Security requirement

End User Devices

Deutsche Telekom Group

Version	5.2
Date	Jul 1, 2024
Status	Released

Publication Details

Published by
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Germany

File name	Document number	Document type
	3.33	Security requirement
Version	**State**	**Status**
5.2	Jul 1, 2024	Released
Contact	**Validity**	**Released by**
Telekom Security psa.telekom.de	Jul 1, 2024 - Jun 30, 2029	Stefan Pütz, Leiter SEC-T-TST

Summary
End User Devices

Copyright © 2024 by Deutsche Telekom AG.
All rights reserved.

Table of Contents

1.	Introduction	4
2.	User Interaction	5
3.	System hardening	8
4.	Hardware	10
5.	Bootloader	12
6.	System Update	13
7.	Firmware	18
8.	Operating System, Web Browser, Application Execution Environments	19
8.1.	Operating System	19
8.2.	Web Browser	22
8.3.	Application Execution Environment	24
9.	Special device capabilities	29
9.1.	Firewall	29
9.2.	Wireless LAN (WLAN / WiFi)	30
9.2.1.	Wireless LAN access point functionality	31
9.2.2.	Wireless LAN (WLAN) client functionality	32
9.3.	Bluetooth functionality	32
9.4.	Web administration interface	34
9.5.	Passwords	35
9.6.	Logging	37

1. Introduction

This security requirement has been prepared based on the general security policies of the Group. The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions in units which do not participate in the PSA process. These requirements shall be taken into account from the very beginning, including during the planning and decision-making processes. When implementing these security requirements, the precedence of national, international and supranational law shall be observed.

2. User Interaction

Req 1	The device must inform the user about the security relevant status.

The device has to ensure that users are informed about the security relevant state and state changes, including relevant error messages and guidance, especially before a user is prompted for a decision. The integrity and authenticity of this information has to be ensured by the device. If necessary, the device may generate audit logs for security-relevant events.

Motivation: The user has to be aware of the device's state whenever the state may influence security or creates costs.

Implementation example: Note that not all example given here need to be fulfilled by all devices!

- Indication of unencrypted or only one-sided authenticated connections (e.g. unsecure Wireless LAN, GSM as opposed to 3G and higher)
- Indication of active network connections (Bluetooth, Wireless LAN, mobile radio network, VPN)
- Indication of active applications and active specific sensors (e.g. GPS): Access to GPS is indicated by means of a symbol in the status bar.
- The usage of TLS is displayed by a lock symbol in the browser.
- The user is pointed to unencrypted GSM connections by means of the Ciphering Indicator.
- A device that can establish internet connections over different bearers (like mobile radio network, Wireless LAN, USB or Bluetooth tethering, IPSec VPN tunnel, different mobile network operators, different technologies like UMTS or higher vs. GSM, roaming, dial-up network, premium rate number, TLS vs. plain HTTP) with different security properties must indicate what bearer and/or carrier is in use (e.g. in the status bar). This aids the user in deciding whether information with a need of protection are allowed to be transmitted and enables him to terminate unwanted actions (e.g. ones that create costs): The status bar of the device displays the status of connections over different bearers.
- The device prompts the user before internet connections are established while roaming and/or the device provides a user-configurable option to disable data roaming.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Unnoticeable feasible attacks

ID: 3.33-1/5.2

Req 2	If the device generates logs for security-relevant events or informs the user about such events together with a timestamp, then the device must maintain a reliable system time (date and time).

Depending on the use case and usage scenario, the device can synchronize its time using network connections (e.g. using NTP, network time protocol) or the user can set date and time manually. If a time synchronisation is technically posibble (e.g. device has internet connection), then the device should support this.

Motivation: A reliable system time is a prerequisite for generating meaningful audit log data; this is the only way how events can be put into chronological order and be tracked over time.

Implementation example: A device with internet connectivity uses this connection in regular interval (e.g. once per week) to synchronize its real time clock (system time) using NTP. In addition, the device allows the user to manually set date and time, such that system time can also be set and corrected, even if there is currently no internet connection available. Alternatively, the device may also set its system clock based on DCF-77 or GPS radio signals.

For this requirement the following threats are relevant:
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.33-2/5.2

Req 3	Access to system resources must be displayed to the user.

On operating systems, where access to system resources with a need of protection (like PIM data, location information, file system ...) is not displayed and prompted for acceptance to the user, the application must notify the user about the fact and the type of data accessed. It must not be possible to misuse / abuse client-side paid resources, e.g. by establishing internet connections, placing calls to premium numbers or by performing buy transactions (e.g. music, applications, online content, NFC ticketing). It has to be decided per application, if prompting or notification should occur only once on or after installation, on each start of the application or on each access of the data. If making a good decision is hard, the device should use defaults and provide a user interface to configure the options.

Motivation: It is good security practice if the user is made aware and has to authorise access to his data with a need of protection.

Implementation example:

- The device prompts the user before execution of actions that were triggered automatically (through application integration / automation, as opposed to direct user interaction). Before invoking an internal (= provided by the operating system) protocol handler (e.g. tel://...), the user is prompted by the operating system whether he wants to proceed with the action or not; before invoking an externally defined protocol handler (e.g. skype:// ...), the user is prompted either by the operating system or alternatively by the application. User prompting must take place whenever an app wants to perform an action that may generate costs (e.g. sending SMS, outbound telephone call).
- The user can define for each app, which local resources (GPS, address book, ...) it is allowed to access.

For this requirement the following threats are relevant:
- Unauthorized use of services or resources
- Unnoticeable feasible attacks

ID: 3.33-3/5.2

Req 4	The device must provide secure interfaces for security relevant end user interaction (e.g. input of passwords and keys), which guarantee integrity and confidentiality of the data transmitted.

The device must support secure input facilities for secrets and provide APIs for their appropriate processing. The exact kind of support depends on the device, its intended operational environment, its features and the device's need for processing secrets.

Motivation: End user devices often perform end user interaction, including security-relevant interaction. Operational security strongly depends on the ease of use of functions for handling secrets.

Implementation example:

- The device's operating system supports user authentication by entering a PIN or a device password. Users will only use robust authentication procedures (e.g. strong keys, long passwords), if they are easy to use.
- In order to protect authentication secrets (PIN, password) against others looking over the user's shoulder, the characters entered by the user are replaced by an asterisk before being displayed. Depending on the keyboard space (and probability of unnoticed typing errors), characters may also be displayed for a short instant

before they are replaced.

- If an end user device supports cryptographic functions, it will often need a possibility to input cryptographic keys (shared secrets, hash values, activation keys, etc.) or to perform the necessary key management. Such a device could feature a keyboard, touchpad or similar that allows for easy input. Use cases also include pairing devices and controlling access to shared resources.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data

ID: 3.33-4/5.2

3. System hardening

Req 5	The user must be able to deactivate unused services and protocols.

It must be readily identifiable for the user what services and protocols are activated. The user must be able to independently deactivate unwanted services and protocols (e.g. in the device settings).

Motivation: Modern end user devices offer a multitude of functions where often convenience conflicts with data security or data privacy. At any time the user must be able to adjust usage of such functions according to his needs.

Implementation example: Based on a good user interface, or with the help of the user manual, the user can adjust the following settings according to his needs: Usage of temporary storage of secrets (e.g. password-storage in a browser), activation of interfaces (Bluetooth, WLAN, NFC, GPS, ...), usage of cloud-storage services including optional data synchronisation across devices.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-5/5.2

Req 6	The system must be implemented robustly against unexpected inputs.

Data transferred to the system must first be validated before further processing to ensure that the data corresponds to the expected data type and format. This is intended to eliminate the risk of manipulation of system processes and states by appropriately constructed data content. Validation must be carried out for any data that is transferred to the system. Examples include user input, values in data fields, and log contents.

The following typical implementation mistakes must be avoided:
- lack of validation of the length of passed data
- Incorrect assumptions about the format of data
- lack of validation of received data for conformity with the specification
- Inadequate handling of protocol deviations in received data
- Insufficient limitation of recursion when parsing complex data formats
- Insufficient implementation of whitelisting or escaping to protect against inputs outside the valid value range

Motivation: An attacker can use specifically engineered data content to try to put a system that does not sufficiently validate received data before internal processing into an unstable state or to trigger unauthorized actions within the system. The damage potential of such attacks depends on the individual system, but has a theoretical range from uncontrolled system crashes to a controlled execution of specially injected code and the resulting complete compromise of a system.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Unnoticeable feasible attacks

ID: 3.01-11/8.0

Req 7	The device must not provide accounts with predefined authentication attributes.

If accounts with predefined authentication attributes exist, then either by configuration (preferably prior to delivery) or by administration of the end user the authentication attribute must be changed to a value which is only known to the user, or alternatively the account has to be deleted or deactivated.

Motivation: Predefined authentication attributes are often known to attackers and then allow unauthorized access.

Implementation example: The user manual points the user to the fact that his device has an account with a predefined password, and the manual comprehensibly describes how the password can be changed.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.33-7/5.2

4. Hardware

Req 8	If the devices processes data or keys with a need for protection so high that physical attacks pose a real threat for those data, the device must provide protection against manipulation for the intended usage scenario.

The protection can be achieved by a combination of physical protection (e.g. housing, tamper detection) and algorithmic countermeasures (e.g. checksums, code integrity verification). If the protection provided by the device differs from what a user would typically expect, then the user must be informed about this fact in the user guidance delivered with the product.

Motivation: It has to be avoided that expectations of end users about the physical protection provided by the device are not met. The user must be able to decide whether he can install a device in the public (e.g. wireless LAN router in a restaurant) or whether additional protective measures are necessary.

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Unnoticeable feasible attacks

ID: 3.33-8/5.2

Req 9	If the devices processes data or keys with a need for protection so high that physical attacks pose a real threat for those data, the embedded processor or the motherboard of the device must contain a hardware security module (HSM) or an interface to an external HSM.

The HSM may be an external dedicated module like a SIM card or it may be part of the embedded processor or the motherboard of the device (e.g. like a trusted platform module, TPM, or a secure element, SE). If no HSM is present in the device, the solution used instead must provide comparable security.

Motivation: A Hardware Security Module typically provides mechanisms that enable secure storage of confidential data such as keys in the device, generate strong random numbers for various applications (e.g. generation of session IDs) and accelerate cryptographic computations.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unnoticeable feasible attacks

ID: 3.33-9/5.2

Req 10	If the device contains a hardware security module (HSM) or an interface to an external HSM, the operating system or some lower-layered firmware must restrict accesses to the HSM interfaces to processes that are trusted (e.g. parts of the operating system).

Motivation: This helps to prevent misuse of the HSM's functionality by untrusted applications or malware.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data

ID: 3.33-10/5.2

Req 11	If the device processes data or keys with a need for protection so high that physical attacks pose a real threat for those data, hardware access ports intended for testing purposes must be deactivated securely.

Motivation: Test access ports enable powerful reverse engineering capabilities and they might enable the disclosure of confidential data stored in the firmware image.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-11/5.2

5. Bootloader

Req 12	If the devices processes data or keys with a need for protection so high that manipulation of executable code poses a real threat for those data, the device must facilitate a secure bootloader that verifies the integrity of executable program code prior to its execution, and doing so enforces a secure boot process.

The secure boot process ensures that the system is used according to its designation and that it has not been tampered with. Whether a secure bootloader must be present in a given end user device has to be decided based on the product category. It is already state-of-the-art technology now for IPTV set top boxes as well as for most smartphones.

Motivation: Attacks against operating systems showed that the operating system protection measures could be circumvented by patching the boot loader.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-12/5.2

6. System Update

Req 13	Software and hardware of the system must be covered by security vulnerability support from the supplier.

Only software and hardware products for which there is security vulnerability support by the supplier may be used in a system.
Such support must include that the supplier

- continuously monitors and analyzes the product for whether it has been affected by security vulnerabilities,
- informs immediately about the type, severity and exploitability of vulnerabilities discovered in the product
- timely provides product updates or effective workarounds to remedy the vulnerabilities.

The security vulnerability support must be in place for the entire period in which the affected product remains in use.

Support phases with limited scope of services
Many suppliers optionally offer time-extended support for their products, which goes beyond the support phase intended for the general market, but is often associated with limitations. Some suppliers define their support fundamentally in increments, which may include limitations even during the final phase before the absolute end date of regular support.
If a product is used within support phases that are subject to limitations, it must be explicitly ensured that these restrictions do not affect the availability of security vulnerability support.

Open Source Software and Hardware
Open Source products are often developed by free organizations or communities; accordingly, contractually agreed security vulnerability support may not be available. In principle, it must also be ensured here that the organization/community (or a third party officially commissioned by them) operates a comprehensive security vulnerability management for the affected product, which meets the above-mentioned criteria and is considered to be reliably established.

Motivation: Hardware and software products for which there is no comprehensive security vulnerability support from the supplier pose a risk. This means that a product is not adequately checked to determine whether it is affected by further developed forms of attack or newly discovered vulnerabilities in technical implementations. Likewise, if there are existing security vulnerabilities in a product, no improvements (e.g. updates, patches) are provided. This results in a system whose weak points cannot be remedied, so that they remain exploitable by an attacker in order to compromise the system or to adversely affect it.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-1/8.0

Req 14	Known vulnerabilities in the software or hardware of the system must be fixed or protected against misuse.

Known vulnerabilities in software and hardware components must be fixed by installing available system updates from the supplier (e.g. patches, updates/upgrades). Alternatively, the use of workarounds (acute solutions that do not fix the vulnerability, but effectively prevent exploitation) is permissible. Workarounds should only be used temporarily and should be replaced by a regular system update as soon as possible in order to completely close the vulnerabilities.

Components that contain known, unrecoverable vulnerabilities must not be used in a system.

The treatment of newly discovered vulnerabilities must also be continuously ensured for the entire deployment phase of the system and implemented in the continuous operating processes of security patch management.

Motivation: The use of components without fixing contained vulnerabilities significantly increases the risk of a successful compromise. The attacker is additionally favored by the fact that, as a rule, not only detailed information on vulnerabilities that have already become known is openly available, but often also already adapted attack tools that facilitate active exploitation.

Implementation example: Following the initial installation of an operating system from an official installation medium, all currently available patches and security updates are installed.

Additional information:
The primary sources of known vulnerabilities in software/hardware are lists in the release notes as well as the security advisories from the official reporting channels of the supplier or independent CERTs. In particular, the reporting channels are sensibly integrated into continuous processes of security patch management for a system, so that newly discovered vulnerabilities can be registered promptly and led into operational remedial measures.
As a complementary measure to the detection of potentially still contained types of vulnerabilities that have in principle already become known, targeted vulnerability investigations of the system can be carried out. Particularly specialized tools such as automated vulnerability scanners are suitable for this purpose. Examples include: Tenable Nessus, Qualys Scanner Appliance.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-10/8.0

Req 15	The software used must be obtained from trusted sources and checked for integrity.

The software used on the system must be obtained from trusted sources and checked for integrity before installation.
This requirement applies to all types of software:

- Firmware and microcode for hardware components
- Operating systems
- Software Libraries
- Application Software
- Pre-integrated application solutions, such as software appliances or containers

as well as other software that may be used.

Trusted Sources
Trusted sources are generally considered to be:

- the official distribution and supply channels of the supplier
- third party distributors, provided they are authorized by the supplier and are a legitimate part of the supplier´s delivery channels
- internet downloads, if they are made from official provisioning servers of the supplier or authorized distributors
  (1) If the provisioning server offers various forms of downloads, those protected by encryption or cryptographic signatures must be preferred to those without such protection.
  (2) If the provisioning server secures the transport layer using cryptographic protocols (e.g. HTTPS, sftp), the associated server certificates or server keys/fingerprints must be validated with each download to confirm the

identity of the provisioning server; if validation fails, the download must be cancelled and the provisioning server has to be considered an untrusted source.

Integrity Check

The integrity check is intended to ensure that the received software is free of manipulation and malware infection. If available, the mechanisms implemented by the supplier must be used for checking.
Valid mechanisms are:

- physical seals or permanently applied certificates of authenticity (if the software is provided on physical media)
- comparison of cryptographic hash values (e.g. SHA256, SHA512) of the received software against target values, which the supplier provides separately
- verification of cryptographic signatures (e.g. GPG, certificates) with which the supplier provides its software

In addition, a check of the software using an anti-virus or anti-malware scanner is recommended (if the vendor has not implemented any of the aforementioned integrity protection mechanisms for its software, this verification is mandatory).

Extended integrity checking when pulling software from public registries

Public registries allow developers to make any of their own software projects available for use. The range includes projects from well-known companies with controlled development processes, as well as from smaller providers or amateur developers.
Examples of such registries are:

- Code registries (e.g. GitHub, Bitbucket, SourceForge, Python Package Index)
- Container registries (e.g. Docker Hub)

Software from public registries must undergo an extended integrity check before deployment.
In addition to the integrity check components described in the previous section, the extended check is intended to explicitly ensure that the software actually performs its function as described, does not contain inherent security risks such as intentionally implemented malware features, and is not affected by known security vulnerabilities. If the software has direct dependencies on third-party software projects (dependencies are very typical in open source software), which must also be obtained and installed for the use of the software, these must be included in the extended integrity check.

Methods for an extended integrity check can be, for example:

- Strict validation of project/package names (avoidance of confusion with deliberately imitated malicious software projects)
- dynamic code analysis / structured functional checks in a test environment
- static code analysis using a linter (e.g. Splint, JSLint, pylint)
- Examination using a security vulnerability scanner (e.g. Qualys, Nessus)
- Examination using a container security scanner (e.g. JFrog Xray, Harbor, Clair, Docker Scan)
- Examination using an SCA (Software Composition Analysis) tool or dependency scanner (e.g. OWASP Dependency Check, Snyk)

The test methods must be selected and appropriately combined according to the exact form of software delivery (source code, binaries/artifacts, containers).

Motivation: Software supply chains contain various attack vectors. An attacker can start at various points to manipulate software or introduce his own routines and damage or control the target environment in which the software is subsequently used. The attack can occur on the transport or transmission path or on the provisioning source itself. Accordingly, an attack is facilitated if software is not obtained from official and controlled sources or if an integrity check is omitted.
There is a particular risk for software obtained from public registries, as these are open to anyone for the provision of software projects. Perfidious attack methods are known, in which the attacker first provides completely inconspicuous, functional software for a while and as soon as it has established itself and found a certain spread, deliberately hidden malicious code is integrated in future versions. Other methods rely on similar-sounding project names for widely used existing projects or overruling version numbers to inject manipulated software into any solutions based on them.

Implementation example: Obtain the software via the official delivery channels of the supplier. Upon receipt of the software, immediately check for integrity using cryptographic checksums, as provided by the supplier, as well as scan for any infections by known malware using anti-malware / anti-virus scanners. Storage of the tested software on an internal, protected file storage and further use (e.g. rollout to the target systems) only from there.

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-2/8.0

Req 16	The device (e.g. its bootloader) must allow updating the whole device's firmware.

Motivation: This enables users to update their devices' firmware in case that errors or vulnerabilities have been found in the firmware.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-16/5.2

Req 17	The device must allow updates/patches of parts of the software.

Motivation: Fine-granular updates (e.g. of single files or applications) allow mitigating security flaws in single components or applications, e.g. in the device's web browser, or in the PDF rendering engine. Installing smaller updates takes less time, data volumen and power and minimises risks, thus there is a significantly higher probability for users to install updates promptly.

Implementation example: Updates which are distributed over-the-air should be as small as possible.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-17/5.2

Req 18	The device must check the image's integrity by signature verification or a procedure with comparable strength of mechanism before executing /storing a firmware image from / into the flash memory.

Motivation: Unchecked firmware updates can introduce malicious code into the system, therefore a strict integrity and authenticity checking is necessary to protect the user from damages of all kind (e.g. SMS attacks, leakage of credentials, key loggers).

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Disruption of availability

For this requirement the following warranty objectives are relevant:
- Integrity

ID: 3.33-18/5.2

Req 19	The device must allow a rollback to the last known good firmware in case that the installation of an update/patch has failed.

Motivation: Rollback functionality ensures that a device will be able to continue working with the old firmware version in the case that the integrity of an updated firmware could not be verified successfully. Customers have less fear to install updates and devices are operated with current firmware versions.

For this requirement the following threats are relevant:
- Disruption of availability

For this requirement the following warranty objectives are relevant:
- Integrity

ID: 3.33-19/5.2

Req 20	The device must provide a means to notify users of available updates.

The user who is notified may be an administrative user (e.g. in the case of a home gateway). If the device supports notifications, then this function should be used (“push”); otherwise the information can be given in the administrative user interface (“pull”).

Motivation: The installation of updates may yield to service interruption. Depending on the device, different preconditions have to be met in order for an update to be installed without difficulty, e.g. power supply, free memory, type of connection (speed, costs incurred) to the update server, etc. Updates should only be pushed out to the device and their installation should only be enforced, if the update is very security-critical. If the update is not installed automatically, the device administrator has to be made aware of available updates.

Implementation example:

- Mobile operating systems offer energy-efficient notification services (iOS: Apple Push Notification Service, Android: Google Firebase Cloud Messaging).
- Notification may also be done via e-mail. The e-mail address can, e.g., be requested during first use of the device.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-20/5.2

7. Firmware

Req 21	If the device stores confidential data in its software/firmware, the device must store these confidential data in encrypted form within the software/firmware.

If the firmware contains confidential data such as cryptographic keys or passwords, then these data must be stored by the device (e.g. by the firmware) in encrypted form using strong algorithms. Whether such data (e.g. private TLS keys to support TR-069 or cryptographic keys to decrypt software updates) are present in the firmware is dependent on the device capabilities.

Motivation: It is very likely that attackers can gain access to firmware images (e.g. download of the images from the internet or read out device's memory content after certain preparations, especially if the device does not provide strong physical protection mechanisms). Additional encryption of confidential data substantially raises the amount of reengineering work that has to be spent before secrets can be extracted.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data

ID: 3.33-21/5.2

Req 22	The device must contain mechanisms (e.g. cryptographic checksums, signatures) that protect data integrity for relevant data.

If the device contains self test mechanisms, transmission errors and simple manipulations can be detected. The application of cryptographic checksums can ensure the integrity of relevant data such as root certificates, public keys, device identifiers, IMEI, etc.

Motivation: This reduces the risk to use corrupt data or malicious firmware images.

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Disruption of availability

ID: 3.33-22/5.2

Req 23	The device must not contain any functionality (test / diagnosis functions, APIs, menu items, comfort features) that enables unauthorized users or applications on the device to read out confidential data from persistent memory.

Motivation: The end user device must protect confidential data from unauthorized disclosure (lost/stolen device, malicious application). Some confidential data like passwords should never be recoverable from the device.

Implementation example: Confidential data may be cryptographic keys or user credentials, or user data like e.g. health data.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data

ID: 3.33-23/5.2

8. Operating System, Web Browser, Application Execution Environments

8.1. Operating System

Req 24	The operating system of the device must enforce a strong memory protection: It must not be possible that one process can corrupt data structures within the address space of another process or within the kernel address space.

Motivation: Strong memory protection enables stable, robust and secure operation.

Implementation example: Kernel and user processes use distinct memory areas. The operating system provides every user process with its own virtual memory, building upon a processor running in protected mode.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data

ID: 3.33-24/5.2

Req 25	If the customer does not explicitly ask for it, any shell access or a similar command-line interface must not be implemented.

This requirement applies to all network interfaces (e.g. WAN, LAN, WLAN, ...) as well as to serial interfaces on the device motherboard. Exception: Business customers explicitly ask for such access.

Motivation: Shell access enables attacks and reverse engineering of internals of the end user device.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-25/5.2

Req 26	If the device has a console interface and/or a GUI, these interfaces must only be used for the output of non-confidential status or log messages.

No confidential data (such as private cryptographic keys, usernames, passwords, credentials, etc.) must be leaked via such an interface. Secrets that were chosen by the user (e.g. WiFi key) can be displayed, but only after successful authentication (e.g. after presentation of the device's admin password).

Motivation: Confidential data stored in an end user device must be protected, even if an attacker gains access to the hardware. Mobile devices might get lost; fixed-line devices might be used in areas that do not provide perfect physical protection.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-26/5.2

Req 27	Depending on the end user device, a device password must either be preconfigured or the user must be prompted to choose a device password during first use of the device.

Whether a device password has to be preconfigured depends on the kind of end user device.

Motivation: An inactivated password or identical standard passwords of mass market devices could lead to attacks against the devices if the user does not change the default password.

Implementation example: For Speedport routers, a device-individual password is preconfigured and written on a backside label. Most smartphones prompt the user to choose a device password during first usage.

For this requirement the following threats are relevant:
- Unauthorized access to the system

ID: 3.33-27/5.2

Req 28	Predefined user accounts that are not required must be deleted or at least disabled.

On many systems, there are predefined but unused user accounts (e.g. "guest") after the initial installation.

These predefined user accounts must be deleted or at least disabled immedately after the initial installation; if these measures are not feasible, the corresponding user accounts must be blocked for remote access. In any case, disabled or blocked user accounts must also be provided with an authentication attribute (e.g. a password or an SSH key) so that unauthorized use of such a user account is prevented in the event of a misconfiguration.

Excempt from the requirement to delete or disable predefined user accounts are user accounts that are used exclusively for internal use on the corresponding system and that are required for the functionality of one or more applications of the system. Even for such a user account, it must be ensured that remote access or local login is not possible and that a user of the system cannot misuse such a user account.

Motivation: User accounts that are predefined by default in a product are typically common knowledge and can be targeted by an attacker for brute force and dictionary attacks. If these user accounts are not needed in a specific system, their existence represents an unnecessary attack surface. A particular risk is posed by predefined user accounts that are preconfigured without a password or with a well-known standard password. Such user accounts can be misused directly by an attacker if their security hardening was missed due to the unplanned use in the specific system.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized use of services or resources
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-7/8.0

Req 29	If the device supports utilization by multiple users, the device must provide the correct user identity to applications and other identity consumers, enforce the user identity for all processes, and provide adequate tools for creation, management and deletion of user profiles.

In cases, where an individual user is to be identified (as opposed to identifying a terminal, PC or client application), appropriate mechanisms to protect this identity must be in place. If the Operating System (OS) offers support for multiple users, this OS functionality should be used to protect the individual identities.

Motivation: Identifying the hardware or software components is quite often not sufficient for use cases, where a particular individual (user) has to be identified. This is given particularly for use cases with data protection relevance.

Implementation example:

- A Home-PC client application which uses personalised preferences and which is used by more than one family

member, clearly benefits from having some kind of multi-user support, e.g. in the form of different profiles from which the user can select at start-up. If the application also generates personal data (e.g. bookmarks in a web browser), then it should in addition restrict access to these data to the rightful data owner, e.g. by requiring a login with a password. However, in this case it would be even better, if the PC's operating system provided this user account management and login functionality, and if the operating system provided user information to the client application (e.g. the web browser), so that no new user management would have to be implemented in the application. The same is true for upcoming applications using the TV screen or in-car infotainment systems, as well as for most tablets used today.

- Home Gateways (internet access devices) are typically used by more than one user. The need for different user profiles arises, if personal data are provided, e.g. when the Home Gateway incorporates NAS functionality.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.33-29/5.2

Req 30	The administrative settings must be clear and easy to understand as well as self-explanatory (as far as possible).

Motivation: This reduces the risk of misconfiguration.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-30/5.2

Req 31	The device must provide a possibility for the user to delete all data stored on the device in such a way, that these data cannot be recovered with justifiable effort, before the device is disposed or sold.

The way how to perform a factory reset must be well-documented, so that the user will be able to execute the factory reset if he wishes to – even if he has forgotten some or all of his credentials. On the other hand inadvertent use of the factory reset functionality must be prevented. - The data to be deleted include personal user data, account data, credentials, and configuration data.

Motivation: The user needs a way to securely delete his personal configuration and erase all user data – otherwise these data might be recovered and misused.

Implementation example:

- The user manual contains a description of the process how to perform the factory reset.
- The GSMA Terminal Steering Group describes recommendations for Local Data Wiping in a Best Practices document.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized use of services or resources

ID: 3.33-31/5.2

8.2. Web Browser

This sub-section only applies to end user devices that contain a web browser. When applications embed browser contexts, these will possibly not fulfil all requirements listed below; deviations have to be discussed with Telekom Security on an individual basis.

Req 32	If the device has a Web browser, the Web browser must support TLS (Transport Layer Security) according to recent standards.

Recent standards are as of today TLS Version 1.2 or higher.

Motivation: TLS (the successor to SSL) is the de facto standard means of securing data transmission at the application layer.

Implementation example: Current state-of-the-art: TLS 1.3 is supported, and for TLS version 1.2 or higher is enforced, and within TLS 1.2 preferrably only cipher suites that provide Perfect Forward Secrecy.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-32/5.2

Req 33	If the device has a Web browser, a current set of TLS root certificates must come pre-installed with the web browser.

Motivation: The root certificates form the trust anchor for all server certificates for web sites.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-33/5.2

Req 34	If the device has a Web browser, it must be possible to maintain the content of the certificate store for the web browser.

Maintenance (updating) of the certificate store can be implemented through timely provision of patches (e.g. for consumer devices) or through a possibility to remotely administrate the certificate store's content (e.g. for devices used and managed by business customers).

Motivation: The root certificates form the trust anchor for all server certificates for web sites. If certificates belonging to certification authorities (CAs) or to root CAs are compromised (which has already happened several times in the past - Comodo, DigiNotar), then this information must be made available within the Web browser.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data

- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-34/5.2

Req 35	If the device has a Web browser, the web browser must clearly indicate to the user security-relevant details of HTTP connections.

The web browser must clearly indicate to the user,

1. whether a connection is secured or not,
2. in case of a secure connection: what is the connection endpoint?,
3. what security objectives are comprised: (a) confidentiality (encryption), (b) integrity protection, (c) authentication.

The web browser must perform complete validation of received TLS certificates. In case of failed validation the browser must present a meaningful and comprehensible warning to the user, indicating that trust cannot be guaranteed for the connection.

Motivation: The user should be aware whether he can input confidential data and trust the content displayed.

Implementation example: Display of a lock symbol (widely used in browsers to indicate TLS connections) together with the domain name or common name / organisation taken from the certificate.

Meanwhile the majority of websites is delivered via HTTPS (with TLS). The browser should clearly point users to insecure connections (using HTTP without TLS), e.g. by using red colour and a hint "insecure" or "insecure connection".

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-35/5.2

Req 36	If the device has a Web browser, the web browser must clearly indicate to the user whether a connection is secured using an Extended Validation certificate (EV-TLS).

Motivation: The issuance of Extended Validation certificates follows more strict criteria, as detailed by the CA/Browser forum (see http://www.cabforum.org/). Extended Validation certificates are widely used for security-critical applications such as online banking.

Implementation example: Displaying the domain name or common name of the certificate in a different colour (preferably green). This display of the domain name is - even on small screens - not scrolled out of sight automatically in order to avoid user deception.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-36/5.2

Req 37	If the device has a Web browser, the web browser must indicate to the user if a secured connection is terminated (redirected to an unsecured connection).

A secured connection is left, when the protocol changes from HTTPS to HTTP. It must be easy to observe for a user when an insecure connection is used.

Motivation: This enables the user to abort transactions that are run over insecure connections.

Implementation example: For unexperienced users a dialog box with an advice is displayed. Experiences users can disable the recurring display of the dialog box. A colour-coding of the URL input field and/or of form fields ensures that also experienced users are always aware whether a web page is currently using an insecure connection.

Meanwhile the majority of websites is delivered via HTTPS (with TLS). The browser should clearly point users to insecure connections (using HTTP without TLS), e.g. by using red colour and a hint "insecure" or "insecure connection".

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data

ID: 3.33-37/5.2

8.3. Application Execution Environment

The requirements in this sub-section apply only to devices that allow the installation of additional applications or that can execute active content or downloaded code. The application execution environment can be the operating system itself, or it can be an additional layer above the operating system (e.g. Java Virtual Machine, Widget runtime).

Note: For special types of end user devices (e.g. mobile phones / smartphones), different approaches have been discussed in the past, see for example the OMTP (Open Mobile Terminal Platform) Application Security Framework (http://wacapps.net/omtp, dated March 2008).

Req 38	If the device provides an execution environment for applications, the device must provide a mechanism that effectively verifies the integrity and authenticity of additional executable code before such code is executed.

Motivation: This helps to prevent the execution of malicious code.

Implementation example:
- Code is being signed and code signatures are verified before code execution.
- Download of signed code from an application store; the device enforces code signature (unsigned code is rejected and cannot be run).

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Unnoticeable feasible attacks

ID: 3.33-38/5.2

Req 39	If the device provides an execution environment for applications, the device must provide mechanisms that enable the user to make an informed decision whether downloaded code shall be executed and whether it shall have access to the requested data (if applicable) and/or local resources.

Motivation: These mechanisms enable the user to efficiently execute control over his data.

Implementation example:
- If applications are typically reviewed before publication or developers are “well-known” (e.g. by app store ecosystems, signing schemes), and if the application is signed, then the device must verify this signature.
- Android displays the permissions (manifest.xml) of new applications to the user before their first start.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-39/5.2

Req 40	If the device provides an execution environment for applications and if it allows installation of additional applications, the device must prevent installation of an application without user consent.

The device may require prior explicit initiation of the user.

Motivation: Every application contains executable code which potentially puts the device at risk.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-40/5.2

Req 41	If the device provides an execution environment for applications, the device must support a recall mechanism for (rogue) applications.

Motivation: If the application ecosystem uses managed networks, the device should support such management.

Implementation example: In case an application turns out to be rogue, the app store provider can recall/revoke the application, which triggers the device to uninstall/remove the rogue application.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.33-41/5.2

Req 42	If the device provides an execution environment for applications, the device must provide applications with appropriate standard mechanisms (e.g. an API) for secure data transmission.

Motivation: Such mechanisms allow the device to protect transmitted data according to its protection needs.

Implementation example:
- The WebView / Widget Runtime of a mobile device implements an Application Programming Interface (API),

such that applications can establish secure TLS connections.
- The device's operating system supports VPN connections according to popular industry standards.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data

ID: 3.33-42/5.2

Req 43	If the device provides an execution environment for applications, the device must provide mechanisms for secure data storage, access control, user authentication, and authorization of access to protected resources.

These mechanisms should not only comprise secure mechanisms to reach security states (through successful authentication, similar to a "login") and to authorize resource usage, but also mechanisms to release such authorization to use resources once they are no longer used (similar to a "logout"), e.g. timeout, logout button.

Motivation: These mechanisms enable the device to enforce discretionary access control to locally stored data.

Implementation example:
- Operating systems allow for role-based definition of access permissions to resources.
- Mobile operating systems (e.g. Android, iOS) provide application-specific persistent local storage. This prevents one malicious application from reading or manipulating / erasing all other user data on the device. iOS7 allows the user to define on a per-application basis whether access to location information, contacts (address book), calendar, reminders, photos, data shared via Bluetooth, the microphone, or activity data (pedometer etc.) shall be possible for this application; the user can revoke his allowance anytime for any given application (see Settings / Privacy).
- Mobile OS (Android, iOS) offer their apps a secure storage for secrets (keychain).
- If authorization is implemented using tokens, there is a means to invalidate these tokens, e.g. through the customer web portal.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.33-43/5.2

Req 44	Stored data in need of protection must be protected against unauthorized access, modification and deletion.

The need for protection of stored data depends on its classification (e.g. according to applicable legal data privacy requirements, regulatory requirements, contractual obligations), the potential damage in the event of its misuse, and other relevant factors (e.g. the location of storage). The nature and extent of protective measures must be appropriately chosen.
Stored authentication attributes such as passwords, private keys, tokens or certificates etc. are generally considered to be in need of protection. Data that determines the functionality and security-relevant behavior of a system (e.g. system configuration files, operating systems and kernels, drivers) are also considered to be fundamentally in need of protection.

Compliance with the protection objectives of confidentiality, integrity and availability must be consistently guaranteed for stored data in need of protection. This also applies during only short-term storage (e.g. when storing in a web cache or in a temporary folder within a data processing chain).

Basically, access to data in need of protection in a system must be fully regulated on the basis of technically implemented authorization assignments and controls.
If such technical access control alone is no longer sufficient to ensure the necessary protection requirements of stored data, or if its effectiveness cannot be consistently ensured, additional cryptographic methods (e.g. encryption, signing, hashing) must be implemented. Cryptographic methods used for the storage of data must be recognised and effective for this purpose and must not have any known vulnerabilities.

Motivation: The storage of data on a system without adequate protection enables an attacker to view, use, disseminate, modify or destroy it without authorization. This potentially opens up additional attack vectors on the immediate and connected other systems and can lead to significant failures, loss of control and damage as well as resulting penalties and loss of reputation towards customers and business partners.

Implementation example: [Example 1]
A system exports data for transport to mobile media. Since the system's technical access control at the file permission level no longer applies as soon as the mobile media is removed from the system, additional measures must be taken to protect the data. Before the system writes the data to the mobile media, it is encrypted accordingly using a suitable algorithm. The associated encryption key is exchanged on a separate channel so that the data can be decrypted and processed again in the legitimate target system. An attacker who takes possession of the mobile media, on the other hand, has no access to the data.

[Example 2]
Only cryptographic hashes of passwords generated with a secure password hashing method are stored in the local user database of a system. For the system, these hashes are sufficient to authenticate users when they log on to the system. However, if an attacker can copy the user database, he does not immediately come into possession of plaintext passwords with which he could log on to the system on behalf of the users.

[Example 3]
On a system, the configuration files of the Web server can only be written by the legitimate admin in which corresponding permissions have been set in the file system. The access control of the operating system kernel thus denies all other users of the system to make changes to the configuration files of the web server; including the web server service account itself, which also reduces the attack surface from the outside in case of vulnerabilities in the web server.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Disruption of availability
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-14/8.0

Req 45	Data in need of protection must be protected against unauthorized access and modification during transmission.

The need for protection of data to be transmitted depends on its classification (e.g. according to applicable legal data privacy requirements, regulatory requirements, contractual obligations), the potential damage in the event of its misuse, and other relevant factors (e.g. transmission via public networks). The nature and extent of the protective measures must be appropriately chosen.
Authentication attributes such as passwords or tokens etc. are generally considered to be in need of protection. Data that determines the functionality and security-relevant behavior of a system (e.g. updates & patches, configuration parameters, remote maintenance, control via APIs) are also considered to be fundamentally in need of protection.

Compliance with the protection objectives of confidentiality and integrity must be consistently guaranteed during the transmission of data in need of protection.

As a rule, this requires the implementation of cryptographic methods (e.g. encryption, signatures, Hashes). Cryptographic methods may

- be applied directly to the data before transmission, which can make subsequent transmission acceptable even via insecure channels
- be used on the transmission channel to create a secure channel and protect any kind of data passing through it
- or be implemented as a combination of both.

Cryptographic methods used for the transmission of data must be recognized and effective for this purpose and must have no known vulnerabilities.

Motivation: The transmission of data without adequate protection enables an attacker to intercept, use, disseminate, modify or remove it from transmission without authorization. This potentially opens up further attack vectors on the immediate target systems as well as connected other systems and can lead to significant failures, loss of control and damage as well as resulting penalty claims and reputational losses towards customers and business partners.

Implementation example: [Example 1]
Confidential documents are encrypted before they are sent by e-mail to the customer.

[Example 2]
An administrator configures a new cloud application over the Internet. Access is via a TLS-encrypted connection ("https").

[Example 3]
A system obtains automatic software updates from an update server. The update server delivers the software updates cryptographically signed. The system can thus validate the received software updates and reliably rule out that they have been manipulated during transmission.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Disruption of availability
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-15/8.0

Req 46	If the device provides an execution environment for applications, the device must execute applications in dedicated execution environments (sandboxing) and provide to the applications clear interfaces to access user data and device capabilities outside of the application context with the possibility to restrict such access.

More extensive accesses (beyond the boundaries imposed by the sandbox) must either be impossible or only possible after confirmation by the user. Access to secrets must only be possible through well-defined protocols.

Motivation: It must be possible to control such access (to data or local resources), either by preconfigured rules of the device or by means of user interaction (e.g. prompting).

Implementation example: The device OS provides well-defined interfaces (API) that enforce the above mentioned restrictions.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data

ID: 3.33-46/5.2

9. Special device capabilities

The requirements in the following sub-sections only apply to the end user device, if it contains the respective functionality.

Req 47	Protocol stacks available on the device must be implemented robust.

Motivation: If protocol stacks like the IP stack or SIP stack are vulnerable to common threats, it may be possible to conduct Denial-of-Service attacks from remote (i.e., from the WAN network), or a remote attacker can take control over the device.

Implementation example: Well-known attacks against the IP protocol stack are detected and mitigated. Fuzzing tests have been performed against the device, yielding evidence of its robustness properties. Denial-of-Service (DoS) protection is enabled on every active network interface. - Information about well-known attacks can for example be found at organisations like www.sans.org.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Disruption of availability
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-47/5.2

9.1. Firewall

This sub-section only applies to end user devices that contain a firewall.

Req 48	If the device has firewall functionality and acts as multi-homed host, the firewall must implement a strong end system model according to RFC1122.

A definition of the term 'multi-homed host' can be found in section 3.3.4.1 of RFC1122.

Motivation: This helps to prevent DNS rebinding attacks that circumvent the same origin policy.

Implementation example: A home gateway acts as a multi-homed host (networks: LAN and internet). A smartphone may act as a multi-homed host, if it shares its mobile radio internet connection with other devices via WiFi (Personal HotSpot) or Bluetooth or USB cable (also known as "tethering").

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-48/5.2

Req 49	If the device has firewall functionality, this functionality must be always active. Neiter a fault condition, nor an operating error nor a user-accessible configuration option must not allow to deactivate the firewall.

This requirement does not prevent the device from offering its administrative users the possibility to configure the firewall rule set in order to allow port forwarding.

Motivation: The firewall is a very important security feature of the device. This service must be always on, stable and fault tolerant.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-49/5.2

Req 50	If the device has firewall functionality, only the required TCP / UDP ports must be open at the network interfaces of the end user device.

All ports that are not required must be delivered to the customer in the default state “closed”.

Motivation: Every open port is a potential security risk.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-50/5.2

Req 51	If the device has firewall functionality, the firewall functionality must dynamically adapt its rule set to the usage of the device.

The firewall must automatically open/close corresponding ports whenever services are activated/deactivated on the device.

Motivation: Every open port is a potential security risk. If a service is not configured and thus not functional, it must not be accessible via network. We must not assume end users to be skilled firewall administrators; therefore every reasonable way to automatically configure the firewall in a secure way has to be implemented.

Implementation example: If a network service (e.g. SIP client) is deactivated by the end user device configuration, the firewall automatically closes all service related TCP / UDP ports (e.g. port 5060/tcp) on all network interfaces to which the service is bound.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-51/5.2

Req 52	If the device has firewall functionality, an IPv4 stack and NAT functionality (network address translation), the firewall must implement "port-restricted cone" NAT.

A definition of port-restricted cone NAT can be found in RFC3489: An external host with source IP address X and source port P can send a packet to a port of the internal host only if the internal host has previously sent a packet to IP address X on port P.

Motivation: This ensures that only answers to requests sent from the end user device will be processed further. Any unsolicited packets are dropped.

Implementation example: A smartphone with tethering / personal hotspot feature must implement "port restricted cone".

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-52/5.2

9.2. Wireless LAN (WLAN / WiFi)

9.2.1. Wireless LAN access point functionality

The following requirements are only applicable, if the end user device provides access via wireless LAN (WLAN / WiFi), i.e., if it contains WLAN access point functionality. Devices with WLAN access point functionality include home gateways, WLAN bridges, WLAN repeaters, etc.

Req 53	If the device has wireless LAN access point functionality, by default the wireless LAN access point functionality must use encryption based on WPA2 or WPA3 and the CCMP protocol.

Motivation: Often users of end user devices will use their devices with the default configuration. Therefore wireless LAN has to be configured securely by default. Attacks against WPA2 (based on AES and CCMP) are possible since 2017 (so called KRACK attacks, see https://www.krackattacks.com/). WPA3 is current state-of-the-art technology (since 2018) and prevents some attacks against WPA2, but not all of them. Final remedy against KRACK attacks will probably only be provided with WPA3.1. Until then, WPA2 is still widely used and to be considered tolerably secure. - The even older technologies WPA (based on RC4 and TKIP) and WEP are insecure and must not be used.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-53/5.2

Req 54	If the device has wireless LAN access point functionality, this functionality is active, and a WPA2 key (or WPA3 key) is pre-configured, then this key must be a random and per device unique value of adequate length and complexity, that does not correlate to other device data (i.e., it cannot be derived from public or deduced data).

The complexity required for a pre-configured key must be equal to or even higher than the complexity enforced for a user-chosen key.

Motivation: A guessable or default WLAN password enables attacks on the WLAN interface. Tools available on the internet support so-called "war-driving" (searching for weakly secured WiFi networks). Dependencies on other device data can be found out through reverse engineering which results in attackers being able to compute and brute-force default keys.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized use of services or resources

ID: 3.33-54/5.2

Req 55	If the device has wireless LAN access point functionality, this functionality is active, and a WPA2 key is not pre-configured, then the user must be prompted for a key value and a minimum strength of the key must be enforced by the end user device.

The complexity requirements (e.g. minimum length, character set) for the WPA2/WPA3 password depend on the end user device, its password-entry capabilities, the use case and other factors such as the readability of characters printed on a label on the backside of the device.

Motivation: Only a strong enough WPA2 password can guarantee that only authorized persons gain access to resources in the wireless LAN. - The wireless LAN provided by a Home Gateway will typically remain visible for long periods of time (months or even years), will be used regularly, and will be stationary. All these aspects make it a more rewarding target for an attacker than an ad-hoc personal HotSpot provided by a smartphone while travelling on the train,

which may be visible only a few minutes (or hours). In addition, special characters might be harder to input on a smartphone than they are on the PC keyboard used for administration of a Home Gateway. All these factors have to be considered when deciding on password complexity.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized use of services or resources

ID: 3.33-55/5.2

9.2.2. Wireless LAN (WLAN) client functionality

The following requirements are only applicable, if the end user device allows connecting to a network (e.g. the internet) via WLAN.

Req 56	If the device has wireless LAN client functionality, by default this functionality must use WPA2 encryption or WPA3 encryption and the CCMP protocol.

Motivation: Attacks against WPA2 (based on AES and CCMP) are possible since 2017 (so called KRACK attacks, see https://www.krackattacks.com/). WPA3 is current state-of-the-art technology (since 2018) and prevents some attacks against WPA2, but not all of them. Final remedy against KRACK attacks will probably only be provided with WPA3.1. Until then, WPA2 is still widely used and to be considered tolerably secure. - The even older technologies WPA (based on RC4 and TKIP) and WEP are insecure and must not be used.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-56/5.2

9.3. Bluetooth functionality

The following requirements are only applicable, if the end user device contains a Bluetooth stack.

Req 57	If the device has Bluetooth functionality, the device must request user authorization (prompting) for every incoming pairing connection request.

The device must not accept connections, files, or other objects from unknown, untrusted sources. Depending on the device type, Bluetooth should also be deactivated by default, and the Bluetooth visibility of the device should also be under control of the user.

Motivation: The user must always retain control of what other Bluetooth devices connect to his device.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.33-57/5.2

Req 58	If the device has Bluetooth functionality, Bluetooth applications and profiles must use configuration and link activity indicators like LEDs or desktop icons.

Motivation: The user must be informed about what connections and profiles are active.

For this requirement the following threats are relevant:
- Unnoticeable feasible attacks

ID: 3.33-58/5.2

Req 59	If the device has Bluetooth functionality, the device must clearly identify all Bluetooth devices.

Motivation: The user must be able to clearly decide with which devices he wants to establish a connection (pairing). Even one single pairing by mistake may lead to a transmission of data with a need of protection, e.g. when the mobile phone's address book is transmitted to the wrong handsfree set in a close-by car.

Implementation example: Example 1: A Bluetooth device displays as device name its product make and model, if need be complemented by a part of the device's MAC address.
Example 2: A passenger car's handsfree set displays as device name the car manufacturer's name complemented by a part of the car serial number.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized use of services or resources
- Unnoticeable feasible attacks

ID: 3.33-59/5.2

Req 60	If the device has Bluetooth functionality, the device must use the available Bluetooth security mechanisms (authentication, encryption).

Depending on the device type and the data transmitted via Bluetooth, the device must initiate Bluetooth authentication (also known as Security Mode 3, Link Level security) immediately after the initial establishment of the Bluetooth connection, and it must activate 128-bit Bluetooth encryption immediately after mutual authentication.

Motivation: Without mutual authentication an attacker can impersonate the other device and perform man-in-the-middle attacks. Without encryption an attacker can eavesdrop the transmitted data.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

ID: 3.33-60/5.2

Req 61	If the device has Bluetooth functionality, the device must generate and store Bluetooth link keys securely.

regarding secure generation: The key negotiation protocol BR/EDR, which is used during pairing of Bluetooth devices, implements a countermeasure against the so-called KNOB attack (see https://knobattack.com/). For example, a minimum entropy length of 16 bytes could be enforced. Alternatively, the integrity of the key negotiation protocol could be protected using the Link Key. Once the Bluetooth Special Interest Group has updated the specification, the countermeasure as specified shall be implemented.regarding secure storage: The possibilities of secure key storage provided by the end user device must be used.

Motivation: regarding secure generation: During a KNOB attack, an attacker takes care during key negotiation that the entropy used for key derivation is only one byte long, which implies that only 256 different keys can be derived; an attacker can easily test all these possible keys and identify the one which is used.regarding secure storage: It has to be prevented that unauthorized processes gain access to Bluetooth link keys; otherwise a device which is controlled by an attacker could impersonate an authenticated Bluetooth device, and thus data with a need for protection might leak.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Unnoticeable feasible attacks

ID: 3.33-61/5.2

9.4. Web administration interface

Req 62	Sessions must be protected against unauthorized takeover ("session hijacking").

Interfaces that provide session functionality to the system must implement technical measures to prevent a legitimate user's session from being taken over and continued by an unauthorized third party.

Such protection can be achieved, for example, by implementing a combination of the following options that makes sense for the specific system:

- At the transport layer: Use of the TCP protocol (with its sequence numbers) and corresponding filter lists
- At the session layer: Use of the TLS Protocol
- At the application layer: Negotiation of a random secret session key between sender and receiver to authorize all session traffic (e.g. session ID, session cookie, session token)
- Use of cryptographic methods to protect session keys from eavesdropping or modification attacks

Motivation: Unprotected sessions can potentially be hijacked and continued by an attacker in order to exercise unauthorized access to the system in the context of the affected user.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-16/8.0

Req 63	The system must allow users to log out of their current session.

The system must have a feature that enables the logged-in user to log out at any time. It must not be possible to resume a logged-out session without re-authenticating the user.

Motivation: A user must retain complete control over the sessions he has established in order to be able to terminate his access to a system at any time according to the situation and thus protect data and functions exposed via this access. In addition, the user must be able to assume that sessions specifically terminated by him cannot subsequently be resumed and continued by unauthorized third parties.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-17/8.0

Req 64	Sessions must be automatically terminated after a period of inactivity adapted to the intended use.

It is necessary that sessions on a system are automatically terminated after a specified period of inactivity.

For this reason, a time-out for sessions must be set. The time period to be selected here depends on the use of the system and, if applicable, the physical environment. For example, the time-out for an application in an unsecured environment must be shorter (a few minutes) than the time-out for an application used by operations personnel for system monitoring tasks in an access-protected area (60 minutes or more).

Motivation: For an open but unused session, there is a risk that an illegitimate user may take over and continue it unnoticed in order to exercise unauthorized access to the system and the data contained therein on behalf of the affected user.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-18/8.0

9.5. Passwords

Req 65	If a password is used as an authentication attribute, users must be able to independently change the password anytime.

The system must offer a function that enables a user to change his password at any time.

When an external centralized system for user authentication is used, it is valid to redirect or implement this function on this system.

Motivation: The fact that a user can change his authentication attribute himself at any time enables him to change it promptly if he suspects that it could have been accessed by a third party.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-29/8.0

Req 66	If a password is used as an authentication attribute, a protection against online attacks like brute force and dictionary attacks that hinder password guessing must be implemented.

Online brute force and dictionary attacks aim for a regular access interface of the system while making use of automated guessing to ascertain passwords for user accounts.

To prevent this, a countermeasure or a combination of countermeasures from the following list must be implemented:
- technical enforcement of a waiting period after a login failed, right before another login attempt will be granted. The waiting period shall increase significantly with any further successive failed login attempt (for example, by doubling the waiting time after each failed attempt)
- automatic disabling of the user account after a defined quantity of successive failed login attempts (usually 5).

However, it has to be taken into account that this solution needs a process for unlocking user accounts and an attacker can abuse this to deactivate accounts and make them temporarily unusable
- Using CAPTCHA ("**C**ompletely**A**utomated**P**ublic**T**uring test to tell**C**omputers and**H**umans**A**part") to prevent automated login attempts by machines ("robots" or "bots") as much as possible. A CAPTCHA is a small task that is usually based on graphical or acoustic elements and is difficult to solve by a machine. It must be taken into account that CAPTCHA are usually not barrier-free.

In order to achieve higher security, it is often meaningful to combine two or more of the measures named here. This must be evaluated in individual cases and implemented accordingly.

Motivation: Without any protection mechanism an attacker can possibly determine a password by executing dictionary lists or automated creation of character combinations. With the guessed password than the misuse of the according user account is possible.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-25/8.0

Req 67	If passwords are used as an authentication attribute, they must not be displayed in plain text during input.

Passwords must not be displayed in legible plain text on screens or other output devices while they are entered. A display while entering must not allow any conclusions to be drawn about the characters actually used in the password.

This requirement applies to all types of password input masks and fields.
Examples of this are dialogs for password assignment, password-based login to systems or changing existing passwords.

Exceptions:
- Within an input field, an optional plain text representation of a password is permitted, provided that this plain-text representation serves a valid purpose, exists only temporarily, has to be explicitly activated by the legitimate user on a case-by-case basis and can also be deactivated again immediately by the latter.
  A valid purpose would be, for example, to allow the legitimate user an uncomplicated visual check, if necessary, that he has entered the password correctly in a login dialog before finally completing the login.
  Such an optional plain text representation of a password must remain fully in the control of the legitimate user so that he can decide on its activation/deactivation according to the situation. In the default setting of the system, the plain text representation must be deactivated.
- The typical behavior on many mobile devices (smartphones) of displaying each individual character very briefly in plain text when entering a password - in order to make it easier for the user to control input - is fundamentally permissible there. However, the full password must never be displayed in plain text on the screen.

Motivation: In the case of a plain text display, there is a risk that third parties can randomly or deliberately spy on a password via the screen output while typing.

Implementation example: When displayed on the screen, each individual character is uniformly replaced by a "*" while entering a password.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-31/8.0

9.6. Logging

Req 68	If the device is completely managed and operated by Deutsche Telekom, then applicable retention and deletion periods must be observed for security-relevant logging data that is recorded locally.

From an IT security perspective, local storage of security-relevant logging data on a system is not mandatory. Since the local storage can be damaged in the event of system malfunctions or manipulated by a successful attacker, it can only be used to a limited extent for security-related or forensic analyses. Accordingly, it is relevant for IT security that logging data is forwarded to a separate log server.

Local storage can nevertheless take place; for example, if local storage is initially indispensable when generating the logging data due to technical processes or if there are justified operational interests in also keeping logging data available locally.

Devices (e.g. Speedport) that Deutsche Telekom leases to consumer customers are not considered as completely managed and operated by Deutsche Telekom. This holds true even if Easy Support (automatic firmware update) is enabled. The responsibility for operation, and thus also for logging data, lies with the end customer. Edge Routers of business customers are an example for devices which are completely managed and operated by Deutsche Telekom.

The following basic rules must be taken into account when storing logging data locally:

- security-related logging data must be retained for a period of 90 days.[1*]
- after 90 days, stored logging data must be deleted immediately.

[1*] This requirement only applies if no additional forwarding to a separate log server is implemented on the system and the logging data is therefore only recorded locally.

Deviances

Different retention periods and deletion periods may exist due to legal or regulatory requirements (especially in connection with personal data) or may be defined by contractual agreements. In these cases, the applicable periods must be agreed individually with a Project Security Manager / Data Privacy Advisor or are specified by them.

Motivation: Logging data is an immensely important IT security tool for preventing, detecting and clearing up system faults, security and data privacy incidents. On the other hand, the recording of logging data, like any other data processing, is also subject to legal and regulatory requirements. Accordingly, guidelines must be adhered to that reconcile the two.

Implementation example: Taking into account the current legal situation and applicable data privacy regulations, the following deletion periods for locally stored security-relevant logging data are implemented on an exemplary telecommunications system:

- Standard System Logs: Deletion after 90 days at the latest
- Logging of public IP addresses: Deletion (or anonymization) after 7 days at the latest
- Logging of the assignment of dynamic public IP addresses by the telecommunication solution: Deletion after 7 days at the latest
- Logging of non-billing-relevant call detail records: Deletion after 7 days at the latest
- Logging of the content of email and SMS: Deletion after 24 hours at the latest

- Logging of the domain queries handled by the DNS server of the telecommunications solution: Deletion after 24 hours at the latest

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.33-68/5.2