**3GPP TSG-SA3 Meeting #124 S3-253721**

**Wuhan, China, 13 – 17 October 2025 (revision of S3-253141)**

**Source: Nokia**

**Title: Pseudo-CR on new test cases on SCAS for CNP**

**Document for: Approval**

**Agenda item: 5.2.8**

**Spec: 3GPP TR 33.730**

**Version: 0.0.0**

**Work Item: FS\_SCAS\_CP**

**Comments**

This pCR aims to introduce the container-based virtualisation approach and is further providing new test cases for testing of basic functionality like handling of passwords and logging data, running vulnerability scans, protection of data, and run-time security.

It is proposed to add new clause 6.1.x on Security functional requirements deriving from containerisation and related test cases ([1]).

[1] TR 33.730, “Study on Security Assurance Specifications (SCAS) for Container-based Products”

# 6 Test cases for Container-based Products

Editor's Note: This clause contains analyses of the test cases in TS 33.117 regarding their applicability to container-based network products. This clause also contains analyses whether existing test cases need to be adapted to container-based network products, and whether new test cases for container-based network products are necessary.

\* \* \* First Change \* \* \* \*

### 6.1.x Security functional requirements deriving from containerization and related test cases

#### 6.1.x.1 Security non-functional requirements related to passwords

All text from TS 33.117 [1], clause 4.2.3.4.3 applies to containerized elements.

#### 6.1.x.2 Security requirements related to logging

All text from TS 33.117 [1], clauses 4.2.3.6.1, 4.2.3.6.2 and 4.2.3.6.3 apply to containerized elements.

*Requirement Name*: Logs from containerized functions are available

*Requirement Description*:

The containerized NF shall provide sufficient logging mechanisms (e.g., stdout/stderr container logs, audit logs, orchestrator audit, audit log from MAC, like AppArmor or SELinux). Security and Audit logs shall be collected and stored allowing security monitoring, forensic and threat detection. The possibility of forwarding relevant Security and Audit logs to external SIEM system must be in place (e.g., Syslog over TLS, REST API over HTTPS, SFTP).

**Test Name:** TC\_SECURE\_CONTAINER\_LOGGING\_CAPABILITIES

**Purpose:**

Ensure that Security and Audit logs are collected and stored allowing security monitoring, forensic and threat detection.

**Execute the following steps:**

1. The tester reviews the documentation provided by the vendor describing how logs from containerized functions are being handled and verifies that this in line with the requirement description

2. The tester verifies the forwarding to an external SIEM by enabling log forwarding, triggering a security event and verifying at the SIEM, that the event has been forwarded.

**Expected format of evidence:**

Snapshots containing the information gathered from documentation.

#### 6.1.x.3 Using trusted image repositories for container image handling

*Requirement Name*: Securing container function source by using trusted image repositories

*Requirement Description*:

The containerized NF shall use trusted/private source image repositories while building the container image.

**Test Name:** TC\_SECURE\_CONTAINER\_IMAGE\_REPOSITORIES

**Purpose:**

Ensure that containers are built using trusted image bases. Images coming from untrusted/public source code repositories (e.g., Public-DockerHub) shall not be used due to risk factors.

- HTTPS protocol for accessing internal repositories shall be used.

- Trust level of image content shall be checked to ensure source and integrity of the image.

**Execute the following steps:**

1. The tester reviews the documentation provided by the vendor describing the container build procedure and listing trusted image repositories.
2. For dynamically built containers the tester reviews the build configuration.

**Expected format of evidence:**

Snapshots of the configuration or documentation.

#### 6.1.x.4 Vulnerability scanning for containerized NF

All text from TS 33.117 [1], clause 4.4.3 applies to containerized elements. Because of the nature of containerized applications and their high dependency on 3rd party software specific vulnerability scanning tools need to be used. Therefore, the test case TC\_BVT\_VULNERABILITY\_SCANNING specified in 4.4.3 need to be enhanced with the testcase below.

*Requirement Name*: Securing container functions by vulnerability scanning

*Requirement Description*:

The containerized NF shall not contain any known vulnerabilities.

**Test Name:** TC\_SECURE\_CONTAINER\_VULNERABILITY\_SCANNING

**Purpose:**

Ensure that containers are not containing any known vulnerabilities. Trust level of image content shall be checked to ensure security and integrity of the image. Vulnerability scanning of container image shall be performed during development phase, discovering the vulnerabilities, and remediating those vulnerabilities before Developer/SO ships the container image to the Container registries. Vulnerabilities shall be resolved, and validated security patches shall be installed in a timely manner by the vendor.

**Execute the following steps:**

1. The tester runs suitable vulnerability analysis tool to scan containers for known vulnerabilities.

**Expected format of evidence:**

Snapshots of the configuration or documentation, snapshots from vulnerability scanner.

#### 6.1.x.5 Containerized NF run-time security

*Requirement Name*: Securing container functions by configuration and hardening testing

*Requirement Description*:

The containerized NF shall not contain any known misconfigurations.

**Test Name:** TC\_SECURE\_CONTAINER\_CONFIGURATION

**Purpose:**

Ensure proper Security hardening was performed. Apart from vulnerability scan of container image, analysis of container security measures implemented for FN in running state shall be performed. Test should prove that all misconfigurations were resolved, and validated security patches were installed.

Container and orchestrator in a running state shall be hardened in relation to security benchmark (e.g., CIS benchmark or other common auditing tools). Network Access Policies shall be configured for securing containerized functions by default. If network segmentation in applicable, related policies preventing lateral movement across containers should be present. Security polices shall be configured for securing PODs and Containers where applicable. Usage of Privileged container, Default Namespace, Ports, Services, Public IP Address etc. shall be restricted.

**Execute the following steps:**

1. The tester runs a benchmark analysis tool to scan container for known misconfigurations.

**Expected format of evidence:**

Snapshots of the configuration or documentation, snapshots from benchmark tool.

#### 6.1.x.6 Data protection in containerized NF

All text from TS 33.117 [1], clause 4.2.3.2.3 applies to containerized elements. Encryption at-rest, in-transit shall be applied for control plane and data plane. Secrets, credentials, keys shall be securely stored in secure way, and the access rights to those secrets, credential, keys shall be restricted rather than keeping them in configuration file.

**Execute the following steps:**

1. Review the documentation provided by the vendor describing data handling procedures.
2. Run container vulnerability analysis tool or a configuration check tool capable of analysing the way secrets are stored by the containerized functions.
3. Ensure secrets, keys, credentials are not stored in plain text.

**Expected format of evidence:**

Snapshots of the configuration or documentation, snapshots from security testing tool.

Editor’s Note: The requirement and threat references will be edited during normative phase.

\* \* \* End of Changes \* \* \* \*