachable over container data-plane networks despite intended
5858 separation.

5859 6. Assessment Evidence

- 5860 • [CONFIG] Network and namespace configuration showing bindings of CRS management/control interfaces
5861 and separation from container data-plane networks.
- 5862 • [DESIGN] Documentation of supported mechanisms for control-plane vs data-plane separation.
- 5863 • [OBSERVATION] / [LOG] Connection tests from container and management contexts demonstrating blocked
5864 access from data-plane and permitted access via the management/host-only paths.

5865 6.4.1.2.3 Assessment Case AC-CRS-CP-ISO-003

5866 1. Assessment Reference

5867 Requirement: REQ-CRS-CP-ISO-003 (Advanced)

5868 2. Assessment Objective

5869 Verify that:

- 5870 1) The CRS supports cryptographically protected and mutually authenticated communication channels for all
5871 remote or network-exposed CRS management and control interfaces.
- 5872 2) When such channels are used, management/control traffic is transmitted only over communication paths that
5873 provide confidentiality, integrity protection and mutual authentication between the CRS and authorized
5874 management entities.
- 5875 3) For CRS management interfaces restricted to local host-only mechanisms, equivalent protection is provided by
5876 underlying platform access controls, consistent with the NOTE.

5877 3. Assessment Preparation

5878 The assessment shall have access to:

- 5879 • Documentation of remote/network-exposed CRS management/control interfaces and supported cryptographic
5880 protocols and mutual authentication options (client certificates, key-based methods).
- 5881 • Documentation for configuring certificates/keys and enabling mutually authenticated channels.
- 5882 • Documentation of host-only management mechanisms (e.g. Unix sockets, loopback-only endpoints) and the
5883 associated OS access-control model.
- 5884 • A test configuration with:
 - 5885 - at least one remote/network-exposed management/control interface;
 - 5886 - representative host-only management mechanisms, if provided;
 - 5887 - one management client with valid credentials and another with invalid/missing credentials.

5888 4. Assessment Activities

5889 The assessment shall at least:

5890 For remote/network-exposed management/control interfaces:

- 5891 1) Configure each such interface to use a cryptographically protected transport with mutual authentication as
5892 documented.
- 5893 2) Establish a management session using a client with valid credentials (certificates/keys) and verify via protocol
5894 inspection and logs that:
 - 5895 - the session is encrypted;
 - 5896 - the CRS authenticates the client; and
 - 5897 - the client authenticates the CRS (e.g. server certificate validation).
- 5898 3) Attempt connections from clients without valid credentials or using disallowed protocol/cipher configurations
5899 and verify that these are rejected and cannot perform management/control actions.
- 5900 4) Verify that no alternative cleartext or unauthenticated network paths exist for CRS management/control in the
5901 assessed configuration (e.g. no parallel HTTP port without TLS).

5902 For host-only management mechanisms (where present):

- 5903 1) Verify that such interfaces are bound only to host-local mechanisms (e.g. Unix domain sockets, loopback-only
5904 addresses) and not reachable from remote networks or container namespaces.
- 5905 2) Verify that OS-level access controls (file permissions, user/group roles, MAC policies) restrict access to these
5906 host-only interfaces to authorized local principals, providing confidentiality, integrity and authenticity
5907 comparable to the cryptographic protections required on remote interfaces.

5908 5. Assessment Verdict

- 5909 • Pass: all remote/network-exposed CRS management/control interfaces can use cryptographically protected,
5910 mutually authenticated channels; management/control traffic over these interfaces is confined to
5911 communication paths providing confidentiality, integrity and mutual authentication; and host-only
5912 management mechanisms rely on OS access controls that provide equivalent protection without exposing
5913 network-reachable cleartext admin paths.
- 5914 • Fail: some remote management/control interfaces cannot be protected cryptographically or cannot use mutual
5915 authentication; cleartext or unauthenticated admin channels remain reachable; or host-only mechanisms are
5916 exposed or insufficiently protected.

5917 6. Assessment Evidence

- 5918 • [CONFIG] Cryptographic and authentication configuration for remote management/control interfaces, and
5919 configuration for host-only management mechanisms (bindings, permissions).
- 5920 • [DESIGN] Documentation of supported management/control protocols, mutual authentication mechanisms
5921 and security model for host-only interfaces.
- 5922 • [LOG] / [OBSERVATION] Protocol traces and logs showing encrypted, mutually authenticated sessions,
5923 failed attempts without valid credentials, absence of cleartext admin paths, and enforcement of OS access
5924 controls on host-only interfaces.

5925 6.4.1.3 Assessment for Network Plane Isolation

5926 6.4.1.3.1 Assessment Case AC-CRS-NP-ISO-001

5927 1. Assessment Reference

5928 Requirement: REQ-CRS-NP-ISO-001 (Basic)

5929 **2. Assessment Objective**

5930 Verify that:

- 5931 1) The CRS maintains logical separation between container network traffic and host/CRS management network
5932 traffic using virtual networking mechanisms under its control.
- 5933 2) The CRS supports configurations in which containers cannot directly access host or CRS management
5934 interfaces/services by default.
- 5935 3) Separate network contexts can be defined for management traffic and container workload traffic.

5936 **3. Assessment Preparation**

5937 The assessment shall have access to:

- 5938 • Documentation of the CRS networking model (container networks, host network, management endpoints,
5939 virtual switches, CNI/plugin-ins, etc.).
- 5940 • Documentation describing how management endpoints (APIs, daemons, sockets) are bound to
5941 host/management networks versus container networks.
- 5942 • A test configuration with:
- 5943 - at least one container network with one or more containers;
- 5944 - CRS/host management interfaces bound to host/management network context;
- 5945 - tools to list network interfaces/namespaces and test connectivity from containers and host.

5946 **4. Assessment Activities**

5947 The assessment shall at least:

- 5948 1) Configure the CRS so that:
- 5949 - containers are attached to one or more container networks; and
- 5950 - CRS/host management interfaces are exposed only on host or dedicated management network contexts,
5951 as documented.
- 5952 2) From within a container, enumerate network interfaces, routing tables and reachable IP/ports, and attempt to
5953 connect to:
- 5954 - CRS management APIs;
- 5955 - CRS management daemons;
- 5956 - host administrative services.
- 5957 Verify that these interfaces/services are not reachable by default.
- 5958 3) From a host/management context, verify that management interfaces are reachable via the intended
5959 management/host network and that container data-plane traffic uses separate network contexts.

5960 **5. Assessment Verdict**

- 5961 • Pass: virtual networking under CRS control provides logical separation; containers cannot directly access host
5962 or CRS management interfaces/services by default; and separate network contexts for management and
5963 container workload traffic can be configured and observed.
- 5964 • Fail: container traffic can reach host/CRS management interfaces by default; or management and container
5965 traffic share the same network context without separation options.

5966 **6. Assessment Evidence**

- 5967 • [SCOPE] Description of the CRS deployment, container networks, and management/host networks used in the
5968 assessment.

- 5969 • [CONFIG] CRS networking configuration (CNI or equivalent), interface bindings for management endpoints,
5970 container network definitions.
- 5971 • [DESIGN] Documentation of CRS network plane separation model and default exposure of management vs
5972 container traffic.
- 5973 • [OBSERVATION] / [LOG] Connectivity tests and command outputs from containers and host showing lack
5974 of default access from containers to management/host interfaces.

5975 6.4.1.3.2 Assessment Case AC-CRS-NP-ISO-002

5976 1. Assessment Reference

5977 Requirement: REQ-CRS-NP-ISO-002 (Elevated)

5978 2. Assessment Objective

5979 Verify that:

- 5980 1) The CRS supports configuration of explicit traffic filtering and access-control rules for:
 - 5981 - flows between containers;
 - 5982 - flows between container networks;
 - 5983 - flows between container networks and host/management networks.
- 5984 2) These rules are enforced at the CRS virtual networking layer or other enforcement points under CRS control.
- 5985 3) Only network flows explicitly permitted by policy are allowed to cross these boundaries.

5986 3. Assessment Preparation

5987 The assessment shall have access to:

- 5988 • Documentation of CRS mechanisms for defining and enforcing network policies (e.g. network policies,
5989 security groups, virtual firewall rules, CNI plug-in ACLs).
- 5990 • Documentation with examples of permitted and denied flows.
- 5991 • A test configuration with:
 - 5992 - at least two container networks and multiple containers;
 - 5993 - host/management network reachable from the CRS;
 - 5994 - the ability to define and modify network policies and to generate traffic between these endpoints.

5995 4. Assessment Activities

5996 The assessment shall at least:

- 5997 1) Define and apply network policies or equivalent rules at the CRS virtual networking layer (or other CRS-
5998 controlled enforcement point) to:
 - 5999 - allow specific flows (e.g. container A in network X to container B in network Y on a given port;
6000 management endpoint to specific container agents); and
 - 6001 - deny all other cross-container, cross-network and container-to-host/management flows by default.
- 6002 2) Generate test traffic:
 - 6003 - flows that match permitted rules;
 - 6004 - flows that should be blocked (e.g. arbitrary container-to-container connections across networks,
6005 container-to-management attempts outside permitted scope).
- 6006 3) Verify that:
 - 6007 - permitted flows succeed as configured;

- 6008 - non-permitted flows are blocked at the CRS-controlled enforcement points; and
- 6009 - adjusting rules (add/remove) changes allowed/denied flows as expected.

6010 5. Assessment Verdict

- 6011 • Pass: the CRS can define and enforce explicit traffic filtering/access-control rules between containers,
6012 container networks and host/management networks, and only flows explicitly permitted by policy can cross
6013 these boundaries.
- 6014 • Fail: policies cannot be defined at the CRS-controlled layer; enforcement is ineffective; or cross-boundary
6015 traffic bypasses the configured rules.

6016 6. Assessment Evidence

- 6017 • [CONFIG] Network policy and access-control rule definitions and their association with container networks
6018 and host/management networks.
- 6019 • [DESIGN] Documentation of CRS virtual networking enforcement mechanisms for inter-container and cross-
6020 plane traffic.
- 6021 • [OBSERVATION] / [LOG] Packet captures, policy logs or connectivity test results showing success of
6022 permitted flows and blocking of non-permitted flows.

6023 6.4.1.3.3 Assessment Case AC-CRS-NP-ISO-003

6024 1. Assessment Reference

6025 Requirement: REQ-CRS-NP-ISO-003 (Advanced)

6026 2. Assessment Objective

6027 Verify that:

- 6028 1) The CRS supports cryptographically protected and mutually authenticated communication channels for
6029 management and control traffic that traverses shared or untrusted networks.
- 6030 2) When such channels are used, CRS management/control traffic is transmitted only over paths that provide
6031 confidentiality, integrity protection and mutual authentication between the CRS and authorized management
6032 entities.
- 6033 3) For management interfaces restricted to local host-only mechanisms, equivalent protection is provided by OS
6034 platform access controls, consistent with the NOTE.

6035 3. Assessment Preparation

6036 The assessment shall have access to:

- 6037 • Documentation of remote/network-exposed CRS management/control interfaces and supported cryptographic
6038 protocols (e.g. TLS, mTLS, SSH, VPN/IPsec) and mutual authentication methods (client certificates, keys,
6039 tokens).
- 6040 • Documentation explaining how to configure these protected channels and how they are bound to
6041 management/control traffic.
- 6042 • Documentation of host-only management interfaces (e.g. Unix domain sockets, loopback-only endpoints) and
6043 their OS-level access-control model.
- 6044 • A test configuration with:
 - 6045 - at least one CRS management/control interface exposed over a shared or untrusted network;
 - 6046 - at least one representative host-only management mechanism;
 - 6047 - management clients with valid and invalid credentials.

6048 **4. Assessment Activities**

6049 For management/control interfaces over shared or untrusted networks, the assessment shall at least:

- 6050 1) Configure each such interface to use cryptographically protected transport with mutual authentication as
6051 documented (e.g. mTLS for control APIs, SSH with key-based admin access, VPN/IPsec tunnels for remote
6052 management).
- 6053 2) From a client with valid credentials, establish a management/control session and verify via protocol inspection
6054 and logs that:
- 6055 - the session is encrypted;
 - 6056 - the CRS authenticates the client;
 - 6057 - the client authenticates the CRS.
- 6058 3) Attempt connections from clients without valid credentials or using disallowed protocol or cipher
6059 configurations and verify that these are rejected and cannot perform management or control actions.
- 6060 4) Verify that no cleartext or unauthenticated alternative management/control paths are reachable over the
6061 shared/untrusted network in the assessed configuration.

6062 For host-only management interfaces, the assessment shall at least:

- 6063 1) Verify that such interfaces are bound only to local mechanisms (e.g. Unix domain sockets, loopback-only
6064 addresses) and are not reachable from container namespaces or external networks.
- 6065 2) Verify that OS-level access controls (file permissions, user/group roles, MAC policies) restrict access to these
6066 interfaces to authorized local principals, providing confidentiality, integrity and authenticity comparable to
6067 what cryptographic protection would provide on a network path.

6068 **5. Assessment Verdict**

- 6069 • Pass: the CRS supports cryptographically protected, mutually authenticated channels for management/control
6070 traffic over shared/untrusted networks; such traffic is carried only over paths providing confidentiality,
6071 integrity and mutual authentication; and host-only management mechanisms rely on appropriate OS access
6072 controls without exposing unprotected network-reachable management paths.
- 6073 • Fail: remote CRS management/control interfaces cannot be protected cryptographically or cannot use mutual
6074 authentication; cleartext or unauthenticated management/control paths exist over shared/untrusted networks; or
6075 host-only mechanisms are exposed beyond the host or lack sufficient OS-level protection.

6076 **6. Assessment Evidence**

- 6077 • [CONFIG] Cryptographic and authentication configuration for remote CRS management/control interfaces;
6078 configuration for host-only interfaces (bindings, permissions).
- 6079 • [DESIGN] Documentation of CRS control-plane security model, supported protocols, mutual authentication
6080 mechanisms and host-only protection model.
- 6081 • [LOG] / [OBSERVATION] Protocol traces and logs demonstrating encrypted, mutually authenticated
6082 management sessions, rejected invalid attempts, absence of cleartext management traffic on shared networks,
6083 and enforcement of OS access controls for host-only mechanisms.

6084 **6.4.1.4 Assessment for Boot Chain Integrity Verification**

6085 **6.4.1.4.1 Assessment Case AC-CRS-B-INT-001**

6086 **1. Assessment Reference**

6087 Requirement: REQ-CRS-B-INT-001 (Basic)

6088 2. Assessment Objective

6089 Verify that:

- 6090 1) The CRS implements an integrity verification mechanism for its core runtime engine and other CRS
6091 executables required to start container execution services, based on trusted reference values.
- 6092 2) The CRS can invoke this integrity verification before enabling container scheduling, container lifecycle
6093 operations, or other container execution interfaces.
- 6094 3) If integrity verification of any covered CRS executable fails, the CRS prevents startup of container execution
6095 services and generates a security-relevant event.

6096 3. Assessment Preparation

6097 The assessment shall have access to:

- 6098 • Documentation describing the integrity verification mechanism for CRS executables, including which
6099 executables are covered and how reference values are established and stored.
- 6100 • Documentation of the CRS startup sequence, showing when integrity verification is invoked relative to
6101 enabling container execution services.
- 6102 • A test configuration that allows:
 - 6103 - a normal startup with unmodified CRS executables; and
 - 6104 - a controlled integrity-verification failure for at least one covered executable (e.g. tampered binary,
6105 invalid signature, test mode), consistent with vendor guidance.

6106 4. Assessment Activities

6107 The assessment shall at least:

- 6108 1) Review documentation and configuration to identify:
 - 6109 - the set of CRS executables covered by integrity checking;
 - 6110 - the verification mechanism and trusted reference values;
 - 6111 - the point in startup where the check is invoked.
- 6112 2) Perform a normal CRS startup and verify that:
 - 6113 - integrity verification is invoked as documented;
 - 6114 - container execution services (scheduling, lifecycle operations, APIs) become available.
- 6115 3) Induce a failure in the integrity verification of one covered executable and restart the CRS. Verify that:
 - 6116 - the verification failure is detected;
 - 6117 - container execution services do not start;
 - 6118 - a security-relevant event is generated and recorded.

6119 5. Assessment Verdict

- 6120 • Pass: integrity verification is implemented for the core runtime engine and required CRS executables; it can be
6121 invoked before container execution services are enabled; and a verification failure prevents startup and
6122 produces a security-relevant event.
- 6123 • Fail: there is no effective integrity verification; integrity checks are not invoked before container execution
6124 services; or verification failures do not block startup and are not logged as security-relevant events.

6125 6. Assessment Evidence

- 6126 • [DESIGN] Description of the CRS executable integrity verification mechanism and startup sequence.

- 6127 • [CONFIG] Configuration showing enabled integrity checks and list of covered executables.
- 6128 • [LOG] / [OBSERVATION] Startup logs and traces for normal and failed verification scenarios, including the
- 6129 recorded security-relevant event.

6130 6.4.1.4.2 Assessment Case AC-CRS-B-INT-002

6131 1. Assessment Reference

6132 Requirement: REQ-CRS-B-INT-002 (Elevated)

6133 2. Assessment Objective

6134 Verify that:

- 6135 1) The CRS implements mechanisms to participate in a verifiable chain of trust for CRS startup components
- 6136 delivered as part of the CRS product and executed before container execution services are enabled.
- 6137 2) Each CRS startup stage delivered with the product and executed prior to exposing container execution
- 6138 interfaces validates the integrity and authenticity of the subsequent CRS stage before transferring control.
- 6139 3) If validation fails for any such startup stage, the CRS prevents startup of container execution services and
- 6140 generates a security-relevant event.

6141 3. Assessment Preparation

6142 The assessment shall have access to:

- 6143 • Documentation of the CRS startup chain for components delivered with the product (e.g. CRS launchers,
- 6144 shims, daemons that initialize container execution services).
- 6145 • Documentation describing how each CRS startup stage validates the integrity and authenticity of the next CRS
- 6146 stage (e.g. signature checks, hash verification) and how this can integrate with a platform or CRS-defined
- 6147 chain of trust.
- 6148 • A test configuration where:
 - 6149 - normal startup with valid CRS startup components is possible;
 - 6150 - a controlled validation failure can be induced for at least one CRS startup stage (e.g. modified or
 - 6151 unsigned binary, untrusted key), consistent with vendor guidance.

6152 4. Assessment Activities

6153 The assessment shall at least:

- 6154 1) Review documentation to identify all CRS startup stages delivered with the product that execute before
- 6155 container execution services are enabled, and the validation performed at each stage.
- 6156 2) Perform a normal CRS startup and verify, using logs or diagnostic outputs, that:
 - 6157 - each CRS startup stage validates the integrity/authenticity of the next stage before passing control;
 - 6158 - container execution services are enabled only after successful validation of all such stages.
- 6159 3) Introduce a controlled integrity/authenticity failure in one CRS startup stage and restart the CRS. Verify that:
 - 6160 - the validation failure at that stage is detected;
 - 6161 - container execution services are not started;
 - 6162 - a security-relevant event is generated and recorded.

6163 5. Assessment Verdict

- 6164 • Pass: all CRS startup stages delivered with the product and executed before container execution services
- 6165 validate the integrity and authenticity of subsequent CRS stages; validation failures are detected; and container
- 6166 execution services are prevented from starting, with a security-relevant event generated.

- 6167 • Fail: some CRS startup stages do not validate subsequent stages; failures are not detected; or validation
- 6168 failures do not block container execution services or produce a security-relevant event.

6169 6. Assessment Evidence

- 6170 • [DESIGN] Documentation of the CRS startup chain, per-stage validation mechanisms and chain-of-trust
- 6171 integration.
- 6172 • [CONFIG] Startup configuration and key/reference-value configuration used for stage validation.
- 6173 • [LOG] / [OBSERVATION] Startup logs or traces from normal and failure scenarios showing successful
- 6174 chained validation, detection of failed validation and prevention of container service startup with an associated
- 6175 security event.

6176 6.4.1.4.3 Assessment Case AC-CRS-B-INT-003

6177 1. Assessment Reference

6178 Requirement: REQ-CRS-B-INT-003 (Advanced)

6179 2. Assessment Objective

6180 Verify that:

- 6181 1) The CRS supports consuming integrity and authenticity information exposed by platform verified boot or
- 6182 integrity measurement mechanisms for operating system components and CRS components that have to be
- 6183 loaded and initialized before container execution services are enabled.
- 6184 2) The CRS supports policies that prevent initialization of container execution services when the reported
- 6185 verification status for any required component in this chain indicates failure or cannot be obtained.

6186 3. Assessment Preparation

6187 The assessment shall have access to:

- 6188 • Documentation of the platform's verified boot or integrity measurement mechanism and the interface through
- 6189 which verification status is exposed (e.g. APIs, status files, attestation logs).
- 6190 • Documentation describing which OS and CRS components are considered "required" before container
- 6191 execution services are enabled and how their verification status is mapped into CRS policy decisions.
- 6192 • Documentation of CRS policies and configuration options that use this verification status to control container
- 6193 startup.
- 6194 • A test configuration where:
 - 6195 - platform verified boot/integrity measurement is enabled and exposes status for required components;
 - 6196 - CRS is configured to consume this status and enforce startup policies;
 - 6197 - verification failures or missing status can be simulated in a controlled way, consistent with
 - 6198 platform/vendor guidance.

6199 4. Assessment Activities

6200 The assessment shall at least:

- 6201 1) Review documentation to identify the required OS and CRS components whose integrity/measurement status
- 6202 is consumed by the CRS and understand how this status is obtained.
- 6203 2) Configure the CRS policy so that container execution services are allowed only when verification status for all
- 6204 required components is reported as successful.
- 6205 3) In a normal scenario, boot the platform and start the CRS, ensuring that platform verified boot/integrity
- 6206 mechanisms report successful status for all required components. Verify that:
 - 6207 - the CRS successfully consumes the verification status;
 - 6208 - container execution services are initialized.

- 6209 4) Induce a failure or unavailable status for one required component (e.g. configuration that causes a component
6210 to fail verification, or removal of its measurement from the platform status interface) and restart. Verify that:
- 6211 - the CRS detects that verification status for that component indicates failure or cannot be obtained;
- 6212 - container execution services are not initialized;
- 6213 - a security-relevant event is generated and recorded if required by CRS logging behaviour.

6214 5. Assessment Verdict

- 6215 • Pass: the CRS can consume platform-exposed verification status for required OS and CRS components; it
6216 supports policies that prevent initialization of container execution services when status indicates failure or is
6217 unavailable; and those policies behave as documented in normal and failure scenarios.
- 6218 • Fail: the CRS cannot effectively consume platform verification status; startup policies do not prevent container
6219 execution services when status is failed or missing; or behaviour does not match documentation.

6220 6. Assessment Evidence

- 6221 • [DESIGN] Documentation of CRS integration with platform verified boot/integrity measurement mechanisms
6222 and the mapping of verification status to startup policy.
- 6223 • [PLT-CAP] Evidence of platform capabilities and interfaces used to expose verification status.
- 6224 • [CONFIG] CRS policy configuration controlling container startup based on verification status.
- 6225 • [LOG] / [OBSERVATION] Logs and status outputs from normal and failure tests showing consumed
6226 verification status and corresponding startup decisions for container execution services.

6227 6.4.1.4.4 Assessment Case AC-CRS-B-INT-004

6228 Applicability:

6229 This assessment applies only when the manufacturer claims that the CRS supports integration with hardware-based
6230 roots of trust for CRS-critical components in accordance with REQ-CRS-B-INT-003.

6231 1. Assessment Reference

6232 Requirement: REQ-CRS-B-INT-004 (Advanced, conditional "may")

6233 2. Assessment Objective

6234 Verify that, when claimed:

- 6235 1) The CRS can integrate with hardware-based roots of trust (e.g. secure elements, dedicated trust anchors,
6236 equivalent components) to strengthen boot-chain integrity and authenticity verification for CRS-critical
6237 components, consistent with the policies required by REQ-CRS-B-INT-003.
- 6238 2) These hardware-based roots of trust provide protection against Unauthorized modification of CRS boot and
6239 startup components.

6240 3. Assessment Preparation

6241 The assessment shall have access to:

- 6242 • Documentation describing the hardware-based root(s) of trust available on the platform (e.g. TPM, secure
6243 element, trust anchor) and how they are used to verify CRS-critical components.
- 6244 • Documentation explaining how CRS integrates with these hardware-based roots of trust (e.g. using
6245 measurements, signed manifests, secure key storage) to obtain integrity/authenticity status for CRS-critical
6246 components.
- 6247 • A test configuration on a platform that provides the claimed hardware-based roots of trust and where CRS
6248 boot/startup verification using those roots can be exercised.

6249 4. Assessment Activities

6250 The assessment shall at least:

- 6251 1) Review documentation to identify the CRS-critical components for which integrity/authenticity is assured via
6252 hardware-based roots of trust and how CRS consumes that assurance as part of the REQ-CRS-B-INT-003
6253 policy.
- 6254 2) In a normal configuration, boot the platform and start the CRS; verify, using logs or status interfaces, that:
 - 6255 - the hardware-based root(s) of trust are used to verify the relevant CRS boot/startup components;
 - 6256 - verification is reported as successful;
 - 6257 - container execution services are enabled accordingly.
- 6258 3) Where feasible and consistent with vendor guidance, introduce an Unauthorized modification to a CRS-critical
6259 boot/startup component protected by the hardware-based root of trust (e.g. unsanctioned binary change that
6260 breaks signature/measurement) and restart. Verify that:
 - 6261 - the hardware-based verification indicates failure or untrusted state;
 - 6262 - the CRS, applying its boot-chain integrity policy, prevents initialization of container execution services
6263 for this condition;
 - 6264 - a security-relevant event is recorded, if required by CRS logging behaviour.

6265 5. Assessment Verdict

- 6266 • Pass: when support is claimed, the CRS can use hardware-based roots of trust as a source of integrity and
6267 authenticity assurance for CRS-critical boot/startup components; Unauthorized modifications are detected
6268 through those roots; and the CRS enforces the same startup-prevention policy as required by REQ-CRS-B-
6269 INT-003.
- 6270 • Fail: claimed hardware-based integration is not functional; tampering with protected components is not
6271 detected; or the CRS does not enforce appropriate startup-prevention behaviour in response to hardware-based
6272 verification failures.

6273 6. Assessment Evidence

- 6274 • [DESIGN] Documentation of CRS integration with hardware-based roots of trust and the list of CRS-critical
6275 components covered.
- 6276 • [PLT-CAP] Evidence of the hardware-based roots of trust present and configured on the platform.
- 6277 • [CONFIG] CRS and platform configuration enabling hardware-based verification of CRS-critical components.
- 6278 • [LOG] / [OBSERVATION] Logs and status outputs from normal and tampered scenarios showing hardware-
6279 based verification results and corresponding CRS startup decisions.

6280 6.4.1.5 Assessment for Container Image Integrity Verification

6281 6.4.1.5.1 Assessment Case AC-CRS-IMG-INT-001

6282 1. Assessment Reference

6283 Requirement: REQ-CRS-IMG-INT-001 (Basic)

6284 2. Assessment Objective

6285 Verify that:

- 6286 1) The CRS validates that the image manifest and all referenced layers match their declared cryptographic digests
6287 before instantiation.
- 6288 2) This integrity verification is invoked automatically before creating or starting any container from an image.

- 6289 3) If any mismatch, corruption, or removal of a referenced layer or manifest entry is detected, the CRS rejects the
6290 image, prevents container instantiation from that image, and records a security-relevant event.

6291 3. Assessment Preparation

6292 The assessment shall have access to:

- 6293 • Documentation describing the container image integrity verification mechanism, including how manifest and
6294 layer digests are obtained and validated.
- 6295 • Documentation of the container creation/startup sequence showing where integrity verification is invoked.
- 6296 • A test environment with:
 - 6297 - at least one image available through the normal image distribution mechanism (e.g. local store or
6298 registry);
 - 6299 - the ability to create a valid image with consistent manifest and layer digests;
 - 6300 - the ability to create an inconsistent or corrupted image (e.g. modified layer content without updating the
6301 digest, missing layer, or modified manifest entry) in a controlled manner.

6302 4. Assessment Activities

6303 The assessment shall at least:

- 6304 1) Review documentation and configuration to identify:
 - 6305 - which images are subject to integrity verification;
 - 6306 - how the CRS validates manifest and layer digests;
 - 6307 - that verification is invoked automatically prior to container creation/start.
- 6308 2) Using a valid image whose manifest and layers match their declared digests:
 - 6309 - request creation or start of a container from that image;
 - 6310 - verify, using logs or diagnostics if available, that integrity verification is performed;
 - 6311 - confirm that the container is successfully instantiated when verification succeeds.
- 6312 3) Using an image modified so that at least one layer or manifest entry does not match its declared digest, or a
6313 referenced layer is unavailable:
 - 6314 - request creation or start of a container from that image;
 - 6315 - verify that the CRS detects the mismatch or missing component;
 - 6316 - confirm that the image is rejected, container instantiation is prevented, and a security-relevant event is
6317 recorded.

6318 5. Assessment Verdict

- 6319 • Pass: the CRS automatically validates manifest and layer digests before container creation/start; images with
6320 consistent digests are allowed; images with mismatches, corruption, or missing layers are rejected, containers
6321 are not instantiated, and a security-relevant event is recorded.
- 6322 • Fail: integrity verification is not applied, not automatic, or can be bypassed; or images with digest mismatches
6323 or missing layers can be used to instantiate containers without rejection and without a recorded security event.

6324 6. Assessment Evidence

- 6325 • [DESIGN] Documentation of the container image integrity verification mechanism and its position in the
6326 container lifecycle.
- 6327 • [CONFIG] CRS configuration enabling image integrity verification and describing image sources.

- 6328 • [LOG] Records of successful verification and container start for valid images, and records of detected
6329 mismatches, rejected images, and corresponding security-relevant events.
- 6330 • [OBSERVATION] Test execution logs or command outputs demonstrating behaviour for valid and corrupted
6331 images.

6332 6.4.1.5.2 Assessment Case AC-CRS-IMG-INT-002

6333 1. Assessment Reference

6334 Requirement: REQ-CRS-IMG-INT-002 (Elevated)

6335 2. Assessment Objective

6336 Verify that:

- 6337 1) The CRS verifies both the integrity and the authenticity of container images before execution.
- 6338 2) Authenticity verification establishes that each image originates from a trusted publisher or approval authority
6339 and has not been altered after approval.
- 6340 3) The CRS prevents execution or instantiation of any container image whose integrity or authenticity cannot be
6341 successfully verified according to the configured trust policy.

6342 3. Assessment Preparation

6343 The assessment shall have access to:

- 6344 • Documentation describing the image trust model and verification mechanisms, including:
 - 6345 - how trusted publishers or approval authorities are represented and configured;
 - 6346 - how integrity and authenticity checks are combined (e.g. signatures, certificates, provenance metadata,
6347 equivalent cryptographic mechanisms).
- 6348 • A test environment with:
 - 6349 - at least one container image produced and "approved" according to the documented process (e.g.
6350 properly signed or otherwise bound to a trusted publisher);
 - 6351 - one or more images for which integrity or authenticity will fail, such as:
 - 6352 ▪ an image modified after approval;
 - 6353 ▪ an image missing authenticity information;
 - 6354 ▪ an image signed or bound to an untrusted publisher, prepared in a controlled manner;
 - 6355 - a configured trust policy specifying which publishers/authorities are trusted.

6356 4. Assessment Activities

6357 The assessment shall at least:

- 6358 1) Review documentation and configuration to identify:
 - 6359 - the trust anchors or metadata used to represent trusted publishers/authorities;
 - 6360 - how the CRS enforces the trust policy when verifying images.
- 6361 2) For an image that is correctly produced and approved under the trust policy:
 - 6362 - request creation or start of a container from that image;
 - 6363 - verify that both integrity and authenticity verification are applied and succeed;
 - 6364 - confirm that the container is instantiated.
- 6365 3) For each test image that should fail integrity or authenticity (modified content, missing/invalid signature,
6366 untrusted publisher, or equivalent):

- 6367 - request creation or start of a container from that image;
- 6368 - verify that the CRS detects the failure to verify integrity and/or authenticity according to the configured
- 6369 trust policy;
- 6370 - confirm that the CRS prevents execution or instantiation of the container from that image.

6371 5. Assessment Verdict

- 6372 • Pass: integrity and authenticity verification is applied to container images before execution; only images from
- 6373 trusted publishers/authorities that have not been modified after approval can be instantiated; and images whose
- 6374 integrity or authenticity cannot be successfully verified under the trust policy are blocked.
- 6375 • Fail: authenticity is not enforced or can be bypassed; untrusted or modified images can be instantiated; or
- 6376 failure to verify integrity/authenticity does not prevent container instantiation.

6377 6. Assessment Evidence

- 6378 • [DESIGN] Documentation of the container image trust model, including integrity and authenticity verification
- 6379 and the trust policy.
- 6380 • [CONFIG] Trust configuration (trusted publishers/authorities, key material, policy settings).
- 6381 • [LOG] Records showing successful verification and container instantiation for approved images, and failed
- 6382 verification with blocked instantiation for images that do not satisfy the trust policy.
- 6383 • [OBSERVATION] Test execution outputs demonstrating enforcement of the trust policy.

6384 6.4.1.5.3 Assessment Case AC-CRS-IMG-INT-003

6385 1. Assessment Reference

6386 Requirement: REQ-CRS-IMG-INT-003 (Advanced)

6387 2. Assessment Objective

6388 Verify that:

- 6389 1) The CRS performs integrity and authenticity verification of container images using trust anchors stored as
- 6390 protected trust material.
- 6391 2) Protected trust material is not modifiable during normal operational state and cannot be modified by container
- 6392 workloads, unprivileged users, or routine runtime operations.
- 6393 3) Protected trust material is updated or replaced only through a controlled, authenticated administrative process.
- 6394 4) If verification of the integrity or authenticity of a container image using this protected trust material fails, or
- 6395 the verification status cannot be established, the CRS prevents execution or instantiation of that container
- 6396 image and records a security-relevant event.

6397 3. Assessment Preparation

6398 The assessment shall have access to:

- 6399 • Documentation describing:
 - 6400 - what constitutes protected trust material (keys, certificates, trust anchors);
 - 6401 - how and where this material is stored;
 - 6402 - what protections prevent modification during normal operation;
 - 6403 - the process and required privileges for controlled update or replacement of trust material.
- 6404 • A test environment with:
 - 6405 - a CRS deployment configured to use protected trust material for image verification;
 - 6406 - at least one container image that verifies successfully against this material;

- 6407 - the ability to simulate conditions where an image fails verification against the protected trust material or
6408 where verification status is indeterminate, in a controlled manner.

6409 **4. Assessment Activities**

6410 The assessment shall at least:

- 6411 1) Review documentation and configuration to confirm:
- 6412 - which trust anchors are used for image verification;
 - 6413 - where they are stored and how access is restricted;
 - 6414 - that updates require a controlled, authenticated administrative process distinct from routine runtime
6415 operations.
- 6416 2) In normal operational state, attempt to modify or replace trust material using:
- 6417 - container workloads;
 - 6418 - unprivileged user contexts;
 - 6419 - standard administrative or runtime interfaces that are not part of the documented controlled update
6420 process.
- 6421 3) Verify that such attempts fail or are blocked.
- 6422 4) Using a container image correctly approved under the current protected trust material, request container
6423 creation/start and verify that:
- 6424 - integrity and authenticity verification succeed;
 - 6425 - the container is instantiated.
- 6426 5) Create a test condition such that verification against the protected trust material fails or becomes indeterminate
6427 for a container image (e.g. image modified after approval, image associated with no matching trust anchor)
6428 and then request container creation/start. Verify that:
- 6429 - the CRS detects the verification failure or inability to establish verification status;
 - 6430 - the container image is not allowed to execute or be instantiated;
 - 6431 - a security-relevant event is recorded.

6432 **5. Assessment Verdict**

- 6433 • Pass: the CRS uses protected trust material for image integrity/authenticity verification; this material cannot be
6434 modified by workloads, unprivileged users or routine operations; only controlled, authenticated processes can
6435 update it; and images that fail or cannot be verified against this material are blocked and generate a security-
6436 relevant event.
- 6437 • Fail: trust anchors are stored in locations modifiable during normal operation; workloads or unprivileged users
6438 can alter them; or images that fail verification against the protected trust material can still be instantiated
6439 without a recorded security event.

6440 **6. Assessment Evidence**

- 6441 • [DESIGN] Documentation of protected trust material, its storage, protection model and update procedures.
- 6442 • [CONFIG] CRS configuration showing trust stores or equivalent structures used for image verification and
6443 their access-control settings.
- 6444 • [LOG] Records showing successful verification and container instantiation for valid images, and failed
6445 verification with corresponding security-relevant events for images that cannot be verified.
- 6446 • [OBSERVATION] Test outputs demonstrating blocked modification attempts on trust material and blocked
6447 instantiation of images that fail verification.

6448 6.4.1.5.4 Assessment Case AC-CRS-IMG-INT-004

6449 **Applicability:**

6450 This assessment case applies only when the manufacturer claims that the CRS supports anchoring container image
6451 verification in hardware-assisted or platform-provided trust services.

6452 **1. Assessment Reference**

6453 Requirement: REQ-CRS-IMG-INT-004 (Advanced, conditional "may")

6454 **2. Assessment Objective**

6455 Verify that, when claimed:

- 6456 1) The CRS can integrate with hardware-assisted or platform-provided trust services (e.g. protected key storage,
6457 secure elements, platform trust services) to store or validate trust anchors used for container image
6458 verification.
- 6459 2) These mechanisms provide resistance against Unauthorized modification of verification keys and trust anchors
6460 used for container image integrity and authenticity verification.

6461 **3. Assessment Preparation**

6462 The assessment shall have access to:

- 6463 • Documentation describing the hardware-assisted or platform trust services used (e.g. secure key store, secure
6464 element, platform trust service) and how they are exposed to the CRS.
- 6465 • Documentation of CRS integration with those services for storage or validation of trust anchors used for
6466 container image verification.
- 6467 • A test environment on a platform that provides the claimed trust services and where CRS integration for image
6468 verification can be exercised.

6469 **4. Assessment Activities**

6470 The assessment shall at least:

- 6471 1) Review documentation to identify:
 - 6472 - which trust anchors or verification keys for container images are stored in or validated by the hardware-
6473 assisted/platform trust services;
 - 6474 - how the CRS uses those services as part of the verification process described in REQ-CRS-IMG-INT-
6475 002/003.
 - 6476 2) In a normal configuration, configure the CRS to use trust anchors managed by the hardware-assisted/platform
6477 services and:
 - 6478 - verify, through logs or diagnostics, that image verification operations consult or rely on these services;
 - 6479 - request container creation/start from an image correctly approved under these trust anchors and confirm
6480 that verification succeeds and the container is instantiated.
 - 6481 3) Where feasible and consistent with vendor guidance, attempt to:
 - 6482 - modify or replace verification keys/trust anchors via interfaces that should not have permission to change
6483 hardware-protected material (e.g. container workloads, unprivileged users, routine admin interfaces);
 - 6484 - use images requiring unregistered or untrusted keys.
- 6485 Verify that:
- 6486 - trust material under control of the hardware-assisted/platform services cannot be altered through these
6487 paths;
 - 6488 - images that do not verify against the hardware-protected trust anchors are rejected and containers are not
6489 instantiated.

6490 5. Assessment Verdict

6491 • Pass: when support is claimed, the CRS integrates with hardware-assisted or platform trust services to store or
 6492 validate trust anchors; verification keys/trust anchors in these services cannot be modified via Unauthorized
 6493 paths; and images that cannot be verified against hardware-protected anchors are rejected and cannot be used
 6494 to instantiate containers.

6495 • Fail: claimed integration with hardware/platform trust services is not functional; verification keys or trust
 6496 anchors stored in those services can be modified without proper controls; or images that should fail
 6497 verification against hardware-protected anchors can still be instantiated.

6498 6. Assessment Evidence

6499 • [DESIGN] Documentation of CRS integration with hardware-assisted or platform trust services for container
 6500 image verification.

6501 • [PLT-CAP] Evidence of the presence and configuration of the relevant platform trust services or secure
 6502 hardware used.

6503 • [CONFIG] CRS configuration showing use of those services for trust anchor storage/validation.

6504 • [LOG] / [OBSERVATION] Logs and test outputs demonstrating normal verification using hardware-protected
 6505 trust anchors and rejection of images that cannot be verified against them.

6506 6.4.1.6 Assessment for Runtime Integrity Protection

6507 6.4.1.6.1 Assessment Case AC-CRS-RP-INT-002

6508 1. Assessment Reference

6509 Requirement: REQ-CRS-RP-INT-002 (Elevated)

6510 2. Assessment Objective

6511 Verify that:

6512 1) The CRS enforces integrity protection for its configuration and control plane so that only authenticated and
 6513 authorized modifications can be applied.

6514 2) The CRS generates audit events for any security-relevant changes to configuration or internal control logic.

6515 3. Assessment Preparation

6516 The assessment shall have access to:

- 6517 • Documentation describing the CRS configuration and control plane, including:
 - 6518 - which configuration items and control-plane elements are considered security-relevant;
 - 6519 - which roles or identities are permitted to modify them;
 - 6520 - which local or remote interfaces are used for such modifications.
- 6521 • Documentation of the CRS logging/auditing mechanisms for configuration and control-plane changes.
- 6522 • A CRS deployment with:
 - 6523 - at least one authenticated administrative account/role with privileges to change security-relevant
 6524 configuration or control-plane logic;
 - 6525 - at least one authenticated non-privileged or Unauthorized account/role;
 - 6526 - access to the configuration and control-plane interfaces under test;
 - 6527 - logging/audit facilities enabled and accessible for inspection.

6528 **4. Assessment Activities**

6529 The assessment shall at least:

- 6530 1) Review documentation and CRS configuration to identify:
- 6531 - the set of security-relevant configuration parameters and control-plane elements;
- 6532 - the authentication and authorization mechanisms governing their modification;
- 6533 - how audit events are produced and stored when such elements are changed.
- 6534 2) Using an authenticated and authorized administrative account:
- 6535 - perform representative security-relevant configuration or control-plane changes (for example
- 6536 enabling/disabling security features, changing access-control policies, or modifying isolation-related
- 6537 settings);
- 6538 - verify that the CRS accepts these changes;
- 6539 - verify that corresponding audit events are generated, including at least the acting identity, the type of
- 6540 change, and the time of the event.
- 6541 3) Using an unauthenticated context, a non-privileged account, or an interface that should not allow such
- 6542 changes:
- 6543 - attempt to perform the same or equivalent security-relevant modifications;
- 6544 - verify that these attempts are rejected and that configuration/control-plane state is not altered;
- 6545 - verify, where specified by product design, that failed or denied modification attempts are logged.

6546 **5. Assessment Verdict**

- 6547 • Pass: only authenticated and authorized entities can perform security-relevant configuration or control-plane
- 6548 changes; attempts by unauthenticated or Unauthorized entities are rejected without altering state; and audit
- 6549 events are generated for security-relevant changes to configuration or internal control logic.
- 6550 • Fail: security-relevant changes can be made without proper authentication or authorization; such changes do
- 6551 not generate audit events; or Unauthorized attempts are not prevented.

6552 **6. Assessment Evidence**

- 6553 • [DESIGN] Documentation of the CRS configuration and control-plane model, including the definition of
- 6554 security-relevant changes and the access-control model.
- 6555 • [CONFIG] CRS configuration showing role/permission assignments for configuration and control-plane
- 6556 interfaces and logging/audit settings.
- 6557 • [LOG] Audit or event logs capturing successful authorized changes and rejected Unauthorized attempts, with
- 6558 timestamps and identifiers.
- 6559 • [OBSERVATION] Test execution records (for example CLI/API outputs, management UI screenshots)
- 6560 demonstrating acceptance of authorized changes and rejection of Unauthorized ones.

6561 **6.4.1.6.2 Assessment Case AC-CRS-RP-INT-003**

6562 **1. Assessment Reference**

6563 Requirement: REQ-CRS-RP-INT-003 (Advanced)

6564 **2. Assessment Objective**

6565 Verify that:

- 6566 1) The CRS maintains the integrity of its security-critical runtime components and detects Unauthorized
- 6567 modification of configuration state or control logic used to enforce security policies or isolation.

- 6568 2) The CRS maintains a trusted baseline of the integrity state of these security-critical runtime components and
6569 performs periodic or continuous integrity validation against this baseline.
- 6570 3) Upon detecting a deviation from the trusted baseline, the CRS automatically triggers protective actions (such
6571 as isolating the affected component, disabling affected functions, or restoring the component from a trusted
6572 state) and generates a security alert that cannot be suppressed by the affected component.

6573 3. Assessment Preparation

6574 The assessment shall have access to:

- 6575 • Documentation that:
- 6576 - identifies which runtime components are considered security-critical (for example processes or modules
6577 enforcing container isolation, access control, security policies, integrity checks, and configuration data
6578 that directly influences these functions);
 - 6579 - describes how the trusted baseline for these components is established, stored, and updated;
 - 6580 - explains whether integrity validation is periodic, continuous, or event-driven, and how it is configured;
 - 6581 - describes the protective actions that can be triggered when deviations are detected;
 - 6582 - describes the alerting mechanism and how alerts are exposed to administrators or monitoring systems.
 - 6583 • A CRS deployment where runtime integrity validation is enabled and observable (for example via logs, status
6584 commands, or telemetry).
 - 6585 • A controlled means, consistent with vendor guidance, to simulate or induce an integrity deviation affecting at
6586 least one security-critical runtime component or its configuration state (for example a supported test hook,
6587 debug mode, test image, or vendor-provided integrity-test procedure).

6588 4. Assessment Activities

6589 The assessment shall at least:

- 6590 1) Review documentation and configuration to confirm:
- 6591 - which runtime components are treated as security-critical and covered by the trusted baseline;
 - 6592 - how the baseline is created, protected, and updated;
 - 6593 - the schedule or trigger conditions for integrity checks;
 - 6594 - the configured protective actions and alerting behaviour when deviations are detected.
- 6595 2) In a normal, non-tampered configuration:
- 6596 - operate the CRS under representative conditions for a suitable period;
 - 6597 - verify from logs, status outputs, or telemetry that integrity validation is active for the defined security-
6598 critical components (for example periodic checks or continuous monitoring events);
 - 6599 - confirm that no integrity deviation alerts are reported while the baseline remains intact.
- 6600 3) Introduce a controlled integrity deviation affecting a security-critical runtime component or its configuration
6601 state using the sanctioned test method:
- 6602 - allow the CRS to perform its runtime validation;
 - 6603 - verify that the deviation from the trusted baseline is detected;
 - 6604 - verify that the documented protective actions (such as isolating or disabling the affected component or
6605 restoring it from a trusted state) are automatically triggered;
 - 6606 - verify that a security alert is generated and remains visible via CRS or management/monitoring
6607 interfaces, and that this alert cannot be cleared or suppressed solely by the affected component.

6608 5. Assessment Verdict

- 6609 • Pass: security-critical runtime components and associated configuration are covered by a trusted baseline; the
6610 CRS performs periodic or continuous validation against this baseline; controlled deviations are detected;
6611 documented protective actions are automatically applied; and a security alert is generated that cannot be
6612 suppressed by the affected component.
- 6613 • Fail: runtime integrity validation does not cover security-critical components; is not active or observable;
6614 deviations are not detected; protective actions are not triggered; or alerts can be suppressed or removed by the
6615 affected component.

6616 6. Assessment Evidence

- 6617 • [DESIGN] Documentation describing security-critical runtime components, trusted baseline creation and
6618 protection, runtime integrity mechanisms, and protective and alerting behaviour.
- 6619 • [CONFIG] Configuration enabling runtime integrity validation, including monitored components, check
6620 parameters, and alerting/protection settings.
- 6621 • [LOG] Integrity-check logs and security alerts from normal operation and from the controlled deviation
6622 scenario, including timestamps and identifiers of affected components.
- 6623 • [OBSERVATION] Test or monitoring outputs showing runtime integrity checks in operation, detection of the
6624 induced deviation, execution of protective actions, and persistence of the generated alert.
- 6625 • [PLT-CAP] Where hardware- or platform-assisted mechanisms are used as part of runtime integrity validation,
6626 evidence of the underlying platform capabilities and their configuration.

6627 6.4.1.6.3 Assessment Case AC-CRS-RP-INT-004

6628 **Applicability:**

6629 This assessment case applies only when the manufacturer claims that the CRS supports the use of hardware-based
6630 platform mechanisms or externally protected trust material to strengthen runtime integrity validation and detection of
6631 Unauthorized modification of security-critical runtime components.

6632 **1. Assessment Reference**

6633 Requirement: REQ-CRS-RP-INT-004 (Advanced, conditional "may")

6634 **2. Assessment Objective**

6635 Verify that, when claimed:

- 6636 1) The CRS can integrate with hardware-based platform mechanisms or externally protected trust material (such
6637 as dedicated secure hardware modules, hardware-supported memory protection, or equivalent platform
6638 features) to strengthen runtime integrity validation and detection of Unauthorized modification of security-
6639 critical runtime components.
- 6640 2) These mechanisms provide additional assurance of runtime integrity beyond software-only mechanisms.

6641 **3. Assessment Preparation**

6642 The assessment shall have access to:

- 6643 • Documentation describing:
 - 6644 - the specific hardware-based mechanisms or external trust services used (for example secure hardware
6645 modules, protected key stores, memory-protection features, platform trust services);
 - 6646 - which security-critical runtime components are protected or monitored using these mechanisms;
 - 6647 - how the CRS integrates these mechanisms into its runtime integrity validation as defined for REQ-CRS-
6648 RP-INT-003.
- 6649 • Evidence that the test platform or deployment environment provides the claimed mechanisms and that they are
6650 enabled.

- 6651 • A CRS deployment configured to use these hardware-based mechanisms or external trust services to support
- 6652 runtime integrity validation, in accordance with product documentation.

6653 4. Assessment Activities

6654 The assessment shall at least:

- 6655 3) Review documentation and platform information to confirm:
 - 6656 - the availability and configuration of the relevant hardware-based mechanisms or external trust services;
 - 6657 - the mapping between these mechanisms and the set of security-critical runtime components covered by
 - 6658 runtime integrity checks;
 - 6659 - how they contribute to detection and alerting for Unauthorized modifications.
- 6660 4) With the hardware-based mechanisms or external trust material enabled, operate the CRS in a normal
- 6661 configuration and verify, using logs, status outputs, or diagnostics, that:
 - 6662 - runtime integrity validation for the stated components explicitly uses these mechanisms (for example
 - 6663 hardware-backed measurements, protected memory regions, verification using externally protected
 - 6664 keys);
 - 6665 - no integrity violations are reported when the system is intact.
- 6666 5) Where feasible and consistent with vendor guidance, either:
 - 6667 - compare runtime integrity behaviour with and without the mechanisms enabled; or
 - 6668 - induce a controlled deviation targeting a component protected by these mechanisms.

6669 In either case, verify that use of the hardware-based mechanisms or externally protected trust material

6670 contributes to detection or prevention of Unauthorized modification and that any resulting alerts or protective

6671 actions are generated as described.

6672 5. Assessment Verdict

- 6673 • Pass: when support is claimed, the CRS integrates hardware-based platform mechanisms or externally
- 6674 protected trust material into runtime integrity validation for security-critical components; these mechanisms
- 6675 provide additional protection or detection capability beyond software-only mechanisms; and attempts to
- 6676 modify protected components are detected or prevented in line with the documented design.
- 6677 • Fail: the claimed mechanisms are not effectively integrated with runtime integrity validation; provide no
- 6678 observable contribution to protection or detection; or protected components can be modified without detection
- 6679 or prevention despite the claimed support.

6680 6. Assessment Evidence

- 6681 • [DESIGN] Documentation of the integration between CRS runtime integrity validation and hardware-based
- 6682 platform mechanisms or external trust services, including the set of protected components and expected
- 6683 assurances.
- 6684 • [PLT-CAP] Evidence of platform capabilities and configuration of secure hardware modules, memory-
- 6685 protection features, or equivalent mechanisms, or of external trust services used.
- 6686 • [CONFIG] CRS configuration showing that hardware-assisted or externally anchored runtime integrity options
- 6687 are enabled and associated with the relevant components.
- 6688 • [LOG] / [OBSERVATION] Logs, diagnostic outputs, or test results demonstrating use of these mechanisms
- 6689 during runtime integrity validation and, where tested, detection or prevention of Unauthorized modifications to
- 6690 protected components.

6691 6.4.1.7 Assessment for Remote Attestation

6692 6.4.1.7.1 Assessment Case AC-CRS-RA-INT-003

6693 1. Assessment Reference

6694 Requirement: REQ-CRS-RA-INT-003 (Advanced)

6695 2. Assessment Objective

6696 Verify that:

- 6697 1) The CRS supports measured launch and generates attestation evidence describing its core runtime engine,
6698 security-critical configuration, and other runtime state relevant to the integrity of container execution.
- 6699 2) The attestation evidence is cryptographically verifiable by an authorized verifier and is bound to the specific
6700 CRS instance for which the measurements were produced.
- 6701 3) Attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay.
- 6702 4) The attestation evidence is limited to the minimum information necessary to verify the integrity and
6703 configuration state of the CRS.
- 6704 5) Where platform or external trust services are used, the CRS uses them as trust anchors for attestation keys and
6705 measurements in accordance with the product documentation.

6706 3. Assessment Preparation

6707 The assessment shall have access to:

- 6708 • Documentation describing:
 - 6709 - the measured-launch and attestation design, including which CRS components and configuration items
6710 are measured (core runtime engine, security-critical configuration, relevant runtime state);
 - 6711 - how measurements are collected, stored, and associated with a CRS instance;
 - 6712 - the format and export mechanism of the attestation evidence (for example data fields, encoding, transport
6713 or API);
 - 6714 - the cryptographic mechanisms used to protect attestation evidence and bind it to a CRS instance (for
6715 example signature algorithms, keys, certificate chains);
 - 6716 - the protection model for attestation keys and measurement data, including which entities are permitted to
6717 read, manage, or update them;
 - 6718 - any use of platform trust services or externally protected trust material as trust anchors for attestation
6719 keys or measurements.
- 6720 • A test configuration with:
 - 6721 - a CRS deployment with measured launch and attestation enabled according to product documentation;
 - 6722 - an interface to request attestation evidence (for example CLI, API, or management interface);
 - 6723 - a verification tool or reference verifier configured with the relevant trust anchors to validate CRS
6724 attestation evidence;
 - 6725 - at least one container workload or non-privileged host context that can be used to attempt Unauthorized
6726 access to attestation keys or raw measurement data;
 - 6727 - where applicable, a platform or environment that provides the claimed trust services used as anchors for
6728 attestation.

6729 4. Assessment Activities

6730 The assessment shall at least:

- 6731 1) Review documentation and CRS settings to identify:
- 6732 - the set of CRS elements covered by measurements (core runtime engine, security-critical configuration,
6733 relevant runtime state);
 - 6734 - how these measurements are represented in the attestation evidence;
 - 6735 - which cryptographic algorithms and keys are used to protect and sign the evidence;
 - 6736 - how the evidence is bound to a specific CRS instance (for example instance identifiers, platform identity,
6737 or key binding);
 - 6738 - how attestation keys and measurement data are stored and access-controlled;
 - 6739 - where applicable, which platform or external trust services are used as trust anchors.
- 6740 2) After a normal startup of the CRS:
- 6741 - request attestation evidence using the documented interface;
 - 6742 - use the verification tool to confirm that the evidence verifies successfully under the configured trust
6743 anchors and corresponds to the expected CRS instance;
 - 6744 - compare fields in the evidence with the deployed CRS software and configuration (for example runtime
6745 engine version, key security-critical settings) to confirm consistency;
 - 6746 - review the content of the evidence to confirm that it is limited to the information required to verify
6747 integrity and configuration state and does not unnecessarily expose unrelated sensitive operational data.
- 6748 3) From a container workload, non-privileged host account, or other context that should not have direct access:
- 6749 - attempt to read or export attestation private keys or raw measurement data using available interfaces;
 - 6750 - verify that such attempts are rejected or blocked and that keys and measurements cannot be obtained
6751 through these paths.
- 6752 4) Where anti-replay behaviour is documented:
- 6753 - capture a valid attestation response;
 - 6754 - attempt to reuse or replay this response in a context that expects fresh attestation;
 - 6755 - verify, using the verifier or tool, that replayed evidence is detected or rejected in accordance with the
6756 documented behaviour.

6757 5. Assessment Verdict

- 6758 • Pass: the CRS generates attestation evidence covering the core runtime engine, security-critical configuration,
6759 and relevant runtime state; the evidence is cryptographically verifiable and bound to the specific CRS instance;
6760 attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay; the
6761 evidence content is limited to the minimum necessary to verify the integrity and configuration state; and,
6762 where platform or external trust services are claimed as anchors, their use is consistent with the documentation
6763 and visible in the attestation design.
- 6764 • Fail: attestation evidence is not generated as specified; cannot be cryptographically verified or is not clearly
6765 bound to a specific instance; Unauthorized entities can access or export attestation keys or measurement data
6766 or successfully replay stale evidence where freshness is required; the evidence content includes unnecessary
6767 sensitive information beyond what is required to verify integrity and configuration state; or claimed use of
6768 platform or external trust services as trust anchors is not realized in the implementation.

6769 6. Assessment Evidence

- 6770 • [DESIGN] Documentation of the CRS measured-launch and attestation design, including measured
6771 components, evidence format, cryptographic protection, instance binding, protection model for keys and
6772 measurements, and any use of platform or external trust services as trust anchors.
- 6773 • [CONFIG] CRS and, where applicable, platform configuration enabling measured launch and attestation and
6774 defining trust anchors used by verifiers.
- 6775 • [LOG] Records of attestation requests and responses and any logged events related to attestation key access
6776 failures or replay handling.
- 6777 • [OBSERVATION] Captured attestation evidence samples and verifier outputs showing successful validation
6778 for normal runs and rejection or detection of replayed or tampered evidence.
- 6779 • [PLT-CAP] Evidence of platform capabilities or external trust services used to support CRS attestation, where
6780 the implementation relies on such services.

6781 6.4.1.7.2 Assessment Case AC-CRS-RA-INT-004

6782 1. Assessment Reference

6783 Requirement: REQ-CRS-RA-INT-004 (Advanced)

6784 2. Assessment Objective

6785 Verify that, where CRS attestation evidence is intended to be used in remote verification scenarios and this capability is
6786 claimed:

- 6787 1) The CRS supports configuration of privacy-preserving attestation options that reduce linkability and
6788 correlation between attestation events.
- 6789 2) These privacy-preserving options preserve the ability of authorized verifiers to assess the integrity and
6790 configuration state of the CRS.
- 6791 3) The CRS allows configuration of attested information, triggering conditions, and attestation intervals in order
6792 to support data minimization and to prevent excessive or Unauthorized monitoring.

6793 3. Assessment Preparation

6794 The assessment shall have access to:

- 6795 • Documentation describing:
 - 6796 - the CRS remote attestation model, including how CRS identity and instance information are represented
6797 in attestation evidence;
 - 6798 - the privacy-preserving options available (for example limiting attested information, configuration of
6799 attestation triggers and intervals, use of pseudonymous identifiers or separation of identity and integrity
6800 information);
 - 6801 - how these options are configured;
 - 6802 - the expected impact of each option on attestation evidence fields, attestation frequency, and verifier
6803 behaviour.
- 6804 • A test configuration with:
 - 6805 - a CRS deployment where remote attestation to one or more authorized verifiers is enabled;
 - 6806 - the ability to configure and modify the documented privacy-preserving options;
 - 6807 - tools or procedures to trigger multiple attestation events (for example on-demand, periodic, or event-
6808 driven) and capture the resulting attestation evidence under different privacy and scheduling
6809 configurations;
 - 6810 - access to an authorized verifier or verification tool capable of validating integrity and configuration state
6811 from the attestation evidence in both baseline and privacy-enhanced modes.

6812 4. Assessment Activities

6813 The assessment shall at least:

- 6814 1) Review documentation and CRS settings to identify:
- 6815 - the default behaviour of remote attestation when privacy-preserving options are not enabled;
 - 6816 - the available privacy-preserving options and their documented effects on identifiers, metadata, attested
6817 content, and attestation frequency or trigger conditions;
 - 6818 - any prerequisites or constraints for enabling and configuring these options.
- 6819 2) With privacy-preserving options disabled or in their default state:
- 6820 - trigger multiple attestation events over time according to the default model;
 - 6821 - collect the corresponding attestation evidence;
 - 6822 - Analyze the evidence to determine the stability and linkability of identifiers and other potentially
6823 identifying fields across events, and the amount of configuration and runtime information included;
 - 6824 - use the authorized verifier or tool to confirm that the integrity and configuration state of the CRS can be
6825 correctly assessed from this baseline evidence set.
- 6826 3) Enable and configure the documented privacy-preserving options (for example limiting data fields, adjusting
6827 attestation triggers or intervals, enabling pseudonymous identifiers or separation of identity and integrity
6828 information) as described by the manufacturer, then:
- 6829 - trigger multiple attestation events under the privacy-enhanced configuration;
 - 6830 - collect the resulting attestation evidence;
 - 6831 - compare this evidence set with the baseline to verify that linkability and correlation between attestation
6832 events are reduced in the documented manner and that attested content is limited in line with data-
6833 minimization objectives;
 - 6834 - verify, using the authorized verifier or tool, that the integrity and configuration state of the CRS can still
6835 be reliably assessed using the privacy-enhanced evidence and any associated metadata or policies.

6836 5. Assessment Verdict

- 6837 • Pass: when remote attestation is used and privacy support is claimed, the CRS provides configurable privacy-
6838 preserving attestation options that limit attested information, allow control of attestation triggers and intervals,
6839 and, where applicable, support techniques such as pseudonymous identifiers or separation of identity and
6840 integrity information; when enabled, these options reduce linkability and correlation between attestation events
6841 as documented while still allowing authorized verifiers to correctly assess the integrity and configuration state
6842 of the CRS.
- 6843 • Fail: privacy-preserving options are not available despite being claimed; configuration changes do not
6844 meaningfully reduce linkability or correlation between attestation events or do not support data minimization
6845 and control of attestation frequency; or enabling these options prevents authorized verifiers from correctly
6846 assessing the integrity and configuration state of the CRS.

6847 6. Assessment Evidence

- 6848 • [DESIGN] Documentation of the CRS remote attestation model, identity representation, and privacy-
6849 preserving options, including their expected effects on attestation evidence, frequency, and verifier behaviour.
- 6850 • [CONFIG] CRS and verifier configuration showing remote attestation settings, selected privacy-preserving
6851 options, and configured triggers or intervals.
- 6852 • [LOG] Records of attestation events and related configuration changes, including timestamps and any
6853 metadata relevant to privacy options and scheduling.

- 6854 • [OBSERVATION] Sets of attestation evidence collected before and after enabling privacy-preserving options,
6855 with verifier outputs showing successful integrity and configuration assessment in both modes and reduced
6856 linkability or correlation when privacy options are enabled.

6857 6.4.1.8 Assessment for Administrative Authentication

6858 6.4.1.8.1 Assessment Case AC-CRS-ADMIN-AUTH-001

6859 1. Assessment Reference

6860 Requirement: REQ-CRS-ADMIN-AUTH-001 (Basic)

6861 2. Assessment Objective

6862 Verify that:

- 6863 1) CRS administrative interfaces require authentication before any administrative operation.
- 6864 2) Administrative accounts use unique, non-default credentials and default credentials are disabled or changed
6865 before use.
- 6866 3) Password complexity and lockout mechanisms for administrative accounts are configurable and enforced to
6867 mitigate brute-force attempts.
- 6868 4) Where password-based authentication is used, administrative passwords are stored only as salted hashes.

6869 3. Assessment Preparation

6870 The assessment shall have access to:

- 6871 • Declared CRS product composition and list of administrative interfaces (CLI, APIs, dashboards/consolas).
- 6872 • Documentation on administrative account model, default credentials, password policy and lockout options, and
6873 password storage mechanism.
- 6874 • At least one CRS configuration with administrative accounts and access to identified administrative interfaces.

6875 4. Assessment Activities

6876 The assessment shall at least:

- 6877 1) Identify administrative interfaces from documentation/configuration and verify that each requires
6878 authentication in the assessed configuration.
- 6879 2) Review installation/hardening guidance and verify in the assessed configuration that any default administrative
6880 credentials are disabled or enforced to be changed before administrative access is granted.
- 6881 3) Confirm that password complexity and lockout policies for administrative accounts can be configured and
6882 verify that non-compliant passwords are rejected and that lockout is applied according to the configured
6883 thresholds.
- 6884 4) Review documentation and, where feasible, configuration/implementation details to confirm that CRS
6885 administrative passwords are stored only as salted hashes (no cleartext).

6886 5. Assessment Verdict

- 6887 • Pass: CRS administrative interfaces require authentication; administrative accounts do not rely on unchanged
6888 default credentials; password complexity and lockout are configurable and enforced; and administrative
6889 passwords are stored only as salted hashes.
- 6890 • Fail: any administrative interface allows unauthenticated administrative operations; or unchanged default
6891 credentials can be used; or password complexity/lockout cannot be configured or are not enforced; or
6892 administrative passwords are stored in cleartext.

6893 6. Assessment Evidence

- 6894 • [SCOPE] CRS product composition and list of administrative interfaces.
- 6895 • [CONFIG] Administrative authentication, password policy and lockout configuration.

- 6896 • [DESIGN] Description of administrative access model, default account handling and password storage
6897 mechanism.
- 6898 • [LOG] / [OBSERVATION] Evidence of authentication enforcement, rejection of weak passwords and logout
6899 behaviour.

6900 6.4.1.8.2 Assessment Case AC-CRS-ADMIN-AUTH-002

6901 1. Assessment Reference

6902 Requirement: REQ-CRS-ADMIN-AUTH-002 (Elevated)

6903 2. Assessment Objective

6904 Verify that:

- 6905 1) CRS administrative interfaces require authenticated access and support cryptographic key-based authentication
6906 for administrative accounts.
- 6907 2) Password-based login for CRS administrative interfaces can be disabled.
- 6908 3) Private keys used for CRS administrative authentication are protected from extraction or reuse by container
6909 workloads or other unprivileged components.
- 6910 4) Algorithms and key lengths for key-based administrative authentication conform to the cryptographic profile
6911 declared by the manufacturer, aligned with [2].

6912 3. Assessment Preparation

6913 The assessment shall have access to:

- 6914 • Documentation on key-based administrative authentication for each CRS administrative interface.
- 6915 • Documentation on applicable cryptographic profiles for administrative authentication (algorithms, key lengths,
6916 disallowed options).
- 6917 • Documentation of key management and key storage for administrative private keys, including separation from
6918 container workloads and unprivileged components.
- 6919 • At least one CRS configuration where administrative accounts use key-based authentication on relevant
6920 interfaces and password-based login can be disabled.

6921 4. Assessment Activities

6922 The assessment shall at least:

- 6923 1) Confirm from documentation/configuration that administrative accounts can use key-based authentication on
6924 relevant CRS interfaces and verify that an administrative account can obtain administrative access using the
6925 key-based mechanism.
- 6926 2) Confirm that the CRS can disable password-based login for administrative interfaces and verify that, where
6927 disabled, password-only attempts are rejected while key-based authentication still permits authorized access.
- 6928 3) Review configuration and, where applicable, protocol characteristics to confirm that algorithms, key lengths
6929 and protocol versions used for key-based administrative authentication comply with the declared cryptographic
6930 profile and that weaker parameters marked as non-compliant are not used.
- 6931 4) Review key storage design and configuration to confirm that container workloads and other unprivileged
6932 components cannot read, export or misuse administrative private keys.

6933 5. Assessment Verdict

- 6934 • Pass: CRS administrative interfaces require authenticated access and support key-based administrative
6935 authentication; password-based login can be disabled and is effectively blocked where configured;
6936 cryptographic parameters comply with the declared profile; and administrative private keys are protected from
6937 container workloads and unprivileged components.

- 6938 • Fail: key-based administrative authentication is not available or not functional; or password-based login cannot
6939 be disabled or remains effective when disabled; or cryptographic parameters are weaker than the declared
6940 profile; or administrative private keys are accessible to container workloads or other unprivileged components.

6941 6. Assessment Evidence

- 6942 • [CONFIG] CRS configuration for key-based administrative authentication, password-login disablement and
6943 crypto settings.
- 6944 • [DESIGN] Documentation of key management and key storage architecture and applied cryptographic profile.
- 6945 • [LOG] / [OBSERVATION] Evidence of successful key-based administrative authentication and rejection of
6946 password-only attempts when disabled.
- 6947 • [PLT-CAP] Where relevant, evidence of platform capabilities used for key protection.

6948 6.4.1.8.3 Assessment Case AC-CRS-ADMIN-AUTH-003

6949 1. Assessment Reference

6950 Requirement: REQ-CRS-ADMIN-AUTH-003 (Advanced)

6951 2. Assessment Objective

6952 Verify that:

- 6953 1) Strong, cryptographically verifiable authentication is supported and enforceable for CRS administrative
6954 accounts on administrative interfaces in scope.
- 6955 2) CRS administrative identities can be bound to a trusted, managed credential source (e.g. PKI or external
6956 identity provider).
- 6957 3) The CRS integrates with external identity or credential management systems to support controlled issuance,
6958 rotation and revocation of administrative credentials.
- 6959 4) The CRS checks validity and revocation (or equivalent invalidation) status of externally managed
6960 administrative credentials before granting administrative access.

6961 3. Assessment Preparation

6962 The assessment shall have access to:

- 6963 • Documentation of strong administrative authentication mechanisms (e.g. certificate-based) for CRS
6964 administrative interface.
- 6965 • Documentation of supported integrations with external identity/credential management systems (PKI, IdP,
6966 directory), including mapping of external identities to CRS administrative roles/accounts.
- 6967 • Documentation describing how externally managed administrative credentials are issued, rotated and revoked,
6968 and how the CRS consumes this information (validity, revocation checks).
- 6969 • At least one CRS configuration where administrative accounts are bound to a trusted external credential
6970 source, with usable valid and invalid/expired/revoked credentials (or equivalent conditions) for testing.

6971 4. Assessment Activities

6972 The assessment shall at least:

- 6973 1) Confirm from documentation/configuration that strong, cryptographically verifiable credentials can be
6974 enforced for administrative accounts on CRS administrative interfaces in scope and verify that administrative
6975 access is granted only when such a valid credential is presented.
- 6976 2) Confirm that CRS administrative identities can be bound to a trusted external credential source (e.g. PKI-
6977 issued certificates, IdP-managed identities) and that this binding is reflected in the CRS role/access model.
- 6978 3) Confirm that the CRS integrates with external identity/credential management systems for administrative
6979 credentials and that it can handle credential issuance and rotation as defined by those systems.

- 6980 4) Verify in the assessed configuration that the CRS checks credential validity period and, where supported,
 6981 revocation or equivalent invalidation status, and denies administrative access when credentials are invalid,
 6982 expired, revoked or otherwise not trusted.

6983 5. Assessment Verdict

- 6984 • Pass: strong, cryptographically verifiable administrative authentication is enforceable on CRS administrative
 6985 interfaces in scope; administrative identities are bound to a trusted external credential source; integration with
 6986 external identity/credential management supports issuance, rotation and revocation; and the CRS verifies
 6987 validity and revocation (or equivalent) of such credentials before granting administrative access.
- 6988 • Fail: strong administrative authentication cannot be enforced on administrative interfaces in scope; or
 6989 administrative identities cannot be effectively bound to a trusted external credential source; or integration with
 6990 external identity/credential management is insufficient for controlled issuance, rotation and revocation; or the
 6991 CRS does not properly verify validity and revocation (or equivalent) before granting administrative access.

6992 6. Assessment Evidence

- 6993 • [CONFIG] CRS configuration for strong administrative authentication, bindings to external credential sources,
 6994 and identity/credential integration settings.
- 6995 • [DESIGN] Documentation of CRS administrative authentication architecture, including external
 6996 identity/credential integration and credential lifecycle handling.
- 6997 • [LOG] / [OBSERVATION] Evidence of successful administrative access with valid external credentials and
 6998 denial of access with invalid, expired or revoked credentials.
- 6999 • [PLT-CAP] Where relevant, evidence of platform or infrastructure capabilities used to support strong
 7000 administrative authentication and external identity integration.

7001 6.4.1.9 Assessment for Service Authentication

7002 6.4.1.9.1 Assessment Case AC-CRS-SERV-AUTH-001

7003 1. Assessment Reference

7004 Requirement: REQ-CRS-SERV-AUTH-001 (Basic)

7005 2. Assessment Objective

7006 Verify that:

- 7007 1) External services accessing CRS APIs or control interfaces (e.g. orchestration, monitoring, backup) are
 7008 authenticated.
- 7009 2) Each service account uses distinct, non-default credentials and no built-in shared or hard-coded service
 7010 credentials are relied upon.
- 7011 3) Configurable lockout and/or rate-limit mechanisms exist for service authentication attempts and are enforced.
- 7012 4) Service credentials are stored in a way that prevents exposure via configuration files, container images or logs.

7013 3. Assessment Preparation

7014 The assessment shall have access to:

- 7015 • Declared CRS product composition and list of external services/interfaces accessing CRS APIs or control
 7016 interfaces.
- 7017 • Documentation on the service account model, credential provisioning, any default/built-in service credentials,
 7018 and authentication lockout/rate-limiting options.
- 7019 • Documentation on how service credentials are stored (e.g. secrets store, environment variables, KMS) and how
 7020 configuration, images and logs are handled.
- 7021 • At least one configuration with representative external services integrated using service accounts.

7022 **4. Assessment Activities**

7023 The assessment shall at least:

- 7024 1) Identify external services accessing CRS APIs or control interfaces and verify that each authenticates
7025 successfully before invoking management or control operations.
- 7026 2) Review configuration and documentation to confirm that each service account has distinct credentials and that
7027 normal operation does not require shared or hard-coded service credentials; verify for at least one service that
7028 its credentials are unique and changeable.
- 7029 3) Confirm that lockout and/or rate-limit thresholds for service authentication attempts can be configured; verify
7030 that repeated failed authentication attempts from a service trigger lockout or rate-limiting as configured.
- 7031 4) Review the documented storage locations and mechanisms for service credentials and inspect representative
7032 configuration files, container images and logs to confirm that credentials are not stored or exposed in cleartext
7033 or other directly reusable form in these artefacts.

7034 **5. Assessment Verdict**

- 7035 • Pass: external services accessing CRS APIs or control interfaces require authentication; each service account
7036 uses distinct, non-default credentials with no reliance on shared/hard-coded credentials; lockout or rate-limit
7037 mechanisms for service authentication attempts are configurable and enforced; and service credentials are not
7038 exposed via configuration files, container images or logs.
- 7039 • Fail: unauthenticated service access to CRS APIs/control interfaces is possible; or shared/default/hard-coded
7040 service credentials are required or used; or lockout/rate-limit mechanisms are not available, not configurable,
7041 or not enforced; or service credentials are exposed in configuration files, images or logs.

7042 **6. Assessment Evidence**

- 7043 • [SCOPE] List of external services and CRS APIs/control interfaces in scope.
- 7044 • [CONFIG] Service authentication, credential, lockout and rate-limiting configuration.
- 7045 • [DESIGN] Documentation of the service account and credential storage model, including handling of
7046 configuration, images and logs.
- 7047 • [LOG] / [OBSERVATION] Evidence of enforced authentication, triggered lockout or rate-limiting, and
7048 absence of exposed credentials in sampled configuration files, images and logs.

7049 **6.4.1.9.2 Assessment Case AC-CRS-SERV-AUTH-002**

7050 **1. Assessment Reference**

7051 Requirement: REQ-CRS-SERV-AUTH-002 (Elevated)

7052 **2. Assessment Objective**

7053 Verify that:

- 7054 1) The CRS supports mutual cryptographic authentication for service-to-CRS interactions that access
7055 management functions or data, allowing both parties to verify each other's identity.
- 7056 2) Cryptographic materials used for service authentication (keys, certificates, tokens) are protected from
7057 Unauthorized extraction or reuse by container workloads or other unprivileged components.
- 7058 3) Algorithms and key lengths used for service authentication conform to the cryptographic profile declared by
7059 the manufacturer, aligned with [2].

7060 **3. Assessment Preparation**

7061 The assessment shall have access to:

- 7062 • Documentation on mutual cryptographic authentication mechanisms for service-to-CRS interactions (e.g.
7063 mTLS, signed tokens).
- 7064 • Documentation on applicable cryptographic profiles for service authentication (approved algorithms,
7065 minimum key lengths, disallowed options).

7066 • Documentation on key/token management and storage (e.g. keystores, HSM/KMS, secret stores) and how
7067 separation from container workloads/unprivileged components is enforced.

7068 • At least one configuration where representative services access CRS management functions or data using
7069 mutual cryptographic authentication.

7070 **4. Assessment Activities**

7071 The assessment shall at least:

7072 1) Confirm from documentation/configuration that mutual cryptographic authentication is supported for service-
7073 to-CRS interactions that access management functions or data; verify in the assessed configuration that a
7074 service using mutual authentication can establish an authenticated session and access management functions or
7075 data only after successful mutual verification.

7076 2) Review configuration/protocol traces to confirm that algorithms, key lengths and protocol versions used for
7077 mutual authentication comply with the declared cryptographic profile and that non-compliant algorithms or
7078 key sizes are not negotiated.

7079 3) Review the design and configuration of storage for service authentication keys/certificates/tokens and confirm
7080 that container workloads and other unprivileged components cannot read, export or misuse these materials
7081 according to documented isolation and access-control mechanisms.

7082 **5. Assessment Verdict**

7083 • Pass: mutual cryptographic authentication is available and functional for service-to-CRS interactions that
7084 access management functions or data; cryptographic parameters used conform to the declared profile; and
7085 keys/certificates/tokens used for service authentication are protected from extraction or reuse by container
7086 workloads and other unprivileged components.

7087 • Fail: mutual cryptographic authentication is not available or not functional where required; or cryptographic
7088 parameters are weaker than the declared profile; or cryptographic materials for service authentication can be
7089 accessed or reused by container workloads or other unprivileged components.

7090 **6. Assessment Evidence**

7091 • [CONFIG] Mutual authentication configuration for services (certificates/keys/tokens and crypto settings).

7092 • [DESIGN] Documentation of service authentication architecture, key/token management and storage, and
7093 applied cryptographic profile.

7094 • [LOG] / [OBSERVATION] Evidence of successful mutual authentication and rejection of connections using
7095 non-compliant parameters or missing authentication.

7096 • [PLT-CAP] Where relevant, evidence of platform capabilities used to protect service authentication materials.

7097 **6.4.1.9.3 Assessment Case AC-CRS-SERV-AUTH-003**

7098 **1. Assessment Reference**

7099 Requirement: REQ-CRS-SERV-AUTH-003 (Advanced)

7100 **2. Assessment Objective**

7101 Verify that:

7102 1) The CRS enforces a verifiable binding between service identities and cryptographic credentials for service
7103 interactions that access management functions or data.

7104 2) Service credentials are issued, rotated and revoked through trusted credential or identity management systems.

7105 3) The CRS automatically rejects revoked or expired service credentials during authentication.

7106 4) The CRS supports integration with external identity or credential management systems to provide traceability
7107 of service identities across deployments and to ensure that only services with valid, managed identities can
7108 invoke CRS management or control functions.

7109 **3. Assessment Preparation**

7110 The assessment shall have access to:

- 7111 • Documentation describing how service identities are represented (e.g. certificate subject/SAN, token claims,
7112 identity IDs) and how they are bound to cryptographic credentials.
- 7113 • Documentation of trusted credential/identity management systems used (PKI, IdP, secret/credential manager)
7114 and the processes for issuing, rotating and revoking service credentials.
- 7115 • Documentation on how the CRS validates credential status (expiry, revocation, or equivalent) during
7116 authentication.
- 7117 • Documentation on integration with external identity/credential management for traceability (e.g. logs,
7118 mappings, cross-deployment identifiers).
- 7119 • At least one configuration where service credentials are managed by an external trusted system, and valid,
7120 expired and revoked credentials (or equivalent conditions) can be exercised.

7121 **4. Assessment Activities**

7122 The assessment shall at least:

- 7123 1) Confirm that the CRS binds service identities to cryptographic credentials (e.g. through certificate identity or
7124 token claims) and verify that, in the assessed configuration, access to management or control functions
7125 depends on presenting a credential that matches the bound identity.
- 7126 2) Verify that service credentials used with the CRS are issued, rotated and revoked by a trusted
7127 credential/identity management system; confirm, for at least one service, that credential updates (rotation) and
7128 revocation performed in the external system are reflected in CRS behaviour.
- 7129 3) Verify that the CRS checks credential expiry and revocation (or equivalent status) during authentication and
7130 automatically rejects expired or revoked service credentials before granting access.
- 7131 4) Verify that integration with the external identity/credential management system allows tracing service
7132 identities across deployments (e.g. consistent identifiers or mappings) and that attempts to invoke CRS
7133 management or control functions from services without valid, managed identities are denied.

7134 **5. Assessment Verdict**

- 7135 • Pass: the CRS enforces verifiable binding of service identities to cryptographic credentials for service
7136 interactions accessing management functions or data; service credentials are managed (issued, rotated,
7137 revoked) by trusted systems; revoked or expired credentials are automatically rejected; and integration with
7138 external identity/credential management provides traceability and ensures that only services with valid,
7139 managed identities can invoke CRS management or control functions.
- 7140 • Fail: service identity is not reliably bound to cryptographic credentials; or service credentials are not managed
7141 by trusted external systems; or revoked/expired credentials are not automatically rejected; or integration with
7142 external identity/credential management does not provide effective traceability or allow the CRS to block
7143 services without valid, managed identities.

7144 **6. Assessment Evidence**

- 7145 • [CONFIG] CRS configuration for service identity binding, use of externally managed credentials and
7146 authentication status checks.
- 7147 • [DESIGN] Documentation of the service identity and credential-lifecycle model, including integration with
7148 external identity/credential management and cross-deployment identity handling.
- 7149 • [LOG] / [OBSERVATION] Evidence of successful authentication with valid managed credentials, rejection of
7150 expired/revoked credentials, and denial of access for services without valid managed identities.
- 7151 • [PLT-CAP] Where relevant, evidence of platform capabilities used to support identity-bound credentials and
7152 status checking (e.g. PKI libraries, identity client libraries).

7153 6.4.1.10 Assessment for Administrative Authorization

7154 6.4.1.10.1 Assessment Case AC-CRS-ADMIN-AUTHZ-001

7155 1. Assessment Reference

7156 Requirement: REQ-CRS-ADMIN-AUTHZ-001 (Basic)

7157 2. Assessment Objective

7158 Verify that:

- 7159 1) All administrative actions are restricted to authenticated administrative accounts.
- 7160 2) A default-deny policy is enforced for access to CRS management functions, permitting only explicitly
7161 authorized actions.

7162 3. Assessment Preparation

7163 The assessment shall have access to:

- 7164 • Documentation of CRS administrative accounts, roles and management interfaces.
- 7165 • Documentation of the administrative authorization model, including default behaviour.
- 7166 • At least one configuration with multiple administrative accounts and access to CRS management functions.

7167 4. Assessment Activities

7168 The assessment shall at least:

- 7169 1) Verify that unauthenticated users and non-administrative accounts cannot perform CRS administrative actions
7170 on any management interface.
- 7171 2) Confirm from configuration and tests that administrative actions not explicitly granted to an authenticated
7172 administrative account are denied by default.

7173 5. Assessment Verdict

- 7174 • Pass: only authenticated administrative accounts can perform CRS administrative actions and any non-
7175 authorized administrative action is denied by default.
- 7176 • Fail: unauthenticated/non-administrative users can perform administrative actions; or actions not explicitly
7177 authorized are allowed.

7178 6. Assessment Evidence

- 7179 • [CONFIG] Administrative authorization and default policy configuration.
- 7180 • [DESIGN] Description of CRS administrative authorization model and default-deny behaviour.
- 7181 • [LOG] / [OBSERVATION] Evidence of denied operations for unauthenticated/non-admin users and permitted
7182 operations only when explicitly authorized.

7183 6.4.1.10.2 Assessment Case AC-CRS-ADMIN-AUTHZ-002

7184 1. Assessment Reference

7185 Requirement: REQ-CRS-ADMIN-AUTHZ-002 (Elevated)

7186 2. Assessment Objective

7187 Verify that:

- 7188 1) Granular RBAC or ABAC is enforced for CRS administrative accounts.
- 7189 2) Each administrative account is granted only the minimum privileges required for its operational role (least
7190 privilege).

7191 **3. Assessment Preparation**

7192 The assessment shall have access to:

- 7193 • Documentation describing RBAC/ABAC for CRS administrative accounts (roles, attributes, permissions).
- 7194 • At least one configuration with several administrative roles/attributes showing clearly differentiated privilege levels.

7196 **4. Assessment Activities**

7197 The assessment shall at least:

- 7198 1) Confirm that administrative permissions are defined and enforced via RBAC or ABAC (e.g. role-/attribute-based rules), not via a single full-privilege administrator profile.
- 7199
- 7200 2) For selected administrative accounts with different roles/attributes, verify that each account can perform only those actions associated with its defined role/attributes and that higher-privilege actions are denied, demonstrating least-privilege enforcement.
- 7201
- 7202

7203 **5. Assessment Verdict**

- 7204 • Pass: RBAC/ABAC is in place and selected administrative accounts are restricted to the minimum privileges defined for their roles/attributes.
- 7205
- 7206 • Fail: RBAC/ABAC is not effectively enforced; or administrative accounts have broader privileges than defined; or least-privilege cannot be demonstrated.
- 7207

7208 **6. Assessment Evidence**

- 7209 • [CONFIG] RBAC/ABAC configuration for administrative accounts (roles/attributes and permissions).
- 7210 • [DESIGN] Documentation of the CRS administrative authorization model and least-privilege approach.
- 7211 • [LOG] / [OBSERVATION] Evidence of allowed actions within role scope and denied actions outside that scope.
- 7212

7213 **6.4.1.10.3 Assessment Case AC-CRS-ADMIN-AUTHZ-003**

7214 **1. Assessment Reference**

7215 Requirement: REQ-CRS-ADMIN-AUTHZ-003 (Advanced)

7216 **2. Assessment Objective**

7217 Verify that:

- 7218 1) Fine-grained authorization policies are enforced for CRS administrative accounts.
- 7219 2) Separation of duties (SoD) for administrative functions is supported and can be enforced.
- 7220 3) Time-bound or just-in-time (JIT) privilege elevation is supported and correctly revoked when no longer needed.
- 7221
- 7222 4) Integration with external identity and policy decision systems is supported to enable centralized definition and enforcement of administrative authorization policies.
- 7223

7224 **3. Assessment Preparation**

7225 The assessment shall have access to:

- 7226 • Documentation of policy-based administrative authorization (policy language/rules, SoD and JIT models).
- 7227 • Documentation of integration with external identity and policy decision systems (if supported).
- 7228 • At least one configuration with:
 - 7229 - distinct administrative roles/functions demonstrating SoD; and
 - 7230 - an example of time-bound or JIT privilege elevation.

7231 **4. Assessment Activities**

7232 The assessment shall at least:

- 7233 1) Verify that administrative authorization rules are expressed as fine-grained policies and that different
7234 administrative functions can be assigned to different accounts to support SoD (e.g. one account configuring
7235 changes and another approving or deploying them).
- 7236 2) Verify that time-bound or JIT elevation can be configured for an administrative account, that elevation is
7237 limited in scope and duration, and that privileges are automatically revoked after expiry or completion.
- 7238 3) Where integration with external identity and policy decision systems is supported, verify that the CRS can use
7239 those systems to obtain administrative identity/attributes and authorization decisions, and that changes to
7240 central policies are reflected in CRS administrative authorization behaviour.

7241 **5. Assessment Verdict**

- 7242 • Pass: administrative authorization is fine-grained and policy-driven; SoD and time-bound/JIT elevation are
7243 supported and function as documented; and, where applicable, external identity and policy decision integration
7244 is effective.
- 7245 • Fail: fine-grained policies cannot be used to control administrative authorization; SoD cannot be enforced;
7246 JIT/time-bound elevation is not available or not effectively revoked; or supported external integration cannot
7247 be used for centralized administrative authorization.

7248 **6. Assessment Evidence**

- 7249 • [CONFIG] Administrative policy configuration showing fine-grained rules, SoD and time-bound/JIT
7250 elevation, including any external PDP integration settings.
- 7251 • [DESIGN] Documentation of policy-driven administrative authorization, SoD model and external
7252 identity/policy decision integration.
- 7253 • [LOG] / [OBSERVATION] Evidence of SoD enforcement, temporary elevation and revocation, and, where
7254 applicable, authorization decisions influenced by external systems.

7255 **6.4.1.11 Assessment for Service Authorization**

7256 **6.4.1.11.1 Assessment Case AC-CRS-SERV-AUTHZ-001**

7257 **1. Assessment Reference**

7258 Requirement: REQ-CRS-SERV-AUTHZ-001 (Basic)

7259 **2. Assessment Objective**

7260 Verify that:

- 7261 1) A default-deny policy is enforced for all CRS service accounts.
- 7262 2) Service accounts are only permitted to perform actions explicitly authorized by CRS configuration or policy.

7263 **3. Assessment Preparation**

7264 The assessment shall have access to:

- 7265 • List of CRS service accounts and service-facing interfaces.
- 7266 • Documentation of the authorization model and default behaviour for service accounts.
- 7267 • At least one configuration with several service accounts, including one with no explicit permissions.

7268 **4. Assessment Activities**

7269 The assessment shall at least:

- 7270 1) Confirm from documentation/configuration that the default behaviour for service accounts is to deny any
7271 action not explicitly authorized.

7272 2) Verify that a service account with no explicit permissions cannot perform management or data-related
7273 operations.

7274 3) For a service account with defined permissions, verify that actions within its configured permissions are
7275 allowed and that non-authorized actions are denied.

7276 5. Assessment Verdict

7277 • Pass: default behaviour for service accounts is deny; unprivileged service accounts cannot perform operations;
7278 and only explicitly authorized actions are permitted.

7279 • Fail: unspecified actions are allowed for service accounts; or a service account without explicit permissions
7280 can perform operations.

7281 6. Assessment Evidence

7282 • [SCOPE] List of CRS service accounts and interfaces in scope.

7283 • [CONFIG] Service-account authorization and default policy configuration.

7284 • [DESIGN] Description of the CRS service-account authorization model and default-deny behaviour.

7285 • [LOG] / [OBSERVATION] Evidence of denied operations for unprivileged service accounts and successful
7286 execution of explicitly authorized actions.

7287 6.4.1.11.2 Assessment Case AC-CRS-SERV-AUTHZ-002

7288 1. Assessment Reference

7289 Requirement: REQ-CRS-SERV-AUTHZ-002 (Elevated)

7290 2. Assessment Objective

7291 Verify that:

7292 1) Granular authorization is enforced for CRS service accounts, with permitted actions bound to the authenticated
7293 service identity.

7294 2) Least-privilege permissions are enforced for each service account.

7295 3. Assessment Preparation

7296 The assessment shall have access to:

7297 • Documentation on how service identities are represented and bound to permissions.

7298 • Documentation on the permission or role model for CRS service accounts.

7299 • At least one configuration with several service accounts having different, clearly scoped permissions.

7300 4. Assessment Activities

7301 The assessment shall at least:

7302 1) Confirm that authorization decisions for service accounts are based on the authenticated service identity, not
7303 solely on generic network or system properties.

7304 2) For at least two service accounts with different permissions, verify that each account can perform only the
7305 actions explicitly bound to its identity and that actions belonging to another service identity are denied.

7306 3) Verify that a service account configured with a minimal permission set cannot perform operations outside that
7307 defined set, demonstrating least privilege.

7308 5. Assessment Verdict

7309 • Pass: authorization for service accounts is identity-bound and granular, and least-privilege permission sets can
7310 be configured and are enforced for individual service accounts.

7311 • Fail: actions are not clearly bound to service identity; or service accounts can perform operations beyond their
7312 assigned permissions; or least-privilege cannot be demonstrated.

7313 6. Assessment Evidence

- 7314 • [CONFIG] Service-account authorization configuration showing bindings between specific identities and
7315 allowed actions, including at least one minimal-permission account.
- 7316 • [DESIGN] Documentation of the CRS service-account authorization model, identity binding and least-
7317 privilege approach.
- 7318 • [LOG] / [OBSERVATION] Evidence of allowed operations within each service account's permissions and
7319 denied operations outside that scope.

7320 6.4.1.11.3 Assessment Case AC-CRS-SERV-AUTHZ-003

7321 1. Assessment Reference

7322 Requirement: REQ-CRS-SERV-AUTHZ-003 (Advanced)

7323 2. Assessment Objective

7324 Verify that:

- 7325 1) Fine-grained, policy-driven authorization is enforced for CRS service accounts.
- 7326 2) Permissions for service accounts are bound to cryptographic service identities.
- 7327 3) Context-aware enforcement (e.g. namespace, cluster, labels or similar attributes) is supported for service-
7328 account authorization decisions.
- 7329 4) Integration with external authorization systems is supported, where available, to enable centralized
7330 management and evaluation of service-account policies.

7331 3. Assessment Preparation

7332 The assessment shall have access to:

- 7333 • Documentation of policy-driven authorization for CRS service accounts (policy language/rules, conditions,
7334 effects).
- 7335 • Documentation on how cryptographic service identities (e.g. certificate identities, token claims) are bound to
7336 permissions.
- 7337 • Documentation on context attributes available to policy evaluation (e.g. environment, namespace, workload
7338 labels).
- 7339 • Documentation on integration with external authorization systems (PDPs/policy engines), if supported.
- 7340 • At least one configuration where:
 - 7341 - service-account policies reference cryptographic identities and at least one context attribute; and
 - 7342 - where supported, an external authorization system is integrated and used for policy evaluation.

7343 4. Assessment Activities

7344 The assessment shall at least:

- 7345 1) Verify that CRS service-account authorization can be expressed and enforced via fine-grained policies and that
7346 permissions for at least one service account are defined in such policies.
- 7347 2) Verify that policies can reference cryptographic identifiers of service accounts and that, in the assessed
7348 configuration, authorization decisions for at least one service depend on its cryptographic identity.
- 7349 3) Verify that at least one context attribute (e.g. namespace, cluster, label) influences authorization decisions for
7350 service accounts and that access is allowed or denied based on that context in the assessed configuration.
- 7351 4) Where integration with an external authorization system is supported, verify that the CRS consults or delegates
7352 service-account authorization decisions to that system and that policy changes in the external system are
7353 reflected in CRS authorization outcomes.

7354 **5. Assessment Verdict**

- 7355 • Pass: service-account authorization is fine-grained and policy-driven; permissions are bound to cryptographic
7356 service identities; context-aware enforcement is supported; and, where applicable, external authorization
7357 integration allows centralized policy management and evaluation.
- 7358 • Fail: policies cannot express fine-grained service-account authorization; or permissions are not effectively
7359 bound to cryptographic identities; or context attributes cannot influence authorization decisions; or supported
7360 external authorization integration is not effective.

7361 **6. Assessment Evidence**

- 7362 • [CONFIG] Policy configuration for service-account authorization referencing cryptographic identities and
7363 context attributes, and, where applicable, external authorization integration settings.
- 7364 • [DESIGN] Documentation of the policy-driven CRS service authorization architecture, including
7365 cryptographic identity binding, context handling and external authorization interfaces.
- 7366 • [LOG] / [OBSERVATION] Evidence of authorization decisions depending on cryptographic identity and
7367 context, and of changes resulting from external authorization policy updates where supported.

7368 **6.4.1.12 Assessment for Confidentiality Protection**7369 **6.4.1.12.1 Assessment Case AC-CRS-CONF-001**7370 **1. Assessment Reference**

7371 Requirement: REQ-CRS-CONF-001 (Basic)

7372 **2. Assessment Objective**

7373 Verify that:

- 7374 1) The CRS provides access-control mechanisms to restrict access to sensitive data under its control, including
7375 container images, configuration data, stored secrets, and security-relevant logs.
- 7376 2) Access to such sensitive data is permitted only to authenticated and authorized entities through CRS-controlled
7377 interfaces and APIs, and not directly accessible to unauthenticated or Unauthorized contexts (including
7378 container workloads).

7379 **3. Assessment Preparation**

7380 The assessment shall have access to:

- 7381 • Documentation describing:
- 7382 - the types of sensitive data handled by the CRS (for example container images under CRS control,
7383 configuration objects, secrets, security-relevant logs);
- 7384 - the storage locations and repositories where such data is kept;
- 7385 - the CRS interfaces and APIs through which this data can be accessed or managed;
- 7386 - the access-control model, including roles or identities authorized to access each type of sensitive data.
- 7387 • A test configuration with:
- 7388 - at least one administrative account/role authorized to access and manage sensitive data;
- 7389 - at least one non-privileged or operational account/role with limited permissions;
- 7390 - one or more container workloads under CRS control;
- 7391 - CRS management interfaces and APIs reachable for testing.

7392 **4. Assessment Activities**

7393 The assessment shall at least:

- 7394 1) Review documentation and CRS configuration to identify:
- 7395 - which CRS repositories or objects correspond to container images, configuration data, stored secrets, and
 - 7396 security-relevant logs;
 - 7397 - which CRS-controlled interfaces and APIs are intended for accessing or managing each type of sensitive
 - 7398 data;
 - 7399 - which roles/accounts are authorized to use these interfaces.
- 7400 2) Using an authenticated administrative account:
- 7401 - access representative examples of each sensitive data type via the documented CRS interfaces/APIs (for
 - 7402 example listing and retrieving container images, viewing configuration objects, reading secrets via a CRS
 - 7403 secret-management API, accessing security logs);
 - 7404 - verify that such access is successful only via the CRS-controlled interfaces and behaves as documented.
- 7405 3) Using a non-privileged account, unauthenticated context, or from within a container workload:
- 7406 - attempt to access or modify the same sensitive data via CRS APIs without appropriate permissions;
 - 7407 - where underlying storage paths are visible, attempt direct access to data repositories outside of CRS-
 - 7408 controlled interfaces;
 - 7409 - verify that these attempts are denied or blocked and do not result in disclosure or modification of
 - 7410 sensitive data.

7411 **5. Assessment Verdict**

- 7412 • Pass: access to container images, configuration data, stored secrets, and security-relevant logs is restricted to
- 7413 authenticated and authorized entities via CRS-controlled interfaces; attempts by unauthenticated,
- 7414 Unauthorized, or workload-level contexts to access or modify such data are prevented.
- 7415 • Fail: sensitive data can be read or modified without using CRS-controlled interfaces, or by unauthenticated or
- 7416 Unauthorized contexts; or the implemented access-control behaviour differs from the documented model.

7417 **6. Assessment Evidence**

- 7418 • [DESIGN] Documentation of sensitive data types, storage locations, CRS interfaces/APIs, and the access-
- 7419 control model.
- 7420 • [CONFIG] CRS configuration showing role/permission assignments and access-control settings for sensitive
- 7421 data.
- 7422 • [LOG] Audit or event logs capturing successful authorized access and rejected Unauthorized attempts, where
- 7423 such logging is implemented.
- 7424 • [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating
- 7425 authorized access succeeding and Unauthorized access failing.
- 7426 • [SCOPE] Evidence identifying which CRS components and data stores were considered in-scope for sensitive
- 7427 data.

7428 6.4.1.12.2 Assessment Case AC-CRS-CONF-002

7429 **1. Assessment Reference**

7430 Requirement: REQ-CRS-CONF-002 (Elevated)

7431 **2. Assessment Objective**

7432 Verify that:

7433 1) The CRS supports encryption of sensitive data at rest, including container images, configuration data, stored
7434 secrets, and security-relevant logs, either via native capabilities or by integrating with encrypted storage,
7435 registries, or secret-management services.

7436 2) When configured, encryption is effectively applied to the relevant sensitive data repositories.

7437 3) Encryption keys used for data-at-rest protection are stored and managed in a manner that prevents access by
7438 container workloads and other unprivileged components.

7439 **3. Assessment Preparation**

7440 The assessment shall have access to:

7441 • Documentation describing:

7442 - the CRS model for encrypting sensitive data at rest (native encryption, use of encrypted
7443 storage/registries, integration with secret-management services);

7444 - which data types and repositories are covered;

7445 - supported key-management approaches (for example internal key store, external KMS, platform
7446 services);

7447 - how encryption and key management are configured and monitored;

7448 - restrictions that prevent container workloads and other unprivileged components from accessing keys.

7449 • A test configuration with:

7450 - one or more CRS data stores (for example image registry, configuration store, secret store, log store)
7451 configured to use the documented data-at-rest protection mechanisms;

7452 - an administrative account authorized to configure and monitor encryption and key management;

7453 - one or more container workloads and a non-privileged host context for attempting access to encrypted
7454 data and keys;

7455 - where applicable, access to underlying storage or diagnostic tools to observe whether data is stored
7456 encrypted.

7457 **4. Assessment Activities**

7458 The assessment shall at least:

7459 1) Review documentation and configuration to identify:

7460 - which sensitive data categories and storage locations are expected to be protected at rest;

7461 - how encryption is enabled and verified for each type (for example flags in CRS configuration, storage-
7462 layer encryption indicators);

7463 - where encryption keys are stored and how access to them is restricted.

7464 2) With data-at-rest protection enabled according to product guidance:

7465 - create or update representative container images, configuration objects, stored secrets, and security logs;

7466 - use platform or storage diagnostics (for example tools showing block- or file-level contents, provider
7467 encryption status APIs) to verify that these items are stored in encrypted form and not in recoverable
7468 plaintext;

- 7469 - verify that disabling or revoking access to encryption keys prevents normal decryption and access to the
7470 protected data as expected.
- 7471 3) From container workloads and other non-privileged contexts:
- 7472 - attempt to access encryption keys or key stores via CRS APIs, environment variables, mounted volumes,
7473 or host-level resources;
- 7474 - verify that such attempts are denied and that key material is not readable or exportable by these contexts.
- 7475 4) Where external or platform key-management services are used:
- 7476 - confirm via their interfaces that keys used for CRS data-at-rest protection reside in those services;
- 7477 - verify that the CRS only obtains derived or wrapped keys as documented, and does not expose raw key
7478 material to workloads.

7479 5. Assessment Verdict

- 7480 • Pass: the CRS supports and correctly applies data-at-rest protection to the configured categories of sensitive
7481 data; protected data is not stored in plaintext on underlying storage; and encryption keys are stored and
7482 managed in a way that prevents access by container workloads and other unprivileged components.
- 7483 • Fail: sensitive data configured to be protected at rest is stored unencrypted or in recoverable plaintext; keys are
7484 accessible to workloads or non-privileged contexts; or behaviour diverges from the documented data-at-rest
7485 and key-management model.

7486 6. Assessment Evidence

- 7487 • [DESIGN] Documentation of the CRS data-at-rest encryption model, including covered data types and key-
7488 management design.
- 7489 • [CONFIG] CRS configuration showing enabled encryption settings for image, configuration, secret, and log
7490 stores, and key-management configuration.
- 7491 • [LOG] Events or audit trails related to enabling/disabling encryption, key rotation, and failed access attempts
7492 to protected data or keys.
- 7493 • [OBSERVATION] Test records and storage or provider-tool outputs demonstrating that data is stored
7494 encrypted and is inaccessible when keys are revoked.
- 7495 • [PLT-CAP] Evidence of platform, storage, registry, or secret-management capabilities used to provide data-at-
7496 rest encryption.

7497 6.4.1.12.3 Assessment Case AC-CRS-CONF-003

7498 1. Assessment Reference

7499 Requirement: REQ-CRS-CONF-003 (Elevated)

7500 2. Assessment Objective

7501 Verify that:

- 7502 1) The CRS supports the use of encrypted communication for CRS management and control interfaces, for
7503 communication with image registries, and for other CRS APIs that handle sensitive data or control operations.
- 7504 2) When these communications are configured to use encrypted transport, network traffic is protected with
7505 appropriate cryptographic mechanisms that ensure confidentiality, integrity, and peer authentication.
- 7506 3) The CRS prevents fallback or downgrade from configured encrypted transport to unencrypted or weakly
7507 protected channels.

7508 **3. Assessment Preparation**

7509 The assessment shall have access to:

- 7510 • Documentation describing:
- 7511 - CRS management and control interfaces and APIs that handle sensitive data or control operations;
- 7512 - communication paths between the CRS and image registries, secret stores, or other services handling
- 7513 sensitive data;
- 7514 - supported encrypted transport mechanisms (for example TLS configurations, integration with service
- 7515 meshes or platform-provided secure channels);
- 7516 - how anti-fallback or anti-downgrade behaviour is implemented and configured.
- 7517 • A test configuration with:
- 7518 - CRS management and control interfaces reachable over the network;
- 7519 - a configured image registry and any other CRS APIs that handle sensitive data;
- 7520 - valid certificates or other credentials for establishing encrypted channels as documented;
- 7521 - tools to capture and inspect network traffic on relevant links.

7522 **4. Assessment Activities**

7523 The assessment shall at least:

- 7524 1) Review documentation and CRS configuration to identify:
- 7525 - which CRS interfaces and API endpoints are in scope for encrypted communication;
- 7526 - the recommended or required secure transport protocols and cipher configurations;
- 7527 - any configuration settings that enforce or relax requirements for encryption.
- 7528 2) Configure encrypted transport for:
- 7529 - at least one CRS management or control interface;
- 7530 - communication between the CRS and a configured image registry;
- 7531 - - at least one additional CRS API that handles sensitive data or control operations.
- 7532 For each case, initiate typical management, registry, or API operations while capturing network traffic, and
- 7533 verify that:
- 7534 - payload data is encrypted and not visible in clear text on the wire;
- 7535 - negotiated cryptographic parameters conform to the configuration and are not weaker than documented.
- 7536 3) Attempt to force downgrade or fallback by:
- 7537 - misconfiguring clients to request unencrypted protocols;
- 7538 - proposing weak or deprecated cipher suites if the protocol allows negotiation;
- 7539 - modifying CRS configuration, where possible, to request or allow unencrypted access despite a
- 7540 requirement for encryption.
- 7541 Verify that, when encrypted transport is mandated, the CRS does not silently fall back to unencrypted or
- 7542 weakly protected channels and instead either maintains encrypted communication or fails the connection in a
- 7543 documented way.

7544 **5. Assessment Verdict**

- 7545 • Pass: CRS management/control interfaces, registry communication, and relevant APIs can be configured to
7546 use encrypted transport; captured traffic for these configured channels is encrypted and uses acceptable
7547 cryptographic parameters; and attempts to downgrade to unencrypted or weakly protected channels are
7548 prevented or cause connection failure as documented.
- 7549 • Fail: encrypted transport cannot be enabled as documented; traffic configured to be encrypted is observed in
7550 plaintext or with weaker protection than configured; or the CRS silently falls back to unencrypted or weakly
7551 protected channels under downgrade attempts.

7552 **6. Assessment Evidence**

- 7553 • [DESIGN] Documentation of secure communication mechanisms for CRS management, registry access, and
7554 sensitive APIs, including downgrade-prevention behaviour.
- 7555 • [CONFIG] CRS configuration showing enabled encrypted transport settings and certificate or credential
7556 configuration.
- 7557 • [LOG] Events or audit records relating to secure channel establishment, certificate or key errors, and rejected
7558 downgraded connections.
- 7559 • [OBSERVATION] Packet captures and analysis outputs demonstrating encrypted traffic for configured
7560 channels and failed or rejected downgrade attempts.
- 7561 • [PLT-CAP] Evidence of platform or service-mesh capabilities used to provide secure communication, where
7562 applicable.

7563 **6.4.1.12.4 Assessment Case AC-CRS-CONF-004**7564 **1. Assessment Reference**

7565 Requirement: REQ-CRS-CONF-004 (Advanced)

7566 **2. Assessment Objective**7567 Verify that, where the underlying platform provides mechanisms for isolated or protected execution environments and
7568 this capability is used:

- 7569 1) The CRS supports provisioning and management of execution environments for containerized workloads in
7570 which confidential data associated with those workloads is protected, during execution, from observation,
7571 inspection, or access through the CRS, the host operating system, and other workloads.
- 7572 2) The CRS provides configuration and API interfaces to deploy workloads into such protected execution
7573 environments.
- 7574 3) The CRS supports secure provisioning of secrets and key material to those workloads such that the secrets and
7575 key material are not accessible to the CRS itself, the host operating system, or other workloads.

7576 **3. Assessment Preparation**

7577 The assessment shall have access to:

- 7578 • Documentation describing:
- 7579 - which confidential-execution mechanisms are supported (for example trusted execution environments,
7580 confidential containers, hardware-backed isolation mechanisms);
 - 7581 - how the CRS exposes configuration and API interfaces to deploy workloads into these environments;
 - 7582 - how secrets and key material are provisioned to protected workloads and which entities can access them;
 - 7583 - any limitations on CRS or host visibility into confidential workload memory and execution state.
- 7584 • A test configuration with:
- 7585 - a platform providing the documented confidential-execution capabilities;

- 7586 - the CRS configured to enable and manage confidential or protected workloads as described;
- 7587 - at least one containerized workload deployable both as a normal container and as a confidential/protected
- 7588 instance;
- 7589 - CRS, host, and workload-level tools that would normally be used to inspect container state, logs, or
- 7590 memory.

7591 **4. Assessment Activities**

7592 The assessment shall at least:

- 7593 1) Review documentation and CRS configuration to identify:
 - 7594 - how an operator selects confidential or protected execution for a workload (for example annotations,
 - 7595 profiles, or API parameters);
 - 7596 - which platform features are used to enforce additional confidentiality;
 - 7597 - how secret and key provisioning is designed to avoid exposure to CRS or host components.
- 7598 2) Deploy the test workload as a normal container and, using standard CRS and host tools:
 - 7599 - confirm that the workload's memory, process state, and secrets (for example environment variables,
 - 7600 mounted secret volumes) are accessible as documented for non-confidential workloads.
- 7601 3) Deploy the equivalent workload into a confidential or protected execution environment via CRS configuration
- 7602 or APIs and:
 - 7603 - verify via platform or CRS indicators that the workload is running in the protected mode;
 - 7604 - attempt to access workload memory, process state, or confidential data using CRS and host inspection
 - 7605 tools;
 - 7606 - verify that direct inspection of confidential memory or secrets is blocked or that retrieved data is
 - 7607 unintelligible and consistent with the confidential-execution model.
- 7608 4) For secrets and key material provisioned to the confidential workload:
 - 7609 - examine CRS and host configurations to confirm that secrets are delivered through mechanisms that do
 - 7610 not expose their plaintext to CRS control-plane components or to other workloads;
 - 7611 - attempt to read these secrets from CRS logs, configuration stores, or unrelated workloads;
 - 7612 - verify that such attempts fail or reveal only non-sensitive metadata.

7613 **5. Assessment Verdict**

- 7614 • Pass: the CRS can deploy workloads into platform-provided confidential or protected execution environments;
- 7615 in such environments, confidential workload data in use cannot be observed or accessed via normal CRS or
- 7616 host inspection mechanisms; and secrets and key material are provisioned in a way that avoids exposure to
- 7617 CRS and other workloads, consistent with the documented model.
- 7618 • Fail: workloads deployed as confidential can still be inspected in clear text by CRS or host tools; secrets or
- 7619 keys intended to be confined to protected workloads are visible to CRS or other workloads; or documented
- 7620 confidential-execution behaviour is not achieved in practice.

7621 **6. Assessment Evidence**

- 7622 • [DESIGN] Documentation of the CRS integration with platform confidential-execution mechanisms and the
- 7623 secret/key provisioning model for protected workloads.
- 7624 • [CONFIG] CRS and platform configuration showing how confidential workloads are enabled and identified,
- 7625 and how secret delivery is configured.
- 7626 • [LOG] Events and audit records related to creation, configuration, and failure of confidential workloads,
- 7627 including any attempted misuse of confidential-execution options.

- 7628 • [OBSERVATION] Test records and screenshots showing normal versus confidential workload behaviour,
7629 including failed attempts to inspect confidential memory or secrets.
- 7630 • [PLT-CAP] Evidence of platform confidential-computing capabilities used by the CRS.

7631 6.4.1.12.5 Assessment Case AC-CRS-CONF-005

7632 1. Assessment Reference

7633 Requirement: REQ-CRS-CONF-005 (Advanced, optional) - Hardware-backed confidentiality

7634 2. Assessment Objective

7635 Verify that, where claimed and where hardware-based or platform-provided mechanisms are available:

- 7636 1) The CRS supports integration with such mechanisms to strengthen protection of cryptographic keys and
7637 confidential execution state used for data-at-rest, data-in-transit, and data-in-use protection.
- 7638 2) These mechanisms (for example dedicated secure hardware modules, memory-encryption features, or
7639 equivalent platform components) are used in a way that provides protected storage for keys and resistance
7640 against Unauthorized access to confidential workload data, consistent with the product documentation.

7641 3. Assessment Preparation

7642 The assessment shall have access to:

- 7643 • Documentation describing:
 - 7644 - which hardware-based or platform-provided mechanisms are supported (for example TPMs, HSMs,
7645 secure enclaves, memory-encryption engines);
 - 7646 - how the CRS integrates with these mechanisms for key storage and management and for protection of
7647 confidential execution state;
 - 7648 - configuration and operational procedures for enabling and monitoring hardware-backed confidentiality.
- 7649 • A test configuration with:
 - 7650 - a platform that provides the claimed hardware or platform security mechanisms;
 - 7651 - the CRS configured to use these mechanisms according to product documentation;
 - 7652 - tools or interfaces to query the hardware modules or platform features (for example vendor utilities, OS
7653 interfaces, or CRS management commands);
 - 7654 - container workloads and non-privileged host contexts that can be used to attempt access to keys or
7655 protected confidential state.

7656 4. Assessment Activities

7657 The assessment shall at least:

- 7658 1) Review documentation and CRS configuration to identify:
 - 7659 - which keys and confidential state are intended to be protected by hardware-backed mechanisms (for
7660 example master keys for storage encryption, keys for secure channels, protected execution state);
 - 7661 - which hardware modules or platform features are used and how they are bound to the CRS deployment;
 - 7662 - documented behaviour if hardware support is not present or becomes unavailable.
- 7663 2) On a platform with the relevant mechanisms present and enabled:
 - 7664 - configure the CRS to use these mechanisms for key storage and/or confidential-state protection as
7665 described;
 - 7666 - use platform or hardware tools to confirm that key material or protected state is stored or managed by the
7667 hardware module (for example key handles in a TPM/HSM, memory-encryption status indicators);

- 7668 - verify that the CRS does not expose the corresponding raw key material to workloads or non-privileged
7669 host contexts.
- 7670 3) From container workloads and other non-privileged contexts:
- 7671 - attempt to access hardware-protected keys or confidential state through normal system interfaces or CRS
7672 APIs;
- 7673 - verify that such attempts are rejected or only yield non-extractable handles consistent with the hardware
7674 security model.
- 7675 4) Where possible, simulate absence or failure of the hardware mechanism and verify that the CRS behaviour
7676 matches documentation (for example disabling certain confidentiality features, refusing to start protected
7677 workloads, or reporting an error condition).

7678 5. Assessment Verdict

- 7679 • Pass: when hardware-backed confidentiality is claimed and mechanisms are available, the CRS correctly
7680 integrates with the documented hardware or platform features to protect cryptographic keys and confidential
7681 state; these assets are not exposed in raw form to container workloads or non-privileged contexts; and fallback
7682 or error behaviour matches the documented model.
- 7683 • Fail: claimed hardware-backed mechanisms are not actually used; keys or confidential state intended to be
7684 hardware-protected are exposed to software-only contexts or to workloads; or the CRS behaviour in the
7685 absence or failure of hardware support diverges from the documentation.

7686 6. Assessment Evidence

- 7687 • [DESIGN] Documentation of CRS integration with hardware-backed or platform-provided confidentiality
7688 mechanisms, including key and state protection models.
- 7689 • [CONFIG] CRS and platform configuration showing use of hardware modules or features for confidentiality
7690 protection.
- 7691 • [LOG] Events related to initialization, use, and failure of hardware-backed mechanisms, including any
7692 reported errors when such mechanisms are unavailable.
- 7693 • [OBSERVATION] Test records and tool outputs demonstrating that keys and confidential state are managed
7694 by hardware or platform security mechanisms and not exposed to workloads.
- 7695 • [PLT-CAP] Evidence of the underlying hardware or platform security capabilities and their configuration.

7696 6.4.1.13 Assessment for Availability and Resilience

7697 6.4.1.13.1 Assessment Case AC-CRS-AVAIL-001

7698 1. Assessment Reference

7699 Requirement: REQ-CRS-AVAIL-001 (Basic)

7700 2. Assessment Objective

7701 Verify that:

- 7702 1) The CRS maintains resource isolation between container workloads and protects CRS control-plane processes
7703 from CPU and memory exhaustion caused by container workloads.
- 7704 2) Failure or abnormal termination of an individual container is contained within its execution context and, under
7705 normal conditions and configured limits, does not compromise CRS control-plane operation or cause other
7706 containers to terminate unexpectedly.

7707 3. Assessment Preparation

7708 The assessment shall have access to:

- 7709 • Documentation of CRS resource isolation mechanisms (namespaces, cgroups/quotas, limits) and how they
7710 apply to CPU and memory.

- 7711 • Documentation of CRS control-plane components and their protection from workload resource use.
- 7712 • Documentation of how container failures are handled (restart policy, isolation, impact).
- 7713 • A test configuration with multiple containers, including at least one container capable of generating high load
- 7714 and one or more normal workloads.

7715 4. Assessment Activities

7716 The assessment shall at least:

- 7717 1) Confirm that CPU and memory limits or equivalent isolation mechanisms can be configured per container or
- 7718 workload, and that CRS control-plane processes are not scheduled within the same resource domain as
- 7719 arbitrary workloads.
- 7720 2) Configure a container with aggressive CPU/memory usage within defined limits, generate high load, and
- 7721 verify that CRS control-plane functions (e.g. scheduling, API server, management operations) remain
- 7722 operational and responsive.
- 7723 3) Induce abnormal termination of a selected container (e.g. crash or kill) and verify that:
- 7724 - the CRS control plane remains operational; and
- 7725 - other containers continue to run unaffected under normal operating conditions and configured limits.

7726 5. Assessment Verdict

- 7727 • Pass: CPU and memory isolation mechanisms protect CRS control-plane processes from workload-induced
- 7728 exhaustion; and failure of an individual container does not compromise CRS control-plane operation or trigger
- 7729 unexpected termination of other containers under normal conditions.
- 7730 • Fail: control-plane processes are starved or disrupted by workload resource usage; or a single container failure
- 7731 propagates to the control plane or causes other containers to terminate unexpectedly.

7732 6. Assessment Evidence

- 7733 • [CONFIG] Resource isolation (limits/quotas) configuration for containers and control-plane components.
- 7734 • [DESIGN] Documentation of CRS resource isolation and failure-handling architecture.
- 7735 • [LOG] / [OBSERVATION] Evidence showing control-plane availability under workload stress and
- 7736 containment of individual container failures.

7737 6.4.1.13.2 Assessment Case AC-CRS-AVAIL-002

7738 1. Assessment Reference

7739 Requirement: REQ-CRS-AVAIL-002 (Elevated)

7740 2. Assessment Objective

7741 Verify that:

- 7742 1) The CRS provides automatic recovery mechanisms to restart failed containers based on defined policies,
- 7743 without manual intervention.
- 7744 2) Where supported by the deployment environment, these mechanisms can be used with orchestration or
- 7745 clustering systems to restore workloads on the same or an alternative host.

7746 3. Assessment Preparation

7747 The assessment shall have access to:

- 7748 • Documentation of CRS container restart policies and automatic recovery behaviour (e.g. "always", "on-
- 7749 failure", back-off behaviour).
- 7750 • Documentation of integration with orchestration or clustering systems, where supported (e.g. Kubernetes,
- 7751 Swarm, other orchestrators).

- 7752 • A test environment with several containers, and where supported, at least two hosts under common
- 7753 orchestration/cluster control.

7754 4. Assessment Activities

7755 The assessment shall at least:

- 7756 1) Configure one or more containers with automatic restart policies as described in documentation.
- 7757 2) Induce container failure (e.g. crash, forced termination) and verify that the CRS automatically restarts the
- 7758 container according to the defined policy, without manual actions.
- 7759 3) Where orchestration/clustering is supported, simulate host or node unavailability for a container workload and
- 7760 verify that the orchestration/cluster in conjunction with CRS mechanisms restores the workload on the same or
- 7761 an alternative host according to the configuration.

7762 5. Assessment Verdict

- 7763 • Pass: failed containers configured with restart policies are automatically restarted without manual intervention;
- 7764 and, where supported, orchestration/clustering combined with CRS behaviour restores workloads on the same
- 7765 or another host as documented.
- 7766 • Fail: container restart requires manual intervention in contradiction with configured policies; or supported
- 7767 orchestration/clustering cannot restore workloads as intended.

7768 6. Assessment Evidence

- 7769 • [CONFIG] Container restart policy configuration and, where applicable, orchestration/cluster configuration.
- 7770 • [DESIGN] Documentation of CRS automatic recovery mechanisms and integrations with
- 7771 orchestration/clustering systems.
- 7772 • [LOG] / [OBSERVATION] Evidence of automatic container restarts and, where applicable, restoration of
- 7773 workloads on alternative hosts.

7774 6.4.1.13.3 Assessment Case AC-CRS-AVAIL-003

7775 1. Assessment Reference

7776 Requirement: REQ-CRS-AVAIL-003 (Elevated)

7777 2. Assessment Objective

7778 Verify that:

- 7779 1) The CRS implements resource scheduling or reservation mechanisms that allow configuration of minimum
- 7780 resource allocations for CRS security and management functions (CPU, memory and I/O).
- 7781 2) These mechanisms protect the CRS control plane from resource starvation caused by container workloads or
- 7782 external contention, ensuring that security and management functions remain able to operate under load.

7783 3. Assessment Preparation

7784 The assessment shall have access to:

- 7785 • Documentation of CRS resource scheduling/reservation or priority mechanisms (e.g. control-plane
- 7786 reservations, priorities, QoS classes).
- 7787 • Documentation describing how minimum CPU, memory and I/O resources can be reserved or prioritized for
- 7788 CRS security and management components.
- 7789 • A test configuration with CRS security/management functions active and multiple container workloads
- 7790 capable of generating high CPU, memory and I/O load.

7791 4. Assessment Activities

7792 The assessment shall at least:

- 7793 1) Confirm that resource scheduling or reservation mechanisms exist and that minimum CPU, memory and I/O
- 7794 allocations (or equivalent prioritization) can be configured for CRS security and management functions.

- 7795 2) Configure such minimum allocations/priorities in the assessed environment.
- 7796 3) Generate high resource load from containers and/or external sources (CPU- and I/O-intensive workloads) and
7797 verify that:
- 7798 - CRS management and security interfaces remain responsive; and
- 7799 - critical control-plane operations (e.g. container lifecycle operations, logging, policy enforcement)
7800 continue to function correctly under load.

7801 5. Assessment Verdict

- 7802 • Pass: resource scheduling/reservation mechanisms support configuration of minimum CPU, memory and I/O
7803 allocations (or equivalent prioritization) for CRS security and management functions; and under high load
7804 from container workloads or external contention, these functions remain able to operate as intended.
- 7805 • Fail: minimum allocations/priorities for security/management functions cannot be configured; or under load,
7806 the CRS control plane or security/management functions become starved and unable to operate.

7807 6. Assessment Evidence

- 7808 • [CONFIG] Resource scheduling/reservation or prioritization configuration for CRS security and management
7809 components.
- 7810 • [DESIGN] Documentation of CRS resource scheduling architecture and its application to control-plane
7811 protection.
- 7812 • [LOG] / [OBSERVATION] Evidence that CRS management and security functions continue to operate
7813 correctly under stress conditions.

7814 6.4.1.13.4 Assessment Case AC-CRS-AVAIL-004

7815 1. Assessment Reference

7816 Requirement: REQ-CRS-AVAIL-004 (Advanced)

7817 2. Assessment Objective

7818 Verify that:

- 7819 1) Where the deployment environment supports workload relocation, the CRS supports checkpoint and restore of
7820 the runtime state of selected container workloads, including process state and relevant in-memory execution
7821 context.
- 7822 2) This checkpoint/restore capability can be used, together with orchestration or clustering mechanisms, to enable
7823 failover with minimal disruption to workload execution when relocating to a standby instance or alternative
7824 host.

7825 3. Assessment Preparation

7826 The assessment shall have access to:

- 7827 • Documentation of CRS checkpoint/restore mechanisms for container workloads (scope of state, limitations,
7828 required environment support).
- 7829 • Documentation of how orchestration or clustering systems can coordinate relocation and failover using
7830 checkpoint/restore.
- 7831 • A test environment where workload relocation is supported (e.g. compatible hosts, shared or consistent
7832 storage/network configuration) and at least one container workload configured for checkpoint/restore and
7833 relocation.

7834 4. Assessment Activities

7835 The assessment shall at least:

- 7836 1) Configure a selected container workload to use CRS checkpoint/restore as documented, including any required
7837 integration with the orchestrator or cluster.

- 7838 2) Create a checkpoint of the running workload and verify that process state and relevant in-memory execution
7839 context are captured according to the CRS documentation.
- 7840 3) Simulate a control-plane or host-level failure (or initiate a planned relocation) and verify that the workload is
7841 restored from checkpoint on a standby instance or alternative host using orchestrator/cluster coordination.
- 7842 4) Observe workload behaviour and confirm that disruption is limited to the minimal interruption inherent to the
7843 checkpoint/restore and relocation process, rather than a full re-deployment from scratch.

7844 5. Assessment Verdict

- 7845 • Pass: checkpoint and restore of runtime state for selected container workloads is supported; when combined
7846 with orchestration/clustering, the mechanism enables relocation/failover to a standby or alternative host with
7847 minimal disruption to workload execution as documented.
- 7848 • Fail: checkpoint/restore of runtime state cannot be configured or does not function as specified; or
7849 relocation/failover requires full restart without using checkpointed state; or workload disruption is inconsistent
7850 with the intended "minimal disruption" behaviour.

7851 6. Assessment Evidence

- 7852 • [CONFIG] Checkpoint/restore and relocation configuration for selected container workloads, including any
7853 orchestrator/cluster settings.
- 7854 • [DESIGN] Documentation of CRS checkpoint/restore and relocation architecture, including scope of captured
7855 state and expected failover behaviour.
- 7856 • [LOG] / [OBSERVATION] Evidence of checkpoint creation, restore operations and observed workload
7857 behaviour during and after relocation/failover.

7858 6.4.1.14 Assessment for Logging

7859 6.4.1.14.1 Assessment Case AC-CRS-LOG-001

7860 1. Assessment Reference

7861 Requirement: REQ-CRS-LOG-001 (Basic)

7862 2. Assessment Objective

7863 Verify that:

- 7864 1) The CRS generates audit logs for security-relevant administrative actions and security events, including at
7865 least:
- 7866 - authentication attempts to CRS administrative or control interfaces;
- 7867 - configuration changes affecting CRS behaviour or security posture;
- 7868 - container lifecycle operations (create, start, stop, delete, restart) initiated through the CRS.
- 7869 2) Audit logs are not writable or directly modifiable by container workloads or other unprivileged processes
7870 running within container namespaces.

7871 3. Assessment Preparation

7872 The assessment shall have access to:

- 7873 • Documentation of CRS logging/auditing capabilities and log storage locations.
- 7874 • Documentation of which actions/events are treated as security-relevant.
- 7875 • A configuration with:
- 7876 - CRS administrative/control access enabled;
- 7877 - at least one container workload that can be created, started, stopped, restarted and deleted.

7878 **4. Assessment Activities**

7879 The assessment shall at least:

- 7880 1) Identify audit log sources and storage locations from documentation/configuration.
- 7881 2) In the assessed configuration, perform:
- 7882 - successful and failed authentication attempts to CRS administrative/control interfaces;
 - 7883 - CRS configuration changes affecting behaviour/security posture;
 - 7884 - container lifecycle operations (creation, start, stop, restart, deletion) via CRS mechanisms.
- 7885 Verify that corresponding audit records are generated, including at least action/event type, affected
- 7886 object, timestamp and responsible account or process where applicable.
- 7887 3) From a container workload and from another unprivileged process, attempt to write to, modify or delete the
- 7888 CRS audit log files or endpoints and verify that these operations are denied.

7889 **5. Assessment Verdict**

- 7890 • Pass: the CRS generates audit logs for the specified security-relevant actions/events; and audit logs cannot be
- 7891 written or directly modified by container workloads or other unprivileged processes in container namespaces.
- 7892 • Fail: one or more required events are not logged; or container workloads/unprivileged processes can write or
- 7893 modify audit logs.

7894 **6. Assessment Evidence**

- 7895 • [CONFIG] CRS logging configuration and log storage settings.
- 7896 • [DESIGN] Documentation of audit logging behaviour, logged event categories and log isolation from
- 7897 workloads.
- 7898 • [LOG] / [OBSERVATION] Sample audit entries for authentication attempts, configuration changes and
- 7899 container lifecycle operations, and evidence of denied write/modify attempts from containers/unprivileged
- 7900 processes.

7901 **6.4.1.14.2 Assessment Case AC-CRS-LOG-002**

7902 **1. Assessment Reference**

7903 Requirement: REQ-CRS-LOG-002 (Elevated)

7904 **2. Assessment Objective**

7905 Verify that:

- 7906 1) CRS audit logs are protected from Unauthorized access, modification or deletion.
- 7907 2) Access to CRS audit logs is restricted to authorized administrative roles or trusted system processes under
- 7908 CRS or host OS control.
- 7909 3) The CRS supports secure export of audit logs to external log management or SIEM systems using
- 7910 authenticated communication channels.

7911 **3. Assessment Preparation**

7912 The assessment shall have access to:

- 7913 • Documentation on audit log storage (files, services, APIs) and access-control mechanisms (roles, permissions,
- 7914 OS controls).
- 7915 • Documentation identifying which administrative roles and trusted system processes are allowed to
- 7916 access/manage CRS audit logs.
- 7917 • Documentation of CRS log export mechanisms and supported authenticated transport options.

- 7918 • A configuration with audit logging enabled and a reachable external log management/SIEM endpoint (test
- 7919 instance) that can accept logs over an authenticated channel.

7920 4. Assessment Activities

7921 The assessment shall at least:

- 7922 1) Review configuration and underlying platform permissions to confirm that audit log storage can only be
- 7923 accessed by authorized administrative roles or trusted system processes, and that modification/deletion by
- 7924 other users/processes is prevented.
- 7925 2) Attempt to read, modify or delete audit logs using non-authorized accounts or processes and verify that these
- 7926 attempts are denied.
- 7927 3) Configure export of CRS audit logs to the external log/SIEM endpoint using an authenticated communication
- 7928 channel as documented (e.g. authenticated TLS, authenticated agent protocol).
- 7929 4) Verify, using configuration and where feasible protocol inspection, that:
 - 7930 - the log export uses authenticated communication; and
 - 7931 - the external system receives log records corresponding to those generated locally.

7932 5. Assessment Verdict

- 7933 • Pass: only authorized administrative roles/trusted processes can access CRS audit logs; Unauthorized
- 7934 access/modification/deletion is prevented; and CRS audit logs can be exported to external systems over
- 7935 authenticated channels.
- 7936 • Fail: Unauthorized users/processes can access or tamper with audit logs; or access controls on audit logs are
- 7937 insufficient; or log export does not use authenticated channels or cannot be securely configured.

7938 6. Assessment Evidence

- 7939 • [CONFIG] Audit log storage permissions, role/process mappings and export configuration, including
- 7940 authentication settings.
- 7941 • [DESIGN] Documentation of CRS audit log protection model and external export mechanisms.
- 7942 • [LOG] / [OBSERVATION] Evidence of denied access attempts by non-authorized entities and successful
- 7943 authenticated export to an external log/SIEM system.

7944 6.4.1.14.3 Assessment Case AC-CRS-LOG-003

7945 1. Assessment Reference

7946 Requirement: REQ-CRS-LOG-003 (Advanced)

7947 2. Assessment Objective

7948 Verify that:

- 7949 1) The CRS supports cryptographic protection and trusted time-stamping of audit logs so that origin and integrity
- 7950 of audit records can be verified during forensic analysis.
- 7951 2) Where external time or signing services are used, audit records include sufficient metadata to identify and
- 7952 verify the trust source used for time-stamping and integrity protection.

7953 3. Assessment Preparation

7954 The assessment shall have access to:

- 7955 • Documentation describing cryptographic mechanisms used to protect CRS audit logs (e.g. signatures, MACs,
- 7956 hash chains or log sealing), including how verification is performed.
- 7957 • Documentation of trusted time-stamping (e.g. internal secure time source, external time service, signing
- 7958 service) and the metadata recorded in audit logs about the trust source.

- 7959 • A configuration with cryptographic log protection and trusted time-stamping enabled, with the ability to export
7960 or access raw audit log data and any associated metadata for verification.

7961 **4. Assessment Activities**

7962 The assessment shall at least:

- 7963 1) Enable cryptographic protection and trusted time-stamping for CRS audit logs as documented.
- 7964 2) Generate a set of representative audit events (e.g. logins, configuration changes, container lifecycle operations)
7965 and collect the corresponding audit records and associated metadata.
- 7966 3) Using the documented verification tools or procedures, verify that:
 - 7967 - cryptographic checks over the audit records succeed, demonstrating integrity and origin as designed;
 - 7968 - each audit record includes a trusted timestamp; and
 - 7969 - where external time or signing services are used, the log metadata clearly identifies the trust source (e.g.
7970 service identifier, certificate, or similar) and allows verification against that source.
- 7971 4) Modify a copy of the audit log (e.g. change or remove records) and confirm that cryptographic and/or time-
7972 stamp verification fails or indicates tampering.

7973 **5. Assessment Verdict**

- 7974 • Pass: cryptographic mechanisms and trusted time-stamping for CRS audit logs are available; audit records
7975 contain sufficient metadata to verify the origin, integrity and trust source; and tampering with logs is
7976 detectable through the provided verification process.
- 7977 • Fail: cryptographic protection or trusted time-stamping cannot be enabled; audit records lack metadata needed
7978 to verify the trust source; or modifications to audit logs cannot be reliably detected.

7979 **6. Assessment Evidence**

- 7980 • [CONFIG] Configuration enabling cryptographic protection and trusted time-stamping of CRS audit logs,
7981 including any external signing/time service integration.
- 7982 • [DESIGN] Documentation of CRS log protection and time-stamping architecture, including metadata format
7983 and verification procedures.
- 7984 • [LOG] / [OBSERVATION] Audit logs and verification outputs showing successful checks for unmodified
7985 logs and detection of tampering and/or invalid trust source.

7986 **6.4.1.15 Assessment for Patches and Updates**

7987 **6.4.1.15.1 Assessment Case AC-CRS-UPD-001**

7988 **1. Assessment Reference**

7989 Requirement: REQ-CRS-UPD-001 (Basic)

7990 **2. Assessment Objective**

7991 Verify that:

- 7992 1) The CRS supports applying security patches and updates to CRS components without a full reinstallation of
7993 the CRS.
- 7994 2) The CRS can be updated without requiring a full reinstallation of the host system.
- 7995 3) Applicable update mechanisms in the evaluated solution (e.g. host package manager, management appliance,
7996 orchestration pipeline) can update CRS binaries and related components in-place.

7997 **3. Assessment Preparation**

7998 The assessment shall have access to:

- 7999 • Documentation of the CRS update mechanisms used in the evaluated solution (host package manager,
8000 management appliance, orchestration/CI pipeline, image-based updates).
- 8001 • Documentation indicating which binaries and components are considered CRS components in scope.
- 8002 • A test environment where a newer CRS version or security patch is available for application using the
8003 supported mechanism(s).

8004 **4. Assessment Activities**

8005 The assessment shall at least:

- 8006 1) Identify the mechanism(s) responsible for delivering and installing CRS component updates in the evaluated
8007 solution.
- 8008 2) Apply a CRS update or security patch using the documented mechanism(s) and verify that the update
8009 completes without performing a full CRS or host system reinstallation.
- 8010 3) After the update, verify that:
 - 8011 - the CRS version/build reflects the applied update; and
 - 8012 - existing CRS configuration and container workloads behave according to the product's documented
8013 update behaviour.

8014 **5. Assessment Verdict**

- 8015 • Pass: CRS components are updated through the documented mechanisms without requiring full CRS or host
8016 system reinstallation, and the updated CRS is active afterwards.
- 8017 • Fail: CRS updates require reinstalling the CRS or host system; or CRS components cannot be updated using
8018 the documented product update mechanism.

8019 **6. Assessment Evidence**

- 8020 • [SCOPE] Description of CRS components in scope and the update mechanism(s) used.
- 8021 • [CONFIG] Update mechanism configuration (repositories, appliance/orchestrator settings, image update
8022 policies).
- 8023 • [DESIGN] Documentation of the CRS update model and its relation to host OS/appliance/orchestration
8024 services.
- 8025 • [LOG] / [OBSERVATION] Evidence of a successful in-place CRS update (before/after version, update logs,
8026 continuity of workloads).

8027 **6.4.1.15.2 Assessment Case AC-CRS-UPD-002**

8028 **1. Assessment Reference**

8029 Requirement: REQ-CRS-UPD-002 (Elevated)

8030 **2. Assessment Objective**

8031 Verify that:

- 8032 1) The CRS, or the update mechanisms in the evaluated solution, verify authenticity and integrity of all CRS
8033 patches and updates using digital signatures validated against trusted keys or certificates configured for the
8034 CRS.
- 8035 2) If authenticity or integrity verification fails, installation of the CRS patch or update is prevented and a security-
8036 relevant event is recorded.
- 8037 3) Trusted keys or certificates used for CRS update verification cannot be modified by container workloads or
8038 other unprivileged components during normal operation.

8039 **3. Assessment Preparation**

8040 The assessment shall have access to:

- 8041 • Documentation describing CRS update verification (signature formats, trust anchors, key/certificate
8042 management).
- 8043 • Documentation showing where and how trusted keys/certificates for CRS update verification are stored and
8044 protected.
- 8045 • Ability to obtain:
 - 8046 - a valid, correctly signed CRS update; and
 - 8047 - an update artefact with invalid/missing signatures or with intentional modification to simulate
8048 verification failure.

8049 **4. Assessment Activities**

8050 The assessment shall at least:

- 8051 1) Apply a valid signed CRS update using the documented mechanism and verify that signature/integrity
8052 verification is performed and the update is accepted and installed.
- 8053 2) Attempt to apply an invalid or tampered CRS update (e.g. corrupted or unsigned package) and verify that:
 - 8054 - authenticity/integrity verification fails;
 - 8055 - the update is not installed; and
 - 8056 - a security-relevant event is recorded in the CRS or host audit logs.
- 8057 3) Review configuration and underlying platform controls for the trusted keys/certificates used in CRS update
8058 verification and verify that container workloads or other unprivileged components cannot modify or replace
8059 these trust anchors under normal operation (e.g. by file permissions, store isolation, or hardware protection).

8060 **5. Assessment Verdict**

- 8061 • Pass: authenticity and integrity of CRS updates are verified against trusted keys/certificates; updates that fail
8062 verification are not installed and generate a security-relevant log; and trust anchors for CRS update verification
8063 cannot be modified by container workloads or other unprivileged components.
- 8064 • Fail: CRS updates can be installed without effective signature/integrity verification; or failed verification does
8065 not block installation and/or is not logged as a security event; or trusted keys/certificates can be modified by
8066 unprivileged components.

8067 **6. Assessment Evidence**

- 8068 • [CONFIG] CRS update verification configuration, trust store settings and access controls on keys/certificates.
- 8069 • [DESIGN] Documentation of the CRS update verification process and trust model, including how failures are
8070 logged.
- 8071 • [LOG] / [OBSERVATION] Evidence of successful installation of a valid update, blocked installation of a
8072 tampered update, and corresponding security-relevant log entries.
- 8073 • [PLT-CAP] Where relevant, evidence of platform capabilities used to protect trust anchors (e.g. OS keystore,
8074 HSM, read-only image).

8075 6.4.1.15.3 Assessment Case AC-CRS-UPD-003

8076 1. Assessment Reference

8077 Requirement: REQ-CRS-UPD-003 (Advanced)

8078 2. Assessment Objective

8079 Verify that, where version rollback is supported for CRS components:

- 8080 1) The update mechanism implements rollback protection to prevent Unauthorized installation of outdated,
8081 revoked or replayed CRS updates.
- 8082 2) Authorized rollback to a previously trusted CRS version is only permitted when:
 - 8083 - the rollback image/package is verified for integrity and authenticity against a trusted key/certificate;
 - 8084 - the rollback action is explicitly authorized by an administrator with elevated privileges or equivalent
8085 authorized control function; and
 - 8086 - the rollback operation is recorded as a security-relevant event in the audit logs, including the version
8087 reverted from and the version reverted to.

8088 3. Assessment Preparation

8089 The assessment shall have access to:

- 8090 • Documentation of rollback capabilities for CRS components and rollback protection mechanisms (version
8091 tracking, anti-replay, revocation checks).
- 8092 • Documentation of the roles or control functions authorized to perform rollback and of the audit logging of
8093 update/rollback operations.
- 8094 • A test configuration where:
 - 8095 - CRS rollback is supported by the evaluated update mechanism; and
 - 8096 - forward updates and rollback to a previous trusted version can be exercised.

8097 4. Assessment Activities

8098 The assessment shall at least:

- 8099 1) Confirm from documentation/configuration that the update mechanism tracks CRS versions and prevents
8100 installation of outdated, revoked or replayed updates except via an explicit, authorized rollback procedure.
- 8101 2) Attempt to install an older or revoked CRS update package without following the documented rollback
8102 procedure and verify that the mechanism blocks this as Unauthorized rollback/replay.
- 8103 3) Perform an authorized rollback using the documented procedure and verify that:
 - 8104 - the rollback image/package is subject to integrity and authenticity verification before use;
 - 8105 - the rollback requires elevated administrative privileges or equivalent control function; and
 - 8106 - the rollback event is recorded as a security-relevant audit record, including at least the previous CRS
8107 version and the rollback target version.
- 8108 4) After rollback, verify that the CRS runs the expected previous trusted version and that normal operation is
8109 restored according to product documentation.

8110 5. Assessment Verdict

- 8111 • Pass: Unauthorized installation of outdated/revoked/replayed CRS updates is prevented; and authorized
8112 rollback requires integrity/authenticity verification of the rollback image, elevated authorization, and produces
8113 an audit log recording from/to version.
- 8114 • Fail: outdated or revoked CRS updates can be installed without appropriate controls; or rollback does not
8115 require elevated authorization; or rollback events are not logged with sufficient detail.

8116 6. Assessment Evidence

- 8117 • [CONFIG] CRS update/rollback configuration, including version control, rollback policy and administrative
8118 authorization settings.
- 8119 • [DESIGN] Documentation of rollback protection mechanisms and the authorized rollback process.
- 8120 • [LOG] / [OBSERVATION] Evidence of blocked Unauthorized rollback attempts and of logged authorized
8121 rollback operations, including version information before and after rollback.

8122 6.4.1.16 Assessment for Secure Configuration and Default

8123 6.4.1.16.1 Assessment Case AC-CRS-CFG-001

8124 1. Assessment Reference

8125 Requirement: REQ-CRS-CFG-001 (Basic)

8126 2. Assessment Objective

8127 Verify that:

- 8128 1) By default, the CRS disables interfaces and features not required for container execution and CRS
8129 administration, including remote debugging endpoints, unauthenticated control sockets, and developer
8130 interfaces exposing management or control capabilities.
- 8131 2) Only administrators can explicitly enable such interfaces and features through CRS configuration.

8132 3. Assessment Preparation

8133 The assessment shall have access to:

- 8134 • Documentation listing CRS interfaces/features (remote debugging, control sockets, developer interfaces,
8135 admin APIs/UIs) and identifying those required for normal container execution and CRS administration.
- 8136 • Documentation of CRS configuration mechanisms and role/permission model for modifying configuration.
- 8137 • A fresh/default CRS deployment in its documented default configuration, with at least one non-administrative
8138 account or context (e.g. unprivileged user, container workload).

8139 4. Assessment Activities

8140 The assessment shall at least:

- 8141 1) In the default configuration, attempt to access all documented non-essential interfaces/features (remote
8142 debugging endpoints, unauthenticated control sockets, developer interfaces) and verify they are disabled or not
8143 reachable.
- 8144 2) Confirm that interfaces/features required for container execution and CRS administration are available as
8145 documented.
- 8146 3) Using administrative credentials, enable one or more of the disabled interfaces/features via CRS configuration
8147 and verify that:
 - 8148 - the configuration change is accepted; and
 - 8149 - the corresponding interface/feature becomes available.
- 8150 4) Attempt to perform the same enable operations using a non-administrative account/context and verify that the
8151 configuration change is rejected or not permitted.

8152 5. Assessment Verdict

- 8153 • Pass: non-essential/debug/developer interfaces and unauthenticated control sockets are disabled by default;
8154 required interfaces for execution/administration are available; and only administrators can enable additional
8155 interfaces/features.
- 8156 • Fail: non-essential or insecure interfaces are enabled by default; or non-administrative users can enable such
8157 interfaces/features.

8158 6. Assessment Evidence

- 8159 • [CONFIG] Default and modified CRS configuration showing interface/feature states and permissions required
8160 for changes.
- 8161 • [DESIGN] Documentation of CRS interfaces, default exposure, and configuration/role model.
- 8162 • [OBSERVATION] / [LOG] Connection attempts and CLI/UI output showing:
 - 8163 - non-availability of non-essential interfaces by default;
 - 8164 - successful enable by admin;
 - 8165 - failed enable attempts by non-admin.

8166 6.4.1.16.2 Assessment Case AC-CRS-CFG-002

8167 1. Assessment Reference

8168 Requirement: REQ-CRS-CFG-002 (Elevated)

8169 2. Assessment Objective

8170 Verify that:

- 8171 1) The CRS validates configuration parameters under its control before applying them.
- 8172 2) Configuration validation rejects settings that would:
 - 8173 - disable or bypass mandatory security requirements defined in the present document; or
 - 8174 - compromise container-to-container isolation; or
 - 8175 - compromise controlled access to host resources.
- 8176 3) When a configuration change is rejected for security reasons, the CRS provides clear feedback to the
8177 administrator indicating the reason.

8178 3. Assessment Preparation

8179 The assessment shall have access to:

- 8180 • Documentation of CRS configuration parameters and validation behaviour, including examples of security-
8181 relevant invalid settings (e.g. disabling required controls, removing isolation, granting unrestricted host
8182 access).
- 8183 • Documentation of mandatory CRS security controls, isolation guarantees and host-access restrictions.
- 8184 • A test environment where CRS configuration can be modified via supported administrative interfaces.

8185 4. Assessment Activities

8186 The assessment shall at least:

- 8187 1) Identify a set of configuration parameters where insecure values would:
 - 8188 - disable/bypass mandatory security controls (e.g. turning off mandatory auth/logging/admission checks);
 - 8189 - weaken or remove container-to-container isolation; or
 - 8190 - grant uncontrolled access to host resources that should remain restricted.
- 8191 2) Attempt to apply such insecure configuration changes and verify that the CRS refuses to apply them as part of
8192 its validation logic.
- 8193 3) Capture the feedback presented to the administrator and verify that it clearly indicates the security-related
8194 reason for rejection (e.g. reference to disabled control, isolation violation, or host-access violation).
- 8195 4) Apply one or more valid configuration changes and verify they are accepted and enforced, demonstrating that
8196 validation focuses on insecure cases rather than blocking configuration in general.

8197 5. Assessment Verdict

8198 • Pass: configuration parameters are validated before application; insecure settings that disable/bypass
8199 mandatory controls or compromise isolation/host access are rejected; and clear, security-related feedback is
8200 provided to the administrator.

8201 • Fail: unsafe configurations can be applied; or validation does not cover key security/isolation/host-access
8202 guarantees; or error messages do not clearly explain the security reason.

8203 6. Assessment Evidence

8204 • [CONFIG] Examples of attempted insecure and valid configuration changes and their outcomes.

8205 • [DESIGN] Documentation of CRS configuration validation logic and its linkage to mandatory controls,
8206 isolation, and host-access rules.

8207 • [LOG] / [OBSERVATION] CLI/UI output showing rejection of insecure configurations with explicit reasons,
8208 and acceptance of secure configurations.

8209 6.4.1.16.3 Assessment Case AC-CRS-CFG-003

8210 1. Assessment Reference

8211 Requirement: REQ-CRS-CFG-003 (Advanced)

8212 2. Assessment Objective

8213 Verify that:

8214 1) The CRS supports defining and using security configuration baselines that specify secure settings for:

8215 - CRS daemon configuration; and

8216 - default container security profiles, including at least isolation-related options and default access to host
8217 resources.

8218 2) The CRS can automatically validate current CRS configuration and default container profile settings against a
8219 selected security baseline.

8220 3) The CRS generates alerts or audit events when non-compliant settings are detected.

8221 3. Assessment Preparation

8222 The assessment shall have access to:

8223 • Documentation describing security configuration baselines (how they are defined, stored, selected and scoped
8224 to daemon config and default container profiles).

8225 • Documentation of baseline content, including isolation-related options (e.g. default capabilities, privilege
8226 levels, namespaces) and default host resource access (e.g. host volumes, host network).

8227 • Documentation of automatic baseline validation mechanisms and alert/audit integration.

8228 • A test environment where at least one security baseline can be defined/selected and CRS/default-profile
8229 settings can be altered to introduce non-compliance.

8230 4. Assessment Activities

8231 The assessment shall at least:

8232 1) Define or select a security baseline that includes secure settings for CRS daemon configuration and default
8233 container security profiles, covering isolation and host-access defaults as supported by the product.

8234 2) Align CRS configuration and default container profiles with the baseline and invoke the baseline validation
8235 function; verify that the CRS reports compliance and no non-compliance alerts/events are generated.

8236 3) Introduce deliberate deviations from the baseline (e.g. relaxing default container isolation, enabling privileged
8237 default containers, broadening default host resource access, changing a daemon setting that weakens isolation).

- 8238 4) Run baseline validation again and verify that:
- 8239 - the CRS detects one or more non-compliant settings; and
- 8240 - alerts or audit log entries are generated identifying which settings deviate from the selected baseline.

8241 5. Assessment Verdict

- 8242 • Pass: security baselines can be defined and used for CRS daemon configuration and default container security
- 8243 profiles; automatic validation against a selected baseline is available; and detected non-compliance results in
- 8244 alerts or audit events that identify the deviations.
- 8245 • Fail: security baselines cannot be defined or do not cover both daemon config and default profiles; automatic
- 8246 validation cannot be performed or is ineffective; or non-compliant settings do not trigger clear alerts/audit
- 8247 events.

8248 6. Assessment Evidence

- 8249 • [CONFIG] Example security baseline definition and corresponding CRS configuration/default profile before
- 8250 and after introducing deviations.
- 8251 • [DESIGN] Documentation of CRS baseline management, validation logic and alert/audit integration.
- 8252 • [LOG] / [OBSERVATION] Validation outputs and alert/audit records showing detection and reporting of non-
- 8253 compliant settings.

8254 6.4.1.17 Assessment for Data Minimization

8255 6.4.1.17.1 Assessment Case AC-CRS-DM-001

8256 1. Assessment Reference

8257 Requirement: REQ-CRS-DM-001 (Basic)

8258 2. Assessment Objective

8259 Verify that:

- 8260 1) CRS logging, telemetry and diagnostic outputs are limited to information necessary for runtime operation,
- 8261 troubleshooting and security monitoring.
- 8262 2) Logs/telemetry/diagnostics do not unintentionally expose sensitive security material such as plaintext
- 8263 credentials, full cryptographic keys or complete container memory contents.
- 8264 3) Payload content from container workloads is not included in CRS logs/telemetry/diagnostics by default.
- 8265 4) Where payload inclusion for debugging is supported, it is only enabled through explicit administrative
- 8266 configuration and this state is clearly indicated to the administrator.

8267 3. Assessment Preparation

8268 The assessment shall have access to:

- 8269 • Documentation of CRS logging, telemetry and diagnostic capabilities, including any "debug", "deep
- 8270 inspection" or "payload capture" options.
- 8271 • Documentation specifying which types of data are explicitly excluded (credentials, keys, memory dumps,
- 8272 application payloads).
- 8273 • A test configuration with:
- 8274 - administrative access to configure logging/telemetry/diagnostics;
- 8275 - at least one container workload processing identifiable test payload data (e.g. specific strings);
- 8276 - the ability to generate authentication events and crypto operations with known test values.

8277 **4. Assessment Activities**

8278 The assessment shall at least:

- 8279 1) Review documentation and default configuration to identify:
- 8280 - which events and fields are logged/collected by default;
- 8281 - any options that may enable payload or extended context capture.
- 8282 2) In default configuration, perform:
- 8283 - successful and failed authentication attempts to CRS admin/control interfaces using known test
- 8284 usernames/password patterns;
- 8285 - cryptographic operations (e.g. key generation/use) with identifiable test keys;
- 8286 - normal container workload activity with recognizable test payload data.
- 8287 Then inspect logs/telemetry/diagnostics to verify that they do not contain:
- 8288 - plaintext credentials;
- 8289 - full cryptographic keys or complete container memory contents;
- 8290 - application/payload data from the containers.
- 8291 3) Where the CRS supports payload inclusion for debugging:
- 8292 - enable the relevant debug/payload option using administrative configuration;
- 8293 - verify that only an administrator can enable it and that the configuration/status clearly indicates that
- 8294 payload capture is active;
- 8295 - generate new test payload data and verify that such payload appears in logs/diagnostics as documented.

8296 **5. Assessment Verdict**

- 8297 • Pass: by default, logs/telemetry/diagnostics are limited to necessary operational and security data, without
- 8298 plaintext credentials, full keys, complete memory dumps or container payload; and any payload capture is only
- 8299 enabled via explicit administrative configuration and is clearly indicated.
- 8300 • Fail: sensitive security material or container payload appears in default logs/telemetry/diagnostics; or payload
- 8301 capture can be enabled without explicit admin action or clear indication.

8302 **6. Assessment Evidence**

- 8303 • [CONFIG] Default and modified CRS logging/telemetry/diagnostic configuration, including any
- 8304 debug/payload options.
- 8305 • [DESIGN] Documentation of data minimization behaviour and explicit exclusions (credentials, keys, memory,
- 8306 payload).
- 8307 • [LOG] / [OBSERVATION] Samples of logs/telemetry/diagnostics under default and debug/payload-enabled
- 8308 configurations, showing absence/presence of test secrets and payload as expected.

8309 **6.4.1.17.2 Assessment Case AC-CRS-DM-002**

8310 **1. Assessment Reference**

8311 Requirement: REQ-CRS-DM-002 (Basic)

8312 **2. Assessment Objective**

8313 Verify that:

- 8314 1) The CRS provides administrative controls to disable or restrict collection of specific categories of log,
- 8315 telemetry and runtime metric data.

- 8316 2) These controls allow administrators to disable or limit particular log types, telemetry streams, metric sets, or
8317 debug payload capture to align data collection with operational and regulatory requirements.

8318 3. Assessment Preparation

8319 The assessment shall have access to:

- 8320 • Documentation describing configurable categories of logs (e.g. audit vs debug), telemetry streams, metric
8321 groups and any debug/payload capture options.
- 8322 • Documentation of administrative roles/permissions required to modify data collection configuration.
- 8323 • A test configuration where different log/telemetry/metric categories can be toggled and where events can be
8324 generated to exercise each category.

8325 4. Assessment Activities

8326 The assessment shall at least:

- 8327 1) Identify, from documentation and current configuration, the distinct categories controllable by administrators
8328 (e.g. specific log types, telemetry streams, metric collections, payload capture).
- 8329 2) Using administrative controls, configure one or more categories (e.g. a debug log type, a telemetry stream, a
8330 metrics group, or payload capture) to be disabled or restricted.
- 8331 3) Generate CRS and workload activity that would normally produce data in those categories and verify that:
8332 - no new records are produced for disabled categories; and
8333 - non-disabled categories continue to produce data as configured.
- 8334 4) Re-enable one or more previously disabled categories and verify that data for those categories is again
8335 produced when new events occur.

8336 5. Assessment Verdict

- 8337 • Pass: administrators can selectively disable or restrict specific log types, telemetry streams, metrics and any
8338 debug payload capture; disabled categories stop producing data while others continue; and re-enabling restores
8339 data collection as configured.
- 8340 • Fail: data collection cannot be selectively controlled; or categories configured as disabled continue to produce
8341 data.

8342 6. Assessment Evidence

- 8343 • [CONFIG] CRS data-collection configuration before and after disabling/re-enabling specific
8344 log/telemetry/metric/payload categories.
- 8345 • [DESIGN] Documentation of administrative controls and category granularity for data collection, including
8346 how these support operational/regulatory alignment.
- 8347 • [LOG] / [OBSERVATION] Evidence showing presence/absence of data from selected categories when
8348 enabled vs disabled, demonstrating effective restriction of collection.

8349 6.4.2 CE

8350 [To be completed in a future version]

8351 6.4.3 CO

8352 [To be completed in a future version]

Annex A (informative): Relationship between the present document and the requirements of Regulation (EU) 2024/2847

The present document has been prepared under the Commission's standardisation request C(2025)618 [i.3] to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) [i.1]

Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance with the normative clauses of the present document given in Table A.1 confers, within the limits of the scope of the present document, a presumption of conformity with the corresponding requirements of that Regulation and associated EFTA regulations.

**Table A.1: Relationship between the present document and
the requirements of Regulation (EU) 2024/2847**

Harmonised Standard ETSI EN 304 635					
Requirement				Requirement Conditionality	
No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
1	The design, development, and production of products with digital elements ensures an appropriate level of cybersecurity based on the risks.	Annex I, Part I, (1)	Hypervisor: 5.1.1.1 5.1.1.2 5.1.1.3 5.1.1.4 5.1.1.5 5.1.1.6 CRS: 5.2.1.1 5.2.1.2 5.2.1.3 5.2.1.4 5.2.1.5 5.2.1.6	U	-
2	Products with digital elements are made available on the market without known exploitable vulnerabilities.	Annex I, Part I, (2)(a)	Hypervisor: 5.1.1.2 5.1.1.8 5.1.1.9 5.1.1.10 CRS: 5.2.1.2 5.2.1.8 5.2.1.9 5.2.1.10	U	
3	Products with digital elements are made available on the market with a secure by default configuration, unless otherwise agreed between manufacturer and business user in relation to a tailor-made product with digital elements, including the possibility to reset the product to its original state.	Annex I, Part I, (2)(b)	Hypervisor: 5.1.1.9 5.1.1.10 CRS: 5.2.1.9 5.2.1.10	U	

Harmonised Standard ETSI EN 304 635					
Requirement				Requirement Conditionality	
No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
4	Products with digital elements ensure that vulnerabilities can be addressed through security updates, including, where applicable, through automatic security updates that are installed within an appropriate timeframe enabled as a default setting, with a clear and easy-to-use opt-out mechanism, through the notification of available updates to users, and the option to temporarily postpone them.	Annex I, Part I, (2)(c)	Hypervisor: 5.1.1.8 CRS: 5.2.1.8	U	
5	Products with digital elements ensure protection from Unauthorized access by appropriate control mechanisms, including but not limited to authentication, identity or access management systems, and report on possible Unauthorized access.	Annex I, Part I, (2)(d)	Hypervisor: 5.1.1.3 5.1.1.4 CRS: 5.2.1.3 5.2.1.4	U	
6	Products with digital elements protect the confidentiality of stored, transmitted or otherwise processed data, personal or other, such as by encrypting relevant data at rest or in transit by state of the art mechanisms, and by using other technical means.	Annex I, Part I, (2)(e)	Hypervisor: 5.1.1.5 CRS: 5.2.1.5	U	
7	Products with digital elements protect the integrity of stored, transmitted or otherwise processed data, personal or other, commands, programs and configuration against any manipulation or modification not authorized by the user, and report on corruptions.	Annex I, Part I, (2)(f)	Hypervisor: 5.1.1.2 5.1.1.7 CRS: 5.2.1.2 5.2.1.7	U	
8	Products with digital elements process only data, personal or other, that are adequate, relevant and limited to what is necessary in relation to the intended purpose of the product with digital elements (data minimization).	Annex I, Part I, (2)(g)	Hypervisor: 5.1.1.11 CRS: 5.2.1.11	U	

Harmonised Standard ETSI EN 304 635					
Requirement				Requirement Conditionality	
No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
9	Products with digital elements protect the availability of essential and basic functions, also after an incident, including through resilience and mitigation measures against denial-of-service attacks.	Annex I, Part I, (2)(h)	Hypervisor: 5.1.1.6 CRS: 5.2.1.6	U	
10	Products with digital elements minimize the negative impact by the products themselves or connected products on the availability of services provided by other products or networks.	Annex I, Part I, (2)(i)	Hypervisor: 5.1.1.1 5.1.1.6 CRS: 5.2.1.1 5.2.1.6	U	
11	Products with digital elements are designed, developed and produced to limit attack surfaces, including external interfaces.	Annex I, Part I, (2)(j)	Hypervisor: 5.1.1.1 5.1.1.10 CRS: 5.2.1.1 5.2.1.10	U	
12	Products with digital elements are designed, developed and produced to reduce the impact of an incident using appropriate exploitation mitigation mechanisms and techniques.	Annex I, Part I, (2)(k)	Hypervisor: 5.1.1.1 5.1.1.2 5.1.1.6 CRS: 5.2.1.1 5.2.1.2 5.2.1.6	U	
13	Products with digital elements provide security related information by recording and monitoring relevant internal activity, including the access to or modification of data, services or functions, with an opt-out mechanism for the user.	Annex I, Part I, (2)(l)	Hypervisor: 5.1.1.7 5.1.1.11 CRS: 5.2.1.7 5.2.1.11	U	
14	Products with digital elements provide the possibility for users to securely and easily remove on a permanent basis all data and settings and, where such data can be transferred to other products or systems, ensure that this is done in a secure manner.	Annex I, Part I, (2)(m)	Hypervisor: 5.1.1.5 5.1.1.9 CRS: 5.2.1.5 5.2.1.9	U	
...	CRA Annex I - Part II: Vulnerability Handling Requirements		5.3		

8367

8368 **Key to columns:**8369 **Requirement:**8370 **No** A unique identifier for one row of the table which may be used to identify a requirement.

8371	Description	A textual reference to the requirement.
8372	Requirements of Regulation	
8373		Identification of article(s) defining the requirement in the Regulation.
8374	Clause(s) of the present document	
8375		Identification of clause(s) defining the requirement in the present document unless another
8376		document is referenced explicitly.
8377	Requirement Conditionality:	
8378	U/C	Indicates whether the requirement is unconditionally applicable (U) or is conditional upon the
8379		manufacturer's claimed functionality of the equipment (C).
8380	Condition	Explains the conditions when the requirement is or is not applicable for a requirement which is
8381		classified "conditional".
8382	Presumption of conformity stays valid only as long as a reference to the present document is maintained in the list	
8383	published in the Official Journal of the European Union. Users of the present document should consult frequently the	
8384	latest list published in the Official Journal of the European Union.	
8385	Other Union legislation may be applicable to the product(s) falling within the scope of the present document.	
8386		

8387
8388

Annex B (informative): Cybersecurity Risk Assessment Methodology

8389

B.0 Overview

8390
8391
8392

This clause is informative and non-binding. It provides a risk assessment methodology for assessing the cybersecurity risk of a product based on its intended use case and deployment environment. This approach helps determine the risk level for various use cases, guiding the application of appropriate security requirements.

8393

Manufacturers can use this methodology as guidance or apply their own established risk assessment methodologies.

8394
8395

The risk assessment methodology used herein is a semi-quantitative approach, combining qualitative analysis with numerical scoring. It provides a framework for:

8396
8397
8398
8399

- Identifying product cybersecurity threats.
- Determining the likelihood and magnitude of impact if those threats materialize.
- Evaluating the overall cybersecurity risk for the product within its specific use case and deployment environment.

8400
8401

This methodology aligns with the general principles for assessing product cybersecurity risk, as defined in prEN 40000-1-2 [i.11].

8402

B.1 Risk Assessment Process

8403
8404

The risk assessment process focuses on evaluating the product within its defined use case and deployment environment, following a logical sequence of steps:

8405
8406
8407

- 1) Define the product and its context: Identify the product being assessed, its intended use case, and the specific deployment environment it is intended for. This provides the foundational context for the assessment.
- 2) Identify product threats: Identify potential cybersecurity threats that could exploit vulnerabilities.

8408

For each identified threat, the following steps should be performed:

8409
8410
8411
8412
8413
8414

- 3) Analyze risk factors for likelihood and impact as defined in clause B.2: This step determines the product's cybersecurity risk by evaluating its characteristics related to potential threats. For each identified threat, and for every relevant risk factor, the following guiding question should be applied: *"How does this factor influence the likelihood or impact if the threat were to occur?"*. Each threat is evaluated against a defined set of risk factors covering both Likelihood and Impact. These factors are scored on a three-level scale (Low = 1, Medium = 2, High = 3) using the definitions provided in clause B.2.

8415

The analysis involves the following sub-steps:

8416
8417
8418
8419

- Identifying key risk factors: Determine the main aspects of the product's operation within that environment that influence its cybersecurity risk (e.g. connectivity, administrative context, configuration volatility, impact of compromise, integration with critical systems). These include factors that affect either:

8420
8421

- 1) the likelihood of threat exploitation; or
- 2) the severity of consequences if the threat materializes.

8422

These factors serve as proxies for the likelihood of a threat materializing and the magnitude of its impact.

8423
8424

- Defining scoring: Assign numerical scores (1 for low risk, 2 for medium, 3 for high) to each risk factor based on how it influences the likelihood of a threat event and the magnitude of its potential impact.

- 8425 4) Derive overall likelihood score: Sum the scores of all risk factors identified as influencing "Likelihood". Map
8426 this total score to the Likelihood level (1-3) using the thresholds defined in clause B.3.1.
- 8427 5) Derive overall impact score: Sum the scores of all risk factors identified as influencing "Impact". Map this
8428 total score to the Impact level (1-3) using the thresholds defined in clause B.3.2.
- 8429 6) Determine final risk level: Using the derived overall Likelihood score and overall Impact score, consult the
8430 risk matrix (clause B.4) to determine the final, overall risk level (Low, Medium, High) for the specific use
8431 case.
- 8432 7) Document and review: Record all assessment findings, including the rationale for scores and the determined
8433 risk level.

8434

B.2 Risk Factor Scoring

8435 Table B.2-1 provides the detailed scoring rubric to be used for assessing cybersecurity risks of products within their use
8436 cases and deployment environments. Each risk factor is explicitly linked to whether it primarily influences likelihood or
8437 impact. The 1-3 ratings of the risk factors are summed per type (Likelihood or Impact) to derive the overall likelihood
8438 and impact scores.

Table B.2-1: Risk factors

Tag	Factor	Type	Description	Low (1)	Medium (2)	High (3)
EXT	External Network Exposure	Likelihood	Measures how reachable the VES/CES is from untrusted networks, including public internet and cross-tenant networks. Higher exposure increases the probability of remote exploitation.	Internal-only access; management on isolated VLAN/bastion; strict firewall segmentation.	Limited/controlled external access (VPN, allow-list, jump host).	Public internet-facing endpoints or broad cross-tenant network paths.
TEN	Tenant Multiplicity & Trust	Likelihood	Captures the number of tenants and their trust level. Multi-tenancy with untrusted tenants increases the chance of malicious insiders or cross-tenant attacks.	Single trusted tenant.	Few semi-trusted tenants.	Many untrusted tenants (public/multi-organization).
CTL	Control-Plane Exposure, Access Control & Detectability	Likelihood	Evaluates the exposure, strength of authentication/authorization on management interfaces, and the extent to which malicious actions on the control plane would be detectable.	Admin access isolated, MFA + RBAC enforced, centralized logging alerts on anomalous admin activity.	Internal network access, standard RBAC, partial monitoring or periodic audits.	Externally reachable control plane, weak RBAC, or limited/no monitoring of administrative actions.
SUP	Update & Supply-Chain Assurance	Likelihood	Assesses the security of update mechanisms, software/image sources, and supply-chain validation. Weak assurance can allow malicious updates or images.	Signed/verified only; private mirrors; staged/canary updates.	Vendor-signed updates/images; occasional direct pulls.	Direct pulls from public registries/vendor repositories; unsigned allowed.
PRV	Attack Complexity & Privileged Footprint	Likelihood	Represents the attacker skill/resources required to exploit the privileged software surface. Higher privileged footprint or weaker hardening → lower attacker skill required.	Minimal attack surface, compiler + runtime hardening (ASLR, SMEP/SMAP/CET), requires advanced attacker capability.	Mixed hardening, moderate privileged surface, requires skilled attacker with tools.	Large privileged surface, device pass-through, weak/no hardening — exploitable by unsophisticated attackers.
CHG	Change/Workload Volatility	Impact	Evaluates how frequently the configuration and workloads change. Higher volatility increases the likelihood of misconfigurations or unpatched instances.	Static, rarely changes.	Periodic changes (weekly/monthly).	Highly dynamic (daily/hourly), autoscaling, frequent reconfiguration.
ISO	Isolation Breakout Consequence	Impact	Assesses the damage if the VM/container isolation boundary is broken. Higher values mean more severe lateral movement or system compromise.	No sensitive assets exposed; minimal lateral movement possible.	Exposure of internal systems/data; contained blast radius.	Cross-tenant or critical system compromise; admin plane reachable.

Tag	Factor	Type	Description	Low (1)	Medium (2)	High (3)
DAT	Data/Safety/Criticality	Impact	Measures the sensitivity of the data and any safety or regulatory implications of compromise. Higher scores indicate severe operational, safety, or compliance impacts.	Public or low-sensitivity data; no safety impact.	Confidential/business data; moderate operational disruption.	Regulated/PII/financial/eHealth data; safety hazard or critical infrastructure outage.
ORB	Orchestration Dependency & Blast Radius	Impact	Evaluates how much damage an orchestrator compromise could cause. Higher values reflect large-scale or critical workload control.	Stand-alone host; no orchestrator or read-only integration.	Basic orchestrator; tenant- or namespace-scoped control.	Full control of critical workloads across multi-node clusters/sites.
DRC	Detection & Recovery Capability	Impact	Measures the ability to detect, contain, and recover from a compromise. Poor detection/recovery increases the residual impact of an incident.	Centralized logging/alerting; immutable backups; tested recovery plans.	Adequate logging/backups; periodic recovery tests.	Minimal logging/alerting; weak or untested recovery procedures.

B.3 Calculation and Mapping to Likelihood/Impact Levels

B.3.0 General

This clause defines how the total scores from Table B.2-1 in clause B.2 are translated into the 1-3 Likelihood and Impact levels used in the risk matrix (clause B.4).

B.3.1 Likelihood Score

Table B.2-1 in clause B.2 defines five Likelihood factors:

- EXT (External Network Exposure)
- TEN (Tenant Multiplicity & Trust)
- CTL (Control-Plane Exposure, Access Control & Detectability)
- SUP (Update & Supply-Chain Assurance)
- PRV (Privileged Footprint & Hardening)

Each factor is scored from 1 (Low) to 3 (High).

- Maximum possible Likelihood score: 5 factors $\times$ 3 = 15
- Minimum possible Likelihood score: 5 factors $\times$ 1 = 5

The resulting total score is mapped to three levels as follows:

Table B.3.1-1: Likelihood score

Total Likelihood Score Range	Likelihood Level	Description
5 - 8	Low (1)	The event is highly unlikely to occur under normal circumstances.
9 - 12	Medium (2)	The event may occur occasionally in enterprise or cloud deployments.
13 - 15	High (3)	The event is expected to occur or has a high probability of occurring.

B.3.2 Impact Score

Table B.2-1 in clause B.2 defines five Impact factors:

- CHG (Change/Workload Volatility)
- ISO (Isolation Breakout Consequence)
- DAT (Data/Safety/Criticality)
- ORB (Orchestration Dependency & Blast Radius)
- DRC (Detection & Recovery Capability)

Each factor is scored from 1 (Low) to 3 (High).

- Maximum possible Impact score: 5 factors $\times$ 3 = 15
- Minimum possible Impact score: 5 factors $\times$ 1 = 5

The resulting total score is mapped to three levels as follows:

Table B.3.2-1: Impact score

Total Impact Score Range	Impact Level	Description
5 - 8	Low (1)	Minimal disruption. No significant loss of confidentiality, integrity, or availability. No impact on safety or critical infrastructure.
9 - 12	Medium (2)	Noticeable disruption requiring recovery. Moderate impact on confidentiality, integrity, or availability. Potential for indirect impact on safety or critical infrastructure.
13 - 15	High (3)	Severe disruption, service interruption, or data loss. Major impact on confidentiality, integrity, or availability. May directly affect safety or critical infrastructure.

B.4 Risk Matrix

The final overall risk level is determined by combining the calculated likelihood level (clause B.3.1) and impact level (clause B.3.2) using the matrix below:

Table B.4-1: Risk matrix

Impact \ Likelihood	1 = Low	2 = Medium	3 = High
3 = High	Medium	High	High
2 = Medium	Low	Medium	High
1 = Low	Low	Low	Medium

B.5 Risk Methodology Applied to Use Cases

B.5.0 General

This clause illustrates the application of the methodology to products deployed within the VES use cases (UC-V1, UC-V2, UC-V3) and CES use cases (UC-C1, UC-C2, UC-C3). The use cases are defined in clause 4.3, while the corresponding threats applicable to these products are described in clause 4.4.

B.5.1 Risk Evaluation of VES use cases

B.5.1.1 Hypervisor Type I

Table B.5.1.1-1: Hypervisor Type I - Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP1-INTF-ESC: Inter-VM interference / VM escape														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	In UC-V1 the Type-1 hypervisor runs on a dedicated, isolated host with a very small set of non-critical VMs, limited external connectivity (EXT=1, TEN=1) and strictly local administration. Even if a VM escape occurs, lateral movement remains confined to that single host and low-value workloads, so both likelihood and impact are assessed as Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	In UC-V2, the Type-1 hypervisor is part of an enterprise cluster with more VMs per tenant, shared management networks and standard hardening (EXT=1, TEN=2, CTL=3). A successful VM escape would expose all VMs owned by that enterprise tenant on that cluster, making both the feasibility and the consequences noticeably higher than in UC-V1.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, Type-1 hypervisors underpin multi-tenant critical clouds where untrusted tenants co-exist and regulated or safety-relevant workloads are hosted (TEN=3, DAT=3, ORB=3). A VM escape here enables cross-tenant compromise and direct access to high-criticality data, so both likelihood and impact are rated High.
T-HYP1-INTEG-COMP: Integrity compromise of workloads / hypervisor components														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	For UC-V1, tampering with hypervisor binaries or VM images typically requires local access to a single host; workloads are static and non-critical, and orchestration blast radius is limited (ORB=1). As a result, integrity compromise remains a localized issue with Low impact on operations.
V2	1	2	3	2	2	2	2	2	2	3	10 → Med	11 → Med	Medium	In UC-V2, hypervisor binaries, kernel modules and VM images are centrally managed for business workloads. If an attacker tampers with these components on an enterprise VES, they can gain persistent control over all VMs of that tenant (ISO=2, ORB=2) and manipulate business services, so both likelihood and impact are Medium.
V3	3	3	3	3	3	3	2	3	3	3	15 → High	14 → High	High	In UC-V3, a compromise of Type-1 hypervisor components (e.g. signed modules, microcode integrations) affects multi-tenant, high-assurance environments. Any persistent backdoor or binary tampering undermines the entire trust anchor for critical workloads

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														across tenants (ISO=2 but DAT=3, ORB=3), justifying High likelihood and High impact.
T-HYP1-UNAUTH-ADMIN: Unauthorized access to management interfaces														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	In UC-V1, direct admin interfaces of the Type-1 hypervisor are usually reachable only from an isolated management network or via local console, and the host runs non-critical VMs. Even if authentication is weak, the lack of broad network exposure and the low value of the workloads keep both likelihood and impact Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, the Type-1 management plane (e.g. vSphere, vendor APIs) is reachable from the enterprise network and is a high-value target (EXT=1, CTL=3). A compromise of admin credentials or interfaces gives full control over all enterprise VMs, but the effects are still contained within that tenant/domain, leading to Medium risk.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, Type-1 management APIs, even if segregated, are critical control points for multi-tenant and regulated workloads. Unauthorized access to these interfaces would allow massive cross-tenant operations, uncontrolled VM management, and potential breach of critical systems, warranting High likelihood and High impact.
T-HYP1-UNAUTH-FUNC: Unauthorized privileged actions														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	Within UC-V1, a misconfigured RBAC or over-permissive role on the Type-1 hypervisor can allow privilege abuse (e.g. VM snapshots, power-off) but only on a few local, non-critical VMs. Given strong isolation from external networks and the limited blast radius, both likelihood and impact remain Low.
V2	1	2	3	1	3	2	2	2	2	2	10 → Med	10 → Med	Medium	In UC-V2, enterprise deployments may assign operator roles to multiple administrators or automation tools. If authorization checks are weak (CTL=3, PRV=3), a compromised low-privilege account can perform administrative actions (e.g. attach disks, modify vSwitches) across the tenant's VMs, making both likelihood and impact Medium.
V3	2	3	2	3	3	3	3	3	3	3	13 → High	15 → High	High	In UC-V3, weak or inconsistent authorization on the Type-1 management plane allows a compromised operator account to perform privileged actions affecting many untrusted tenants (TEN=3, ORB=3) and high-criticality workloads. This drives both likelihood and impact to High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP1-DATA-DISC: Data disclosure														
V1	1	1	2	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, Type-1 hypervisors host non-critical workloads with low-sensitivity data (DAT=1) and minimal connectivity. Even if hypervisor misconfigurations or weak encryption expose VM disks or inter-VM traffic, the resulting confidentiality impact is limited, so both likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, the Type-1 hypervisor stores and processes business data and may support backups or snapshots. Misconfigurations or vulnerabilities leading to access to VM images or memory dumps would significantly affect the enterprise tenant, but not multiple independent tenants, leading to Medium risk.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, Type-1 hypervisors underpin clouds handling regulated and critical workloads (e.g. eHealth, finance, telecom core). A breakout that accesses VM RAM, disks, or east-west traffic can disclose highly sensitive multi-tenant data (DAT=3, ISO=3, ORB=3), resulting in High likelihood and High impact.
T-HYP1-AVAIL-DOS: DoS via resource exhaustion														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, resource exhaustion attacks generally require local or tightly scoped access, target a small number of VMs, and affect functions that are not business- or safety-critical. Even if CPU, memory or I/O are saturated, only a single host and low-value workloads are impacted, so likelihood and impact are Low.
V2	1	2	2	2	2	2	2	2	2	2	9 → Med	10 → Med	Medium	For UC-V2, noisy-neighbour effects or malicious workloads can cause CPU steal, memory pressure, or shared-I/O bottlenecks on Type-1 hosts that carry many enterprise workloads. While blast radius is still tenant-scoped, such events can cause notable service disruption (CHG=2, ORB=2), driving likelihood and impact to Medium.
V3	2	3	3	2	3	3	3	2	3	2	13 → High	13 → High	High	In UC-V3, shared Type-1 hypervisors run heterogeneous, often untrusted tenants at high density. Misconfigured overcommitment or deliberate resource exhaustion can create systemic availability issues across many critical services (TEN=3, CHG=3, ORB=3). Therefore, both likelihood and impact are assessed as High.
T-HYP1-LOG-INT: Insufficient logging / log tampering														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	Low	In UC-V1, logging on the Type-1 hypervisor may be minimal, and local administrators or malware could clear or modify logs. However, the environment has low data sensitivity and limited orchestration (DAT=1, ORB=1), so the inability to trace events has a Low overall impact.
V2	1	2	2	2	2	2	1	2	2	2	9 → Med	9 → Med	Medium	In UC-V2, insufficient audit trails on hypervisor admin actions, VM lifecycle events, and configuration changes delay detection of targeted attacks on business workloads. Enterprise security monitoring exists but may not fully cover hypervisor logs (DRC=2), so both likelihood and impact are Medium.
V3	2	3	3	2	3	2	2	3	3	3	13 → Med	13 → High	High	In UC-V3, gaps or tampering in Type-1 logging and audit mechanisms seriously undermine forensic readiness and incident containment. In multi-tenant critical clouds, missing or untrustworthy logs mean that cross-tenant attacks may remain undetected and unbounded in time, resulting in High likelihood but High impact.
T-HYP1-UPD-COMP: Malicious or compromised update														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, updates to the Type-1 hypervisor may be done manually or infrequently, sometimes from offline media or vendor repositories. A compromised image would affect a single host with non-critical workloads, and some manual verification is often possible, so both likelihood and impact stay Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, centralized yet sometimes ad-hoc update procedures (SUP=2) mean that delayed or inconsistent patching can allow malicious or tampered hypervisor updates to be applied across an enterprise cluster. This can compromise all VMs owned by that tenant, giving Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, Type-1 hypervisors rely on highly automated, large-scale update pipelines. A compromise of the update supply chain or signing infrastructure would propagate to many critical hosts simultaneously (SUP=3, ORB=3), making both likelihood and impact High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP1-CFG-COMP: Insecure or Unauthorized configuration														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, misconfigurations (e.g. leaving a debug interface enabled or using permissive vSwitch settings) are usually applied locally and affect a small, static VM set. The environment is not exposed to untrusted tenants, so while isolation can be weakened, the overall risk remains Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Med	10 → Med	Medium	For UC-V2, centralized management of many VMs introduces more frequent changes and complex configuration (CHG=2). Errors such as exposing management services, disabling security features or misconfiguring virtual networks can expose sensitive enterprise workloads, leading to Medium likelihood and Medium impact.
V3	2	3	3	2	3	3	3	3	3	3	13 → High	15 → High	High	In UC-V3, configuration baselines have to enforce strong isolation, least privilege, and hardened defaults. Insecure or Unauthorized changes (e.g. tenant-visible management networks, overly permissive resource mappings) can immediately affect multiple critical tenants and services, so both likelihood and impact are High.
T-HYP1-SC-COMP: Supply chain compromise														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a compromised driver, library, or firmware component used by the Type-1 hypervisor yields a backdoor on one isolated host. Since workloads are low-value and connectivity is limited, the damage is confined, so likelihood and impact are Low.
V2	2	2	2	3	2	2	2	2	2	2	11 → Med	10 → Med	Medium	In UC-V2, the Type-1 stack depends on a broader supply chain (Virtualization platform, OS components, drivers). A poisoned component can introduce persistent backdoors affecting all enterprise VMs on the cluster, but still within one tenant domain, resulting in Medium likelihood and Medium impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, supply-chain compromise of hypervisor components (e.g. low-level drivers or cryptographic libraries) scales across many tenants and critical services. Every VM on the affected hosts inherits the compromised code path, giving the attacker a privileged foothold with High likelihood and High impact.

B.5.1.2 Hypervisor Type II

Table B.5.1.2-1: Hypervisor Type II - Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP2-HOST-ABUSE: Guest-to-host escalation														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a Type-2 hypervisor runs on a single host OS with few non-critical VMs and limited services exposed. An attacker in a guest can theoretically abuse host APIs or shared services, but low exposure and low data value keep both likelihood and impact Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, the host OS provides shared services (e.g. file shares, clipboard, device redirection) to multiple enterprise VMs. Weak isolation or host hardening allows a malicious VM to escalate into the host OS, from which it can access all internal workloads on that machine, giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, hosted Virtualization may underpin critical or multi-tenant workloads; a successful guest-to-host escalation immediately compromises the host OS and all VMs it controls (ISO=3, DAT=3, ORB=3). Given the large attack surface of general-purpose OSEs, both likelihood and impact are High.
T-HYP2-INTEG-COMP: Integrity compromise of hosted hypervisor														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-V1, tampering with the Type-2 hypervisor binary or VM images on the host filesystem typically requires local access and affects only non-critical workloads. The blast radius is limited to one host, so likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, compromise of host OS or hypervisor application binaries directly affects the integrity of all VMs on that host and may propagate through shared images or templates within the enterprise. However, the effect is still scoped to one internal tenant environment, leading to Medium risk.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, hosted Virtualization is used for high-assurance workloads; any manipulation of binaries, modules or VM images via the host OS or packaging system undermines trust in multiple critical services. This justifies High likelihood and High impact.
T-HYP2-UNAUTH-ADMIN: Unauthorized admin access														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	Low	In UC-V1, the host OS admin account and the Type-2 hypervisor console are typically accessible only locally or on a restricted network, and control non-critical VMs. Even if passwords are weak, compromise remains local to a single device, making both likelihood and impact Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	For UC-V2, host OS and hypervisor admin interfaces are reachable from the enterprise network and control multiple business VMs (CTL=3). Unauthorized access to these accounts enables full control over internal workloads, but still within one enterprise boundary, so the risk is Medium.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, internet-reachable or widely exposed host OS admin accounts and Type-2 management consoles represent a systemic control point. Their compromise yields administrative control over critical or regulated workloads and potentially multiple tenants, so both likelihood and impact are High.
T-HYP2-UNAUTH-FUNC: Unauthorized privileged actions														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, misconfigured permissions in the Type-2 application or host OS may allow local users to perform limited privileged actions, but the small number of non-critical VMs and limited connectivity keep the overall risk Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, enterprise deployments often have multiple operators or automation agents interacting with the hypervisor application. Weak separation of roles or coarse-grained permissions allow Unauthorized privileged actions that affect many internal workloads (PRV=2, ISO=2), giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, insufficient authorization controls in the Type-2 hypervisor or host OS allow a compromised user or process to perform high-impact operations (e.g. mount host paths, change networking, access snapshots) across critical tenants. This drives both likelihood and impact to High.
T-HYP2-DATA-DISC: Sensitive data disclosure														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, VMs on a hosted hypervisor process low-sensitivity data and are often isolated physically or logically. Even if an attacker

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														reads VM image files or host-mediated traffic, the confidentiality impact is minimal, resulting in Low likelihood and Low impact.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, a Type-2 hypervisor can host enterprise workloads whose VM image files, snapshots, or memory may contain confidential business data. Exploiting host OS weaknesses or hypervisor flaws to access these artefacts affects the enterprise tenant significantly, but not other independent tenants, leading to Medium risk.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, Type-2 environments may handle highly sensitive or regulated data on general-purpose OSes. If an attacker gains access to VM images, RAM, or inter-VM traffic via the host, multiple critical services and tenants can be affected, so both likelihood and impact are High.
T-HYP2-AVAIL-DOS: Resource exhaustion														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, a DoS caused by a single VM or host-level process consuming CPU, memory, or I/O affects only a standalone host with non-critical workloads, yielding Low likelihood and impact.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	For UC-V2, resource exhaustion on a host OS running a Type-2 hypervisor can disrupt many internal business VMs (e.g. via runaway processes or misconfigured resource limits). However, the effect is still constrained to one enterprise environment, so likelihood and impact are Medium.
V3	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	High	In UC-V3, hosted Virtualization may carry critical services or multi-tenant workloads; noisy-neighbour effects or deliberate resource abuse at host level can degrade or interrupt many high-criticality services simultaneously (CHG=3, ORB=3). This justifies High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP2-LOG-INT: Logging weakness/tampering														
V1	1	1	1	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, logs for the Type-2 hypervisor and host OS are often stored locally and may not be centrally monitored. While this allows some tampering, the low-sensitivity nature of the workloads limits the overall impact, so risk is Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, weak or inconsistent logging of VM lifecycle events, admin actions, or host OS changes makes enterprise incident detection slower. Attackers can persist longer within internal workloads (DRC=2), which supports Medium likelihood and impact.
V3	2	3	3	2	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, insufficient or tamperable logs on the host OS and Type-2 hypervisor significantly weaken forensic and recovery capabilities across critical workloads. Multi-tenant or regulated services then lack reliable evidence of compromise, resulting in High likelihood and High impact.
T-HYP2-UPD-COMP: Malicious/compromised update														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, updates to the hosted hypervisor application are often done manually or automatically by local admins. A malicious installer can compromise that single host and its few non-critical VMs, but the limited scale and occasional update frequency make the overall risk Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, admins may download and install Type-2 hypervisor updates directly from vendors or repositories on multiple enterprise hosts. A poisoned installer or downgraded package can compromise many internal workloads, so both likelihood and impact are Medium.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, automated or large-scale update processes for hosted hypervisors and host OS components mean a single compromised package or repository can propagate quickly to many critical nodes (SUP=3, ORB=3). This supports High likelihood and High impact.
T-HYP2-CFG-COMP: Insecure or Unauthorized configuration														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, insecure defaults (e.g. permissive host-guest shared folders, bridged networking) or local misconfigurations affect only a small number of non-critical VMs. The lack of multi-tenancy and external exposure keeps likelihood and impact Low.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-V2, Type-2 hypervisors on enterprise hosts may be configured with convenient but insecure settings (e.g. broad host path mappings, weak isolation between lab and production VMs). These configurations enable lateral movement and Unauthorized access between internal workloads, giving Medium risk.
V3	2	3	3	2	3	3	3	3	3	3	14 → High	14 → High	High	In UC-V3, insecure Type-2 configurations (such as open shared folders, permissive host networking, or control channels without authentication) directly break isolation between high-assurance tenants and critical services. Given the strong reliance on configuration correctness, both likelihood and impact are High.
T-HYP2-SC-COMP: Supply chain compromise														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a malicious driver or library used by the Type-2 hypervisor executes only on a single, isolated host. While it provides a backdoor to that system, the limited connectivity and low-value workloads constrain the impact, so likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, enterprises may deploy Type-2 hypervisors at scale using OS packages or vendor installers. A compromised dependency (e.g. kernel module, library) can introduce persistent, hard-to-detect backdoors on multiple hosts and affect many internal workloads, leading to Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, Type-2 hypervisors and host OS components depend on broader upstream ecosystems. A supply-chain attack on drivers, libraries, or packaged modules can propagate widely, compromising host OS and hypervisor instances across critical tenants, thus supporting High likelihood and High impact.

B.5.1.3 M&O System

[To be completed in a future version]

B.5.2 Risk Evaluation of CES use cases

B.5.2.1 CRS

Table B.5.2.1-1: CRS - Risk evaluation of CES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-INTF-ESC: Runtime-level container breakout														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, the CRS runs on isolated devices or personal environments where containers host non-critical workloads and have limited external exposure. A runtime-level escape (e.g. via runc or namespace flaws) is technically possible, but any breakout only affects a single device and low-sensitivity applications, so both likelihood and impact are assessed as Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the CRS underpins enterprise or edge container platforms that host multiple business microservices. A successful runtime-level escape allows a compromised container to interfere with peer containers on that node (and possibly across nodes, if combined with orchestration weaknesses), exposing all business workloads for that tenant. This justifies Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-C3, the CRS is part of high-assurance, multi-tenant container platforms processing critical or regulated workloads. Exploits that break container isolation (e.g. sandbox escapes, kernel namespace bugs) can lead to cross-tenant compromise, access to critical data, and potential safety or regulatory failures, so both likelihood and impact are High.
T-CRS-INTEG-COMP: Runtime-level tampering														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-C1, local tampering with runtime binaries or configuration (e.g. disabling seccomp or cgroup enforcement) requires direct access to a single device and affects only non-critical applications. The orchestrator, if present, is minimal or absent, and the blast radius is small, so both likelihood and impact remain Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the same CRS binaries and configs may be deployed across many enterprise nodes. If an attacker modifies runtime settings (e.g. disables AppArmor profiles, relaxes namespaces or cgroup limits), multiple business services lose isolation and resource

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														controls. Although the effect is tenant-scoped, it impacts many workloads, leading to Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-C3, runtime integrity is a core assurance anchor for critical and regulated container workloads. Tampered runtime binaries or unverified state (e.g. modified seccomp profiles, spoofed namespaces) can silently bypass orchestrator policies across many tenants and nodes. Therefore, both likelihood and impact are rated High.
T-CRS-UNAUTH-ADMIN: Unauthorized runtime control access														
C1	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	Low	In UC-C1, the container runtime is typically driven by local commands (e.g. docker, ctr, podman) on a single, non-critical device. Misuse of these binaries via a local user or shared account is possible but affects only that device and low-value workloads, so likelihood and impact are Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-C2, internal users or automation (CI/CD agents, deployment scripts) may have direct access to runtime control (e.g. privileged socket access or debug binaries). If authentication around such access is weak, insiders or compromised accounts can bypass orchestrator RBAC and start, stop, or modify containers directly on enterprise nodes. This yields Medium likelihood and Medium impact across the tenant's workloads.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	14 → High	High	In UC-C3, any exposed or weakly protected runtime control interface (e.g. an unprotected runtime socket, debug endpoint, or local root account on critical nodes) allows attackers to circumvent orchestrator policy and manipulate containers running critical/regulated workloads. In multi-tenant or federated scenarios, this can impact many tenants and critical services, hence High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-UNAUTH-FUNC: Unauthorized privileged ops														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-C1, users with legitimate but broad permissions (e.g. local "developer" accounts) can perform privileged runtime actions such as starting privileged containers or mounting host paths. However, these actions are confined to a single device and non-critical apps, so overall risk remains Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise or edge environments often delegate limited runtime access to developers, operators, or automation tools. If authorization is insufficiently granular, these principals can escalate to privileged operations (e.g. modifying seccomp profiles, running containers as root, altering cgroup limits) and affect many containers belonging to that tenant, justifying Medium likelihood and Medium impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, inadequate authorization in the CRS enables authenticated but mis-scoped identities to perform powerful runtime operations across nodes hosting critical or regulated workloads. This can override intended least-privilege policies, create new attack surfaces, or directly expose sensitive data, so both likelihood and impact are High.
T-CRS-DATA-DISC: Data disclosure via runtime														
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-C1, containers typically host low-sensitivity data (e.g. local telemetry, hobbyist applications). Even if CRS flaws allow access to container filesystems, mounts, or memory, the confidentiality impact is minimal, so both likelihood and impact are Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the CRS may host business microservices processing confidential data (e.g. HR, analytics, internal APIs). Runtime-level weaknesses that expose mounted volumes, layered filesystems, or shared memory segments can leak sensitive internal data of the enterprise tenant, leading to Medium likelihood and Medium impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, containers may handle financial, e-identity, or medical data in multi-tenant and federated setups. If runtime isolation fails, attackers can extract highly sensitive data from multiple tenants. This supports High likelihood and High impact.
T-CRS-AVAIL-DOS: Resource exhaustion														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-C1, a container that saturates CPU, memory, or I/O via weak cgroup settings will only affect a single device and non-critical services. The resulting DoS is limited to local functionality, so both likelihood and impact are Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise CES nodes often run many business microservices. If the CRS does not enforce appropriate cgroup limits or priority settings, a single misbehaving or malicious container can starve other services on a node, causing noticeable but tenant-scoped outages, which is reflected as Medium likelihood and Medium impact.
C3	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	High	In UC-C3, container platforms run critical or regulated services at scale. Insufficient isolation or resource governance in the CRS (e.g. no hard limits, shared resources across tenants) enables noisy-neighbour or deliberate resource exhaustion that can degrade or interrupt critical services across multiple tenants and edge sites, so both likelihood and impact are High.
T-CRS-LOG-INT: Logging weakness/tampering														
C1	1	1	1	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-C1, CRS logs (e.g. container start/stop events, seccomp denials) may be minimal or stored locally without protection. Attackers running inside containers could modify or delete such logs, but given the low sensitivity and small scale, the inability to reconstruct activity has limited impact, so risk remains Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise CES platforms rely on CRS logs for troubleshooting and incident response (e.g. detecting abnormal restarts, denied syscalls, mount failures). Weak logging or easy log tampering delays detection of attacks against business microservices and complicates forensics, warranting Medium likelihood and Medium impact.
C3	2	3	3	2	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, logging from the CRS is essential to maintain evidential integrity across critical workloads and tenants. If runtime logs are incomplete, not centrally collected, or modifiable by compromised containers, attackers can conduct stealthy operations (e.g. privilege escalation, forbidden syscalls) without trace across multi-tenant environments. This justifies High likelihood and High impact.
T-CRS-UPD-COMP: Malicious/compromised update														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, CRS updates (e.g. runc, containerd) are often applied manually on single devices, sometimes via direct downloads. A compromised package or replayed older runtime can weaken isolation on that device, but the effect is limited to one low-critical node, so likelihood is Low and impact Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprises may use repositories or CI/CD pipelines to distribute runtime updates. If these are compromised or unsigned, a malicious runtime can be rolled out across many nodes, weakening container isolation and seccomp/cgroup enforcement for all workloads of that tenant. This supports Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, large-scale or automated update pipelines for CRS components mean that a poisoned runtime package can rapidly propagate across nodes hosting critical, multi-tenant workloads, effectively undermining isolation guarantees everywhere it is deployed. Hence both likelihood and impact are High.
T-CRS-CFG-COMP: Insecure or Unauthorized configuration														
C1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-C1, insecure runtime settings (e.g. no default seccomp/AppArmor profile, permissive host mounts) stem mainly from local experimentation or convenience. Misconfigurations are limited to one device and non-critical containers, so although isolation is reduced, overall risk remains Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-C2, runtime configuration (default seccomp profiles, namespace settings, resource limits) is often propagated via container engines or orchestrators. Weak defaults (e.g. unbounded CPU/memory, privileged containers, hostPath mounts) or Unauthorized changes can significantly increase the attack surface across enterprise workloads, giving Medium likelihood and Medium impact.
C3	2	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, CRS configuration is a primary control for maintaining strict isolation and resource separation in critical or regulated environments. Insecure or Unauthorized runtime settings can directly enable container escape, lateral movement and high-impact DoS across tenants, so both likelihood and impact are High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-SC-COMP: Supply chain compromise														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, a compromised runtime component (e.g. unpacker library, sandboxing module) affects a single isolated host. While it may permit container escape or tampering, the limited connectivity and low-value workloads keep the overall impact low; the Likelihood is Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprises may deploy CRS packages from shared repositories or base images. A poisoned dependency embedded in the runtime can introduce persistent backdoors or isolation weaknesses across many enterprise nodes, impacting all internal workloads but still within one organizational tenant, hence Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, CRS supply-chain integrity is critical. Compromise of a low-level runtime dependency (e.g. syscall handling library, sandbox module) used in container platforms for multiple tenants means every container instantiation can execute malicious code with high privileges. This can enable systemic escape and cross-tenant compromise in public or federated CES, leading to High likelihood and High impact.

B.5.2.2 CE

[To be completed in a future version]

B.5.2.3 CO

[To be completed in a future version]

B.5.3 Risk Evaluation of common hardware-level threat T-ALL-HW-ACCESS

The threat T-ALL-HW-ACCESS (Compromise via hardware-level or platform trust components) is common to both Virtualization Execution Stacks (VES) and Container Execution Stacks (CES).

It models compromise of the underlying hardware or platform trust components (CPU, firmware, TPM/HSM/TEE, etc.) in a way that can bypass or undermine software-based security mechanisms, including boot integrity verification, runtime integrity checks, attestation, and isolation controls.

Because hardware-level compromise typically requires a high level of attacker sophistication and is heavily dependent on platform supply-chain assurance, the likelihood remains relatively low to medium, while the impact can be catastrophic for high-assurance, multi-tenant or critical deployments.

Table B.5.3-1 applies the risk assessment methodology of clause B to this threat for the VES and CES use cases.

Table B.5.3-1: Risk evaluation of the common hardware-level threat T-ALL-HW-ACCESS across VES and CES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	1	2	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, hypervisors run on isolated hosts with a small number of non-critical VMs and limited connectivity. Hardware-level compromise (e.g. tampered firmware or physical access) would give full control of that single host, but the data is low-sensitivity and there is no large orchestration blast radius (ISO=1, DAT=1, ORB=1). Supply-chain risk exists (SUP=2), but exposure and attacker incentive (EXT=1, TEN=1, CTL=1, PRV=1) are low. Overall likelihood and impact are therefore assessed as Low.
V2	2	2	2	2	1	2	2	2	2	2	9 → Medium	10 → Medium	Medium	In UC-V2, Type-1/Type-2 hypervisors support general business workloads in enterprise/private clouds and connected edge clusters. Commodity hardware and shared infrastructure increase the relevance of hardware supply-chain issues (EXT=2, TEN=2, SUP=2), but the attack still requires high sophistication (PRV=1). A successful hardware-level compromise affects multiple business VMs and nodes within one organization (ISO=2, DAT=2, ORB=2), and standard enterprise monitoring/recovery exists (DRC=2), so both likelihood and impact are Medium.
V3	3	3	2	3	1	3	3	3	3	3	12 → Medium	15 → High	High	In UC-V3, hypervisors are the foundation of high-assurance, multi-tenant infrastructures processing critical or regulated workloads. These platforms are high-value targets for sophisticated hardware-level attacks (EXT=3, TEN=3, SUP=3) even though such attacks remain complex (PRV=1). A compromised CPU, firmware, or hardware security module can bypass secure boot, attestation, and isolation controls, impacting all tenants and critical services across clusters (ISO=3, DAT=3, ORB=3). Detection and recovery from deep hardware compromise are limited (DRC=3). As a result, likelihood is Medium but impact is High, and the overall risk level is High.
C1	1	1	1	2	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-C1, the CES run on devices with limited connectivity (e.g. hobbyist setups, isolated test/dev hosts) that execute non-critical containers. Hardware-level compromise would allow full control of the host and all containers, but the workloads and data are low-sensitivity and there is no meaningful multi-tenant blast radius (ISO=1, DAT=1, ORB=1). Supply-chain risk exists (SUP=2), but exposure and tenant multiplicity are low (EXT=1, TEN=1, CTL=1, PRV=1), so both likelihood and impact are Low.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C2	2	2	2	2	1	2	2	2	2	2	9 → Medium	10 → Medium	Medium	In UC-C2, the CES supports general business microservices and edge workloads. A hardware-level compromise of the underlying platform (e.g. compromised firmware or SoC) affects all CRS instances and containers on affected nodes, exposing sensitive internal data and services within one organization (ISO=2, DAT=2, ORB=2). Commodity hardware and common supply chains (SUP=2) increase the exposure compared to UC-C1, but attacks still require high sophistication (PRV=1). This results in Medium likelihood and Medium impact.
C3	3	3	2	3	1	3	3	3	3	3	12 → Medium	15 → High	High	In UC-C3, the CES underpins critical or regulated containerized workloads in public, federated, or zero-trust environments, including edge sites performing safety-relevant processing. Hardware-level compromise of CPUs, firmware or platform trust components can silently bypass all CRS and orchestrator controls, enabling arbitrary access to containers and data across many tenants and sites (ISO=3, DAT=3, ORB=3). These platforms are attractive targets for well-resourced attackers (EXT=3, TEN=3, SUP=3), and deep hardware compromise is hard to detect and remediate (DRC=3). Accordingly, likelihood is Medium but impact is High, giving an overall High risk.

Annex (informative): Bibliography

Annex (informative): Change history

Date	Version	Information about changes
June 2025	0.0.1	- Added representative use cases for hypervisors and container runtime systems, classified by risk level. - Defined cumulative security levels (Basic, Elevated, Advanced) to guide requirement selection. - Defined threats - Defined generic security requirements - Defined instantiated security requirements for isolation, integrity, authentication, authorization and confidentiality with assigned security levels across product types (Type 1/2 Hypervisors, CRS). - Added Annex A: Risk assessment methodology - Added Annex B: Mapping between GR requirements and CRA. - Added Annex C: Threat-to-Requirement-to-Control mapping table for traceability. - Added Annex D: Illustrative security controls
August 2025	0.0.2	- Extend the scope to cover the entire container execution stack, including the Container Runtime System (CRS), the Container Engine (CE), and the Container Orchestrator (CO). - Extend the scope to cover the entire Virtualization execution stack, including the hypervisor and the orchestration and management system. - Update scope figures. - Replace "security level" with "security category level". - Add new requirements specific to the container engine (CE) and container orchestrator (CO). - Revise existing CRS requirements to align with the extended scope and updated terminology. - Update the risk assessment methodology in Annex A to: - Clarify how threats are factored into risk determination. - Describe how to assess the overall product risk level. - Introduce an "Assessment" clause, including an example test case for one of the defined security requirements. - Replace the "Assumptions" clause with a new "Operational Environment Objectives" clause.
August 2025	0.0.3	- Updated scope figures. - Revised list of threats, now with specific threats defined per component. - New clause on Objectives for the Operational Environment, replacing the previous <i>Assumptions</i> clause. - New and updated requirements, including: - New requirements for the management and orchestration system. - Updates to isolation and integrity requirements for hypervisors. - Addition of three options for structuring the assessment of requirements: - Test case-based approach - Simplified PT2 - ETSI TS 103 645 / ETSI EN 303 645 approach
September 2025	0.0.4	- Addressed new comments received (see Excel sheet v5). - Updated the titles of representative use cases in Clause 5. - Updated requirement categorization and selection approaches in Clause 6: - The label Enhanced has been replaced by Advanced. - A new option for creating a custom security profile has been introduced for cases where the product use case does not fit one of the representative use cases in the present document, or where the manufacturer wishes to conduct a dedicated risk assessment. - A categorization approach: Cumulative, Exclusive, and Single - has been introduced to clarify the risk-based selection of requirements. - Transformed generic requirements into security objectives (Clause 8). - Updated VES and CES requirements following received comments; additional elements have been added, and the new categorization approach has been applied to a subset of them (see Clauses 10 and 11). - Added new requirements on vulnerability handling and SBOM in Clause 12. - Updated the risk assessment methodology: simplification of risk factors and application of the updated risk factors to the assessment of use cases.

Date	Version	Information about changes
		Moved implementation notes intended as guidance for manufacturers from the requirements clauses to Annex D.
September 2025	0.0.5	- Addressed new comments received from ETSI. - Added a time synchronization objective and related requirements for the hypervisor, with implementation notes included in Annex D. - Completed and updated authentication requirements for the hypervisor, covering both administrative and service authentication. - Annex A: CRA essential requirements have been mapped to the existing hypervisor requirements. - Annex C: Now designated as a normative annex. Threats to hypervisor requirements are partially mapped as a sample for presentation; this mapping will be completed once a stable set of requirements with fixed IDs is available. - Reordered clauses: The clause defining labels, categories, and SCLs, along with the approaches for applying requirements, has been moved to appear just before the requirements clause, making the present document easier to follow. - Updated use case descriptions, including illustrative business applications and revised example environments. - Requirement categorization, labelling, and classification approaches have been clarified for manufacturers. The guidance is now more direct and less perspective-based. - The label Standard has been replaced by Elevated.
October 2025	0.0.6	- Conformance assessment approach described, including evaluation of hypervisor isolation and boot chain integrity requirements. - Incorporated new comments received. - Added definitions of Tenant and Multi-Tenancy. - Slightly updated use cases UC-V3 and UC-C3 to also cover multi-tenancy scenarios within the same organization. - Updated risk assessment methodology applied to UC-V1/2/3, with risk evaluated per threat for hypervisors. - Mapped hypervisor threats to corresponding technical requirements to CRA requirements. - Moved implementation notes to Annex D. - Replaced the 'Label' term with 'Class'.
October 2025	0.0.7	- Update following the ETSI/CEN-CENELEC rapporteurs meeting on the alignment of CRA vertical standards: - Document structure made more compact and logically sequenced. - Assessment approach optimized for consistency. - Addition of a new clause: "Product variants" clarifying the scope of products under the present document, identifying core mandatory components, and distinguishing optional elements. - Update of use cases to remove inconsistencies and confusion related to the terms single-tenant and multi-tenant; consolidation achieved through additional illustrative examples. - All requirements in this version now follow a harmonised structure, with the class-based approach applied consistently across all defined requirements. - Update of requirements leveraging hardware-based technologies to ensure they are described in a more neutral, technology-agnostic, and risk-based manner. - CRS requirements are completed. - Addition of a new table clarifying the applicability of hypervisor requirements to Type I, Type II, and Hybrid architectures. - Application of the new assessment approach to both Hypervisor and CRS requirements related to Isolation and Integrity Protection. - Completion of the mapping between CRA essential requirements and the corresponding Hypervisor and CRS requirements. - Completion of the mapping between identified Threats and corresponding Requirements for both Hypervisor and CRS components.

Date	Version	Information about changes
November 2025	0.0.8	• Addressed most of the comments received during the pre-open consultation. • Addressed most of the comments received from the ETSI team working on the present document. • Updated the use cases section, including the addition of edge deployment scenarios. • Clarified the assessment approach and added requirements for the assessment report and the evidence package. • Added applicability notes to several security requirements where relevant. • Added a new clause, "Roles and Responsibilities," clarifying the responsibilities of the manufacturer and the product integrator. • Added additional assessment cases for confidentiality and availability requirements for Hyper and CRS components. • Updated the applicability table for hypervisor requirements, replacing "partially applicable" with "conditionally applicable." • Added references to prEN 40000-1-1, prEN 40000-1-2, and prEN 40000-1-3. • Referenced prEN 40000-1-3 for SBOM and vulnerability handling requirements. • Updated several requirements following received comments, including changes in classification for some of them.
November 2025	0.0.9	• Added a new hardware-related threat and performed the corresponding risk assessment. • Updated the applicability table for requirements on Type I, Type II and Hybrid hypervisors to clarify that a requirement may be implemented directly by the hypervisor or delegated to other OE components. • Updated the isolation, integrity and confidentiality requirements: Advanced class may leverage software- or hardware-based mechanisms; hardware-assisted requirements are now optional ("may"). • Completed the assessment descriptions for all Hypervisor and CRS requirements. • Updated the risk assessment for the use cases and refined the rationales to better reflect the context of the risk factors and use cases. • Performed an end-to-end alignment in preparation for the HAS review.
December 2025	0.0.10	• Revised based on ETSI editing support comments to improve compliance with ETSI drafting requirements.

History

Document history		
V0.0.6	October 2025	Clean-up done by <i>editHelp!</i> E-mail: mailto:edithelp@etsi.org
V0.0.10	December 2025	Clean-up done by <i>editHelp!</i> E-mail: mailto:edithelp@etsi.org