Jailhouse - HAZOP Report

July 14, 2020

Organisation:	OpenTech EDV Research GmbH
Working Group:	Safety Critical Linux Working Group
Author:	Nicholas Mc Guire, Markus Kreidl
Release:	1
Revision:	0
Revision Number:	1.0
Date:	July 14, 2020
Expires:	---
Ref:	IEC 61508 Ed 2
Status:	extbfreviewed, authorized, complete, Release
Format:	LaTeX
Tracking	GIT
QA:	initial review and authorisation
License:	Creative Commons

0.1 Item ALLOCATE RESOURCES

Origin: top-level

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HL_0 No	1. No allocation	1. No allocation (a) Resource exhausted (b) Resource inaccessible (c) Allocation can't be assigned (meta-data unavailable) (d) Required resource not configured	1. No allocation (a) Usage not possible TP_0 (b) Usage fails TP_0 (c) Allocated resource not accessible TP_0 (d) No allocation attempt takes place TP_0	1. No allocation (a) Usage not possible • HLS_0 (b) Usage fails • HLS_1 (c) Allocated resource not accessible • HLS_2 (d) No allocation attempt takes place • ND	1. No allocation (a) Usage not possible • HLS_3 • HLS_4 (b) Usage fails • HLS_5 (c) Allocated resource not accessible • NC (d) No allocation attempt takes place • HLS_36	TOP_Level	1. No allocation (a) Usage not possible • TP_2 -> HLS_0 • TP_4 -> HLS_4 • TP_1 -> HLS_3 (b) Usage fails • TP_2 -> HLS_1 • TP_4 -> HLS_5 (c) Allocated resource not accessible • TP_3 -> HLS_2 (d) No allocation attempt takes place • TP_0 -> HLS_36	Note: During the review of "Late" the possibility of delay in the form of "timed-out" arose - this is formally covered by the analysis of "No" but might be worth a specific treatment if allocations were to be built with a timeout (which is not expected).
HL_0 More	1. Excess resource allocated 2. Multiple requested resources allocated	1. Excess resource allocated (a) Alignment constraints on resources 2. Multiple requested resources allocated (a) Systematic fault in resource management	1. Excess resource allocated (a) Mild demand growth TN_2 2. Multiple requested resources allocated (a) Incomplete management, loss of resources TN_2	1. Excess resource allocated (a) Mild demand growth • HLS_6 • HLS_7 2. Multiple requested resources allocated (a) Incomplete management, loss of resources • ND	1. Excess resource allocated (a) Mild demand growth • NC 2. Multiple requested resources allocated (a) Incomplete management, loss of resources • HLS_8	TOP_Level	1. Excess resource allocated (a) Mild demand growth • TN_3 -> HLS_6 • TN_4 -> HLS_7 2. Multiple requested resources allocated (a) Incomplete management, loss of resources • TN_3 -> HLS_8	
HL_0 Less	1. Truncated resource	1. Truncated resource (a) Request larger than available resource	1. Truncated resource (a) Usage fails with delay FN_0	1. Truncated resource (a) Usage fails with delay FN_0 • HLS_9	1. Truncated resource (a) Usage fails with delay • HLS_10	TOP_Level	1. Usage fails with delay • TP_4 -> HLS_9 • TP_3 -> HLS_10	
HL_0 As well as	1. Unintended resources also allocated 2. Unintended in-use resources also allocated	1. Unintended resources also allocated (a) Systematic fault in resource management 2. Systematic fault in resource management (a) Unintended in-use resources also allocated	1. Unintended resources also allocated (a) Unintended resources blocked TN_2 2. Systematic fault in resource management (a) Conflict in use of unintended resources FN_0	1. Unintended resources also allocated (a) Unintended resources blocked TN_2 • ND 2. Systematic fault in resource management (a) Conflict in use of unintended resources FN_0 • HLS_11	1. Unintended resources also allocated (a) Unintended resources blocked • HLS_8 2. Systematic fault in resource management (a) Conflict in use of unintended resources • HLS_10	TOP_Level	1. Unintended resources also allocated (a) Unintended resources blocked • TN_2 -> HLS_8 2. Systematic fault in resource management (a) Conflict in use of unintended resources • FN_2 -> HLS_10 • FP_2 -> HLS_11	The excess unrelated resource would effectively be permanently blocked and as it is assumed that no safety related system would ever be designed to allocate 100% of its resources this fault could stay undetected. In this scenarios this effectively is a no-effect bug - note that collisions of the excess resources are handled in other cases
HL_0 Part of	1. Subset of requested resources	1. Subset of requested resources (a) Systematic fault in resource management	1. Subset of requested resources (a) Usage fails TP_0	1. Subset of requested resources (a) Usage fails • HLS_9	1. Subset of requested resources (a) Usage fails • HLS_10 • HLS_11	TOP_Level	1. Subset of requested resources (a) Usage fails • TP_4 -> HLS_9 • TP_3 -> HLS_10 • FN_2 -> HLS_11	Note: the term subset of request pertains to a structured resource which is though used as a whole.
HL_0 Other than	1. Action other than allocation takes place	1. Action other than allocation takes place (a) Systematic fault in resource management	1. Action other than allocation takes place (a) Arbitrary action takes place TP_0, FN_0	1. Action other than allocation takes place (a) Arbitrary action takes place • HLS_12	1. Action other than allocation takes place (a) Arbitrary action takes place • HLS_8 • HLS_13	TOP_Level	1. Action other than allocation takes place (a) Arbitrary action takes place • FP_2 ->HLS_12, • FN_0 -> HLS_8, • TP_4 -> HLS_13	Note 1: it is not clear if unique return values are feasible - for the Jailhouse hypervisor though it may be possible to define return-type classes that allow partial coverage. Note 2: even though HLS_8 has no direct impact on the failure mode it is relevant here due to its relation to HLS_12.
HL_0 Early	NA					TOP_Level		Note: allocation of resources is a causal event chain so early makes no sense here.
HL_0 Late	NA					TOP_Level		Note: allocation is assumed to be a blocking call and thus it may be slow, even too slow (timeout), but would not be late. Timeout would though be handled in "No" not here.
HL_0 Before	NA					TOP_Level		Note: Early allocation of a resource, that is before it is needed, is basically innocent from a safety perspective
HL_0 After	1. Allocation takes place after use of resource	1. Allocation takes place after use of resource (a) Systematic fault in use of allocation	1. Allocation takes place after use of resource (a) Usage not possible TP_0	1. Allocation takes place after use of resource (a) Usage not possible • HLS_1	1. Allocation takes place after use of resource (a) Usage not possible • HLS_3 • HLS_4 • HLS_9	TOP_Level	1. Allocation takes place after use of resource (a) Usage not possible • TP_2 -> HLS_1 • TP_4 -> HLS_3 • TP_1 -> HLS_4 • TP_4 -> HLS_9	
HL_0 Interrupted	1. Delay due to resource contention	1. Delay due to resource contention (a) Priority inversion prevents progress	1. Delay due to resource contention (a) Excessive delay in allocation FN_2	1. Delay due to resource contention (a) Excessive delay in allocation • HLS_17 • HLS_18	1. Delay due to resource contention (a) Excessive delay in allocation • HLS_5	TOP_Level	1. Delay due to resource contention (a) Excessive delay in allocation • TP_5 -> HLS_5 • TP_3 -> HLS_17 • TP_3 -> HLS_18	Note: Allocation is assumed to be a blocking call. Interrupted refers to concurrent events consuming resources and thus indirectly blocking allocation

HL_0 Terminate	1. Allocation terminates prematurely	1. Allocation terminates prematurely (a) Concurrent operation triggers failure	1. Allocation terminates prematurely (a) Incomplete allocation TP_0, FN_2	1. Allocation terminates prematurely (a) Incomplete allocation • HLS_9 • HLS_11 • HLS_12	1. Allocation terminates prematurely (a) Incomplete allocation • HLS_8 • HLS_13	TOP_Level	1. Allocation terminates prematurely (a) Incomplete allocation • TP_4 -> HLS_9 • FN_2 -> HLS_11 • FP_2 -> HLS_12 • FN_0 -> HLS_8 • TP_4 -> HLS_13	Note 1: While this is similar in scope to "Part Of" the difference here is that we assume there is no return value that would allow concluding "correct" operation while a failure occurred. Note 2: The verified allocation protocol does not have any direct mitigation capabilities on its own - but is rather a pre- requisite to the other measures taking effect.
HL_0 No	1. No separation of resources between inmates	1. No separation of resources between inmates (a) Protection mechanism fails silently (b) Systematic fault in resource separation	1. No separation of resources between inmates (a) No protection between inmates - risk elevated FN_0 (b) No protection against malicious inmate FN_0	1. No separation of resources between inmates (a) ND (b) ND		TOP_Level	1. No separation of resources between inmates (a) No protection between inmates - risk elevated • TP_2 -> HLS_14 (b) No protection against malicious inmate • FN_0 -> HLS_15 • FN_0 -> HLS_16	

0.2 Item SEPARATE RESOURCES

Origin: top-level

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HL_13 More	NA					TOP_Level		Note 1: that separation is a binary property - so More (quantitative increase) makes little sense.
HL_13 Less	NA					TOP_Level		
HL_13 As well as	1. Unintended separation of an additional resource 2. Unintended mode of separation	1. Unintended separation of an additional resource (a) Systematic fault in resource description 2. Unintended mode of separation (a) Systematic fault in resource management	1. Unintended separation of an additional resource (a) Systematic fault in resource description FN_0 (b) Unintended blocking of resource FN_2 2. Unintended mode of separation (a) Elevated access rights to restricted resources FN_0	1. Unintended separation of an additional resource (a) Systematic fault in resource description • ND (b) Unintended blocking of resource • ND 2. Unintended mode of separation (a) Elevated access rights to restricted resources • ND	1. Unintended separation of an additional resource (a) Systematic fault in resource description • HLS_15 (b) Unintended blocking of resource • HLS_17 • HLS_18 2. Unintended mode of separation (a) Elevated access rights to restricted resources • HLS_19	TOP_Level	1. Unintended separation of an additional resource (a) Systematic fault in resource description • FN_0 -> HLS_15 (b) Unintended blocking of resource • TP_3 -> HLS_17 • TP_3 -> HLS_18 2. Unintended mode of separation (a) Elevated access rights to restricted resources • TP_2 -> HLS_19	Note 1: blocking could occur in the case that a legitimate user of the resource is no longer able to access to to "re-assignment" Note 2: improper separation is assumed to have achieved an undesired separation - thus a later attempt by the legitimate inmate to allocate or access would fail.
HL_13 Part of	1. Resource is only partially protected	1. Resource is only partially protected (a) Systematic fault in resource managemen	1. Resource is only partially protected (a) Violation of resource access restriction FN_0 (b) Elevated access rights to restricted resources FN_0	1. Resource is only partially protected (a) Violation of resource access restriction • ND (b) Elevated access rights to restricted resources • ND	1. Resource is only partially protected (a) Violation of resource access restriction • HLS_15 • HLS_19 (b) Elevated access rights to restricted resources • HLS_19	TOP_Level	1. Resource is only partially protected (a) Violation of resource access restriction • FN_0 -> HLS_15 • TP_2 -> HLS_19 (b) Elevated access rights to restricted resources • TP_2 -> HLS_19	
HL_13 Other than	1. Action other than separation takes place	1. Action other than separation takes place (a) Systematic fault in resource management	1. Action other than separation takes place (a) Arbitrary action takes place TP_0, FN_0	1. Action other than separation takes place (a) Arbitrary action takes place • HLS_12	1. Action other than separation takes place (a) Arbitrary action takes place • HL_8 • HL_13	TOP_Level	1. Action other than separation takes place (a) Arbitrary action takes place • FP_2 -> HLS_12 • FN_0 -> HLS_8 • TP_4 -> HLS_13	Note 1: it is not clear if unique return values are feasible - for the Jailhouse hypervisor though it may be possible to define return type classes that allow partial coverage
HL_13 Early	NA					TOP_Level		Note 1: separation of resources has a sequential but not a temporal dependency - see "Before/After"
HL_13 Late	NA					TOP_Level		Note 1: separation of resources has a sequential but not a temporal dependency - see "Before/After"
HL_13 Before	NA					TOP_Level		Note 1: it could be that separation actually happens after allocation IFF no other access is possible in the time window between allocation and effectiveness of separation. This would need to be explicitly enforced though.
HL_13 After	1. Separation takes effect while resource in use	1. Separation takes effect while resource in use (a) Race condition during initialization FN_0 (b) Race condition during initialization FN_0	1. Separation takes effect while resource in use (a) Temporarily inadequately protected resources (b) Opportunity for malicious inmate	1. Separation takes effect while resource in use (a) Temporarily inadequately protected resources • ND (b) Opportunity for malicious inmate • HLS_38	1. Separation takes effect while resource in use (a) Temporarily inadequately protected resources • HLS_15 (b) Opportunity for malicious inmate • HLS_37	TOP_Level	1. Separation takes effect while resource in use (a) Temporarily inadequately protected resources • FN_0 -> HLS_15 (b) Opportunity for malicious inmate • TN_2 -> HLS_37 • FP_2 -> HLS_38	Note 1: The initialization sequence should be modeled as a resource state-machine.
HL_13 Interrupted	1. Delay due to resource contention	1. Delay due to resource contention (a) Priority inversion prevents progress	1. Delay due to resource contention (a) Excessive delay in separation setup FN_2	1. Delay due to resource contention (a) Excessive delay in separation setup • HLS_17 • HLS_18	1. Delay due to resource contention (a) Excessive delay in separation setup • HLS_5	TOP_Level	1. Delay due to resource contention (a) Excessive delay in separation setup • TP_3 -> HLS_17 • TP_3 -> HLS_18 • TP_4 -> HLS_5	Note: This is effectively identical to Allocation interruption
HL_13 Terminate	1. Separation setup terminates prematurely	1. Separation setup terminates prematurely (a) Concurrent operation triggers unrecoverable failure	1. Separation setup terminates prematurely (a) Incomplete separation setup TP_0, FN_2	1. Separation setup terminates prematurely (a) Incomplete separation setup • HLS_9 • HLS_19 • HLS_24	1. Separation setup terminates prematurely (a) Incomplete separation setup • HLS_13 • HLS_21 • HLS_22	TOP_Level	1. Separation setup terminates prematurely (a) Incomplete separation setup • TP_4 -> HLS_9 • TP_4 -> HLS_19 • TP_3 -> HLS_21 • TP_4 -> HLS_13 • TP_4 -> HLS_24 • TP_3 -> HLS_22	Note 1: While this is similar in scope to "Part Of" the difference here is that we assume there is no return value that would allow concluding "correct" operation while a failure occurred. Note 2: There are further indications possible - we only list tose that seem to be the strongest ones (e.g. return-value-encodes-function is not listed as we consider it weaker than the listed ones.

HL__13 No	1. Resource not shared	1. Resource not shared (a) Resource inaccessible (b) Inconsistent metadata	1. Resource not shared (a) Usage not possible TP__0 (b) Sharing fails TP__0	1. Resource not shared (a) Usage not possible • HLS__1 (b) Sharing fails • HLS__20	1. Resource not shared (a) Usage not possible • HLS__4 (b) Sharing fails • HLS__5 • HLS__3 • HLS__13	TOP__Level	1. Resource not shared (a) Usage not possible • TP__2 -> HLS__1 • TP__1 -> HLS__4 (a) Sharing fails • TP__2 -> HLS__20 • FIX: TP__2 -> HLS__5 • TP__4HLS__3 • TP__5HLS__13	Note 1: Usage not possible if the local attributes can not be set appropriately Note 2: Sharing fails if the other inmate is unable to establish appropriate privilege levels. Note 3: To make an exchange protocol sensible blocking facilities will be needed.

0.3 Item SHARE RESOURCES

Origin: top-level

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HL_24 More	1. Excess resource shared	1. Excess resource shared (a) Alignment constraints on resources (b) Systematic fault in resource management	1. Excess resource shared (a) Mild demand growth TN_2 (b) Inconsistent management, loss of resources TN_2	1. Excess resource shared (a) Mild demand growth • HL_6 • HLS_21 (b) Inconsistent management, loss of resources • ND		TOP_Level	1. Excess resource shared (a) Mild demand growth • TN_3 -> HL_6 • TN_4 -> HLS_21 (b) Inconsistent management, loss of resources • TN_3 ->HLS_8	Note 1: If the shared resource is allocated at both sides and follows a well defined protocol then size-mismatch of resources should be detectable.
HL_24 Less	1. resource incompletely shared	1. resource incompletely shared (a) Systematic fault in resource management	1. resource incompletely shared (a) Inconsistent management, partial loss of resources TN_2	1. resource incompletely shared (a) Inconsistent management, partial loss of resources • HLS_21	1. resource incompletely shared (a) Inconsistent management, partial loss of resources • HLS_8	TOP_Level	1. resource incompletely shared (a) Inconsistent management, partial loss of resources • TN_3 -> HLS_8 • TN_4 -> HLS_21	Note 1: If the shared resource is allocated at both sides and follows a well defined protocol then inconsistent arrangements of resources should be detectable.
HL_24 As well as	1. Unintended resources also shared	1. Unintended resources also shared (a) Systematic fault in resource management	1. Unintended resources also shared (a) Hidden channel present TN_2 (b) Unexpected modification by other inmate possible FN_0	1. Unintended resources also shared (a) Hidden channel present • HLS_21 • HLS_20 (b) Unexpected modification by other inmate possible • HLS_21 • HLS_11	1. Unintended resources also shared (a) Hidden channel present • HLS_3 • HLS_15 (b) Unexpected modification by other inmate possible • HLS_5 • HLS_15	TOP_Level	1. Unintended resources also shared (a) Hidden channel present • TP_4 -> HLS_3 • FN_0 -> HLS_15 • TP_2 -> HLS_20 • TN_4 -> HLS_21 (b) Unexpected modification by other inmate possible • TP_4 -> HLS_5 • FN_2 -> HLS_11 • FN_0 -> HLS_15 • TN_4 -> HLS_21	Note 1: The initial exchange protocol would need to verify that the initial values are actually the expected invalid initialization values
HL_24 Part of	1. Shared resource modes not consistent	1. Systematic fault in resource management	1. Systematic fault in resource management (a) Usage fails with delay FN_0	1. Systematic fault in resource management (a) Usage fails with delay • HLS_22	1. Systematic fault in resource management (a) Usage fails with delay • HLS_9	TOP_Level	1. Systematic fault in resource management (a) Usage fails with delay • TP_4 -> HLS_22 • TP_3 -> HLS_9	Note 1: Shared resource realm matches intent but modes are not consistent with intent or not consistent across resource. Note 2: It is a bit arbitrary to have "Part of" cover the modes and "Less" cover holes in the resource - the motivation here is that less could describe reduced privileges while "Part of" can cover both reduced and elevated privileges. So this split simplifies things a bit.
HL_24 Other than	1. Action other than allocation takes place	1. Action other than allocation takes place (a) Systematic fault in resource management	1. Action other than allocation takes place (a) Arbitrary action takes place TP_0, FN_0	1. Action other than allocation takes place (a) Arbitrary action takes place • HLS_12 • HLS_20	1. Action other than allocation takes place (a) Arbitrary action takes place • HLS_8 • HLS_13	TOP_Level	1. Action other than allocation takes place (a) Arbitrary action takes place • FP_2 -> HLS_12 • TP_2 -> HLS_20 • FN_0 -> HLS_8 • TP_4 -> HLS_13	Note 1: it is not clear if unique return values are feasible - for the Jailhouse hypervisor though it may be possible to define return type classes that allow partial coverage
HL_24 Early	NA					TOP_Level		Note 1: sharing of resources is done in conjunction with other actions on a resource so "Before" seems more adequate than "Early"
HL_24 Late	NA					TOP_Level		Note 1: A late sharing of resources would effectively look like "No". Note 2: Sharing of resources is done in conjunction with other actions on a resource so "After" seems more adequate than "Late"
HL_24 Before	1. Resource marked shared before allocation	1. Resource marked shared before allocation (a) Allocation silently failed	1. Resource marked shared before allocation (a) Invalid access TP_0	1. Resource marked shared before allocation (a) Invalid access • HLS_20 • HLS_21	1. Resource marked shared before allocation (a) Invalid access • HLS_13	TOP_Level	1. Resource marked shared before allocation (a) Invalid access • TP_4 -> HLS_13 • TP_2 -> HLS_20 • TN_4 -> HLS_21	
HL_24 After	NA					TOP_Level		Note 1: Marking a resource as shared happens after allocation and through allocation protocol, after access, so "late" sharing would not hurt (incorrect sharing is addressed by "Part of","More","No" and "Less")

HL_24 Interrupted	1. Delay due to resource contention	1. Delay due to resource contention (a) Priority inversion prevents progress	1. Delay due to resource contention (a) Excessive delay in Sharing setup FN_2	1. Delay due to resource contention (a) Excessive delay in Sharing setup • HLS_17 • HLS_18	1. Delay due to resource contention (a) Excessive delay in Sharing setup • HLS_5 • HDS_4	TOP_Level	1. Delay due to resource contention (a) Excessive delay in Sharing setup • TP_3 -> HLS_17 • TP_3 -> HLS_18 • TP_4 -> HLS_5 • TP_0 -> HDS_4	Note 1: Sharing resource setup is assumed to be a blocking call. Interrupted refers to concurrent events consuming resources and thus indirectly blocking allocation Note 2: HDS_4 does it self not mitigate but in combination with defined failure response on violation of sequential initialization - which would allow for a concurrent/pseudo- concurrent inter- mediate state - and thus permit inconsistent sharing setup - it can prevent this inconsistency. Note 3: There are potentially further indications/mitigations notably the run time testing - but this does not seem necessary and preferably these fault modes are handled at setup.
HL_24 Terminate	1. Shared resource setup terminates prematurely	1. Shared resource setup terminates prematurely (a) Concurrent operation triggers failure	1. Shared resource setup terminates prematurely (a) Incomplete setup of resource sharing TP_0, FN_2	1. Shared resource setup terminates prematurely (a) Incomplete setup of resource sharing • HLS_9 • HLS_22 • HLS_41	1. Shared resource setup terminates prematurely (a) Incomplete setup of resource sharing • HLS_38	TOP_Level	1. Shared resource setup terminates prematurely (a) Incomplete setup of resource sharing • TP_4 -> HLS_9 • TN_4 -> HLS_22 • TP_0 -> HLS_41 • FP_2 -> HLS_38	Note 1: While this is similar in scope to "Part Of" the difference here is that we assume there is no return value that would allow concluding "correct" operation while a failure occurred. Note 2: As unconfigured IUs would stay unconfigured only if there is no overlap in configuration - non-overlapping is mandated here - this probably should have shown up sooner. While the non-overlapping access modes do not in it self provide any indication/mitigation they are a crucial pre- requiesite here.
HL_24 No	1. IU not configured	1. Configuration failed 2. Resource inaccessible	1. Configuration failed (a) Usage Uncontrolled TP_0, FP_1 2. Resource inaccessible (a) Usage fails TP_0	1. Configuration failed (a) Usage Uncontrolled • HLS_14 • HLS_23 2. Resource inaccessible (a) Usage fails • HLS_19	1. Configuration failed (a) Usage Uncontrolled • HLS_13 2. Resource inaccessible (a) Usage fails • HLS_13	TOP_Level	1. Configuration failed (a) Usage Uncontrolled • TP_4 -> HLS_13 • TP_2 -> HLS_14 • FN_3 -> HLS_23 2. Resource inaccessible (a) Usage fails • TP_4 -> HLS_13 • TP_2 -> HLS_19	Note 1: We assume that no shared resources between inmates as well as between inmate and hypervisor are executable HLS_39

0.4 Item CONFIGURE IU

Origin: top-level

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HL_39 More	NA					TOP_Level		Note 1: A quantitative more for configuration does not seem meaningful More as in "additional" will be treated in "As well as"
HL_39 Less	NA					TOP_Level		Note 1: A quantitative less for configuration does not seem meaningful Less as in "partial" will be treated in "Part of"
HL_39 As well as	1. Unintended configuration of an additional resource 2. Unintended mode of configuration	1. Unintended configuration of an additional resource (a) Systematic fault in IU description 2. Unintended mode of configuration (a) Systematic fault in IU management	1. Unintended configuration of an additional resource (a) Violation of isolation requirements FN_0 (b) Unintended blocking of isolated resource FN_2 2. Unintended mode of configuration (a) Undetected access to restricted resources FN_0	1. Unintended configuration of an additional resource (a) Violation of isolation requirements • HLS_16 • HLS_19 (b) Unintended blocking of isolated resource • HLS_18 2. Unintended mode of configuration (a) Undetected access to restricted resources • ND	1. Unintended configuration of an additional resource (a) Violation of isolation requirements • HLS_15 (b) Unintended blocking of isolated resource • HLS_17 • HLS_18 2. Unintended mode of configuration (a) Undetected access to restricted resources • HLS_19	TOP_Level	1. Unintended configuration of an additional resource (a) Violation of isolation requirements • FN_0 -> HLS_15 • FN_0 -> HLS_16 • TP_2 -> HLS_19 (b) Unintended blocking of isolated resource • TP_3 -> HLS_18 • TP_3 -> HLS_17 2. Unintended mode of configuration (a) Undetected access to restricted resources • TP_2 -> HLS_19	Note 1: blocking could occur in the case that a legitimate user of the resource is denied by the IU Note 2: For safety violations of isolation requirements at the level of an IU would need a double-fault to manifest them selves, for security no double fault can be assumed here as the cause is assumed to be a systematic fault.
HL_39 Part of	1. Resource is only partially isolated 2. IU only partially operational	1. Resource is only partially isolated (a) Mismatch of resource description and IU configuration (b) Configuration incomplete (corrupted) 2. IU only partially operational (a) Improper initialization of IU	1. Resource is only partially isolated (a) Corrupted IU TP_0 (b) Incomplete isolation FN_0 2. IU only partially operational (a) Improper isolation FN_0	1. Resource is only partially isolated (a) Corrupted IU TP_0 • HLS_16 (b) Incomplete isolation FN_0 • HLS_16 • HLS_22 2. IU only partially operational (a) Improper isolation FN_0 • HLS_14	1. Resource is only partially isolated (a) Corrupted IU TP_0 • HLS_9 • HLS_19 (b) Incomplete isolation FN_0 • HLS_9 • HLS_19 2. IU only partially operational (a) Improper isolation FN_0 • HLS_14 • HLS_19	TOP_Level	1. Resource is only partially isolated (a) Corrupted IU TP_0 • TP_1 -> HLS_16 • TP_4 -> HLS_9 • TP_2 -> HLS_19 (b) Incomplete isolation FN_0 • TP_1 -> HLS_16 • TN_4 -> HLS_22 • TP_4 -> HLS_9 • TP_2 -> HLS_19 2. IU only partially operational (a) Improper isolation FN_0 • TP_2 -> HLS_14 • TP_2 -> HLS_19	Note 1: Corrupted essentially means partially unknown or some random seedings Note 2: Incomplete refers to those part that are configured are configured as intended though the overall configuration could still be more or less random with respect to the overall effect (high level intent of isolation or protection) Note 3: Improper refers to the configuration being applied as specified but onto a IU in an undefined state that could thus be randomly misconfigured.
HL_39 Other than	1. Action other than configuration IU takes place	1. Action other than configuration IU takes place (a) Systematic fault in resource management (b) Mismatch of assumed and actual IU configuration space	1. Action other than configuration IU takes place (a) Arbitrary action takes place TP_0, FN_0 (b) Arbitrary configuration takes place TP_0, FN_0	1. Action other than configuration IU takes place (a) Arbitrary action takes place • HLS_12 (b) Arbitrary configuration takes place • HLS_23	1. Action other than configuration IU takes place (a) Arbitrary action takes place • HLS_13 (b) Arbitrary configuration takes place • HLS_13	TOP_Level	1. Action other than configuration IU takes place (a) Arbitrary action takes place • TP_4 -> HLS_13 • TP_2 -> HLS_12 (b) Arbitrary configuration takes place • TP_4 -> HLS_13 • TP_2 -> HLS_23	Note 1: it is not clear if unique return values are feasible - for the Jailhouse hypervisor though it may be possible to define return type classes that allow partial coverage Note 2: The UI startup configuration is assumed to be some detectable default setting
HL_39 Early	NA					TOP_Level		Note 1: configuring an IU is done in conjunction with other actions on a related resource so "Before" seems more adequate than "Early"
HL_39 Late	NA					TOP_Level		Note 1: A late configuring of IU would effectively look like "No" with respect to the errors occurring while unconfigured do not allow ANY successful operations ? Note 2: configuring an IU is done in conjunction with other actions on a related resource so "After" seems more adequate than "Late"
HL_39 Before	NA					TOP_Level		Note 1: Configuring IU is assumed to be the first action during configuration Note 2: This is currently ignoring re-configuration while operations take place - this might need to be revisited.
HL_39 After	1. IU configuration takes effect while related resource in use	1. IU configuration takes effect while related resource in use (a) Incorrect order of IU configuration/resource allocation (b) Race condition during initialization	1. IU configuration takes effect while related resource in use (a) Temporarily inadequately protected resources FN_0 (b) Opportunity for malicious inmate FN_0	1. IU configuration takes effect while related resource in use (a) Temporarily inadequately protected resources • ND (b) Opportunity for malicious inmate • ND	1. IU configuration takes effect while related resource in use (a) Temporarily inadequately protected resources • HLS_24 (b) Opportunity for malicious inmate • HLS_24	TOP_Level	1. IU configuration takes effect while related resource in use (a) Temporarily inadequately protected resources • TP_4 -> HLS_24 (b) Opportunity for malicious inmate • TP_4 -> HLS_24	Note 1: periodic testing could provide a protection for the 2nd case but the diagnostic coverage would be indeterminable and most often very low - so mark it as No Detection.

HL__39 Interrupted	1. Delay due to resource contention	1. Delay due to resource contention (a) Priority inversion prevents progress	1. Delay due to resource contention (a) Excessive delay in IU configuration FN__2	1. Delay due to resource contention (a) Excessive delay in IU configuration • HLS__17 • HLS__18	1. Delay due to resource contention (a) Excessive delay in IU configuration • HLS__5 • HLS__38 • HDS__6	TOP__Level	1. Delay due to resource contention (a) Excessive delay in IU configuration • TP__3 -> HLS__17 • TP__3 -> HLS__18 • TP__4 -> HLS__5 • FP__2 -> HLS__38 • TP__4 -> HDS__6	Note: we are assuming that there would be no side- effects with interruption of configuration - for some actual IUs this may not be a realistic assumption - but that would be handled in the tech aware analysis not at this level.
HL__39 Terminate	1. Premature termination results in incomplete IU configuration	1. Premature termination results in incomplete IU configuration (a) Concurrent operation triggers failure	1. Premature termination results in incomplete IU configuration (a) Incomplete setup of IU TP__0, FN__2	1. Premature termination results in incomplete IU configuration (a) Incomplete setup of IU • HLS__9 • HLS__22 • HLS__40	1. Premature termination results in incomplete IU configuration (a) Incomplete setup of IU • HLS__24 • HLS__38 • HDS__4	TOP__Level	1. Premature termination results in incomplete IU configuration (a) Incomplete setup of IU • TP__4 -> HLS__24 • FP__2 -> HLS__38 • TP__4 -> HLS__9 • TN__4 -> HLS__22 • TP__3 -> HLS__40 • TP__4 -> HDS__4	Note 1: Concurrent operations on IU could be a practical cause for such invalid access.
HL__39 No	1. IU not monitored	1. IU not monitored (a) Data not accessible (b) Signals not delivered (c) Data not stored	1. IU not monitored (a) Status of systems unknown TP__0 (b) Updated status not read FN__0 (c) Trending and analysis not possible FN__0	1. IU not monitored (a) Status of systems unknown • HLS__25 (b) Updated status not read • HLS__25 (c) Trending and analysis not possible • ND	1. IU not monitored (a) Status of systems unknown • HLS__27 (b) Updated status not read • HLS__18 (c) Trending and analysis not possible • HLS__28	TOP__Level	1. IU not monitored (a) Status of systems unknown • TP__2 -> HLS__25 • TP__2 -> HLS__27 (b) Updated status not read • TP__4 -> HLS__18 • TP__3 -> HLS__25 (c) Trending and analysis not possible • TP__2 -> HLS__28	

0.5 Item MONITOR IU

Origin: top-level

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HL_52 More	NA					TOP_Level		Note 1: While it may be possible to have a quantitative increase in the sense of reading more data than should have been read - this would not actually prevent the intent of monitoring.
HL_52 Less	NA					TOP_Level		
HL_52 As well as	1. Logging data from other source also processed 2. Unrelated data also processed	1. Logging data from other source also processed (a) Logging resources misconfigured 2. Unrelated data also processed (a) General resource configuration fault	1. Logging data from other source also processed (a) Inconsistent logs FN_0 2. Unrelated data also processed (a) Corrupted logs FN_0	1. Logging data from other source also processed (a) Inconsistent logs • HLS_26 • HLS_30 2. Unrelated data also processed (a) Corrupted logs • HLS_26 • HLS_30	1. Logging data from other source also processed (a) Inconsistent logs • HLS_29 2. Unrelated data also processed (a) Corrupted logs • HLS_29	TOP_Level	1. Logging data from other source also processed (a) Inconsistent logs • TN_4 -> HLS_26 • TN_3 -> HLS_30 • TP_3 -> HLS_29 2. Unrelated data also processed (a) Corrupted logs • TN_5 -> HLS_26 • TN_3 -> HLS_30 • TP_3 -> HLS_29	Note 1: Inconsistent logs refer to the log message it self being valid but association being wrong or sequence being wrong while the corruption case refers to the actual message being damaged. Note 2: For the corrupted log a CRC would probably be needed to detect but for logs that does seem to be too much consequence of identifiable corrupted logs under the assumption that the length is used in the encapsulation would need to be reviewed (collision probability of length only).
HL_52 Part of	1. Logging data only partially processed	1. Logging data only partially processed (a) Logging resources misconfigured (b) Systematic fault in monitor	1. Logging data only partially processed (a) Inconsistent logs FN_0 (b) Corrupted logs FN_0	1. Logging data only partially processed (a) Inconsistent logs • HLS_26 • HLS_30 (b) Corrupted logs • HLS_26 • HLS_30	1. Logging data only partially processed (a) Inconsistent logs • HLS_29 (b) Corrupted logs • HLS_29 • HLS_31	TOP_Level	1. Logging data only partially processed (a) Inconsistent logs • TN_4 -> HLS_26 • TN_3 -> HLS_30 • TP_3 -> HLS_29 (b) Corrupted logs • TN_4 -> HLS_26 • TP_3 -> HLS_29 • TP_3 -> HLS_31	Note 1: probably testing during development would suffice in the case of the corrupted logs if they are checked for consistency during operations. Note 2: for the simple truncation case a length of message in the encapsulation should do.
HL_52 Other than	1. Action other than monitoring takes place	1. Action other than monitoring takes place (a) Systematic fault in monitor	1. Action other than monitoring takes place (a) Arbitrary action takes place TP_0, FN_0	1. Action other than monitoring takes place (a) Arbitrary action takes place • HLS_12	1. Action other than monitoring takes place (a) Arbitrary action takes place • HLS_32 • HLS_13	TOP_Level	1. Action other than monitoring takes place (a) Arbitrary action takes place • FN_0 -> HLS_32 • TP_4 -> HLS_13 • FP_2 -> HLS_12	Note 1: Monitoring of IU is actually already a high-level (intent level) mitigation if inmates log all critical actions - should this be an inmate SAC ? This probably would need to go into the safety manual as an AoU on the inmate. Note 2: it is not clear if unique return values are feasible - for the Jailhouse hypervisor though it may be possible to define return type classes that allow partial coverage Note 3: corruption or loss of monitoring data would not have any immediate impact on the safety properties of the system though it could result in failing to detect a systematic deviation from intent
HL_52 Early	1. Spurious recording	1. Spurious recording (a) False trigger	1. Spurious recording (a) Empty log message recorded FN_0	1. Spurious recording (a) Empty log message recorded • HLS_27	1. Spurious recording (a) Empty log message recorded • NC	TOP_Level	1. Spurious recording (a) Empty log message recorded • TN_4 -> HLS_27	Note 1: It is assumed that monitoring would be triggered by the IU actions including periodic triggering (HLS_27
HL_52 Late	1. Record lost	1. Record lost (a) Lost trigger	1. Record lost (a) Log message not recorded TP_0	1. Record lost (a) Log message not recorded • HLS_27 • HLS_28	1. Record lost (a) Log message not recorded • NC	TOP_Level	1. Record lost (a) Log message not recorded • TN_4 -> HLS_27 • TP_2 -> HLS_28	Note 1: It is assumed that monitoring would be triggered by the IU actions including periodic triggering (HLS_27) sequence numbering would reveal lost log message.
HL_52 Before	NA					TOP_Level		Note 1: Monitoring is triggered and a spurious trigger is handled in "Early"
HL_52 After	NA					TOP_Level		Note 1: Monitoring always happens after some source triggered a loss of message by is handled in "Late"

HL_52 Interrupted	1. Delay due to resource contention 2. Re-ordering	1. Delay due to resource contention (a) Priority inversion prevents progress 2. Re-ordering (a) Suspend-resume induces re-ordering	1. Delay due to resource contention (a) Excessive delay in IU Monitoring FN_2 2. Re-ordering (a) IU Monitoring sequence order change TP_0	1. Delay due to resource contention (a) Excessive delay in IU Monitoring • HLS_27 • HLS_28 • HLS_42 2. Re-ordering (a) IU Monitoring sequence order change • HLS_26 • HLS_27	1. Delay due to resource contention (a) Excessive delay in IU Monitoring • HLS_29 2. Re-ordering (a) IU Monitoring sequence order change • HLS_29	TOP_Level	1. Delay due to resource contention (a) Excessive delay in IU Monitoring • TN_4 -> HLS_27 • TP_2 -> HLS_28 • TN_4 -> HLS_42 • TP_3 -> HLS_29 2. Re-ordering (a) IU Monitoring sequence order change • TN_4 -> HLS_26 • TN_4 -> HLS_27 • TP_3 -> HLS_29	Note 1: IO Monitoring is active during the system lifetime and no serialization can be assumed. Note 2: entry encapsulation check has different mitigation capabilities as in the delay case there is no followup indicator unless timeouts would be reached (not that likely) while re-ordering would be readily detectable by later analysis/post-processing.
HL_52 Terminate	1. Premature termination of IU Monitoring	1. Premature termination of IU Monitoring (a) Concurrent operation triggers failure (b) Concurrent operation interleaves logs	1. Premature termination of IU Monitoring (a) Incomplete logs TP_0, FN_2 (b) Corrupted logs TP_0, FN_2	1. Premature termination of IU Monitoring (a) Incomplete logs • HLS_27 (b) Corrupted logs • HLS_26	1. Premature termination of IU Monitoring (a) Incomplete logs • HLS_13 • HLS_29 (b) Corrupted logs • HLS_13 • HLS_29	TOP_Level	1. Premature termination of IU Monitoring (a) Incomplete logs • TP_4 -> HLS_13 • TP_3 -> HLS_29 • TN_4 -> HLS_27 (b) Corrupted logs • TP_4 -> HLS_13 • TP_3 -> HLS_29 • TN_4 -> HLS_26	Note 1: Concurrent operations of monitored instances reporting asynchronously is assumed here.
HL_52 No	1. Notification not delivered	1. Notification not delivered (a) Inmate not operational (b) Resources inaccessible (c) Notification ignored by inmate	1. Notification not delivered (a) Notification not possible TP_0 (b) Notification fails TP_0 (c) Notification ineffective FN_0	1. Notification not delivered (a) Notification not possible • HLS_33 (b) Notification fails • HLS_9 (c) Notification ineffective • HLS_34	1. Notification not delivered (a) Notification not possible • HLS_13 (b) Notification fails • HLS_2 • HLS_13 (c) Notification ineffective • HLS_13	TOP_Level	1. Notification not delivered (a) Notification not possible • TP_4 -> HLS_13 • TP_2 -> HLS_33 (b) Notification fails • TP_3 -> HLS_2 • TP_4 -> HLS_13 • TP_4 -> HLS_9 (c) Notification ineffective • TP_4 -> HLS_13 • TP_3 -> HLS_34	Note 1: only if the inmate is safety related would entering safe state on non-operation be mandatory

0.6 SACs HLS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated
HLS_0	Resource monitoring	Resources shall be monitored with respect to used/free	HL_0	down	Complex systems will not allow operation in purely statical resources thus a minimum level of free resources to ensure that critical operations can continue is needed. Further the monitoring supports health monitoring of the system (expected vs non-expected resource load/usage)	TBD	TBD
HLS_1	Resource checking	Resources shall be checked on allocation	HL_0	down	To ensure temporal conditions resources allocated need to be checked for availability	TBD	TBD
HLS_2	Pre allocated meta structure	To ensure resources allocated can actually be managed properly (including failure cases) pre allocated meta-data shall be provided.	HL_0	down	Pre- allocated resource meta-data ensures that the internals of the allocation process can be assured to operate correctly even if a resource is not accessible.	TBD	TBD
HLS_3	Defensive structures	As a general precaution against faults defensive structures shall be used for all resource allocations	HL_0	up	Unhandled allocation resources can lead to more or less undefined behaviors so as a standard umbrella mitigation against faults defensive structures/processes are used.	TBD	TBD
HLS_4	Defined initialization	The system initialization sequence shall be fully defined e.g. via configuration file so that the correctness of initialization can be confirmed before entering actual operational phase	HL_0	down	If the systems state after initialization is not well defined - atleast at the level of resources allocated and privilege levels assigned - it is not possible to asses the success of the initialization process.	TBD	TBD
HLS_5	Defined failure response	The failure response to defined functionality not achieving its intent shall be well defined and consistent at system level. Specifically no collisions of return values and/or ambiguous failure modes shall occur.	HL_0	up	Failure response is a system property and not a local function property so it needs to be defined at a system level and mapped to appropriate system response (e.g. continue/log-and-continue/log-and-halt)	TBD	TBD
HLS_6	Defined alligment constraints	Alignment of data and meta-data shall be well defined	HL_1	up	To allow for simple sanity checking of allocations as well as simplify pre- allocation of meta-data objects alignment needs to be well defined. Note we mark this as "up" because this essentially would need to go into guidelines and from there into implementation.	TBD	TBD
HLS_7	Allocation fills allignment constraints	Resources allocation is strictly on alligment boundaries	HL_1	down	To allow effective checking of adherence to alignment constraints of resources	TBD	TBD
HLS_8	Verified allocation procedure	The allocation procedure shall be verified against the privilege levels for sharing and separation to ensure that allocation is limited to the intended request.	HL_1	down	Verification of allocation needs to ensure that the expected (configured) separation and sharing privilege levels are retained at runtime - this can only be approximately ensured (non- maliciousness assumed) by ensuring consistency of the resources range, privilege level and mode (shared/private)	TBD	TBD
HLS_9	Invalid access triggers response	Access to a legitimate range that was not properly initialized or a resource that is inaccessible (range, mode or privilege level) shall trigger a response.	HL_2	up	The isolation can only be effective if the violation of configured ranges, configured modes or privilege levels triggers a system level (high privileged level) response. The response is not defined though as this depends on the resource as well as the condition (e.g. a non-safe babbling idiot should not be able to shut down the system).	TBD	TBD
HLS_10	Allocation fills resource	Writable resources allocated shall be fully accessed by writing to the full range.	HL_2	down	To ensure temporal constraints and prevent loss of data written to resources during operation.	TBD	TBD
HLS_11	Resources pre initialized to invalid values	To ensure that a non-initialized resource is runtime detectable by the resource manager as well as by inmates, all resources shall be initialized to detectable invalid values.	HL_3	down	While it not be possible to actually determine an invalid value for every resource that can ensure access would fail under all conditions it seems feasible to initiate resources to values that would make it highly likely that non-initialization could be detected by inspection of the resource it self independent of the state of privileged resource management.	TBD	TBD
HLS_12	Return value encodes function	To make detection of a wrong function invocation likely the return values shall strive to be encoded in unique return values that identify the called function.	HL_5	up	While this may not be possible for all system level functions a high level of diversity of return values may allow a very high probability of catching an unintended invocation of a wrong function. Note: This must be resolved at the requirements and design level	TBD	TBD
HLS_13	Mismatch triggers safe state	A mismatch of the return value from the expected range of permissible return values shall trigger a safe state of the system.	HL_5	down	If one assumes that a control flow error occurred and is the cause of the invalid return value, the system is effectively in an undefined state and the only possible reaction is to enter a safe state. The noteworthy exception to this will necessarily be return values received from non-safety related inmates • for that case not the system but the inmate would need to be shut down as that would be the most likely reason for receiving an invalid return value (CFG-fault induced errors are assumed to be very unlikely).	TBD	TBD
HLS_14	Protection periodically invoked	Protection mechanisms shall be periodically invoked during operation to verify that protection is actually still actively enforced.	HL_12	down	Protection mechanisms that are not directly bound to functional behaviors can silently fail without any detectable impact on normal operations however the residual risk of the system without the operational protection is immediately and significantly increased, thus a periodic verification of protection availability is needed to limit the residual risk to a tolerable level.	TBD	TBD
HLS_15	Well defined design process	TBD	HL_12	up	TBD	TBD	TBD
HLS_16	Protection mechanism testing	The protection mechanisms shall be tested under the assumption of malicious inmates with knowledge of the inner workings of protection mechanisms.	HL_12	up	Notably non-safety related inmates must be assumed to be exposed to considerable security threats with the aggravating condition of full access to the inner workings of the protection mechanisms (essentially this is what Schneier defines as prerequisite to security anyway). Only if with this knowledge the separation mechanisms can not be overcome in an unnoticed manner can the safety of the system be assured.	TBD	TBD
HLS_17	Timeout supervision for initialization	Initialization of resources or their configuration shall be protected by a timeout.	HL_15	down	While the time needed for initialization or (re-)configuration of a resource may legitimately vary a upper bounds is needed to assure proper interaction of resources as well as adequate error response. Note that for some cases timeouts also may be relevant to ensure security properties that potentially attack intermediate states of resource (re-)configuration.	TBD	TBD

HLS_18	Timeout supervision for critical operations	All tasks deemed safety related or safety critical shall be supervised by an independent timeout mechanism.	HL_15	down	Any safety related or safety critical task can ultimately fail, thus to allow never the less bounding the system risk a quantifiable maximum response time to the failure of any such task is needed to allow achieving system level tolerable risk.	TBD	TBD
HLS_19	Startup selftest of access level	Startup self tests of resources that provide different access levels (privilege levels and modes of access) shall verify these properties by startup self tests.	HL_15	down	As a minimum for any protection that could possibly fail it needs to be assured that the mechanism atleast initially was operational and responded to violations as expected. This can be assured by startup tests that exercise deliberate privilege or access mode violations. Note that for some relatively short operating units startup self tests may be sufficient to cover the assurance needs for the entire cycle time of the unit (notably automotive with the 1h operation stunt).	TBD	TBD
HLS_20	Initial exchange protocol	Resource separation and sharing shall be verified after configuration by applying an initial exchange protocol for dummy data.	HL_24	down	By exercising the resource in the intended configuration misconfiguration respectively some types of side-effects can be detected before the resource is actually used for operations. This is not only relevant to ensure safety properties but also may touch security needs.	TBD	TBD
HLS_21	Initialization protocol fills resource	To detect invalid order of allocation, configuration and activation of resource access privileges and/or modes, a resource initialization protocol shall attempt modification (writing) the full extent of the resource marked modifiable.	HL_25	down	While operational monitoring would catch invalid access (resource, privilege or mode violation), the protection by runtime measures is conditioned on the initially sound state of resources, thus the initial validity needs to be ensured so as to reduce the reliance of operational checking and response.	TBD	TBD
HLS_22	Initialization protocol accesses resource	To detect invalid order of allocation, configuration and activation of resource access privileges the resource initialization protocol shall access the full extent of the resource in the intended mode.	HL_28	down	While operational monitoring would catch invalid access (resource type, privilege or mode violation), the protection by runtime measures is conditioned on the initially sound state of resources, thus the initial validity needs to be ensured so as to reduce the reliance of operational checking and response.	TBD	TBD
HLS_23	Readback of IU configuration	Any configuration of isolation units shall be read back so as to ensure that the intended settings actually are present in the isolation units configuration.	HL_36	down	Any configuration could have silent failure modes respectively corupt settings in an unexpected way. Writing configurations without reading them back would not allow to rely on the setting from the very start of operations.	TBD	TBD
HLS_24	Unconfigured IU fails all access	As initially isolation units will be unconfigured it shall be ensured that such an unconfigured IU does not allow any modifying or retrieving access of the resource it self.	HL_45	down	IUs will not all be configured instantaneously and may not even be fully configured at system level before some operations commence, thus a basic protection of the unconfigured IU should provided by denying all access to resources that have not actively been enabled by the configuration process.	TBD	TBD
HLS_25	Periodic satus update check	The status of all inmates shall be periodically checked and accordingly system status updated.	HL_48	down	At any point in time the system level health needs to be assured with the uncertainty limited to the last check-period. Note that this interval may need to be adjusted globally to the tolerable residual risk of the highest integrity element.	TBD	TBD
HLS_26	Log entries encapsulated	Log entries shall be encapsulated to allow integrity verification during log post-processing.	HL_51	down	Many of the log messages will probably not be processed in real-time and may also be needed for post- mortem analysis, thus suitable encapsulation should ensure that defective/truncated/corrupted log messages are reasonably reliably detected and do not lead to wrong conclusions.	TBD	TBD
HLS_27	Status with timestamp and sequence number	Log entries shall carry timestamps and sequence numbers to allow detection of lost,duplicate,late or early status messages.	HL_60	down	System health monitoring needs reliable status data on all inmates independent of their safety role to allow detection of system level anomalies. As status messages are simple "OK" statements in the good case the message content needs to be assured current and updated	TBD	TBD
HLS_28	Record periodic keep alive messages	A keep-alive message shall be periodically recorded from each inmate to ensure that the inmate is in some form operational - atleast responsive.	HL_60	down	System health monitoring needs reliable information on the operational capabilities of an inmate. For this a periodic message shall be recorded from each inmate (for non-safe inmates in polling mode to prevent babbling idiot issues) that provides a defined minimum indication of proper operation.	TBD	TBD
HLS_29	Log entries encapsulation check	Log message encapsulation is verified at runtime to ensure validity of log entries.	HL_51	down	While the runtime processing of log messages will need to be limited a minimal check of message validity will be needed to allow deriving global system state from log messages.	TBD	TBD
HLS_30	Inmates carry unique ID	Every inmate activated shall carry a unique ID that identifies it at system level.	HL_51	down	To allow tagging any resource meta-data, log-messages or other objects associated with inmates, each inmate needs an ID that is unique at system level and ensures that the data/event can be reliably traced to the related inmate. Note that this implies that the unique ID is at instance level - with other words restarting an inmate would mandate that a new again system wide unique ID were in use.	TBD	TBD
HLS_31	Monitoring mechanism testing	Any inmate monitoring mechanism shall be runtime testable	HL_52	down	Inmate monitoring is a key aspect in assuring system level properties, but as the inmate monitoring does require some reliable action on the side of the inmate this creates a certain dependency on the processing encapsulated within the inmate, thus the assurance claims only are valid as long as the monitoring of the inmate can be assured by runtime checking. This could be by a reasonable challenge-response type test.	TBD	TBD
HLS_32	Verified monitoring procedure	To assure that the monitoring induced actions actually took place the monitoring process shall be verified.	HL_53	down	Any monitoring actions that are intended to take place on an inmate could also divert unexpectedly thus a defined monitoring process shall ensure that the monitoring actions triggered the expected response which would imply that the procedure is being adhered to. Note that this in it self would not exclude that aside from the intended behavior other behavior also took place though. Again this could possibly be achieved by a reasonable challenge-response type process.	TBD	TBD
HLS_33	Inmate status checkable	The status of an inmate shall be checkable	HL_60	down	While this leaves the status definitions open - what ever these status definitions are, the status of an inmate must be checkable. This also shall include the ability to check status of a failed/inactive inmate!	TBD	TBD
HLS_34	Notification acked by inmate	Notification of an inmate shall be reliably acknowledged by the same	HL_60	down	Requests to an inmate may impact system state (e.g. a request to terminate) - thus any assumptions made after notification only can by reasonably assured if inmates acknowledge recieving the message. Note that this in it self does not necessarily imply that the inmate did anything reasonably with the presented message though.	TBD	TBD

HLS_35	Sequence number on notification	Notifications shall carry a unique sequence number to ensure that out-of-order processing or late/duplicate message processing is detectable	HL_63	down	The reliance on the messaging of an inmate is bound to the ablity to ensure that the inmate actually received the message that is assumed sent, this can be achieved by a unique sequence number on the message respectively on the response.	TBD	TBD
HLS_36	Static resource configuration	Resources configuration shall be statically pre-defined	HL_72	down	It is assumed that at the hypervisor level the resources are partitioned statically and the overall allocation of resources to inmates is known before system launch.	TBD	TBD
HLS_37	Trusted inmates initialized first	All safety related inmates are assumed to be trust able inmates and all resources of the same are to be initialized first.	HL_81	down	For trusted inmates the errors assumed are unintentional or residual technical errors. Thus having those initialize first enables the protection mechanisms in sound state prior to any malicious or deliberate misconfiguration attempts.	TBD	TBD
HLS_38	Defined initialization sequence	Initialization of separation shall take place without any intrusive operations concurrently taking place on the resource being configured, thus initialization shall be strictly defined sequential with respect to separation of resources and actions that may interact with the same.	HL_81	down	Permitting concurrent configuration of separation and operation on those resources could insecure all sorts of hard to handle and hard to analyze corner case therefor we simply disallow any such access during the initial resource configuration.	TBD	TBD
HLS_39	Shared resources not executable	Non-invasive sharing of resources shall be limited to sharing of non-executable resources.	HL_72	down	There would be no way to ensure protection against asynchronous manipulation if a executable is shared between inmates or inmates and the hypervisor it self.	TBD	TBD
HLS_40	Config protection mechanism testing	A configured IU shall be tested before entering operation.	HL_75	down	While general protection mechanism testing as part of the development is called for in HLS_16 to ensure that the mechanism in principle works, this systematic testing can not cover all possible configurations. Given the static nature of the configuration it is reasonable to test precisely the configuration selected with a tolerable effort at time of IU-configuration.	TBD	TBD
HLS_41	Non overlapping resource access modes	Resources configuration shall not allow overlapping of access modes that are incompatible. Appropriate extended modes shall be specified.	HL_35	up	Sharing of resources only makes sense if multiple inmates can actually access the resources. While this access may be encoded in usual modes or R, W, X type permissions - this would not allow detection of unintended overlapping. This implies that modes would need to address this explicidly e.g. RW and RO would need to be extended to RWS (Read-write-shared) and ROS (read-only-shared) even though RW (inmate 0) RO (inmate 1) might seem sufficient, respectively and exclusive E bit might allow prohibiting sharing.	TBD	TBD
HLS_42	Record carries timestamp of record receiving	UI message recordings shall carry the time stamp of when the IU Monitor recorded the message (at least for the periodic keep-alive messages)	HL_58	down	This allows detecting if the IU Monitoring system is operating at a proper load and/or if inversions occurred - even if this only were a post-analysis it would probably suffice.	TBD	TBD

0.7 Item RESOURCE PREP

Origin: allocate-resources

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HD_0 No	1. Resources not prepared (in undefined state)	1. Resources not prepared (in undefined state) (a) Systematic fault in resources	1. Resources not prepared (in undefined state) (a) Initialization out-of-spec FN_0, TP_0 (b) Resource not available FN_0, TP_0	1. Resources not prepared (in undefined state) (a) Initialization out-of-spec • HLS_3 (b) Resource not available • HLS_3 • HLS_1	1. Resources not prepared (in undefined state) (a) Initialization out-of-spec • HDS_0 • HLS_13 (b) Resource not available • HDS_0 • HLS_13	• HLS_2 • HLS_11 • HLS_19	1. Resources not prepared (in undefined state) (a) Initialization out-of-spec • TP_4 -> HLS_3 • TP_4 -> HDS_0 • TP_4 -> HLS_13 (b) Resource not available • TP_5 -> HLS_1 • TP_4 -> HDS_0 • TP_4 -> HLS_13 • TP_4 -> HLS_3	Note 1: Defensive structures are of course only of limited protection against a systematic fault in the design of the element it self. Note 2: Safe state in the context of Resource Prep only can mean "halt" and never reaching operational state. Note 3: HLS_1 protection refers to the followup action of allocate resources.
HD_0 More	NA					• HLS_2 • HLS_11 • HLS_19		Note 1: Resource preparation refers to all resources to a quantitative more makes little sense.
HD_0 Less	NA					• HLS_2 • HLS_11 • HLS_19		Note 1: Resource preparation refers to discretized resources so a reduced coverage is covered in "Part of"
HD_0 As well as	1. Resources modified beyond intent of preparation	1. Resources modified beyond intent of preparation (a) Unknown side effects of intended operations (b) Systematic fault in resource preparation (c) Systematic fault in resource	1. Resources modified beyond intent of preparation (a) Resource in undefined state TP_0, FN_0 (b) Resource systematically modified TP_0, FN_0 (c) Resource systematically modified TP_0, FN_0	1. Resources modified beyond intent of preparation (a) Resource in undefined state • HLS_36 • HDS_1 (b) Resource systematically modified • HDS_1 (c) Resource systematically modified • HDS_1	1. Resources modified beyond intent of preparation (a) Resource in undefined state • HLS_10 • HLS_9 (b) Resource systematically modified • HLS_10 • HLS_9 (c) Resource systematically modified • HLS_10 • HLS_9	• HLS_2 • HLS_11 • HLS_19	1. Resources modified beyond intent of preparation (a) Resource in undefined state • FN_2 -> HLS_10 • TP_4 -> HLS_9 • TP_0 -> HLS_36 • TP_3 -> HDS_1 (b) Resource systematically modified • FN_2 -> HLS_10 • TP_4 -> HLS_9 • TP_3 -> HDS_1 (c) Resource systematically modified • FN_2 -> HLS_10 • TP_4 -> HLS_9 • TP_3 -> HDS_1	Note 1: While HLS_36 has no direct effect on severity it is a pre- requisite and thus listed here.
HD_0 Part of	1. Only part of resource initialized	1. Only part of resource initialized (a) Systematic fault in resource preparation (b) Systematic fault in resource	1. Only part of resource initialized (a) Resource in undefined state TP_0, FN_0 (b) Resource systematically modified TP_0, FN_0	1. Only part of resource initialized (a) Resource in undefined state • HLS_36 • HDS_1 (b) Resource systematically modified • HDS_1	1. Only part of resource initialized (a) Resource in undefined state • HLS_10 • HLS_9 (b) Resource systematically modified • HLS_10 • HLS_9	• HLS_2 • HLS_11 • HLS_19	1. Only part of resource initialized (a) Resource in undefined state • FN_2 -> HLS_10 • TP_4 -> HLS_9 • TP_0 -> HLS_36 • TP_3 -> HDS_1 (b) Resource systematically modified • FN_2 -> HLS_10 • TP_4 -> HLS_9 • TP_3 -> HDS_1	Note 1: Filling resource pertains to the configuration boundaries, if some resources stay unconfigured that is in principle possible if the configuration file did not address these.
HD_0 Other than	1. Action other than resource preparation takes place	1. Action other than resource preparation takes place (a) Systematic fault in resource preparation	1. Action other than resource preparation takes place (a) Arbitrary action takes place TP_0, FN_0	1. Action other than resource preparation takes place (a) Arbitrary action takes place • HLS_12 • HLS_36 • HDS_1	1. Action other than resource preparation takes place (a) Arbitrary action takes place • HLS_8 • HLS_13	• HLS_2 • HLS_11 • HLS_19	1. Action other than resource preparation takes place (a) Arbitrary action takes place • TN_2 -> HLS_8 • TP_4 -> HLS_13 • FP_3 -> HLS_12 • TP_0 -> HLS_36 • TP_3 -> HDS_1	Note 1: it is not clear if unique return values are feasible - for the Jailhouse hypervisor though it may be possible to define return type classes that allow partial coverage Note 2: even though HLS_8 has no direct impact on the failure mode it is relevant here due to its relation to HLS_12
HD_0 Early	NA					• HLS_2 • HLS_11 • HLS_19		Note 1: Resource preparation is the first action in the system thus early makes no sense Note 2: This may change during design refinement - in that case the interpretation may need a review
HD_0 Late	1. Resource preparation slowed	1. Resource preparation slowed (a) Resource defective	1. Resource preparation slowed (a) Excessive delay	1. Resource preparation slowed (a) Excessive delay • HDS_2	1. Resource preparation slowed (a) Excessive delay • HDS_3	• HLS_2 • HLS_11 • HLS_19	1. Resource preparation slowed (a) Excessive delay • TP_5 -> HDS_3 • FN_1 -> HDS_2	Note 1: iWDT is assumed to transit system into safe state
HD_0 Before	NA					• HLS_2 • HLS_11 • HLS_19		Note 1: Resource preparation is the first action in the system thus early makes no sense Note 2: This may change during design refinement - in that case the interpretation may need a review
HD_0 After	NA					• HLS_2 • HLS_11 • HLS_19		Note 1: It does not seem reasonable to assume that the resource preparation would take place at some later stage since all resources depend on preparation - covered in "Part of"

HD_0 Interrupted	1. Resources not prepared (in undefined state)	1. Resources not prepared (in undefined state) (a) Concurrent task preempts preparation	1. Resources not prepared (in undefined state) (a) Resource not available FN_0, TP_0	1. Resources not prepared (in undefined state) (a) Resource not available • HLS_1 • HDS_4	1. Resources not prepared (in undefined state) (a) Resource not available • HDS_6	• HLS_11 • HLS_19 • HLS_2	1. Resources not prepared (in undefined state) (a) Resource not available • TP_4 -> HDS_6 • TP_3 -> HLS_1 • TP_0 -> HDS_4	
HD_0 Terminate	1. Resource preparation terminates prematurely	1. Resource preparation terminates prematurely (a) Concurrent operation triggers failure	1. Resource preparation terminates prematurely (a) Incomplete preparation TP_0, FN_2	1. Resource preparation terminates prematurely (a) Incomplete preparation • HDS_0 • HDS_1	1. Resource preparation terminates prematurely (a) Incomplete preparation • HLS_13	• HLS_2 • HLS_11 • HLS_19	1. Resource preparation terminates prematurely (a) Incomplete preparation • TP_4 -> HDS_0 • TP_2 -> HDS_1 • TP_4 -> HLS_13	
HD_0 No	1. Monitoring does not take place	1. Monitoring does not take place (a) Monitor not launched (b) Monitor failed	1. Monitoring does not take place (a) No monitoring TP_0 (b) Incomplete monitoring TP_0	1. Monitoring does not take place (a) No monitoring • HDS_4 (b) Incomplete monitoring • HDS_5	1. Monitoring does not take place (a) No monitoring • HDS_2 • HDS_6 (b) Incomplete monitoring • HDS_2 • HDS_6	• HLS_0 • HLS_28 • HLS_31 • HLS_32	1. Monitoring does not take place (a) No monitoring • FN_1 -> HDS_2 • TP_4 -> HDS_6 • TP_4 -> HDS_4 (b) Incomplete monitoring • FN_1 -> HDS_2 • TP_4 -> HDS_6 • TP_5 -> HDS_5	Note: While HDS_4 has no effect on its own it is effective in concert with HDS_6

0.8 Item SYSTEM MONITOR

Origin: monitor-iu

GW	Interpretation	Cause	Consequence	Indication	Mitigation	Consolidates	Severity	Question
HD__12 More	1. Monitor pats iWDT at high frequency 2. Monitor polls state at high frequency	1. Monitor pats iWDT at high frequency (a) Time-base problems 2. Monitor polls state at high frequency (a) Time-base problems	1. Monitor pats iWDT at high frequency (a) Monitor block system TP__0 2. Monitor polls state at high frequency (a) System overload TP__0 (b) Excessive logs TP__0	1. Monitor pats iWDT at high frequency (a) Monitor block system • HDS__7 2. Monitor polls state at high frequency (a) System overload • HLS__14 • HLS__25 (b) Excessive logs • HLS__29	1. Monitor pats iWDT at high frequency (a) Monitor block system • HDS__2 • HLS__18 2. Monitor polls state at high frequency (a) System overload • HDS__2 • HLS__18 (b) Excessive logs • HDS__2 • HLS__18	• HLS__0 • HLS__28 • HLS__31 • HLS__32	1. Monitor pats iWDT at high frequency (a) Monitor block system • FN__1 -> HDS__2 • TP__3 -> HLS__18 • TP__0 -> HDS__7 2. Monitor polls state at high frequency (a) System overload • FN__1 -> HDS__2 • TP__3 -> HLS__18 • TP__3 -> HLS__14 • TP__3 -> HLS__25 (b) Excessive logs • FN__1 -> HDS__2 • TP__3 -> HLS__18 • TP__3 -> HLS__29	Note 1: HDS__7 has no direct effect but rather is a constraint on the selected iWDT Note 2: It is assumed that while the Monitor is operating at a too high frequency it would still be checking status and if inmates make no progress then their status would indirectly trigger safe state transition (hence TP__3)
HD__12 Less	1. Monitor polls state at to low frequency		1. Monitor polls state at to low frequency (a) Fault tolerance times violated TP__0	1. Monitor polls state at to low frequency (a) Fault tolerance times violated • HDS__5	1. Monitor polls state at to low frequency (a) Fault tolerance times violated • HDS__2	• HLS__0 • HLS__28 • HLS__31 • HLS__32	1. Monitor polls state at to low frequency (a) Fault tolerance times violated • TP__5 -> HDS__5 • FN__1 -> HDS__2	
HD__12 As well as	NA					• HLS__0 • HLS__28 • HLS__31 • HLS__32		Note 1: While it could happen that the monitor monitors something that is not present this is considered a somewhat obscure case aside from it not having a credible impact on the system.
HD__12 Part of	1. Part of the resources are monitored 2. A resource is only partially monitored	1. Part of the resources are monitored (a) Incorrect configuration of monitor 2. A resource is only partially monitored (a) Systematic fault in monitor	1. Part of the resources are monitored (a) i. Not all resources monitored FN__0, TP__0 2. A resource is only partially monitored (a) i. Resource incompletely monitored FN__0, TP__0	1. Part of the resources are monitored (a) i. Not all resources monitored • HDS__8 2. A resource is only partially monitored (a) i. Resource incompletely monitored • HLS__29	1. Part of the resources are monitored (a) i. Not all resources monitored • HDS__9 2. A resource is only partially monitored (a) i. Resource incompletely monitored • HLS__26	• HLS__0 • HLS__28 • HLS__31 • HLS__32	1. Part of the resources are monitored (a) i. Not all resources monitored • TP__4 -> HDS__9 • TP__0 -> HDS__8 2. A resource is only partially monitored (a) i. Resource incompletely monitored • TN__4 -> HLS__26 • TP__3 -> HLS__29	Note 1: It is perceivable that in addition to the minimum logset other resource logs would be obtained it is though assumed that the safety related ones are predefined and not dynamically negotiated.
HD__12 Other than	1. Action other than monitoring takes place	1. Action other than monitoring takes place (a) Systematic fault in system monitor	1. Action other than monitoring takes place (a) Arbitrary action takes place TP__0, FN__0	1. Action other than monitoring takes place (a) Arbitrary action takes place • HLS__12 • HLS__3	1. Action other than monitoring takes place (a) Arbitrary action takes place • HLS__13	• HLS__0 • HLS__28 • HLS__31 • HLS__32	1. Action other than monitoring takes place (a) Arbitrary action takes place • TP__4 -> HLS__13 • FP__3 -> HLS__12 • TP__4 -> HLS__3	
HD__12 Early	NA					• HLS__0 • HLS__28 • HLS__31 • HLS__32		Note 1: System monitoring is periodic so "Early" is addressed in "More" interpreted as "too high frequency"
HD__12 Late	NA					• HLS__0 • HLS__28 • HLS__31 • HLS__32		Note 1: System monitoring is periodic so "Late" is addressed in "Less" interpreted as "too low frequency"
HD__12 Before	NA					• HLS__0 • HLS__28 • HLS__31 • HLS__32		Note 1: System monitoring is a conceptually asynchronous action in the system and not in an order relation to other actions.
HD__12 After	NA					• HLS__0 • HLS__28 • HLS__31 • HLS__32		Note 1: System monitoring is a conceptually asynchronous action in the system and not in an order relation to other actions.
HD__12 Interrupted	1. System monitor delayed 2. System monitor events reorder	1. System monitor delayed (a) Concurrent task preempts monitor 2. System monitor events reorder (a) Concurrent monitor event preempts running monitor event	1. System monitor delayed (a) Monitoring event delayed FN__0, TP__0 2. System monitor events reorder 2. Monitoring event reordered FN__0, TP__0	1. System monitor delayed (a) Monitoring event delayed • HDS__5 2. System monitor events reorder 2. Monitoring event reordered • HLS__26	1. System monitor delayed (a) Monitoring event delayed • HDS__2 2. System monitor events reorder 2. Monitoring event reordered • HDS__9	• HLS__0 • HLS__28 • HLS__31 • HLS__32	1. System monitor delayed (a) Monitoring event delayed • TP__4 -> HDS__5 • TP__4 -> HDS__2 2. System monitor events reorder 2. Monitoring event reordered • TN__4 -> HLS__26 • TP__4 -> HDS__9	Note 1: The monitor is already the second line of defense so we refrain from listing all possible indication/mitigation and focus on a set seen as sufficient.

HD__12 Terminate	1. System monitor terminates prematurely 2. System monitor event handling terminates prematurely	1. System monitor terminates prematurely (a) Concurrent task blocks monitor 2. System monitor event handling terminates prematurely (a) Concurrent monitor event blocks/terminates running monitor event	1. System monitor terminates prematurely (a) Monitoring fails FN__0, TP__0 2. System monitor event handling terminates prematurely (a) Monitoring event lost FN__0, TP__0	1. System monitor terminates prematurely (a) Monitoring fails • HDS__5 2. System monitor event handling terminates prematurely (a) Monitoring event lost • HLS__26 • HLS__27	1. System monitor terminates prematurely (a) Monitoring fails • HDS__2 2. System monitor event handling terminates prematurely (a) Monitoring event lost • HDS__9	• HLS__0 • HLS__28 • HLS__31 • HLS__32	1. System monitor terminates prematurely (a) Monitoring fails • TP__4 -> HDS__5 • TP__4 -> HDS__2 2. System monitor event handling terminates prematurely (a) Monitoring event lost • TN__4 -> HLS__26 • TN__4 -> HLS__27 • TP__4 -> HDS__9	
HD__12 No	1. Watchdog not active	1. Watchdog not active (a) Watchdog failed silently	1. Watchdog not active (a) Risk reduction not assured TP__0	1. Watchdog not active (a) Risk reduction not assured • HDS__10 • HDS__5	1. Watchdog not active (a) Risk reduction not assured • HDS__2	• HLS__14 • HLS__17 • HLS__18 • HLS__25 • HLS__28	1. Watchdog not active (a) Risk reduction not assured • TP__5 -> HDS__10 • TP__5 -> HDS__5 • FN__1 -> HDS__2	Note 1: The risk reduction that is implemented by different measures and monitored by the watchdog would not be assured if the watchdog had silently.

0.9 SACs HDS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated
HDS__0	Preperation self checking	Resource preparation actions shall self-check all operations	HD__0	down	As at this point no system level monitoring is present that could protect against deviation only self-protection can ensure desired properties. On failure of self-checking a safe-state transition is to be triggered.	TBD	TBD
HDS__1	Allocation checks initialization on allocation	Allocation shall check presence of expected canary (value) of resources	HD__3	down	Essentially there is little that can be done against the resource preparation failing silently other than detecting its expected outcome - the resources (where possible) set an pre- defined "invalid" or carnary value.	TBD	TBD
HDS__2	Independent watchdog supervision	An independent watchdog (iWDT) shall be provided	HD__7	down	While self-checking may catch some faults, serious diversions from expected behavior mandate detection by an independent observer. For this stage a relatively simple iWDT seems sufficient.	TBD	TBD
HDS__3	Resource preperation pats iwdt	The resource preparation shall pat the independent watchdog at suitable intervals.	HD__7	down	At this point in the system bring-up there is not other protection to prevent faults leading to inactivity or extensive delays - both of which indicate a significant diversion from intent thus safe-state transition is the only reasonable response	TBD	TBD
HDS__4	Sequencial initialization state tracking	The initialization sequence states shall be tracked by all initialization steps.	HD__12	down	Transition of strictly serialized initialization shall not be allowed out-of-order or with partial completion - thus the state of initialization is tracked allowing followup steps to detect incomplete initialization phases.	TBD	TBD
HDS__5	Monitor pats iwdt	The system monitor shall periodically path the iWDT	HD__12	down	Overload situations as well as failure situations of the system monitor can be detected if the periodic patting of the iWDT is mandated. Note that other internal actions of the system monitor may be handled by conditionally patting the iWDT as well. Note that the system monitor may generally use the Watchdog for assurance - the iWDT shall through also be addressed with a defined frequency to have a second indication source.	TBD	TBD
HDS__6	Initialization sequence violation triggers safe state	The system monitor shall periodically path the iWDT	HD__12	down	Overload situations as well as failure situations of the system monitor can be detected if the periodic patting of the iWDT is mandated. Note that other internal actions of the system monitor may be handled by conditionally patting the iWDT as well.	TBD	TBD
HDS__7	iWDT configurable maximum frequency	The iWDT shall also trigger on too high frequency	HD__13	down	Babbling idiot effects as well as time-base issues can result in too high patting frequency as an indicator of the system operating out-of-spec	TBD	TBD
HDS__8	Minimum monitored resource set predefined	The minimum mandatory set of resources to be runtime monitored shall be statically pre- defined.	HD__16	down	Detection of a non-monitored resource would be hard if the list of resources is not know. For safety critical inmates it can be assumed that the resources to be monitored are predetermined and thus the system monitor can use that basis to ensure that critical resources are monitored.	TBD	TBD
HDS__9	Incomplete logging triggers safe state	Incomplete logging events of the watchdog (sequence numbers) shall trigger safe state transition	HD__16	down	As there are multiple events to supervise not all events can be bound to relative timestamps - but all events must take place so if an inmates logs are incomplete it is an indication of a fault in the system that potentially may lead to missing an event that should be acted upon -> safe state transition initialized as the only option as the severity of the missed event is unknown.	TBD	TBD
HDS__10	Watchdog pats iwdt	The watchdog shall periodically path the iWDT	HD__24	down	Failure of the watchdog function or overload of the same would lead to not satisfying the minimum confirmation frequency and thus allow the iWDT to detect grave deviations from intent of the watchdog. This two layer approach allso the iWDT to be relatively simple. Note that the system monitor also directly interacts with the iWDT to achieve a higherassurance than would be achievable with the watchdog only due to its low achieved coverage.	TBD	TBD

0.10 SACs UTS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated

0.11 SACs UDS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated

0.12 SACs STS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated

0.13 SACs SDS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated

0.14 SACs SDFS

SAC ID	Short SAC	SAC	Origin	Direction	Rational	Consolidates	Consolidated