**3GPP TSG-SA3 Meeting #114Adhoc-e *S3-24XXXX***

**Online, 22 - 26 January 2024**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **33.117** | **CR** | **TBD** | **rev** | **-** | **Current version:** | **18.2.0** |  |
|  | | | | | | | | |
| *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:*** | Clarifications to test cases in TS 33.117 clauses 4.4.1 - 4.4.4. | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | MITRE Corporation | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SCAS\_5G\_Ph3 | | | | |  | ***Date:*** | | | 2024-01-15 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | | Rel-18 |
|  | *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 can be 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:*** | | Clarifications to provide consistency, and objectivity of Basic Vulnerability Testing (BVT) test cases in TS 33.117 clauses 4.4.2 – 4.4.4.  Removes endpoint scanning from BVT in clause 4.4.1. These clarifications where recommended in S3-234423. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Provide clarifications on the testing tools, tool configuration, and procedures in BVT clauses 4.4.1, 4.4.2, 4.4.3, and 4.4.4 | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | BVT test cases will yield inconsistent results and will be difficult to reproduce. BVT tests will yield inconclusive results that do not verify the presence, or lack of, basic vulnerabilities in 3GPP network products | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 4.4.1, 4.4.2, 4.4.3, 4.4.4 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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:*** | |  | | | | | | | | |

\*\*\*\*\*\*\*\*\*\*\*\*\* Start of 1st Change \*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.4 Basic vulnerability testing requirements

### 4.4.1 Introduction

Basic Vulnerability Testing activities consist of requirements for running automated FOSS and COTS security testing tools against the external interfaces of a Network Product. These activities cover at least three aspects: port scanning, vulnerability scanner by using vulnerability scanners, and robustness/fuzz testing. For each of these aspects, test requirements and test results are described in the present clause.

NOTE 1: The individual tools used for Basic Vulnerability Testing are selected by the evaluator. The SECAM accreditation body will ensure during accreditation of the testers lab that the testing tools meet the conditions in clause 4.1.2.

NOTE 2: The ports referred to in this clause are identifiers for services running on transport layer protocols (e.g., HTTPS, port 443, FTP, port 21).

### 4.4.2 Port Scanning

*Requirement Name*: Port scanning

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*:

It shall be ensured that on all network interfaces, only documented ports on the transport layer respond to requests from outside the system. Testing is performed on all ports from 0-65535, including those marked as optional or not recommended.

The test for this requirement can be carried out using a port scanning tool (e.g., Nmap, Angry IP Scanner) tool or manually performed as described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool provides functionality equivalent to the steps described below.

*Threat References*: TR 33.926 [4]

*Test Case*:

**Test Name**: TC\_BVT\_PORT\_SCANNING

**Purpose:**

To ensure that on all network interfaces, only documented ports on the documented transport layer protocols respond to requests from outside the system.

**Procedure and execution steps:**

**Pre-Conditions:**

A list of the following information shall be included in the documentation accompanying the Network Product:

1. all network interfaces providing IP-based protocols;

2. the available transport layer protocols on these interfaces that provide IP based protocols;

3. the transport layer protocol open ports; and

4. a free-form description describing the purposes for each interface, transport protocol, and open port.

The port scanning tool that is used shall be configured to scan all the transport layer ports and detect open ports for each network interface as identified in steps 1 to 4. above.

NOTE 2: 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 in some circumstances it might not be efficient to do external port scanning, e.g. if there are security measures to limit the rate a system can be probed. In those cases the tester determines another means suitable to verify which ports are open.

**Execution Steps**

The tester is required to execute the following steps:

1. Verification of the compliance to the prerequisites:

a. Verification that the list in Pre-Conditions (i.e., 1 to 4) is available in the documentation of the Network Product

b. Validation that all entries in the list in Pre-Conditions are necessary for the operation of the Network Product (e.g., see clause 4.3.2.1 No unnecessary or insecure services / protocols)

c. Verification that the port scanning method is configured to scan all transport layer protocol ports starting from 0 and ending at 65535

2. Identification of the open ports by means of port scanning tools or other suitable testing means.

NOTE: Other suitable test means are to be documented including any procedures.

3. Verification that the list of identified open ports matches the list in Pre-Conditions in the documentation accompanying the Network Product.

**Expected Results:**

The used tool(s) name, their unambiguous version (also for plug-ins if applicable), used settings/configurations (e.g., scan type, syn scan, UDP scan), and the relevant output containing all the technically relevant information about test results (e.g., open/closed ports, service address, network interface) is evidence and shall be part of the testing documentation.

All discrepancies between the list of identified open ports as identified in Pre-Conditions (i.e., 1 to 4) and open ports per network IP interface, transport protocol that have been highlighted in the testing documentation that have not been documented per Pre-Conditions.

**Expected format of evidence:**

Output of portscan and list of identified discrepancies.

### 4.4.3 Vulnerability scanning

*Requirement Name:* Vulnerability scanning

*Requirement Reference*: In accordance with industry best practice

*Requirement Description*:

The purpose of vulnerability scanning is to ensure that there are no known vulnerabilities (or that relevant vulnerabilities are identified and remediation plans in place to mitigate them) on the Network Product, both in the OS and in the applications installed, that can be detected by means of automatic testing tools via the Internet Protocol enabled network interfaces.

Vulnerability scanning tools may also report false positives and they shall be investigated and documented in the test report.

The test for this requirement can be carried out using a vulnerability assessment tool (e.g., OpenVAS, Nessus) or manually performed as described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool actually provides functionality equivalent to the steps described below.

*Threat References*: TR 33.926 [4]

*Test case*:

**Test Name**: TC\_BVT\_VULNERABILITY\_SCANNING

**Purpose:**

The purpose of vulnerability scanning is to ensure that there are no known vulnerabilities (or that relevant vulnerabilities are identified and remediation plans in place to mitigate them) on the Network Product that can be detected by means of automatic testing tools via the Internet Protocol enabled network interfaces.

**Procedure and execution steps:**

**Pre-Conditions:**

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

- all interfaces providing IP-based protocols;

- the available transport layer protocols on these interfaces;

- their open ports and associated services;

- and a free-form description of their purposes (In the event the service is partially or wholly based on third-party or FOSS code, the name and version of the code should be included in the description when available).

NOTE 1: This list is to be validated as part of the BVT port scanning activity (see clause 4.4.2).

The used vulnerability scanning tool shall be configured to assess each network service to detect known vulnerabilities (e.g., CVE) on common services for each network interface, such as OAM and SBA interfaces. The used vulnerability information shall be updated within 15 days of the time of testing.

Vulnerability testing that is network product specific is identified in the respective SCAS specification.

**Execution Steps**

The tester is required to execute the following steps:

1. Execution of the suitable vulnerability scanning tool against all interfaces providing IP-based protocols of the Network Product.
2. For each network service the tester shall test against all known vulnerabilities (e.g.,CVEs) related to that service or its underlying software components (e.g., GTP-U, open source software).
3. Evaluation of the results based on their severity (e.g., using CVSS (Common Vulnerability Scoring System)).

**Expected Results:**

The used tool(s) name, their unambiguous version (also for plug-ins if applicable), used settings, and the relevant output is evidence and shall be part of the testing documentation.

The discovered vulnerabilities (including source, example CVE ID), together with a rating of their severity, shall be highlighted in the testing documentation.

COTS Vulnerability scanners, by their nature, (e.g. depending on how they are configured) may result in false findings/positives. The tool’s documentation may even mention that the failing test shall be repeated to check whether it is really a recurring problem or not. The tester shall make best effort to determine if there is an issue with NE or the test tool and if necessary, work with the vendor of the network product to come to a consensus on the test result outcome.

NOTE 2: This testing documentation is input to the vulnerability mitigation process (that may include patching). This is part of the product lifecycle management process developed by GSMA SECAG.

**Expected format of evidence:**

Output of vulnerability assessment tool.

### 4.4.4 Robustness and fuzz testing

*Requirement Name:* Robustness and fuzz testing

*Requirement Reference:* 4.2.6.2.2. – Interface Robustness

*Requirement Description*:

It shall be ensured that externally reachable services are robust enough to detect or dismiss unexpected or malformed input and remain operational.

*Threat References*: TR 33.926 [4]

*Test case*:

**Test Name**: TC\_BVT\_ROBUSTNESS AND FUZZ TESTING

**Purpose:**

To verify that the network product provides externally reachable services which are robust against unexpected or malformed input. The target of this test are the protocol stacks (e.g. diameter stack) rather than the applications (e.g. web app).

**Procedure and execution steps:**

**Pre-Conditions:**

- The tester has the privileges to log in the network product and to access all system resources (e.g. log files)

- A list of the following information shall be included in the documentation accompanying the Network Product:

- all network interfaces providing IP-based protocols;

- the available transport layer protocols on these interfaces that provide IP based protocols;

- the transport layer protocol open ports and associated services;

- and a free-form description describing the purposes for each interface, transport protocol and open port combination.

NOTE: This list is to be validated as part of the BVT port scanning activity (see clause 4.4.2).

- The robustness and fuzzing tools that are selected for this test shall be capable to identify input which causes the Network Product to behave in an unspecified, undocumented, or unexpected manner.

- Fuzz testing tools are a highly sophisticated technology and adaptation to the individual protocols in question is needed to be effective. Therefore, there is a lack of effective fuzz testing tools available especially for protocols proprietary to the Telco industry. Taking into account NOTE 4 of TR 33.916's clause 7.2.4, test labs shall acquire fuzz testing tools for those protocols when at least two capable and commercially available tools are available.

- It needs to be taken into account that fuzz testing tools might show drastic differences in terms of effectiveness. The tester is expected to recognize faults, misuse, or crashes in the protocol under test to determine the level of effectiveness of the available tools.

- A network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product and on a tester machine is available.

**Execution Steps**

The tester is required to execute the following steps:

1. Execution of fuzzing tools against the protocols available via interfaces providing IP-based protocols of the Network Product for at least 12 hours, or until the fuzzing test algorithms become indistinct or exhausted.

2. Execution of robustness test tools (e.g., unit testing, load testing) against the protocols available via interfaces providing IP-based protocols of the Network Product until the robustness test algorithms become indistinct or exhausted.

3. For both step 1 and 2:

a. Using a network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product, the tester verifies that the packets are processed correctly by the network product.

b. The testers verifies that the network product does not crash.

c. The execution of tests shall be repeated at least once to validate the results.

**Expected Results:**

A list of all the protocols of the network product reachable externally on an IP-based interface, together with an indication whether robustness and fuzz testing tools have been used against them, shall be part of the testing documentation. If no tool can be acquired for a protocol, a free form statement should explain why not.

The used tool(s) name, their unambiguous version (also for plug-ins if applicable), used settings, and the relevant output is evidence and shall be part of the testing documentation.

Any input causing unspecified, undocumented, or unexpected behaviour, and a description of this behaviour shall be highlighted in the testing documentation.

COTS fuzzing tools, by their nature, may have an acceptable failure rate (e.g. 0.1%) due to different non-deterministic variables in their implementation. In some cases, the tool’s documentation may indicate that the failing test shall be repeated to check whether it is a recurring problem or not. The tester shall make best effort to determine whether there is an issue with the Network Element or the test tool and if necessary, work with the vendor of the network product to come to a consensus on the test result outcome.

**Expected format of evidence:**

A testing report provided by the testing agency which shall consist of the following information:

- Description of the test;

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

- Settings and configurations used for the test tool(s) and Network product;

- The output log file of the chosen tool that displays the results of the test;

- Screenshot(s);

- Test result (Passed or not);

- Log/evidence tracing possible crashes, faults, or unexpected behaviour; and passes. and

- Any input causing unspecified, undocumented, or unexpected behaviour on the network product.

\*\*\*\*\*\*\*\*\*\*\*\*\* End of 1st Change \*\*\*\*\*\*\*\*\*\*\*\*\*