Federal Office
for Information Security

Developer Documentation ALC - Life-Cycle Support

de.fac2 - FIDO U2F Authenticator Applet, v1.34

1.4 27.02.2020

Federal Office for Information Security
Post Box 20 03 63
D-53133 Bonn

Internet: https://www.bsi.bund.de

Table of Contents

Figures

1 Introduction

This document describes the life-cycle support during the TOEs development and maintenance. All tools, development techniques and equipment which are used during the TOEs life-cycle are very reduced to use by one single developer using one single developer computer. We follow the KISS principle "Keep it simple, stupid" as a design principle which also means that the selected tools and processes are kept very small and simple as much as possible but strong enough to ensure a stable and secure development and maintenance process.

2 CM Capabilities

2.1 TOE Identification

The name and version of the de.fac2 applet are printed on the card body. It is also possible to get this information from the applet itself by performing the authenticity check procedure as described in [AGD].

The identity of the platform can be check by following the instructions in chapter 7 “Identification of the TOE configuration” from [SmartCafe_UserGuidance]

2.2 Configuration items Identification and Access

All configuration items are managed and tracked with the Git version control system. All source code, libraries and scripts as well as all evaluation documentation are stored as objects in a Git Repository. Git creates a unique identifier for each object (configuration item) that is under the version control. The unique identifier a SHA1 based value of the objects file type and its content. The objects identifiers are stored together with their contents in the Git repository. Git stores all versions of the same object (configuration item) in its internal history. The Git history is stored in such a way that the identifier of a particular version depends upon the complete development history leading up to the current one. Once it is published, it is not possible to change the old versions without it being noticed. It’s always possible to track the changes of different versions and restore specific versions.

All configuration items are added to the Git version control by using the *git add* command. Changes and updates to the added items will be observed. With the *git commit* command the changes will be stored in the Git repository. The *git log* command shows all committed changes.

All changes and commit to the Git repository are documented in the Git log. The Log contains the affected files, the changes that was made, time, date, username and commit message. As the development computer has only one user account which has access to the system, changes can only be performed from this developer account.

TOE versions are labeled as releases in git. All sources and their versions can be mapped to a release and to a specific TOE version.

For all TOE versions the same test setup has to be performed. No deviations on the test cases are needed for different TOE versions.

2.3 TOE production

The JavaCard cap file which will be installed on Sm@rtCafe platform is build from the de.fac2 source code by using an Ant build script. The Ant-Script builds the class files from the source code and converts them to the cap file. For testing purpose the script can also be used to upload the applet to the platform and set a test attestation certificate. The build script doesn’t need any user interaction.

2.4 CM plan

The following tools must be available for the developer for the configuration management:

- Git (Version 2.14.1)
- gitea (web gui for git repositories and issue management)
- BorgBackup backup (Version 1.0.11)

The TOE development includes the following activities:

2.4.1 Creation of a configuration item

New configuration items will be create inside the Integrated Development Environment (IDE). The IDE creates a new file in the file system of the operation system (OS) and adds it to the git repository. Alternatively a new configuration item can be created with other editors. In this case the item must be added manually to the git repository by calling the following command:

```
git add <filename>
```

To store the content of new added configuration items in the git repository, the addition must be confirmed and committed to git by either using the IDE build-in commit function, or manually by calling the following command:

```
git commit -m “<commit message>”
```

The commit message shall contain a description of the purpose of the newly added item.

2.4.2 Modification of a configuration item

Configuration item can be modified by using the associated editor. Changes on configuration items which are already under version control in the git repository, will be automatically detected by git. Uncommitted changes can be displayed with the following git command:

```
git diff
```

To store the changes to the git repository the commit command as described in chapter 2.4.1 must be used.

2.4.3 Deletion of a configuration item

Configuration items which are already added to the git repository can be removed from the repository and the file system by using the following git command:

```
git rm <filename>
```

2.4.4 Backup of configuration item

All configuration items will be automated backed with BorgBackup up on an external hard disk drive once a day if the development computer is in use. The backups are AES encrypted and integrity and authenticity is ensured by using HMAC-SHA256. In order to have the data stored in an emergency case, backups are stored in another room. A second backup on an encrypted USB token is stored in the BSI Building, Godesberger Allee. This backup will only be replaced after major modifications on the TOE.

2.4.5 Issue / Change Management

Question or problems which occur during the development are stored in the issues tracker system. We use a local installation of the Gitea system (Version 1.3.2) which includes a git repository management, wiki system and issue management system. New issues are kept open as long as they are not resolved. Issues could be assigned to a specific developer which isn’t used in this project since there is only one developer. The access to the issue tracking system is protected by username/password. Furthermore, the issue tracking system operates on the single developer computer which has developer and admin accounts. All roles (Admin, Developer and Tester) can have the own login to the issues tracker. Admin grants / removes access on need.

Planned changes for further versions of the applet are also tracked by the issue tracking system.

3 CM Scope

The TOE consists of the de.fac2 JavaCard applet and the Sm@rtCafe Expert C3 platform.
The assets to be protected are:

- Source Code (confidentiality, integrity, authenticity): The applet developer is responsible to ensure these protections.
- Evaluation Documentation (confidentiality, integrity, authenticity): The editors of these documents are responsible to ensure these protections.
- User Guidance (integrity, authenticity): The editors of this document are responsible to ensure these protections.
- G&D Smart Café Javacard Libraries (confidentiality, integrity, authenticity): The applet developer is responsible to ensure these protections.

The configuration list with the applets source code and development tool chain can be found in [ALC Configuration List].

4 Delivery

The general description of the delivery of the TOE (de.fac2 applet on Sm@rtCafé® Expert C3 platform) is described in this chapter.

THE CONTENT OF THIS CHAPTER HAS BEEN REMOVED FOR PUBLICATION BECAUSE IT CONTAINED CONFIDENTIAL CONTENT OF THIRD PARTIES.

5 Development Security

This chapter contains the description of different measures to keep the confidentiality, integrity, and authenticity of the TOEs sources and ensure only authorized personnel have access to the development system. The developer shall ensure that these protections are set up correctly and checks frequently that these policies are met and will update if necessary.

5.1 Physical Security

The development and composite product integration site is located at [redacted]

There is only one development PC, without peripheral devices, which is located in an access restricted room. The room can only be entered with a smartcard token with enabled access rights to this room. [redacted] Other personnel like cleaning personnel do not have any access rights to this room. [redacted]

[redacted]

Guest and personnel, that have to enter the room in case of technical or IT problems, may enter the development room only accompanied by a BSI D15 employee. Each access from guests are logged handwritten.

5.2 Procedural Security

Only the developers (see roles in chapter 5.4) have access to all configuration items on the development PC. The roles developer and composite product integrator are taken by the same persons. The OS of the development PC has only one active user which has access to the development environment. Only the admin and developer of the applet knows the file system encryption key and the user password.

The access to the development PC is controlled by the admin role. In the case the developer changes, the admin revokes the access for this current developer and switches the access to the development environment to the new developer. In the case one admin changes all admin accesses and passwords will be given to the new admin. The given passwords have then to be replaced by new passwords. There shall be at least two users with admin access.

The admin selects all team members and gives them access to the development environment. Team members must be BSI employees, that have signed a “statement of secrecy” when they were hired at BSI.

5.3 IT Security

The applet is developed on a single Desktop PC with one screen. This PC was specially procured for the development of the applet. The preinstalled operating system was removed and a fresh [redacted] operating system was installed. The whole file system is encrypted with dm-crypt and LUKS (cryptsetup version 1.6.6) with the following parameters. Encryption algorithm: AES-256 in XTS-plain64 mode with 512 bit key size, Hash algorithm: SHA-256. The PC will only start the OS after the file system encryption key was entered. Without this key, there is no access to any file on the hard drive, even if the hard drive will be removed from the development PC and connected to a different computer.

The operation system (OS) of the development PC is access controlled. Only users with an activated user profile are able to access the OS. All user profiles are protected with individual passwords. Every login attempt is logged by the OS and can be controlled by the admin.

The development PC isn't connected to any network. No remote access to the PC is possible. To transfer files between the office computers and the development PC USB drives will be used. The USB drives stores Truecrypt/Veracrypt encrypted containers. The container files are encrypted with AES algorithm and SHA512 as hash algorithm. At the development PC and/or the office PCs the containers will be opened and the files to transfer will be placed into these containers. The files themselves are additionally pgp encrypted stored in the container.
Updates and patches for the development tools will be downloaded on the office computers, stored in the encrypted containers on the USB drives and then transferred to the development PC where the patches/updates will be installed.

The applets source code and documentation are stored additionally on an external hard disk as a backup. The backup script runs once a day if the development PC is in use and stores the backup data AES-256 encrypted. The data integrity and authenticity is verified using HMAC-SHA256. The external backup drive will be stored in the developers office in a locked container.

The passwords for the encrypted backup and the OS file system encryption are encrypted stored at the Admins office PC. The passwords are shared by the admin by the "need to know" principle. If team members change, the admin changes the backup encryption password and the OS file encryption password. Knowledge about the password of the admins' office PC have only the admins themselves.

All passwords (for file encryption, PC login, Issue management, git, etc.) must have a length of at least 8 characters and should contain at least the characters A-Z, a-z and 0-9 and shouldn't contain known words in any language. All team members will be instructed to follow these rules.

[redacted]

If, despite all the security measures listed above, unintentional access to the development room (broken door etc.) or unexplained PC-behavior occur, the documents on the development PC are checked and replaced if necessary.

The development PC is also used for the generation of the attestation certificate and the corresponding private key as well as the composite product integration.

5.4 Roles

There are five roles defined:

- Admin: Grants and revokes access rights to the development PC. If there are any reasons the admin can audit logfiles on all systems. Has access to all assets.
- Developer: Has access to the source files and can check in modification to the source code repository. Developer and Admin shouldn't be the same person. This role is the owner of all assets.
- Composite Product Integrator: Has access to the source files and can check in modification to the source code repository. Developer and CPI may be the same person
- Tester: Develop and run test cases for the TOE. Tester and Developer may be the same person.
- Site-Security Manager: Informs all responsible persons about security issues.

6 Life-Cycle Definition

The development of the de.fac2 applet loosely follows the spiral model. All team members following this life cycle model.

- The objectives are determined based on the technical specification in Fido Standard [FIDOSpec] and [FIDO_U2F_Msg]. and the security requirements in [Guide SmartCafe] and [FIDOPP].
- Risk are identified and resolution are planned within each iteration. This includes mainly the identification of security risks of the actual implementation vs. performance requirements.
- The development and implementation design details directly derives from the specifications mentioned above. The modeling of packages and functional units is an iterative process, which leads to the results documented in [ADV_ARC] and [ADV_TDS]. Within each iteration an implementation which passes all tests cases was developed. These implementation versions are all kept as releases in the git repository to be able to switch back to previous versions if needed. New releases of the applet will be delivered to the composite product integrator as described in chapter 4 after all team members have approved the release.
- Next iteration of the process is planned based on the results of the current iteration. This could contain improvements in the performance and implemented algorithms. All next steps are discussed with experts for Fido and cryptography.

7 Tools and Techniques

As the de.fac2 applet is designed to run on the Sm@rtCafe platform, the programming language is JavaCard. The platform supports the JavaCard Classic edition in Version 3.0.4 ([JC304Spec])and so it's used for the implementation of the applet. Additional to the JavaCard Classic Edition SDK libraries, the applet uses the proprietary G&D Sm@rtCafe libraries to call G&D proprietary API as defined in the [SmartCafe_UserGuidance].

For building the binaries from the de.fac2 applet source code the Java Card Software Development Kit [JC_SDK] is used. The build process is controlled by Apache [Ant] build scripts. To simplify the JavaCard SDK build calls out of Ant, an external Ant task tool[1] for building JavaCard CAP files is used.

The cap file is upload and installed to the platform by using the platforms Global Platform Manger as documented in the [SmartCafe_UserGuidance]. For easy use of the Global Platform Manager interface we use the GlobalPlatformPro[2] tool.

To finalize the initialization of the de.fac2 applet an attestation certificate has to be upload to the TOE. This is done by using the applet proprietary ADPU "SET_ATTESTATION_CERT" as defined in [AGD]. For sending this APDU to the TOE the script or tool is used. This tool is part of the "pcsc-tools"[3].

The upload, installation and initialization of the applet on the platform is used for testing purpose and production of the final TOE which is done by the Composite Product Integrator (see chapter 4).

The source code is written in the Eclipse IDE (Version Oxygen 2: 4.7.2). The IDE supports the development with features like

- Keyword and syntax coloring (including inside Javadoc comments)
- Declaration of structure and source code folding
- Code completion proposals
- Import assistance for automatic creation and organization of import declarations
- Refactoring (safe rename for methods, method extraction, etc.)
- Git support

All source code files of the applet are documented with comments that can be interpreted by [Javadoc].

For testing the applets implementation the JUnit framework is used. The test cases are used to ensure each iteration of the implementation works as expected (test-driven software development). They are also part of the functional test described in [ATE].

For the generation of the preliminary attestation certificate and the private key the software XCA[4] is used (Version: 1.3.2 based on OpenSSL 1.0.2g).

1 https://github.com/martinpaljak/ant-javacard
2 https://github.com/martinpaljak/GlobalPlatformPro
3 http://ludovic.rousseau.free.fr/softwares/pcsc-tools/
4 http://sourceforge.net/projects/xca

8 Glossary

U2F Universal Second Factor

For further FIDO related terms see the „FIDO Technical Glossary“ of [FIDOSpec].

Reference Documentation

AGD	BSI: Developer Documentation AGD - User Guidance for de.fac2 Fido Authenticator Applet
SmartCafe_UserGuidance	Giesecke+Devrient Moile Security GmbH: Operation User Guidance Sm@rtCafé Expert 7.0 C3, Version 5.2/Status 07.08.17
ALC Configuration List	Bundesamt für sicherheit in der Informationstechnik: Developer Documentation - Configuration Item List - de.fac2
FIDOSpec	FIDO Alliance: Fido Alliance Universal 2nd Factor 1.1 Specifications, https://fidoalliance.org/specifications/download/
FIDO_U2F_Msg	FIDO Alliance: FIDO U2F Raw Message Formats, https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-raw-message-formats-v1.2-ps-20170411.pdf
Guide SmartCafe	G&D: Operational User Guidance Sm@rtCafé Expert 7.0 C3, Version 5.2
FIDOPP	Bundesamt für Sicherheit in der Informationstechnik: Common Criteria Protection Profile - FIDO Universal Second Factor (U2F), BSI-CC-PP-0096-2017
ADV_ARC	BSI: de.fac2 Developer Documentation ARC - Security Architecture
ADV_TDS	BSI: de.fac2 Developer Documentation TDS - TOE Design Description
JC304Spec	Oracle: Java Card Classic Platform Specification 3.0.4, http://www.oracle.com/technetwork/java/javacard/specs-jsp-136430.html
JC_SDK	Oracle: Java Card Software Development Kit, http://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-2043229.html
Ant	Apache: Apache Ant, http://ant.apache.org/
Javadoc	Oracle: Javadoc Tool, http://www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html
ATE	Bundesamt für Sicherheit in der Informationstechnik: de.fac2 Developer Documentation ATE - Developer Tests