**3GPP TSG-SA3 Meeting #111 *S3-23xxxx***

**Berlin, Germany, 22 -26 May 2023**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** |  | **rev** |  | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HELP***](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 | **X** | Core Network | **x** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Clarification of privilege escalation methods to check for | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Federal Office for Information Security (BSI) | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | eSCAS\_5G | | | | |  | ***Date:*** | | | 2023-05-08 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | F |  | | | | | ***Release:*** | | | Rel-17 |
|  | *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:*** | | This SCAS test checks for privilege escalation methods that need to be strictly limited in number and functionality as well as documented by the vendor.  Two main methods in UNIX/Linux based systems are file system SUID/SGID permissions and process/thread capabilities. To highlight their importance for security and guide the tester to their checking, capabilities should be explicitly named as examples for privilege escalation methods. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Add check of capabilities as additional example in pre-conditions. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The tester could miss the checking of capabilities which would lead to privilege escalation methods not being checked. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 4.2.4.1.2.1 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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 \*\*\*\*\*\*\*\*\*\*

4.2.4.1.2.1 Authenticated Privilege Escalation only

*Requirement Name*: There shall not be a privilege escalation method in interactive sessions (CLI or GUI) which allows a user to gain administrator/root privileges from another user account without re-authentication.

*Requirement Description*:

There shall not be a privilege escalation method in interactive sessions (CLI or GUI) which allows a user to gain administrator/root privileges from another user account without re-authentication. Implementation example: Disable insecure privilege escalation methods so that users are required to (re-)login directly into the account with the required permissions.

*Test Case*:

**Test Name**: TC\_OS\_PRIVILEGE

**Purpose:**

To ensure that privileged operating system functions shall not be used without successful authentication and authorization, and that violations of this requirement are documented and strictly limited in number and functionality.

**Procedure and execution steps:**

**Pre-Conditions:**

1. The manufacturer shall provide documentation of the operating system(s) used in the network product.

2. The manufacturer shall supply a list "A" of operating system functions which a system user can use to explicitly gain higher privileges, and how these functions are configured. Unix® example: sudo command and its configuration file /etc/sudoers or used Linux® capabilities.

3. The manufacturer shall supply a list "B" of operating system commands, GUI functions, and files which will execute specifically limited tasks automatically with higher privileges, even when used by a low-privileged user. List "B" shall also contain:

- configuration of these commands and GUI functions;

- owner and permission settings of files;

- justification for having the command, GUI function or file on the network product  
Unix® example: root-owned files with SUID and SGID permissions or Linux® capabilities.

NOTE: Linux® capabilities can provide a subset of root user privileges to a process rather than granting total root access. Some capabilities can be used for privilege escalation:

- cap\_setuid, cap\_setguid, cap\_setfcap, cap\_sys\_admin, cap\_sys\_module, cap\_sys\_module, cap\_dac\_read\_search, cap\_dac\_override, cap\_chown, cap\_fowner, cap\_sys\_rawio, cap\_sys\_chroot

**Execution Steps**

The accredited evaluator's test lab is required to execute the following steps:

1. The tester logs into the network product and verifies that list "A" is accurate, based on his expert knowledge of the operating system(s) used in the network product, and operating system documentation.

2. The tester verifies that entries in the list "A" require successful authentication for all users without exception, on basis of the user name and at least one authentication attribute.

3. The tester logs into the network product and verifies that list "B" is accurate, based on his expert knowledge of the operating system(s) used in the network product, and operating system documentation. Unix® example: To list files with SUID and SGID permissions and Linux® capabilities, the following commands can be used:

SUID: find / -perm -4000 -type f -exec ls {} \; > suid\_files.txt

SGID: find / -perm -2000 -type f -exec ls {} \; > sgid\_files.txt

Capabilities: getcap -r / 2>/dev/null

4. The tester verifies that file entries in the list "B" do not have write permissions for anyone else than the owner.

5. The tester verifies that entries in the list "B" only allow execution of specifically limited tasks which are needed on this network product, based on his expert knowledge of the operating system(s) used in the network product, and operating system documentation.

6. The tester logs into the network product and tests for every entry in the list "B" that it does not provide a means to execute arbitrary functions with administrator/root privileges, e.g. via a shell escape.

**Expected Results:**

1. The network product does not allow a user to gain administrator/root privileges from another user account without re-authentication.

2. If a network product provides functions and files which execute specifically limited tasks automatically with higher privileges, it ensures that these limits cannot be bypassed.

3. The system documentation about means for a user to gain administrator/root privileges from another user account accurately describes the network product.

**Expected format of evidence:**

A test report provided by the accredited evaluator's test lab which will consist of the following information:

- Documentation provided by the vendor: lists "A" and "B"

- Description of executed tests and commands

- Relevant output (e.g. screenshot or terminal log)

- Test result (passed or not passed)

\*\*\*\*\*\*\*\*\*\* END OF CHANGE \*\*\*\*\*\*\*\*\*\*