**3GPP TSG-SA3 Meeting #123 S3-253016**

Goteborg, Sweden, 25 – 29 August 2025

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **33.117** | **CR** | **Draft CR** | **rev** | **-** | **Current version:** | **19.2.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network |  |

|  |
| --- |
|  |
| ***Title:***  | Iiving document for TS 33.117 |
|  |  |
| ***Source to WG:*** | Huawei, HiSilicon, MITRE-FFRDC, US National Security Agency, Deutsche Telekom, BSI (DE) |
| ***Source to TSG:*** | S3 |
|  |  |
| ***Work item code:*** |  |  | ***Date:*** | 2025-08-18 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-20 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | S3-253015:For clause 4.2.2.2.3.1Access token verification tests do not include verification of the access token subject (i.e., NF service consumer).The token verification steps in TS 33.501 clause 13.4.1.1 are not aligned with the token verification steps in the test case in clause 4.2.2.2.3.1 TC\_AUTHORIZATION\_TOKEN\_VERIFICATION\_FAILURE\_ONE\_PLMN. Two token verification checks are not addressed in this test case including:1. *In the direct communication case, it checks that the NF Instance ID in the subject claim within the access token matches the NF Instance ID in the subjectAltName in the NF Service Consumer's TLS client certificate.*
2. *If the CCA is present in the service request, it may verify the CCA as specified in clause 13.3.8.3 and that the subject claim (i.e., the NF Instance Id of the NF Service Consumer) in the access token matches the subject claim in the CCA.*

Two new test cases F & G are added to this test case to address these access token verification steps that are specified in TS 33.501.Since this test case applies to direct communication in NF-NF communications, the token verification check in 1) is applicable under the pre-condition that TLS is used for authentication. Thus, a pre-condition of mutual authentication using TLS is added for test case F only.Similarly, the token verification check in 2) is applicable under the pre-condition that a *CCA is present in the service request*. Thus, a pre-condition of CCA being present in the service request is added for test case G only.This test case is applicable to both Direct and Indirect communication modes. Thus, the pre-condition that requires mutual authentication cannot hold true in indirect communication according to the specified text in TS 33.501 clause 13.3.2.2 Indirect Communication which says in *NOTE 0: Mutual authentication between NF Service Consumer and NF Service Producer is not achieved with hop-by-hop security.* Therefore, this pre-condition must be altered to hold true in indirect communication. To this end, the network product under test can utilize implicit authentication of the NF service consumer, according to TS 33.501 clause 13.3.2.2 which states *In indirect communication scenarios, the NF Service Producer and NF Service Consumer shall use implicit authentication by relying on authentication between NF Service Consumer and SCP, and between SCP and NF Service Producer, provided by the transport layer protection solution, NDS/IP, or physical security.* Therefore, the pre-condition on mutual authentication is changed to say, “The network product under test has already authenticated the NF service consumer.”S3-252584For steps 4 and 7, the NP shall not send a response back to the tester because the message is supposed to be discarded.S3-252585SS7 and SIGTRAN may no longer be supported and should be marked as such as they may be insecure or may be used to circumvent security measures.S3-253018The interface is defined as an IP address and not explicitly mentioning a port. This can lead to a service being reachable on the same IP address but different ports.S3-253020Multiple test steps for web server test cases relate to web server configuration files to be checked.A web server may not always be a standalone application but could be directly integrated into the network product, missing a web server configuration file and use hardcoded web server settings.S3-253021The test case in chapter 4.3.6.4 does list all requirements for IEs, however does not explicitly define test cases with execution steps and concrete expected results.The status codes are specified to ensure the NF implements the security mechanism needed to fulfil the limitations in 29.501 and the status codes are specified to ensure the NF implements the security mechanisms needed to hold up 29.501.S3-252974Access to firmware should be given to the privileged users only post succesful authetication. |
|  |  |
| ***Summary of change:*** | S3-253015:The following changes are made to clause 4.2.2.2.3.1:1. The Requirement Description is modified to align to the requirements in TS 33.501.
2. The pre-conditions are unified for both Indirect and Direct communication modes by replacing the mutual authentication pre-condition with client authentication performed by the network product under test.
3. Additional Pre-conditions are added, which are only applicable for test cases F & G.

Two new test cases F & G are added to the test case in clause 4.2.2.2.3.1, to address the access token subject verification steps that are specified in TS 33.501.S3-252584Correct TC to indicate no response is sent back to tester after the message is discarded. Rather than “any” response.S3-252585Adds SS7 and SIGTRAN in services that are disabled by default.S3-253018Including IP address and port in the configuration of the interface under testS3-253020Added a NOTE to several test cases to notify the tester to omit test steps in such cases.S3-253021The test case TC\_IE\_VALUE\_FORMAT is split into four test cases for testing the valid format and range of values for IEs in HTTP requests. The execution steps are concretized.S3-252974In the current version of specification, access to firmware/boatloader is given after successful authentication. In worst case, this might be used by a ‘normal user’. From security perspective, the access to write/modify should be given to privilged user only upon successful authentication. |
|  |  |
| ***Consequences if not approved:*** | S3-253015:The required token verification steps in TS 33.501 clause 13.4.1.1, including verification of access token subject, are not met in the test case for NF access token authorization. Additionally, the test case pre-conditions do not hold true for both Indirect and Direct Communications mode.S3-252584Tester may assume a response from network product is expected.S3-252585SS7 and SIGTRAN may be allowed by default and could be used by threat actors to compromise the security of the network product.S3-253018The tester may pass the test when services are reachable on the same IP address but not the same port.S3-253020Test cases may fail due to the lack of web server configuration files regardless of the web server following the security requirements.S3-253021The execution of the test case specification by different testers could differ greatly, which could lead to different results and interpretations of the test case.S3-2529743GPP specification is ambigious which may lead to security consequences, if not addressed as proposed. |
|  |  |
| ***Clauses affected:*** | 4.2.2.2.3.1, 4.2.3.3.2, 4.2.6.2.3, 4.2.6.2.4, 4.3.2.1, 4.3.2.2, 4.3.4.2, 4.3.4.4, 4.3.4.5, 4.3.4.6, 4.3.4.7, 4.3.4.10, 4.3.4.12, 4.3.4.14, 4.3.6.4 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **x** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **x** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | The merger of S3-253015, S3-252584,S3-252585,S3-253018, S3-253020, S3-253021, S3-252974 |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

###### 4.2.2.2.3.1 Authorization token verification failure handling within one PLMN

*Requirement Name*: Authorization token verification failure handling within one PLMN

*Requirement Reference:* TS 33.501 [10], clause 13.4.1.1

*Requirement Description*:

According to TS 33.501 [10], clause 13.4.1.1, the NF Service producer verifies the access token as follows:

- The NF Service producer ensures the integrity of the access token by verifying the signature using NRF’s public key or checking the MAC value using the shared secret. If integrity check is successful, the NF Service producer verifies the claims in the access token as follows:

NOTE: Void.

- In the direct communication case, it checks that the NF Instance ID in the subject claim within the access token matches the NF Instance ID in the subjectAltName in the NF Service Consumer's TLS client certificate.

- It checks that the audience claim in the access token matches its own identity or the type of NF service producer. If a list of NSSAIs or list of NSI IDs is present, the NF service producer checks that it serves the corresponding slice(s).

- If an NF Set ID present, the NF Service Producer checks the NF Set ID in the claim matches its own NF Set ID.

- If the access token contains "additional scope" information (i.e. allowed resources and allowed actions (service operations) on the resources), it checks that the additional scope matches the requested service operation.

- If scope is present, it checks that the scope matches the requested service operation.

- It checks that the access token has not expired by verifying the expiration time in the access token against the current data/time.

- If the CCA is present in the service request, it can verify the CCA as specified in clause 13.3.8.3 and that the subject claim (i.e., the NF Instance Id of the NF Service Consumer) in the access token matches the subject claim in the CCA.

- If the verification is successful, the NF Service producer executes the requested service and responds back to the NF Service consumer. Otherwise, it replies based on OAuth 2.0 error response defined in RFC 6749 [12]. The NF service consumer optionally stores the received token(s). Stored tokens may be re-used for accessing service(s) from producer NF type listed in claims (scope, audience) during their validity time.

*Threat References*: TR 33.926 [4], clause 6.3.3.1, Incorrect Verification of Access Tokens

*Test Case*:

**Test Name:** TC\_AUTHORIZATION\_TOKEN\_VERIFICATION\_FAILURE\_ONE\_PLMN

**Purpose:**

Verify that the NF service producer does not grant service access if the verification of authorization token from a NF service consumer in the same PLMN fails.

**Procedure and execution steps:**

**Pre-Conditions:**

- The tester shall know if the network product supports the following optional access token verification claims. If an optional claim is not supported, the associated sub-test case does not apply:

- S-NSSAI (Test Case H)

- NSI (Test Case I)

- NF Set ID (Test Case J)

- additional scope (Test Case K)

- Test environment with a NF service consumer.

- The NF service consumer may be simulated.

- The network product under test has already authenticated the NF service consumer.

- If either of the following pre-conditions are met, the associated sub-test case applies:

- The network product under test has already mutually authenticated with the NF service consumer using TLS certificate, as specified in TS 33.501[10] clause 13.1.0 (Test Case F).

- A Client Credentials Assertion (CCA) is present in the service request and verified as specified in TS 33.501[10] clause 13.3.8.3, and the Network product supports CCA based authentication (Test Case G).

- The tester has access to the interface between the NF service consumer and the network product under test.

- The tester has the NRF’s private key or the shared key.

- The network product under test is preconfigured with the NRF’s public key or the shared key.

**Execution Steps**

The network product under test receives the access token sent from the NF service consumer, verifies the access token based on OAuth 2.0.

Test Cases A~G are tests on failure handling by the network product under test when the mandatory claims in access token failed verification.

Test Case A: No access token

1) The tester sends a request without a token to the network product under test.

2) The network product under test recognized the absence of the access token and the verification of the access token fails.

Test Case B: Verification failure of the access token integrity

1) The tester computes an access token correctly, except that the signature or the MAC is incorrect, e.g., the signature or the MAC is randomly selected, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.

2) The integrity verification of the access token by the network product under test fails.

Test Case C: Incorrect audience claim in the access token

1) The tester computes an access token correctly, except that the audience claim is incorrect, i.e., the audience claim in the access token does not match the identity or the type of the network product under test, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token is valid. However, the audience claim in the access token does not match its identity or type.

Test Case D: Incorrect scope claim in the access token

1) The tester computes an access token correctly, except that the scope is incorrect, i.e., the scope does not match the requested service operation, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token and the audience claim are valid. However, the scope does not match the requested service operation.

Test Case E: Expired access token

1) The tester computes an access token correctly, except that the expiration time has expired against the current data/time, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience and scope claims are all valid. However, the expiration time in the access token has expired against the current data/time.

Test Case F: Access token subject claim does not match the TLS certificate

1) The tester computes an access token correctly, except that the subject claim does not match the corresponding field in the TLS certificate, i.e., the NF Instance ID in the subject claim within the access token does not match the NF Instance ID in the subjectAltName in the NF Service Consumer's TLS client certificate (which was used to establish the TLS connection), and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope, and expiration time claims are all valid. However, the subject claim in the access token does not match the subjectAltName in the TLS certificate of the NF service consumer.

NOTE: Test case F is applicable only for direct communication case.

Test Case G: Access token subject claim does not match the CCA

1) The tester computes a Client Credentials Assertion (CCA) correctly for the NF Service Request from the NF service consumer to the network product under test.

2) The tester computes an access token correctly, except that the subject claim (i.e., the NF Instance Id of the NF Service Consumer) in the access token does not match the subject claim in the CCA, and then includes the access token and the CCA in the NF Service Request sent from the NF service consumer to the network product under test.

3) The network product under test verifies that the integrity of the access token, the audience, scope, and expiration time claims are all valid. However, the subject claim in the access token does not match the subject claim in the CCA.

Test Cases H~K are tests on failure handling by the network product under test when the optional claims in access token failed verification.

NOTE: The test cases below only apply to the NFs which support identifying and understanding the optional claims in the received access token.

Test Case H: Incorrect list of S-NSSAIs in the access token

1) The tester computes an access token correctly, except that the list of S-NSSAIs is incorrect, i.e., the network product under test does not serve the slices indicated in the list of S-NSSAIs, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the list of S--NSSAIs included in the access token.

Test Case I: Incorrect list of NSIs in the access token

1) The tester computes an access token correctly, except that the list of NSIs is incorrect, i.e., the network product under test does not serve the slices indicated in the list of NSIs, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the list of NSIs included in the access token.

Test Case J: Incorrect NF Set ID in the access token

1) The tester computes an access token correctly, except that the NF Set ID is incorrect, i.e. the NF Set ID in the claim does not match the NF Set ID of the network product under test, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the NF Set ID included in the access token.

Test Case K: Incorrect additional scope in the access token

1) The tester computes an access token correctly, except that the additional scope information is incorrect, i.e. the allowed resources and allowed actions on the resources do not match the requested service operations, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the additional scope included in the access token.

**Expected Results:**

For test cases A~G on verification failure of mandatory claims in the access token, the network product under test rejects the NF service consumer’s service request based on OAuth 2.0 error response defined in RFC 6749 [12].

For test cases H~K on verification failure of optional claims in the access token, if the network product under test understands these optional claims (list of S-NSSAIs, list of NSIs, NF Set ID, additional scope), it rejects the NF service consumer’s service request based on OAuth 2.0 error response defined in RFC 6749 [12].

**Expected format of evidence:**

Evidence suitable for the interface, e.g., packet trace (pcap file).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of 1st CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 2nd CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 4.2.3.3.2 Boot from intended memory devices only

*Requirement name*: Boot from intended memory devices only

*Requirement reference*: In accordance with industry best practice

*Requirement Description*:

The network product can boot only from the memory devices intended for this purpose.

*Test case:*

**Test Name:** TC\_BOOT\_INT\_MEM\_1

**Purpose:**

Verify that the network product can only boot from memory devices intended for this purpose (e.g. not from external memory like USB key).

**Procedure and execution steps:**

**Pre-Conditions:**

A document which contains information regarding the firmware access mechanism supported by the product and about the memory devices from which the network product can boot.

**Execution Steps**

1. The tester verifies that the network product is configured to boot from memory devices declared in the network product document only.

2. The tester verifies that the network product does not boot from any undeclared memory device by preparing a bootable medium for every class of bootable memory device (e.g. CD, USB key, network boot) present in and accessible at the network product and trying to boot from this medium.

3. The tester verifies that attempts to access and modify the firmware of the network product are permitted following successful authentication but prevented without prior successful authentication.

4. The tester verifies that the boot sources can be modified only by administrator/root user.

**Expected Results:**

The network product cannot boot from a memory device that is not configured in its firmware, and access to the firmware is only possible with the correct authentication.

**Expected format of evidence:**

Screenshot of the actual boot device configuration of the network product and firmware access mechanism/authentication.

Textual description of the attempts of booting from prepared memory devices.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of 2nd CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 3rd CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 4.2.6.2.3 GTP-C Filtering

*Requirement Name:* GTP-C Filtering

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*:

The following capability is conditionally required:

- For each message of a GTP-C-based protocol, it shall be possible to check whether the sender of this message is authorized to send a message pertaining to this protocol.

NOTE 1: The check could be performed e.g. against an allow list or disallow list of permitted message type / sender identity combinations.

- At least the following actions should be supported when the check is satisfied:

- Discard: the matching message is discarded.

- Accept: the matching message is accepted.

- Account: the matching message is accounted for, i.e. a counter for the rule is incremented. This action can be combined with the previous ones. This feature is useful to monitor traffic before its blocking.

This requirement is conditional in the following sense: It is required that at least one of the following two statements holds:

- The Network Product supports the capability described above and this is stated in the product documentation.

- The Network Product's product documentation states that the capability is not supported and that the Network Product needs to be deployed together with a separate entity which provides the capability described above.

NOTE 2: Such a separate entity could e.g. be a GTP Firewall.

NOTE 3: Test cases for this separate entity are not provided in the present document, but are believed to be similar to them.

NOTE 4: The test cases are only applicable to all network product classes utilizing GTP-C based protocol.

*Threat References:* TR 33.926 [4], clause 5.3.7, Denial of service

*Test case*:

The test case described here apply only when GTP-C filtering is provided on the Network Product itself.

**Test Name**: TC\_GTP-C\_FILTERING

**Purpose:**

To verify that the network product provides filtering functionalities for incoming GTP-C messages. In particular this test case verifies that:

1. The network product provides filtering of incoming GTP-C messages on any interface.

2. It is possible to block all GTP-C messages on those network product interfaces where they are unwanted.

3. It is possible to specify defined actions for each rule.

**Procedure and execution steps:**

**Pre-Conditions:**

- The network product has at least two physical interfaces, named if1 and if2.

- The tester has the privileges to configure GTP-C filtering on the network product.

- The vendor declares that the GTP-C filtering is supported.

- The vendor includes a guideline to configure the GTP-C filtering in the documentation accompanying the network product.

- A network traffic generator or a pcap file containing the GTP-C messages is available.

- A network traffic analyser on the network product (e.g. tcpdump) is available.

**Execution Steps**

1. The tester log in the network product.

2. The tester configures the network product with the following rules:

a) Accept only GTP-C EchoRequest messages on if1.

b) Discard all GTP-C messages on if2.

c) For each rule above the accounting is also enabled.

3. The tester turns on the network traffic analyser on if2.

4. The tester sends on if2 EchoRequest messages replaying a pcap file or using a network generator.

a) Using the network analyser the tester verifies that the network product correctly receives the EchoRequest messages on if2.

b) Using the accounting, the tester verifies that the messages are discarded and that no response is sent back by the network product.

5. The tester sends to if1 EchoRequest messages replaying a pcap file or using a network generator.

a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

b) The tester verifies that the GTP-C EchoRequest messages are not discarded because EchoResponse messages are sent back by the network product.

6. The tester verifies that the matching messages are correctly accounted for both rules.

7. The tester sends to if1 GTP-C messages different from EchoRequest replaying a pcap file or using a network generator.

a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

b) Using the accounting, the tester verifies that the messages are discarded and that no response is sent back by the network product.

8. The tester deletes the previous rules and configures a new rule, i.e. to accept only GTP-C EchoRequest on if1 coming from a certain IP Address named IP1.

9. The tester sends GTP-C EchoRequest messages with source IP Address set to IP1:

a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

b) The tester verifies that the GTP-C EchoRequest messages are not discarded and EchoResponse messages are sent back by the network product.

10. The tester sends GTP-C EchoRequest messages with source IP Address set to IP2 different from IP1 using a network traffic generator or replaying a pcap file.

a) Using the network analyser the tester verifies that the messages are correctly received by the network product.

b) The tester verifies that the GTP-C EchoRequest messages are discarded and that no EchoResponse messages are sent back.

**Expected Results:**

- For steps 4, 5, 6 and 7 the tester receives GTP-C EchoResponse messages from if1 only.

- For steps 4, 5, 6 and 7 the messages matching the rules are correctly accounted.

- For steps 8, 9, 10 the tester receives GTP-C EchoResponse messages only for the authorized source IP address.

**Expected format of evidence:**

- The used tool(s) name and version information

- Settings and configurations used

- Pcap trace

- Screenshot

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of 3rd CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 4th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 4.2.6.2.4 GTP-U Filtering

*Requirement Name:* GTP-U Filtering

*Requirement Reference:* In accordance with industry best practice

*Requirement Description*:

The following capability is conditionally required:

- For each message of a GTP-U-based protocol, it shall be possible to check whether the sender of this message is authorized to send a message pertaining to this protocol.

NOTE 1: The check could be performed e.g. against an allow list or disallow list of permitted message type / sender identity combinations.

- At least the following actions should be supported when the check is satisfied:

- Discard: the matching message is discarded.

- Accept: the matching message is accepted.

- Account: the matching message is accounted for, i.e. a counter for the rule is incremented. This action can be combined with the previous ones. This feature is useful to monitor traffic before its blocking.

This requirement is conditional in the following sense: It is required that at least one of the following two statements holds:

- The Network Product supports the capability described above and this is stated in the product documentation.

- The Network Product's product documentation states that the capability is not supported and that the Network Product needs to be deployed together with a separate entity which provides the capability described above.

NOTE 2: Such a separate entity could e.g. be a GTP Firewall.

NOTE 3: Test cases for this separate entity are not provided in the present document, but are believed to be similar to them.

NOTE 4: The test cases are only applicable to all network product classes utilizing GTP-U based protocol.

*Threat References:* TR 33.926 [4] clause 5.3.7, Denial of service

*Test case*:

The test case described here apply only when GTP-U filtering is provided on the Network Product itself.

**Test Name**: TC\_GTP-U\_FILTERING

**Purpose:**

To verify that the network product provides filtering functionalities for incoming GTP-U messages. In particular this test case verifies that:

1. The network product provides filtering of incoming GTP-U messages on any interface.

2. It is possible to block all GTP-U messages on those network product interfaces where they are unwanted.

3. It is possible to specify defined actions for each rule.

**Procedure and execution steps:**

**Pre-Conditions:**

- The network product has at least one physical interface named if1 and may have another physical interface named if2.

- The tester has the privileges to configure GTP-U filtering on the network product.

- The vendor declares that the GTP-U filtering is supported.

- The vendor includes a guideline to configure the GTP-U filtering in the documentation accompanying the network product.

- A network traffic generator or a pcap file containing the GTP-U messages is available.

- A network traffic analyser on the network product (e.g. tcpdump) is available.

NOTE: If the network product has only one physical interface named if1, execution steps on if2 are not needed.

**Execution Steps**

1. The tester log in the network product.

2. The tester configures the network product with the following rules:

a) Accept only GTP-U EchoRequest messages on if1.

b) Discard all GTP-U messages on if2.

c) For each rule above the accounting is also enabled.

3. The tester turns on the network traffic analyser on if2.

4. The tester sends on if2 EchoRequest messages replaying a pcap file or using a network generator.

a) Using the network analyser, the tester verifies that the network product correctly receives the EchoRequest messages on if2.

b) Using the accounting, the tester verifies that the messages are discarded and that no response is sent back by the network product.

5. The tester sends to if1 EchoRequest messages replaying a pcap file or using a network generator.

a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

b) The tester verifies that the GTP-U EchoRequest messages are not discarded because EchoResponse messages are sent back by the network product.

6. The tester verifies that the matching messages are correctly accounted for both rules.

7. The tester sends to if1 GTP-U messages different from EchoRequest replaying a pcap file or using a network generator.

a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

b) Using the accounting, the tester verifies that the messages are discarded and that no response is sent back by the network product.

8. The tester deletes the previous rules and configures a new rule, i.e. to accept only GTP-U EchoRequest on if1 coming from a certain IP Address named IP1.

9. The tester sends GTP-U EchoRequest messages with source IP Address set to IP1:

a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

b) The tester verifies that the GTP-U EchoRequest messages are not discarded and EchoResponse messages are sent back by the network product.

10. The tester sends GTP-U EchoRequest messages with source IP Address set to IP2 different from IP1 using a network traffic generator or replaying a pcap file.

a) Using the network analyser the tester verifies that the messages are correctly received by the network product.

b) The tester verifies that the GTP-U EchoRequest messages are discarded and that no EchoResponse messages are sent back.

**Expected Results:**

- For steps 4, 5, 6 and 7 the tester receives GTP-U EchoResponse messages from if1 only.

- For steps 4, 5, 6 and 7 the messages matching the rules are correctly accounted.

- For steps 8, 9, 10 the tester receives GTP-U EchoResponse messages only for the authorized source IP address.

**Expected format of evidence:**

- The used tool(s) name and version information

- Settings and configurations used

- Pcap trace

- Screenshot

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of 4th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 5th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.2.1 No unnecessary or insecure services / protocols

*Requirement Name*: No unnecessary or insecure services / protocols

*Requirement Reference:* In accordance with industry best practice

*Requirement Description*:

The network product shall only run protocol handlers and services which are needed for its operation, and which do not have any known security vulnerabilities. In particular, by default the following services shall be initially configured to be disabled on the network product by the vendor except if services are needed during deployment. In that case those services shall be disabled according to vendor’s instructions after deployment is done. Disabled protocols can still be enabled for other reasons by the network operators, e.g. remote diagnostics.

- FTP

- TFTP

- Telnet

- rlogin, RCP, RSH

- HTTP

- SNMPv1 and v2

- SSHv1

- TCP/UDP Small Servers (Echo, Chargen, Discard and Daytime)

- Finger

- BOOTP server

- Discovery protocols (CDP, LLDP)

- IP Identification Service (Identd)

- PAD

- MOP

- SS7, SIGTRAN

NOTE 1: As an alternative to disabling the HTTP service, it is also possible for this service to remain active for reasons of user friendliness. In this case, however, queries to the web service are not answered directly on this port but from a redirected to HTTPS service.

Note 2: Full documentation of required protocols and services of the network product and their purpose needs to be provided by the vendor as prerequisite for the test case.

*Threat References:* TR 33.926 [4], clause 5.3.7.3, Insecure Network Services

*Test Case*:

**Test Name**: TC\_NO\_UNNECESSARY\_SERVICE

**Purpose:**

To ensure that on all network interfaces, there are no unsecure services or protocols that might be running.

**Procedure and execution steps:**

**Pre-Conditions:**

A list of all required network protocols and services containing at least the following information shall be included in the documentation accompanying the Network Product:

- protocol handlers and services needed for the operation of network product;

- their open ports and associated services;

- and a description of their purposes.

The tool used shall be capable to detect and identify the protocol handlers and running services in the system.

**Execution Steps**

1. Verification of the compliance to the prerequisites:

a. Verification that the list of available network services and protocol handlers is available in the documentation of the Network Product.

b. Validation that all entries in the list are necessary for the operation of the Network Product class.

2. Identification of the network services and protocol handlers by means of tools or any other testing means.

3. Validation that there are no entries in the list of available network services and handlers apart from the ones that have been mentioned for the operation of the Network Product in the attached documentation.

4. The tester shall reboot the network product and re-execute execution steps 2 and 3 without further configuration.

**Expected Results:**

The report will contain:

- The names and version of the tool(s) used.

- Information of all the protocol handlers and services running in the network product.

Result will show:

- There are no unnecessary services running in the network product except for the ones which are necessary for its operation.

- Any undocumented services running on the network product should be highlighted and brought out in the report.

- The network product behaves the same after reboot as before.

**Expected format of evidence:**

- The used tool(s) name and version information;

- Settings and configurations, and commands used (if applicable);

- The output pertaining to the test case performed and

- The test results i.e. services existing or not existing in the Network Product.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of 5th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 6th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4.3.2.2 Restricted reachability of services

*Requirement Name*: Restricted reachability of services

*Requirement Reference:* In accordance with industry best practice

*Requirement Description*:

The network product shall restrict the reachability of services so that they can only be reached on interfaces where their usage is required. On interfaces where services are active, the reachability should be limited to legitimate communication peers. This limitation shall be realized on the network product itself (without measures (e.g. firewall) at network side) according to the requirement detailed in clause 4.2.6.2.1 Packet Filtering.

Example: Administrative services (e.g. SSH, HTTPS, RDP) shall be restricted to interfaces in the management network to support separation of management traffic from user traffic.

*Threat References*: TR 33.926 [4], clause 5.3.7.3, Insecure Network Services

*Test Case*:

**Test Name***:* TC\_RESTRICTED\_REACHABILITY\_OF\_SERVICES

**Purpose:**

To verify that it is possible to bind the services only to the interfaces from which they are expected to be reachable.

Note: The test case developed for the requirement " 4.2.6.2.1 Packet Filtering" implicitly verifies that the network product permits to limit the reachability of the services only to legitimate communication peers.

**Procedure and execution steps:**

**Pre-Conditions:**

- The vendor shall declare, in the documentation accompanying the network product if the network product supports the capability to restrict services reachability to only the nodes authorized to access them. In this case, the vendor shall detail how this capability can be configured.

- A list of all required network protocols and services containing at least the following information shall be included in the documentation accompanying the Network Product:

- protocol handlers and services needed for the operation of network product;

- their open ports and associated services;

- the configuration options;

- and a description of their purposes.

- The network product is configured such that the required network protocols and services (as described in the network product documentation) are setup and each service is bound to an IP address and port of a specific network interface (e.g. IP1 which is the ip address and port of if1). Configuration may occur automatically during the initialization phase of the network product or manually as defined in the network product administration documentation.

- The network product shall have at least two interfaces enabled, if1 and if2 respectively configured with IP Address IP1 and IP2 and respective ports.

- The tester has administrative privileges.

- A tester machine equipped with a network port scanner tool is available.

**Execution Steps**

**For every available interface if\_n:**

1. The tester runs a network port scanner (e.g. nmap) or uses local network interface information on if\_n and verifies that the configured services (according to the vendor documentation) are open/reachable.

2. The tester runs a network port scanner (e.g. nmap) or uses local network interface information on all other available interfaces (except if\_n) and verifies that the services configured for if\_n are not open/reachable.

NOTE: It might not be possible for certain transport layer protocols (like UDP) to unambiguously detect whether a port is open or not by means of external port scanning. Also, external port scanning can be ineffective, if there are security measures present, e.g. like rate limiting. Local port discovery (e.g. with *netstat*, *ss*) in collaboration with collection of local route information (e.g. with *ip route*) could be applied in those cases.

**Expected Results:**

Services can be enabled on per-interface basis.

**Expected format of evidence:**

- The network product configuration showing the mapping between interfaces and configured service.

- Pcap files.

- Screenshot.

- Software name and version of the used port scanner, log of the executed commands.

- Network port scanner results (e.g. files containing this results).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of 6th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 7th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.2 No system privileges for web server

*Requirement Name*: No system privileges for web server.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*:

No web server processes shall run with system privileges. This is best achieved if the web server runs under an account that has minimum privileges. If a process is started by a user with system privileges, execution shall be transferred to a different user without system privileges after the start.

*Threat References*: TR 33.926 [4], clause 5.3.8, Elevation of privilege

*Test Case*:

***Test Name*:** TC\_NO\_SYSTEM\_PRIVILEGES\_WEB\_SERVER

**Purpose:**

Verify that the Web server is not run under system privileges.

**Procedure and execution steps:**

**Pre-Conditions:**

- The tester has needed administrative privileges.

- A tester machine is available.

- Recommended: an automatic assessment tool has been configured /script adapted in line with the Requirement Description.

**Execution Steps**

1. The tester checks that no web server processes run with system privileges. The tester checks that this is the case even for processes that may have been started by a user with system privileges.

a. The tester starts the web server process as web server user and checks process privileges.

b. If possible, the tester starts the web server process with system privileges and check if process privileges get dropped.

2. The tester checks in relevant system settings and web server configurations that a web server user is configured with minimal privileges needed to run the web server and the web server is executable by that user.

NOTE: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

**Expected Results:**

- There are no findings of web server processes that run with system privileges.

- System settings are set to ensure that no processes will run with system privileges.

**Expected format of evidence:**

- Log files / command line output and screen shots of test executions

- Part of web server and/or system configuration (plain text or screenshot) showing the configured user for the web server process

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 7th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* START 8th of CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.4 No unused add-ons

*Requirement Name*: No unused add-ons

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: All optional add-ons and components of the web server shall be deactivated if they are not required. In particular, CGI or other scripting components, Server Side Includes (SSI), and WebDAV shall be deactivated if they are not required.

*Threat References*: TR 33.926 [4], clause 5.3.6.11, Unnecessary Services

*Test Case*:

***Test Name*:** TC\_NO\_UNUSED\_ADD-ONS

**Purpose:**

To verify that the Web server has deactivated unneeded add-ons and unneeded scripting components.

**Procedure and execution steps**

**Pre-Conditions:**

- The vendor has supplied a list of add-ons or scripting tools for Web server components needed for system operation, and that therefore need to be exempted from the test investigation.

- The tester has administrative privileges.

- A tester machine is available.

- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

**Execution Steps**

1. Check that the web server is only running and listening on known ports (e.g. tcp port 80 and/or 443). Check that CGI or other scripting components, Server Side Includes (SSI), and WebDAV are deactivated if they are not required. See also guidance under 4.3.4.12.

2. Check that nothing else has been installed than the web server.

3. Check that relevant system settings and configurations are correct to ensure fulfilment of the requirement.

NOTE: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

**Expected Results:**

System settings and configurations have been found, for all Web components of the system, to ensure that all unneeded add-ons or script components are deactivated.

**Expected format of evidence:**

Log files and screen shots of test executions.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 8th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 9th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.5 No compiler, interpreter, or shell via CGI or other server-side scripting

*Requirement Name*: No compiler, interpreter, or shell via CGI or other server-side scripting.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: If CGI (Common Gateway Interface) or other scripting technology is used, the CGI directory - or other corresponding scripting directory - shall not include compilers or interpreters (e.g. PERL® interpreter, PHP interpreter/compiler, Tcl interpreter/compiler or operating system shells).

*Threat Reference*: TR 33.926 [4], clause 5.3.6, Information disclosure

*Test Case*:

***Test Name*:** TC\_NO\_COMPILER\_FOR\_CGI

**Purpose:**

To verify that there are no compilers, interpreters or shell accessible via CGI or other scripting components.

**Procedure and execution steps**

**Pre-Conditions:**

- The tester has administrative privileges

- A tester machine is available.

- Recommended: an automatic assessment tool has been configured /script adapted in line with the Requirement Description.

**Execution Steps**

1. Consult the web server configuration to identify all directories used for CGI or other scripting components.

NOTE: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

2. Check that there are no compilers or interpreters (e.g., PERL® interpreter, PHP interpreter/compiler, Tcl interpreter/compiler or operating system shells) in the directory/directories used for CGI or for other scripting tools (including PERL®, PHP, and others).

**Expected Results:**

There are no compilers, interpreters or shells in directories accessible via CGI or other scripting components.

**Expected format of evidence:**

- Log files and screen shots of test executions.

- Part of web server configuration (plaintext or screenshot) showing all directories accessible by the CGI/scripting components.

- List of files (with types and permissions, if available) inside the directories accessible by the CGI/scripting components.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 9th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 10th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.6 No CGI or other scripting for uploads

*Requirement Name*: No CGI or other scripting for uploads.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: If CGI or other scripting technology is used, all directories where the web server has write permissions shall be distinct from all directories containing CGI/script or executable code.

*Threat References*: TR 33.926 [4], clause 5.3.8.3, Folder Write Permission Abuse

*Test Case*:

**Test Name:** TC\_NO\_CGI\_OR\_SCRIPTING\_FOR\_UPLOADS

**Purpose:**

To ensure that directories with write permissions for the web server do not contain executable code such as CGI scripts.

**Procedure and execution steps:**

**Pre-Condition:**

If the web server is configured with CGI/Scripting on, this test applies.

**Execution Steps**

1. The tester identifies directories where the web server user has write permissions.

2. The tester verifies that these writable directories do not contain any executable scripts, CGI programs, or other executable code.

3. The tester verifies that directories configured for CGI/Scripting do not have write permissions for the web server.

NOTE: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

**Expected Results:**

Web server user writable directories are different from those containing executable code or the ones configured to be used for CGI/scripting.

**Expected format of evidence:**

A part of the configuration file / screenshot of the configuration showing that the web server is properly configured and the corresponding file system permissions.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 10th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 11th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.7 No execution of system commands with SSI

*Requirement Name*: No execution of system commands with SSI.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: If Server Side Includes (SSI) is active, the execution of system commands shall be deactivated.

*Threat Reference*: TR 33.926 [4], clause 5.3.8, Elevation of privilege

*Test Case*:

**Test Name**: TC\_NO\_EXECUTION\_OF\_SYSTEM\_COMMANDS

**Purpose:**

To test whether it is possible to use the exec directive and if so, whether it can be used for system commands.

**Procedure and execution steps:**

**Pre-Condition:**

If the web server is configured with SSI active, this test applies.

**Execution Steps**

1. The tester checks whether execution of system commands is disabled in the web server configuration.

NOTE: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

2. The tester actually attempts to use the exec directive in an SSI file with and without system commands.

**Expected Results:**

- The execution of system commands via SSIs exec directive is disabled in the web server configuration.

- It is impossible to execute system commands via SSIs exec directive.

**Expected format of evidence:**

- A part of the configuration file / screenshot of the configuration showing that the web server is properly configured. For example, a configuration file that shows that the IncludesNOEXEC (Apache HTTP Server®) or ssiExecDisable (Microsoft® IIS) is set.

- Web server log while executing step 2.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 11th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 12th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.10 No directory listings

*Requirement Name*: No directory listings / Directory Browsing.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: Directory listings (indexing) / "Directory browsing" shall be deactivated.

*Threat References*: TR 33.926 [4], clause 5.3.6.9, File/Directory Read Permissions Misuse

*Test Case*:

***Test Name*:** TC\_NO\_DIRECTORY\_LISTINGS

**Purpose:**

To verify that Directory listings / Directory browsing has been deactivated in all Web server components.

**Procedure and execution steps**

**Pre-Conditions:**

- The tester has administrative privileges

- A tester machine is available.

- The tester should have configured a script, or an automatic assessment tool adapted in line with the Requirement Description..

**Execution Steps**

1. The tester checks the web server configuration for Directory listings (indexing) / "Directory browsing" to be deactivated in all Web server components.

NOTE 1: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

2. The tester attempts directory listings on all endpoints (domains, subdomains and directories) offered by the web server.

NOTE 2: Whether directory listings have been deactivated could be done by checking the web server configuration file specifically the parameters related to directory listing. The directory listing could be turned off in the web server configuration file, and there is no activation capability.

NOTE 3: Directory listings could be obtained by entering a valid URL (e.g., /var/www/test\_1) that does not contain any index file.

**Expected Results:**

- Directory listing / Directory browsing has been deactivated in all Web server components configurations.

- The tester is unable to perform Directory listing / Directory browsing on all endpoints (domains, subdomains and directories) offered by the web server.

**Expected format of evidence:**

- Log files and screen shots of test executions

- Text excerpt of the web server configuration showing that directory listing is disabled

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 12th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 13th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.12 Web server information in error pages

*Requirement Name*: Web server information in error pages.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: User-defined error pages shall not include version information about the web server and the modules/add-ons used. Error messages shall not include internal information such as internal server names, error codes, etc. Default error pages of the web server shall be replaced by error pages defined by the vendor.

*Threat References*: TR 33.926 [4], 5.3.6.5, System Fingerprinting

*Test Case*:

***Test Name*:** TC\_NO\_WEB\_SERVER\_ERROR\_PAGES\_INFORMATION

**Purpose:**

To verify that error pages and error messages do not include information about the web server.

**Procedure and execution steps**

**Pre-Conditions:**

- The tester has needed administrative privileges.

- A tester machine is available.

- The vendor provides documentation on user-defined error pages (e.g. location, content, where configured) and messages.

- The vendor provides a list of potential parameters/commands to trigger events resulting in an http status code 3xx, 4xx, 5xx.

- The tester should have configured a script, or an automatic assessment tool adapted in line with the Requirement Description.

**Execution Steps**

1. The tester verifies that the web server configuration does replace default error pages with error pages defined by the vendor.

NOTE 1: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

2. The tester verifies that the vendor defined error pages do not contain information about the web server.

3. The tester triggers and captures at least one occurrence of the following HTTP status code classes:

a) Redirection error response (300-399)

b) Client error response (400-499)

c) Server error response (500-599)

NOTE 2: Possible error pages that could be displayed are: 3xx: redirection, 4xx: client errors, 5xx: server errors.

NOTE 3: The 3xx error pages could be triggered by permanent or temporary move of content to other URL and the page is found because redirected.

NOTE 4: The 4xx error page could be triggered by trying to access a URL pointing to a non-existent or restricted resource.

NOTE 5: The 5xx error page could be triggered by requesting a HTTP method the web server does not support or disabled (e.g. CONNECT, PUT, PATCH).

**Expected Results:**

Generated error pages and error messages do not include information about the web server.

**Expected format of evidence:**

Log files and screen shots of test executions

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 13th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 14th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.4.14 Restricted file access

*Requirement Name*: Restricted file access.

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*: Restrictive access rights shall be assigned to all files which are directly or indirectly (e.g. via links or in virtual directories) in the web server's document directory. In particular, the web server shall not be able to access files which are not meant to be delivered.

*Threat References*: TR 33.926 [4], clause 5.3.6.9, File/Directory Read Permissions Misuse

*Test Case*:

**Test Name:** TC\_RESTRICTED\_FILE\_ACCESS

**Purpose:**

To test whether the restrictive access rights are assigned to all files which are directly or indirectly in the web server's document directory and to verify whether path traversal is made improbable.

**Procedure and execution steps:**

**Pre-Condition:**

The web server is configured according to the manual

**Execution Steps**

1. The tester verifies that access rights on the servable content (meaning directories and files) is set to the following:

a. The files are owned by the user that runs the web server;

b. The files are not writable to others, except the web server's account;

2. The tester verifies that the user running the web server is an unprivileged account;

3. For Operating Systems that have chrooted environments, the tester verifies that the web server runs inside a jail or chrooted environment. If the chrooted environment is not used, the web server or system functionality can be used to restrict access to file directories.

NOTE: If the web server is integrated directly into (parts of) the network product, it could not use configuration files. Configuration could be done via command line parameters or simply be hardcoded into the application. In such cases, the tester can omit test steps or parts of test steps related to web server configuration files.

**Expected Results:**

- Name of user running the web server with the privileges of the account;

- Access rights of files and directories that the web server serves;

- Configuration that shows that the web server is in a chrooted environment, or restricted by accessing to file directories.

**Expected format of evidence:**

A part of the configuration file / screenshot of the configuration showing that the web server, the file access rights and the account running the web server is properly configured.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of 14th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* START of 15th CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 4.3.6.4 The valid format and range of values for IEs

*Requirement Name*: Validation of the IEs limits.

*Requirement Reference:* TS 29.501 Principles and Guidelines for Services Definition [13], clause 6.2; TS29.500 Technical Realization of Service Based Architecture [21], clause 5.2.7.2

*Requirement Description*:

The valid format and range of values for each IE, when applicable, is defined unambiguously:

- For each message the number of leaf IEs does not exceed 2048K.

- The maximum size of the JSON body of any HTTP request does not exceed 16 million octets.

- The maximum nesting depth of leaves does not exceed 32.

as stated in TS 29.501 [13] clause 6.2.

If a received HTTP request contains IEs or query parameters not compliant with the schema defined in the corresponding OpenAPI specification, the NF rejects the request with the appropriate error code, e.g. "400 Bad Request (INVALID\_MSG\_FORMAT)", even when the failed IEs are defined as optional by the schema, as stated in TS 29.500 [21], clause 5.2.7.2.

*Threat References*: TR 33.926 [4], clause 6.3.2.2, JSON Parser not Robust

*Test Case*:

NOTE 1: This requirement can also be verified as part of Robustness and Protocol fuzzing tests as defined in clause 4.4.4 Robustness and fuzz testing according to referenced requirements.

**Test Name:** TC\_IE\_VALUE\_FORMAT

**Purpose:**

Verify that the API implementation fulfils the requirements as specified in 29.501[13], clause 6.2.

**Pre-Conditions:**

Test environment with network product under test so that the tester is able to send HTTP requests with "out of bound IEs" towards the network product under test. Rest of the network may be simulated.

NOTE 2: IEs having invalid format and/or not in the defined range of values can be considered as out of bound IEs.

**Execution Steps**

Test case 1 (JSON object depth exceeds 32):

1. The tester sends an HTTP request with a JSON object that exceeds 32 levels of nesting to the network product under test.

2. The tester checks if the network product under test responds with an appropriate HTTP error code and that it does not contain valid response content for the request.

Test case 2 (Leaf IEs exceed 2048K):

1. The tester sends an HTTP request with leaf IEs that exceed 2048K to the network product under test.

2. The tester checks if the network product under test responds with an appropriate HTTP error code and that it does not contain valid response content for the request.

Test case 3 (JSON body size exceeds 16 million octets):

1. The tester sends an HTTP request with a JSON body that exceeds 16 million octets to the network product under test.

2. The tester checks if the network product under test responds with an appropriate HTTP error code and that it does not contain valid response content for the request.

**Expected Results:**

Test case 1 (JSON dictionary depth exceeds 32):

- The network product under test's response has an appropriate HTTP error code and its body does not contain valid response content for the request.

Test case 2 (Leaf IEs exceed 2048K):

- The network product under test's response has an appropriate HTTP error code and its body does not contain valid response content for the request.

Test case 3 (JSON body size exceeds 16 million octets):

- The network product under test's response has an appropriate HTTP code and its body does not contain valid response content for the request.

**Expected format of evidence:**

- The used tool(s) name and version information,

- Settings and configurations used.

- The output log file of the chosen tool that displays the results (passed/failed).

- Log/evidence tracing possible crashes.

- Information of any input causing unspecified, undocumented, or unexpected behaviour.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of CHANGES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*