



Smart Device Applications

Requirements and test methods

Foreseen as edition VdS 3169-1en : 2015-06 (01)

These draft guidelines are agreed upon with the public experts and may directly be used as base for testing and certification. The final version may be subject to changes.

Publisher: VdS Schadenverhütung GmbH
Publishing house: VdS Schadenverhütung GmbH

Amsterdamer Str. 172-174
50735 Köln, Germany
Phone: +49 221 77 66 0; Fax: + 49 221 77 66 341

Copyright by VdS Schadenverhütung GmbH. All rights reserved.

VdS Guidelines for Computer-based Information Systems

Smart Device Applications

Requirements and test methods

Foreseen as edition VdS 3169-1en : 2015-06 (01)

These draft guidelines are agreed upon with the public experts and may directly be used as base for testing and certification. The final version may be subject to changes.

Content

1	General	5
1.1	Scope.....	5
1.2	Validity.....	5
2	Terms and abbreviations	5
2.1	Terms.....	5
2.2	Abbreviations.....	5
3	Normative references	5
4	Classification	6
5	Requirements	7
5.1	Basic protection measures.....	7
5.1.1	Firewall.....	7
5.1.2	Virus scanner.....	7
5.1.3	User code.....	7
5.1.4	Connection to control and indication functions of technical systems.....	7
5.1.5	Update management.....	7
5.2	Measures against brute force attacks.....	7
5.2.1	Time constant.....	7
5.2.2	Length and composition of user codes.....	8
5.2.2.1	Class „1-star“.....	8
5.2.2.2	Class „2-star“.....	8
5.2.2.3	Class „3-star“.....	8
5.2.3	Complete blocking.....	8
5.3	Measures against reverse engineering.....	8
5.3.1	Standard obfuscation.....	8
5.3.2	Superior graded obfuscation.....	8
5.4	Measures against loss of confidentiality on the transmission path.....	8
5.5	Measures against keylogger.....	9
5.5.1	Individual keypad.....	9
5.5.2	Scramble individual keypad.....	9
5.6	Measures against loss of confidentiality on the smart device.....	9
5.6.1	Encrypted storage.....	9
5.6.2	Protection of the integrity by checksum function.....	9
5.6.3	Protection by secure element.....	9
5.7	Measures against root exploits.....	10

6	Test methods.....	10
6.1	Basic protection measures	10
6.1.1	Firewall.....	10
6.1.2	Virus scanner	10
6.1.3	User code.....	10
6.1.3.1	Requirements on the user code – class „1-star“.....	10
6.1.3.2	Requirements on the user code – class „2-star“.....	10
6.1.3.3	Requirements on the user code – class „3-star“.....	11
6.1.4	Update management	11
6.2	Measures against brute force attacks.....	11
6.2.1	Time constant	11
6.2.2	Complete blocking	11
6.3	Measures against reverse engineering.....	12
6.3.1	Standard obfuscation.....	12
6.3.2	Superior graded obfuscation.....	12
6.4	Measures against loss of confidentiality on the transmission path.....	12
6.5	Measures against keylogger	13
6.5.1	Individual keypad	13
6.5.2	Scramble individual keypad	13
6.6	Measures against loss of confidentiality	13
6.6.1	Encrypted storage.....	13
6.6.2	SHA-256	13
6.6.3	Secure Element	13
6.7	Measures against root exploits.....	13
7	Requirements for the software quality	14
7.1	Documentation obligation of the manufacturer.....	14
7.2	Minimum requirements for the smart device and the system software	14
7.3	Installation instruction/establishing of the App.....	14
7.4	Operation instruction.....	14
7.5	Software documentation	15
7.5.1	Version/Version scheme	15
7.5.2	Development documentation	15
7.5.3	Process description.....	15
7.5.4	Program documentation	16
7.5.5	Measures against attacks from outside	16

1 General

1.1 Scope

Smart device applications are available for multiple appliances. Often, their utility rises only by the interaction with other computerised information systems via public networks. Caused by the exchange of information via public accessible networks and interfaces as well as the mobility and spreading of the smart devices security significant risks are rising for security critical applications.

These guidelines describe the general requirements and test methods for the VdS approval of smart device applications. Excluded from this are applications for the use in connection with fire detection and alarm systems.

Note: In connection with systems for security purposes the guidelines VdS 3169-2 are valid in addition.

1.2 Validity

These guidelines may be used for testing and VdS-approval by 1st October 2014.

2 Terms and abbreviations

2.1 Terms

Secure Element

By connecting with a security module in the smartphone (secure element) an additional protection of sensible data may be reached. This secure element may be realised in form of a special SIM-card, a micro-SD-card or an implementable chip in the smartphone.

Personal Unblocking Key (PUK)

Additional code by which a complete blockade may be cancelled.

2.2 Abbreviations

PUK Personal Unblocking Key

AES Advanced Encryption Standard

3 Normative references

These guidelines contain dated and undated references to other publications. The normative references are cited at the appropriate places in the clauses, the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to these guidelines only when announced by a change of these guidelines. For undated references the latest edition of the publication referred will be applied

VdS 2465-S2 – Richtlinien für das VdS-Übertragungsprotokoll für Gefahrenmeldungen, Version 2, Ergänzung S2: Protokollerweiterung zur Anschaltung an Netze der Protokollfamilie TCP (Guidelines for the Transmission Protocol for Alarm Messages, Version 2, Supplement S2: Protokoll Add-on for Interface Connection on nets of the TCP Family)

4 Classification

The table shows which of the following requirements are to be fulfilled in which class.

Attack vector	Measure(s)	Clause			
			1-star	2-star	3-star
Basic attack vector	Firewall	5.1.1	X	X	X
	Virus scanner	5.1.2	X	X	X
	User code	5.1.3	X	X	X
	Pairing-procedure	5.1.4	*	*	*
	Update management	5.1.5	X	X	X
Brute-force attack	Time constant	5.2.1	X	X	X
	User code length	5.2.2	X	X	X
	Complete blocking	5.2.3			X
Reverse Engineering	Standard obfuscation	5.3.1	X	X	
	Superior graded obfuscation	5.3.2			X
Loss of confidentiality on the transmission path	Encryption	5.4	X	X	X
Key logger	Individual keypad	5.5.1		X	
	Scramble individual keypad	5.5.2			X
Loss of confidentiality on the smart device	Stored in the device in encrypted form	5.6.1	X	X	X
	SHA-256	5.6.2			X
	Secure element	5.6.3			X**
Root-exploit	Prevention/detection	5.7		X	X

Table 4-1: Classification

*) if applicable, see respective clause

***) if technically realisable optionally to 5.6.1 and 5.6.2

5 Requirements

5.1 Basic protection measures

5.1.1 Firewall

The smart device shall, if technically possible, dispose of a firewall which is kept automatically up-to-date.

5.1.2 Virus scanner

The smart device shall, if technically possible, dispose of a virus scanner which is kept automatically up-to-date.

5.1.3 User code

Only authorised persons shall be able to start the application. The authorisation shall be proved by input of a user code or another, similar identification feature (e.g. fingerprint).

5.1.4 Connection to control and indication functions of technical systems

The additional requirements of VdS 3169-2 are to be fulfilled, if applicable.

5.1.5 Update management

The documentation for the operator shall include an explicit hint that always the current version of the operating system of the smart device should be used.

An update management shall provide the application always being up-to-date.

The application shall regularly search for updates and inform the user as soon as an update is available (indication). The user then has a certain time which is calculated from knowledge of the application until availability of the update and results in table 5-1, to operate the update. When exceeding this time, the application shall no more be usable (forced blocking).

Class	Indication	Forced blocking
1-star	immediately	no
2-star	immediately	After 30 days
3-star	immediately	After 7 days

Table 5-1: Update management

Note: The update management may be realised also by a third program, as e.g. App Store®, Google play®.

5.2 Measures against brute-force attacks

5.2.1 Time constant

If a wrong user code has been entered, a time delay shall ensure that the next trial of entering is possible only after a certain time has elapsed. The time t is calculated depending on error trials n after $t = 2^n$ seconds.

5.2.2 Length and composition of user codes

5.2.2.1 Class „1-star“

The user code shall have at minimum 4 characters and be composed of small letters, capitals or digits. The user is to be informed in the documentation of the importance of the choice of a secure code.

5.2.2.2 Class „2-star“

The user code shall have at minimum 6 characters and be composed of small letters, capitals or digits. The user is to be informed in the documentation of the importance of the choice of a secure code.

5.2.2.3 Class „3-star“

The user code shall have at minimum 8 characters and be composed of small letters, capitals or digits. The user is to be informed in the documentation of the importance of the choice of a secure code.

The user should be required automatically in regular intervals to change his user code.

5.2.3 Complete blocking

If five wrong user codes are entered consecutively, the starting of the application is to be blocked completely.

The manufacturer shall foresee a suitable procedure by which a complete blocking may be annihilated.

Note: For annihilation of the complete blocking, a PUK may be polled for example. If a wrong PUK has been entered three times consecutively, all information concerning this application shall be deleted.

5.3 Measures against reverse engineering

5.3.1 Standard obfuscation

The source code shall be protected against reverse engineering by standard obfuscation mechanisms offered by the development system.

5.3.2 Superior graded obfuscation

The source code shall be protected against reverse engineering by superior graded obfuscation mechanisms. The obfuscation procedure which is offered by the development system is not to be used exclusively.

5.4 Measures against loss of confidentiality on the transmission path

The confidentiality and integrity of data which are to be transmitted via data networks is to be ensured. This shall be reached by suitable encryption (e.g. AES and using of checksum functions) of the connection and block chiffré algorithms (in order to display data of same content on the transmission route in different ways thus impeding conclusions on systems states/changes).

Note: Suitable encryptions, checksum functions and blocking chiffre algorithms are described in the Guidelines of the Transmission Protocol for Alarm Systems, Version 2, Amendment S2: Protocol Amendment for Connection to Networks of the Protocol Family TCP, VdS 2465-S2.

5.5 Measures against key logger

5.5.1 Individual keypad

In order to ensure protection against noting of keypad inputs (key logging), the keypad as commonly offered for the model kits for the application development shall not be used. A keypad function shall be implemented by which the information of the keypad inputs is generated and processed only within the application. Instead, an own keypad shall be programmed so that inputs are not fed via interface from the operation system into the application.

5.5.2 Scramble individual keypad

In order to ensure a sufficient protection against noting of keypad inputs (key logging) also for applications of higher classes, an individual keypad (clause 5.5.1) shall be used, which additionally scramble the alignment of the buttons for each input situation (scramble function).

5.6 Measures against loss of confidentiality on the smart device

5.6.1 Encrypted storage

The confidentiality of data which are stored on the smart device for the respective application, is to be ensured.

This shall be reached by the use of specially protected storage areas and suitable encryption (e.g. AES).

If applicable (e.g. for control and display functions of technical equipment) the following is valid:

For security reasons the information stored on the clients should be restricted to the extent which is absolutely necessary. Each information which can be retrieved from the master, should reside there exclusively.

5.6.2 Protection of the integrity by checksum function

The integrity of data which is stored on the smart device for the respective application is to be ensured.

This shall be performed by the use of suitable checksum functions (e.g. SHA-256) of the data.

5.6.3 Protection by secure element

The data which are stored on the smart device for the respective application shall be stored by the provider or a TSM (trusted service manager) in the secure element, if this is technically possible.

5.7 Measures against root exploits

The smart device is to be ensured against known root exploits. Alternatively, the application shall reliably detect if the operator has got root rights and immediately exit the application respectively prevent a request of the application.

6 Test methods

6.1 Basic protection measures

6.1.1 Firewall

It is checked if the manufacturer has sufficiently pointed out in the documentation for the operator of the application the use of a firewall.

Pass/fail criteria: The test is passed if the operation instruction contains a respective hint.

6.1.2 Virus scanner

It is checked if the manufacturer has sufficiently pointed out in the documentation for the operator of the application the use of a virus scanner.

Pass/fail criteria: The test is passed if the operation instruction contains a respective hint.

6.1.3 User code

It is checked if the starting of the application requires the input of a user code according to clause 5.1.3 or the proof of another, similar identification feature (e.g. fingerprint).

6.1.3.1 Requirements on the user code – class „1-star“

It is checked if the minimum requirements on the user code are fulfilled.

Pass/fail criteria: The test is passed if three different and randomly generated user codes entered consecutively are rejected, which do not fulfil the minimum requirements of this class.

It is checked if the user documentation contains a hint on the importance of the choice of a secure code.

Pass/fail criteria: The test is passed if the user documentation contains a note on the importance of the choice of a secure code.

6.1.3.2 Requirements on the user code – class „2-star“

It is checked if the minimum requirements on the user code are fulfilled.

Pass/fail criteria: The test is passed if three different and randomly generated user codes entered consecutively are rejected, which do not fulfil the minimum requirements of this class.

It is checked if the user documentation contains a hint on the importance of the choice of a secure code.

Pass/fail criteria: The test is passed if the user documentation contains a note on the importance of the choice of a secure code.

6.1.3.3 Requirements on the user code – class „3-star“

It is checked if the minimum requirements on the user code are fulfilled.

Pass/fail criteria: The test is passed if three different and randomly generated user codes entered consecutively are rejected, which do not fulfil the minimum requirements of this class.

It is checked if the user documentation contains a hint on the importance of the choice of a secure code.

Pass/fail criteria: The test is passed if the user documentation contains a note on the importance of the choice of a secure code.

It is checked if the user is automatically required in regular intervals to change his user code.

Pass/fail criteria: The test is passed if the user is automatically required in regular intervals to change his user code.

Note: The proof may e.g. given by the existence of a respective routine in the source code.

6.1.4 Update management

It is checked if the manufacturer indicates in the documentation for the operator of the application, that always the current version of the operation system of the smart device shall be used.

Pass/fail criteria: The test is passed if a respective hint is contained in the documentation for the operator.

It is checked if the application searches regularly for updates and informs the user as soon as an update is available.

Pass/fail criteria: The test is passed if a respective notification is given in case of an update respectively this functionality may be retraced on base of the source code.

It is checked if the application can no more be started if more than 7 resp. 30 days have passed since knowledge of the app on the existence of an update.

Pass/fail criteria: The test is passed if the app can no more be started after a defined time resp. this functionality may be retraced on base of the source code.

6.2 Measures against brute force attacks

6.2.1 Time constant

It is checked if a renewed input of a PIN code is delayed by a time constant on the device if a wrong user code has been entered before.

Pass/fail criteria: It is checked if the required delay for wrong inputs is implemented. The test has failed if the delay as specified in clause 5.2.1 has not been implemented in the application logic.

6.2.2 Complete blocking

It is checked if the starting of the application is completely blocked after having entered five times a wrong user code.

Pass/fail criteria: The test is passed if the starting of the app is blocked after having entered five times a wrong user code.

Furthermore, it is checked if the complete blocking may be deactivated by input of a PUK.

Pass/fail criteria: The test is passed if the complete blocking may be released by input of the correct PUK.

Alternatively, the method as specified by the manufacturer is checked with analogical criteria.

After triple input of three wrong PUK it is checked if all information which is lodged in the application are deleted as required.

Pass/fail criteria: The test is passed if after a triple input of a wrong PUK all user related data of the application are deleted and this may be retraced on base of the source code.

Alternatively, the method as specified by the manufacturer is checked with analogical criteria.

6.3 Measures against reverse engineering

6.3.1 Standard obfuscation

On base of the development and compilation system it is checked if the application uses a basic obfuscation function.

Pass/fail criteria: The test is passed if the obfuscation function as offered commonly by the development system of the application or similar mechanisms are used.

Note: A comparable mechanism is for example Apple's FairPlay DRM-System.

6.3.2 Superior graded obfuscation

It is checked if the application uses a higher graded obfuscation function.

Pass/fail criteria: The test is passed if an obfuscation function which is higher graded than that one offered commonly by the development system of the application or similar mechanisms are used.

6.4 Measures against loss of confidentiality on the transmission path

On base of the documentation of the manufacturer or by bypassing the data traffic it is checked if data are transmitted only in encrypted form and the encryption information is lodged such that it is protected against access.

Pass/fail criteria: The test is passed if the use of suitable encryption procedures and the access protected storage of encryption information has been proved.

Note: These may be in time of publishing of these guidelines e.g. AES-128 in connection with CBC or similar/higher graded.

6.5 Measures against key logger

6.5.1 Individual keypad

It is checked if suitable measures are realised in the application which effectively prevent the spying out of keypad inputs, e.g. by a key logger. This is for example the case if an own keypad is used within the application.

Pass/fail criteria: The test is passed if a keypad function has been implemented for which the information on keypad inputs is generated and processed only within the application.

6.5.2 Scramble individual keypad

It is checked if suitable measures are implemented in the application by which a spying out of keypad inputs, e.g. by a key logger are effectively prevented. This is for example the case if an own keypad within the application is used and for this in addition the alignment of buttons for each format is scrambled (scramble function).

Pass/fail criteria: The test is passed if a keypad function has been implemented by which the information of keypad inputs is generated and processed only within the application and the alignment of the keys is sufficiently varied.

6.6 Measures against loss of confidentiality

6.6.1 Encrypted storage

It is checked if data is stored only in encrypted form.

Pass/fail criteria: The test is passed if the use of suitable encryption procedures and the access protected lodging of application information has been proven e.g. on base of the source code.

6.6.2 SHA-256

It is checked if hash numbers are generated, if these correspond at least with SHA-256 and the hash numbers are checked in the application.

Pass/fail criteria: The test is passed if the app generates hash numbers with at least SHA-256 and a manipulation is detected.

6.6.3 Secure element

It is checked if the secure element is used and the data is stored there.

Pass/fail criteria: The test is passed if on base of the source code and – if given – further documentation it may be retraces that data of the application are stored in a secure element.

6.7 Measures against root exploits

It is checked if a method for detection of root exploits (e.g. rooting resp. jail breaking) is implemented in the application and this prevents the further execution of the application in case of successful detection.

Pass/fail criteria: The test is passed if the manufacturer gives proof of the implementation of a respective method.

7 Requirements for the software quality

7.1 Documentation obligation of the manufacturer

The documentation as described in the following shall be made available to the manufacturer of the app at the time of delivery of the app. A procedure shall be described on how the manufacturer updates and amends this documentation.

Pass/fail criteria: The test is passed if the laboratory has documentation on the app and the procedure for the documentation administration is described.

Note: The documentation should be made available in electronic form for the test. No special form of representation is requested.

7.2 Minimum requirements for the smart device and the system software

The following information shall be documented and made available to the user of the app:

- Minimum requirements for the supporting hardware (processor speed, necessary communication modules, storage requirements, etc.)
- Which system software is supported by the app
- Which supporting software if given is additionally necessary on the smart device (e.g. VPN client)

Pass/fail criteria: The test is passed if the a.m. information are contained in the documentation.

7.3 Installation instruction/establishing of the app

An instruction for installation of the app shall be made available. If configuration is to be made at the app, these are also to be documented.

Pass/fail criteria: The test is passed if the documentation contains information on the items installation and configuration.

7.4 Operation instruction

An operation instruction is to be made available to the user.

Pass/fail criteria: The test is passed if an operation instruction is available, which describes the operation of the app in an understandable manner and at least in German language and/or English language. It contains at least:

- Version of the app to which it refers
- Denomination of the CIE for which the app is suitable
- Description of the functions of the app and its use
- General notes for secure handling with personal passwords

Note 1: The operation instruction may be made available to the user in printed or in electronic form.

Note 2: Knowledge in operation and modes of operation of the IAS of the user may be presupposed.

Note 3: The basic knowledge of the user on general operation of the smart device may be presupposed.

7.5 Software documentation

The documentation of the software shall guarantee that construction and program steps (e.g. in case of change of the or one software engineer) are traceable. Furthermore, the documentation should be complete, correct and consistent.

It is pointed out that the test does not present a verification of the software according to the requirements and cannot assure that the software or the documentation is free of faults.

This documentation need not be made available to the operator and user. It presents an internal document of the manufacturer.

7.5.1 Version/version scheme

The manufacturer shall specify and keep a fixed version scheme. A differentiation regarding extent and effects of modifications of programs shall be aligned on base of this scheme. A differentiation of different program versions shall be unambiguously possible here. This shall contain at least the changes at the app.

Pass/fail criteria: The test is passed if a version scheme which fulfils the requirements as described in VdS 2203, clause 5.2.1.1 is available respectively maintained.

7.5.2 Development documentation

A development documentation exists which is maintained by the manufacturer.

Pass/fail criteria: The test is passed if development documentation is available and the manufacturer can give proof how this is maintained. Furthermore, the documentation shall contain regarding the development tools at least details (name, manufacturer/provider, version number) on the software tools (e.g. programming environment, assembler etc.) which are used for programming

7.5.3 Process description

On base of a process description the principle mode of operation of the app may be re-traced. By often parallel processes of modern programs the main program is merely to be recognised. The involved processes are here to be described each at one example for the establishing of a connection of the App with the IAS and the request for a status signal setting/unsetting.

Pass/fail criteria: The test is passed if a traceable process description for the a. m. example is available.

Note 1: Graphic diagrams are possible (e.g. sequence diagrams, flowcharts, UML or classes diagrams, etc.) or descriptions in form of a text.

Note 2: If the app has an easy and sequential construction, or the development documentation contains already a description of the program flow, a description of the program flow may be sufficient.

Note 3: The description of the program should be made such that it is traceable for a software engineer.

7.5.4 Program documentation

The app shall dispose of detailed program documentation. It shall be made available such that the rights of the manufacturer on confidentiality are preserved. The program code shows a structured, modular construction. For each program modules a description is available and an overview on the program architecture shows the interdependencies of the program modules.

Pass/fail criteria: The test is passed if the program code has a structured, modular construction and the program documentation contains descriptions of all program modules including tasks to be performed as implemented in the source code of the program. Furthermore, the program documentation contains an overview on the program architecture which illustrates the interdependencies of the program modules. Also, it is checked on base of at least three samples if a description for a program module exists and is consistent with the source code.

Note: It is not necessary that the descriptions are made in a separate document; they may also be integrated in the source code, e.g. itself or in the programming environment.

7.5.5 Measures against attacks from outside

The manufacturer shall describe protection measures which he has installed in the program against attacks from outside. These should show at least the principles and respective safety objectives. A disclosure of the source code to which it is referred is not necessary.

Pass/fail criteria: The test is passed if the measures against attacks from outside are illustrated in a traceable manner for the software engineer.