Security requirement

Client computers

Deutsche Telekom Group

Version	6.0
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
	3.19	Security requirement
Version	**State**	**Status**
6.0	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
Requirements for safeguarding client computers.

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

Table of Contents

1. Introduction 4
2. System hardening 5
3. Hardware 11
4. System update 13
5. Protecting data and information 16
6. Availability and integrity 22
7. Authentication and authorization 24
8. Logging 33

1. Introduction

This security document 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 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.

This requirement document provides the basis for the client's minimum requirements need to be fulfilled to be integrated to the internal backend infrastructure of Deutsche Telekom AG group. Deviations in projects must be documented accordingly in the SDSK and assessed by the security management of Deutsche Telekom AG.

Furthermore, this requirement document serves to get a validation of the security status of those client's supposed to be integrated to the internal backend infrastructure as part of own operation.

However, further approvals or preliminary services (network, RAS, VPN etc.) are required for the client to gain access to the company network.

The terms ‘client’ or ‘client computer’ refer to all possible hardware types and operating systems, if the device is used in the context mentioned before. However, this document primarily focuses on desktop computers, laptops and tablets. Usually this implies devices running a Microsoft Windows operating system, but other systems may also relevant if used in this context.

2. System hardening

Req 1	BIOS/EFI security mechanisms must be available and configured for secure use.

Among the BIOS/EFI security mechanisms are:

- write protection option
- password protection via password conform to the password guideline of DTAG (see requirement Technical Baseline)
- boot priority
- virus protection, virus warning function
- secure boot
- etc.

Thus, for example, the possibility of booting the workstation from a removable disk must be disabled in the boot priority.

Motivation: BIOS/EFI manipulations are difficult to detect and provide unauthorised access to sensitive information on the client. Thus, for example, booting from a removable disk (e.g. USB drive) before booting Windows/Mac, enables the use of the software Kon-Boot, which provides administration rights on the client.

Implementation example: If the write protection option on the client's BIOS is active, BIOS code in the flash storage cannot be overwritten. Security relevant BIOS options can be protected against manipulation by a special administration password unknown to the workstation's user. For certain attack methods, unauthorised access to installed hard disks - and thus unauthorised inspection of information in need of protection - can be prevented, if the boot priority is clearly defined and can not be changed unwarrently.

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
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.19-1/6.0

Req 2	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 3	Only required software may be used on the system.

In the installation routines for software provided by the supplier, individual components of the software are often preselected as standard installations, which are not necessary for the operation and function of a specific system. This also includes parts of software that are installed as application examples (e.g. default web pages, sample databases, test data), but are typically not used afterwards.

Such components must be specifically deselected (not installed) during the installation of the system or - if deselection during installation is not possible - removed immediately afterwards.

In principle, no software may be used that is not required for the operation, maintenance or function of the system.

Motivation: Vulnerabilities in a system's software are gateways for attackers. By uninstalling unnecessary components, the potential attack surfaces can be significantly reduced.

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

ID: 3.01-3/8.0

Req 4	Unnecessary services must be disabled.

After the installation of systems and software products, supplier-preset, local or network-accessible services are often active that are not required for the operation and functionality of the specific system in the intended operating environment.

However, in principle only the services actually required may be active on a system.

Accordingly, all services that are not required on a system must be completely disabled immediately after installation. It must be ensured that these services remain disabled even after the system is restarted.

Motivation: Active services that are not required unnecessarily increase the attack surface of a system and, as a direct consequence, the risk of a successful compromise. This risk can be further increased if - as is often observed with services that are not required - a targeted examination and optimization of the configuration with regard to security does not take place sufficiently.

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

ID: 3.01-5/8.0

Req 5	Usage of disapproved applications must be prevented on the client.

Using respectively importing of disapproved software and programming codes (installed locally or run portable) must be prevented. This should be prevented by technical measures as far as possible, but can also be realised through restriction of the user environment.

Motivation: By using portable applications (e.g. web browser), security settings of the locally installed web browser can be bypassed, which possibly leads to malicious code getting onto the client. This shall prevent the entering of programms or scripts with unwanted impacts, not only local but also to further systems.

Additionally, system usage beyond the approved range of functions should be also prevented.

Implementation example: For technical solutions on Windows clients using Applocker respectively a 3rd party solution such as EgoSecure Application Control, McAfee Application Control, Trendmicro Endpoint Application Control, AppSense Application Manager, etc.

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: 3.19-5/6.0

Req 6	Usage of disapproved hardware must be prevented on clients.

Using respectively importing of disapproved hardware (connected without need of opening the case) must be prevented. This should be prevented by technical measures as far as possible, but can also be realised through restriction of the user environment.

Motivation: By usage of external hardware malicious software can be transfered to the workstation or sensitive data can be extracted from the it.

Implementation example: Installation of external hardware can be regulated by a specific group policy.

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

ID: 3.19-6/6.0

Req 7	The workstation must be configured in accordance with the hardening guidelines of the operating system manufacturers or vendors or rather in adherence with "Security Best Practices".

Operating system manufacturers or vendors, security organisations, etc. already have appropiate knowledge in hardening workstations and offer corresponding instructions, hardening scripts etc. for use. These "Security Best Practices" can be applied as long as contrary security requirements of the Deutsche Telekom AG group or the respective business unit are not violated.

Motivation: Application of hardening guidelines from the manufacturer as well as "Security Best Practices" reduces vul-

nerabilities due to restrictive configuration settings.

Implementation example: Implementation of CIS security benchmarks (retrieved 04/2015 http://benchmarks.cisecurity.org/index.cfm), standards of the respective manufacturer, e.g. Microsoft (retrieved 04/2015 http://blogs.technet.com/b/secguide/archive/2014/08/13/security-baselines-for-windows-8-1-windows-server-2012-r2-and-internet-explorer-11-final.aspx, etc.) or governmental institution like the Bundesamt für Sicherheit in der Informationstechnik (https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Internetsicherheit/isi-client-Anleitung_Windows-7.pdf?__blob=publicationFile).

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

ID: 3.19-7/6.0

Req 8	Redundant or rather not required tunnel adapters must be removed in the client's operating system.

Motivation: A tunnel adapter packs network packets of a different connection protocol in TCP/IP packets. By doing so, the packets are sent over the internet to its matching counterpart. P2P networks work that way, as well as the translation of IPv4 to IPv6. Because of that, information discharge from the workstation is possible.

Implementation example: On Windows clients, tunnel adapters (6-to-4 or ISATAP adapter) can be made visible, if the command "Show hidden devices" is selected under "View" in the device manager. Nevertheless the devices can only be selected and deleted separately. With the command-line based tool devcon.exe, which also works with newer Windows versions, it is more comfortable.

Implementation of the "Fix it" tools for Windows clients at https://support.microsoft.com/en-us/kb/929852.

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

ID: 3.19-8/6.0

Req 9	Ad-hoc mode via the workstation's WLAN interface must be prevented.

Motivation: Connections between multiple Wi-Fi clients without the use of any access point can be established, when in ad-hoc mode. This is insecure, due to the lack of access control.

Implementation example: Prevention of all connections to the workstation, that are unauthorised and beyond control.

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

ID: 3.19-9/6.0

Req 10	The client must contain a host-based packet filter that blocks all incoming network connections and prevents the spread of malicilous code.

To prevent the spread of malicious code the host-based packet filter may only allow requested and established ser-

vices, such as SRTP (secure realtime protocol) for Cisco Jabber, etc. Any other incoming connection must be blocked.

Motivation: Due to the host-based packet filter, unwanted incoming connection attempts are blocked. For example, so-called port scans, that allow potential attackers to check all possible spots for intrusion. The host-based packet filter should contain the spreading of malicious network programs and control incoming access to the client.

Implementation example: Usage of host-based packet filter or host intrusion prevention system. Configuring the host-based packet filter for incoming connection attempts to defined gateways and services such as SCCM, Active Directory etc.

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
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.19-10/6.0

Req 11	Remote support and administration access to the client must only occur via a secure solution.

Remote support and administration access to the workstation for troubleshooting or maintenance of the client must be integrity protected, encrypted and authenticated securely. Additionally, these connections must be logged and may only be established after the user's approval.

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

ID: 3.19-11/6.0

3. Hardware

Req 12	The integrity of the client's system configuration must be ascertainable.

The system configuration of a client includes hardware components and firmware settings (BIOS/EFI). The integrity of those must be ascertainable.

Motivation: Clients offer a large variety of options for accessing information that need to be protected as well as material values. Malicious code could possibly be installed on the flash memory permanently by BIOS/EFI manipulation (e.g. rootkit lighteater etc.).

Implementation example: To minimise the risk of hardware and firmware manipulation, the clients's case may be protected against unauthorised opening by methods such as sealings (laptops, tablets) and locks (desktop systems). Devices where the hardware and firmware is not accessible with common measures can be used alternatively (e.g. Microsoft Surface, Apple Macbook). Trusted Platform Modules (TPM) and Intel AMT (Active Management Technology) can be used to verify and monitor the integrity 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
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.19-12/6.0

Req 13	Hardware interfaces not needed must be disabled.

Hardware interfaces not needed must be disabled. This applies in particular to unsecure ones such as FireWire.

Motivation: Through hardware interfaces data can be read, extracted and manipulated, sometimes even so when the device is locked. For example, through the FireWire interface it is possible to access the main memory content via DMA (Direct Memory Access) without any authentication.

Implementation example: Disabling the affected interfaces for certain users (e.g. via group policies) or for the system in general via BIOS/UEFI settings.

ID: 3.19-13/6.0

Req 14	A mulimedia device such as camera or microphone must be disabled by default and must not be enabled remotely.

The off-site (meaning not local) control of the multimedia device like the workstation's camera or microphone must not be possible. A clear indication (e.g. light signal on the media device) must be noticeable during recording or live streaming.

The workstation's microphone can be used by those having access rights to the respective device file (at Unix e.g. /dev/audio). At Windows, e.g. the access rights to respective registration keys (HKEY_LOCAL_MACHINE\ HARDWARE\.) determine who is able to activate the computer's microphone. Enabling the control of media devices like cameras and microphones must be restricted to the local user. Therefore the use of a multimedia device must be indicated through a verifiable signal.

Motivation: Through activation of multimedia devices an attacker might have the possibility to monitor people or rooms secretly.

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

- Unauthorized use of services or resources
- Unnoticeable feasible attacks

ID: 3.19-14/6.0

4. System update

Req 15	The workstation's operating system and each installed application must be operated within the product life cycle.

The Microsoft OS of the workstation should be operated within the mainstream support and must not be operated beyond the end of the extended support. The Mac OS should not be operated beyond the Mac OS version backed by Apple Support and its previous version - also backed by the support. A UNIX/LINUX and BSD OS must not be operated beyond the period of unrestricted support by the manufacturer (currently 7 years for Novell SuSE linux and Red Hat Enterprise Linux).

Motivation: Operating systems include critical errors and security vulnerabilities, due to their complexity and design flaws. Such vulnerabilities in older operating systems are known and can be exploited by unauthorised individuals with the intention to inject malicious software into the OS or to us the workstation for their own purposes. Older operating systems are an increased target, due to new technical possibilities and vulnerabilities, and because the manufacturers do not provide any more patches/bugfixes, that could close a known vulnerability.

Implementation example: A workstation with windows OS must be migrated before the end of extended support to an OS backed by mainstream support. Alternatively, the product life cycle can be extended by acquiring the custom support from the manufacturer Microsoft.

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

ID: 3.19-15/6.0

Req 16	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 17	The workstation must feature all security related patches and updates for operating system and applications.

All updates, bugfixes and patches relaesed by the manufacturer / distributor of the operating system or an application must be tested and installed **as soon as possible**. There must be an established process on organisational level to handle all updates, bugfixes and patches for the operating system or an application released beyond that date. In analogy, the identical process is applicable to approved workarounds, as far as no update, bugfix or patch is provided by the manufacturers / distributors.

Motivation: Errors in the operating systems, that are not fixed, can cause instable system statuses and crashes and the ones, that are security relevant, can be used for specific attacks on company networks and data.

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: 3.19-17/6.0

Req 18	Any update of the workstation's operating system or applications must be downloaded from secure update servers.

A workstation must be updated via secure software distribution or rather update mechanisms. It should be ensured, that components of the workstation cannot be updated without any control. The same applies to so-called "add-in-purchases", respectively "reloading functional components".

Motivation: A compromised update might weaken the operating system in a way that leads to data leakage. Such malware could also lead to a disrupt of the workstation's operation with the aim of a possible workstation misuse.

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

ID: 3.19-18/6.0

Req 19	The workstation must have an Endpoint Detection and Response (EDR) solution.

An up-to-date EDR solution must be used on the workstation. An EDR solution collects security-relevant activity data from processes and evaluates it centrally. Alarms can be generated from malicious behavior of processes or by a spe-

cific signature (like classic virus scanners). In addition, it is possible to react directly to suspicious program behavior via a central console. For example, the client can be isolated or the malicious process terminated. Furthermore, an overview of the vulnerabilities of all clients is forwarded to a central location.

Motivation: Normal virus scanners rely purely on signatures, whereas an EDR solution also searches for anomalies in the process behavior. Signatures have the disadvantage that new signatures have to be written for new virus variants before they can be detected.

Implementation example: An example of an EDR solution is Defender for Endpoint.

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
- Unnoticeable feasible attacks

ID: 3.19-19/6.0

5. Protecting data and information

Req 20	The client must be equipped with a secure solution for full disc encryption.

Solutions, that require a multi-factor based user authentication prior to the OS boot (pre-boot authentication) and are centrally managed increase security and should be preferred on highly critical workstations. The keys for encrypting the storage should thereby be stored within a secure location. External devices such as smart cards or USB drives are regarded as particularly secure for that matter (media separation).

Clients restricted to a persistent location of use (at least protection class 1) may be excepted from this requirement.

Motivation: Systems without pre-boot authentication are vulnerable against DMA (direct memory access) and cold-boot attacks, if no appropriate countermeasures (e.g. locking vulnerable ports) are set in place.

By using an alien operating system booted from a USB drive or another a removable storage the client's hard drive can be accessed reading and writing. This could cause unauthorised inspection of information in need of protection or manipulation of the operating system.

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

ID: 3.19-20/6.0

Req 21	The client must feature appropriate tools and applications allowing to process information worth protecting accordingly to the company guidelines and protection classes.

A list of tools that allow the processing of sensitive information according to the company's guide lines and protection classes and that are approved for business use has been published by the central security management: https://yam-united.telekom.com/pages/information-wheel-portal/apps/content/approved-products-2

Alternative tools for business use may be approved by each responsible security devision.

Motivation: Protection against intrusion to information in need of protection on clients (also temporary files).

Implementation example: Installation and usage of container or file encryption. Installation and usage of secure deletion methods, even for temporary folders (%TEMP%).

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

ID: 3.19-21/6.0

Req 22	The client must feature a secure solution for encrypting external drives, such as USB drives.

Motivation: Users can easily transport data from one client to another with the help of external drives. This may quickly lead to unwanted data extraction by unauthorised persons, if they gain access to sensitive information, e.g. through loss or theft of external drives. Accessing this information may be hampered by using an encryption solution for external drives.

Implementation example: Installation and use of encryption solutions for external drives.

ID: 3.19-22/6.0

Req 23	Authentication information sent over the network must be transferred integrity protected and encrypted.

Motivation: Network connections that are unencrypted or use weak encryption algorithms are vulnerable against e.g. man-in-the-middle attacks causing authentication information getting intercepted and thus in the hands of unauthorised persons. In order that authentication information can not be intercepted, eavesdropped or manipulated during transmission via network, the communication must happen via adequately encrypted networks.

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

ID: 3.19-23/6.0

Req 24	If folder redirection or roaming profiles are used, information needing to be protected must be stored and transmitted encrypted.

User settings and user files are usually stored in the local user profile inside the folder **User**. Using folder redirection, the folder's path is redirected to a new location (a folder on the local computer or a directory of a file sharing on the network), so that the information is available on every computer in the network.

Motivation: The workstation's security mechanisms have no protective effect, when stored on a file share on the network. Thus, a unauthorised inspection of information in need of protection is possible.

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

ID: 3.19-24/6.0

Req 25	Keys required for encryption and decryption must be held secure against unauthorised access.

During use, client OS store all keys required for encryption and decryption in a main memory area not being shifted to the pagefile. This should ensure, that the keys will not be compromised, if an unauthorised person gained access to the pagefile.
Usage of "hybernation mode" and "energy saving mode" are crtitical if these functions force writing of whole memory content (RAM) to a file that contains all keys. Therefore those hibernation and energy saving modes must not be used.

Motivation: Access to the encryption and decryption keys located in the main memory area using the USB or firewire interface (cold boot attack).

Implementation example: Storing the encryption and decryption keys on secure encryption modules like TPM or smart cards and disabling hibernation and power-saving mode.

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

ID: 3.19-25/6.0

Req 26	Only protocols and algorithms asserted as safe must be used for encryption.

Protocols and algorithms asserted as insecure must be deactivated on the workstation.

Motivation: Insecure protocols and algorithms like LAN Manager, NTLMv[1], SSL 3.0 etc. offer opportunities for man-in-

the-middle attacks due to known vulnerabilities.

Implementation example: TLS 1.0 or higher with secure cypher suites must be used as encryption protocol for data transfer. For this, see http://www.internet-sicherheit.de/crypto-poster/

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

ID: 3.19-26/6.0

Req 27	When the user is inactive the client must lock the screen automatically after a suitable amount of time and ask the user to log on for reactivation.

The user must lock the client (screen lock) as soon as he leaves it unattended. For mobile devices and workstations operated in areas without protection classes (e.g. reception, info counter, fair stand etc.) it is recommended to set a shorter amount of time.

Motivation: If a user leaves his workstation for a short time, an unhampered access to all information to which the user has permission is possible for third parties. If a user leaves a workstation without activating the screen lock, this setting should ensure protection against unauthorised access.

Implementation example: The automatic screen lock can be configured in the system settings of most operating systems and may even be controlled by group policies. On Windows clients the use of a time-controlled screensaver, that requests a password for reactivation, is recommended.

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

ID: 3.19-27/6.0

Req 28	The client must only feature a single bootable operating system.

Any of the client's dual or multi boot functionality (e.g. Apple Bootcamp) must be disabled. If the use of multiple operating systems is yet required for operational reasons, any operating system must comply with the company's security requirements and be installed on a single encrypted partition separated from other operating systems.

If a hard drive encryption with pre-boot authentication is used, the OS of that hard drive encryption is considered a bootable OS, which boots and subsequently starts up another OS from the partition to be encrypted. Those solutions may be excepted by this requirement.

Motivation: By using dual-boot or multi-boot configuration, it is possible to boot an OS from another partition of the local hard drive. Therefrom it is possible to get read and write access to the hard drive installed in the workstation, peer data or rather manipulate data or the operating system.

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

ID: 3.19-28/6.0

Req 29	Any of the workstation's partitions must use a file system depicting the NTFS semantic and using access control lists.

Windows operating systems offer the possibility of using alternative file systems for backward compatibility. It is only allowed to use the file system NTFS or equal/higher. Filesystems equal to the NTFS security standard may be used on other operating systems.

Motivation: FAT32 cannot offer such level of security, as e.g. the NTFS file system provides. If a FAT32 partition or a FAT32 volume is existing on the workstation, every user with acces to the client can also access all data stored on it. Only FAT32 or an equal file system offer security settings such as rights management or monitoring.

Implementation example: A file system featuring access control lists and journaling / soft updates should be used on the workstation. Contrary to the FAT file system, other file systems like NTFS, ext3, HFS+, JFS, XFS etc. provide a specific access protection on file level as well as a higher level of data security by using journaling.

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

ID: 3.19-29/6.0

Req 30	Folders or file paths effecting the workstation's security functions must be access protected.

Motivation: At Windows workstation a malicious copy of the dynamic link library (dll files) can be placed, when accessing the folder %SystemRoot%\system32 (DLL preloading attack or binary planting attack).

Implementation example: No access to folder %SystemRoot%\system32 for the user. No granting of administrative rights on the workstation giving full access to those directories by default.

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
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.19-30/6.0

Req 31	The autonomous establishment of uncontrolled communication links through the operating system as well as installed applications of the workstation must be prevented.

Operating systems, applications, browsers and browser plugins etc. may offer features like "automatic error reporting", "automatic reload of missing component" etc. These and similar features (call home) of the workstation, respectively its resulting data connection to parties/servers outside the internal network of Deutsche Telekom AG group, must be disabled.

Motivation: By default configuration, several services of operating systems and applications establish a connection to the internet unnoticed by the user. System and/or user-specific data is thereby transferred to the OS manufacturer or other providers. Thereby it is not always obvious what kind of data is transferred in addition to user and report information. Thereby system and/or user-specific data like information about browsing habits etc. can be gathered and forwarded to unauthorised third parties without the user's consent.

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

- Unauthorized modification of data
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.19-31/6.0

Req 32	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 33	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

6. Availability and integrity

Req 34	Intrusion prevention system must be installed and configured to protect the workstation.

The host-based intrusion prevention system must prevent malware infection from outside the network. It also must prevent the spread of malicious code through infected workstations.

Motivation: To detect a potential attack against and on the workstation, a host-based intrusion prevention system uses detailed information about system processes, resource usage and device activity to interrupt the connection or alter the transmitted data. Network-based sensors are lacking those information for the possibility of discarding data packets. The use of a host-based packet filter is inherently limited regarding functionality and resistance. The aim is to reduce the existing residual risks to an acceptable level by using a combination of host-based packet filter and intrusion prevention system.

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
- Unnoticeable feasible attacks

ID: 3.19-34/6.0

Req 35	Functionality of displaying file name extensions must be enabled in the workstation.

Motivation: Windows often hides the file extension, which can be very dangerous! Attackers could rename a mailicious from "harmless.exe" into "harmless.jpg.exe". The user only sees "harmless.jpg" and expects a picture-file, but with double-clicking or other user confirmation the unwanted executable file will be executed.

Implementation example: At Windows Explorer, go to "Folder settings" and select the tab "View". Remove the check mark at "Hide extensions for known file types" and apply changes.

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

ID: 3.19-35/6.0

Req 36	A mobile storage medium/device must be checked for malware by the workstation when accessed.

Motivation: By using so called shared executable files malware secretly spreads through portable devices like USB drives or external hard drives. Users manually pass the malicious code on to others, just like they did in the past with floppy disks. Malicious code is increasingly spread through mobile storage media/devices to infect workstations.

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

ID: 3.19-36/6.0

Req 37	Security features of the workstation like DEP, ASLR etc. must be configured and used by the operating system and the user.

In 2013 Secunia found 1208 security leaks in the 50 most used applications. Only 24 percent of those were found in

Microsoft software. Thus, installed applications must be equipped with an additional protection against security flaws.

Among these security features are:

- DEP (data execution prevention)
- ASLR (adress space layout randomization)
- ASR (attack surface reduction)
- etc.

Motivation: Anti-virus manufacturers cannot take counteraction until new viruses or trojans emerge. This is even more dangerous when security flaws, whose remedy through updates takes time or can only be released with great delay due to internal tests, get known.

In case of so-called "zero-day exploits" and general threats, the use of security features might be helpful to reduce the risks. Those technologies feature special security mechanisms and install barriers an attacker has to break through in order to exploit the software's vulnerabilities.

Implementation example: For Windows workstations installation and configuration of the enhanced mitigation experience toolkit (EMET).

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: 3.19-37/6.0

7. Authentication and authorization

Req 38	The use of system functions that require protection as well as access to internal or confidential data must not be possible without prior authentication and authorization.

The use of functions of the system that require protection as well as access to data classified as internal or confidential must only be possible after the user has been uniquely identified and successfully authenticated by means of the user name and at least one authentication attribute. In addition, it must be verified that the user is authorized to access the affected functions and data within the user role assigned to him or her in the system.

An exception to this are functions and data that may be used publicly without restriction; for example, the area of a website on the Internet where only public information is provided.

Examples of features that require prior authentication include:

- Remote access to network services (such as SSH, SFTP, web services)
- Local access to the management console
- Local use of operating system and applications

Examples of authentication features that can be used:

- Passwords
- cryptographic keys or certificates (e.g., in the form of smart cards)

This requirement also applies without restriction to any machine access to the system (here the implementation is usually carried out by using so-called M2M - "Machine-to-Machine" - user accounts).

Motivation: The unambiguous authentication and authorization of access to a system are elementary to protect functions and data from misuse.

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: 3.01-19/8.0

Req 39	The automatic login (AutoLogon) to the workstation must be prevented.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.19-39/6.0

Req 40	On mobile devices (e.g. laptops, tablets) without direct access to the company network the authentication on the client must include at least two factors.

Clients without direct access to the company network must authenticate the user with at least to factors.

Those factors are
- knowledge (password, PIN, ...)

- ownership (certificate, mobile device, ...)
- inherence (fingerprint, retinal pattern, ...)

which can be combined in any possible way. Methods that keep its factors separated (e.g. smartcard with certificate + PIN input on the client) are regarded as particularly safe (media separation).

Devices that are directly connected to the company's network at a company location (via VLAN Office or CWLAN Office) can be excluded from this requirement.

Motivation: The short term memory of the human brain can only memorise between five and nine characters (average is seven characters). Due to the fact, that a majority of users can not memorise a password with more than five characters, many of them might write it down, causing potential attackers to acquire the credentials.

A long password with more than 14 characters containing randomly selected letters, digits and special characters might be difficult to decrypt. Furthermore rainbow tables for those passwords have not been calculated until now (May 2015). Unfortunate, the human brain is not always able to memorise such passwords. A two-factor authentication reduces the chance of an unauthorised access to the workstation significantly.

By using single sign-on mechanisms of the installed operating system, it is possible to access further services and thus sensitive information unjustifiably after login with an obtained password.

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

ID: 3.19-40/6.0

Req 41	Privileged user accounts must be protected with at least two authentication attributes from different factors.

A privileged user account is a user account with extended authorizations within a system. Extended authorizations enable access to configuration settings, functions or data that are not available to regular users of the system. In direct dependence on the special tasks that are carried out via a privileged user account within a system, the assigned extended authorizations can be specifically restricted or include completely unrestricted system access.

Examples of privileged user accounts:
- Accounts for administration, maintenance or troubleshooting tasks
- Accounts for user administration tasks (e.g. creating/deleting users; assigning permissions or roles; resetting passwords)
- Accounts that are authorized to legitimize, initiate or prevent business-critical processes
- Accounts that have access to data classified as SCD (Sensitive Customer Data) in the interests of Group Deutsche Telekom, its customers or the public
- Accounts that have extensive access to data defined as "personal" according to the EU-GDPR (e.g. mass retrieval of larger parts or the complete database)

A single authentication attribute for privileged user accounts with their extended authorizations is usually no longer sufficient.

In order to achieve an adequate level of protection, at least two mutually independent authentication attributes must be used. The authentication attributes must come from various factors (knowledge, ownership, inherence). A combination of authentication attributes of the same factor (e.g. two different passwords) is not permitted

This approach is commonly referred to as MFA (Multi-Factor Authentication).
A specific form of MFA is 2FA (2-factor authentication), which combines exactly two authentication attributes.

Motivation: Privileged user accounts represent an increased risk to the security of a system. If an attacker successfully compromises such a user account, he receives extensive authorizations with which he can bring the system or system

parts under his control, disrupt system functions, view/manipulate processed data or influence business-critical processes. The combination of multiple authentication attributes of different types significantly minimizes the risk of a user account being compromised.

Implementation example: Very popular is 2FA in a variant consisting of an attribute that the user knows (factor KNOWLEDGE) and an attribute that the user possesses (factor OWNERSHIP).
Examples of such a 2FA are:

- smartcard (e.g. MyCard) plus PIN
- private key plus passphrase
- classic password plus hardware token for the generation of OTPs

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: 3.01-21/8.0

Req 42	The local administration account of the workstation must be disabled.

Motivation: On a workstation with Windows OS the amended account name of the integrated admin account is identified through a unique SID and multiple login attempts on the client can be realised through brute-force attacks. All Windows OS are being shipped with a prime local admin account, whose technical security ID (SID) is always identical to all Windows OS. This account cannot be locked, if invalid login attempts are made. A change of the account name does not increase the protection.

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

ID: 3.19-42/6.0

Req 43	The predefined local guest account must be disabled.

Motivation: By exploit it is possible for an attacker to add the guest account to the group of administrators or to assign further rights to the account. The predefined local guest account represents a vulnerability and with its identity it is possible to gain information about the operating system, which benefits further attacks.

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

ID: 3.19-43/6.0

Req 44	Predefined authentication attributes must be changed.

After the takeover or initial installation of a system, there are usually predefined authentication attributes (e.g. passwords, SSH keys, SSL/TLS Certificates) in the system, as assigned by manufacturers, developers, suppliers or automated installation routines.

Such predefined authentication attributes must be changed to new, individual values immediately after the takeover or installation of the system.

Motivation: Values predefined by third parties in authentication attributes cannot be trusted because they do not represent a controlled secret. Affected authentication attributes can be misused by unauthorized persons to access and compromise systems. This risk is significantly increased if commonly known default values are used for authentication attributes (e.g. a default password for the administrator user account in a particular software product).

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: 3.01-8/8.0

Req 45	The principal of minimal rights respectively "need to know" principal must be used for accounts, applications and interfaces.

It must be found out, which jobs the user has to do on the workstation and which minimal rights derive of those. The workstation must be restricted in his rights, so that they do not contain anything beyond the derived minimum of rights. Access from locally logged-in users to sensitive directories of the workstation (other user's directories, system and program directories of the OS etc.) must be prevented. In addition to that, the unjustified enabling or disabling of functionalities, programs or interfaces by a user must also be prevented.

Applications or services on the client must be installed with as many rights as possible. Write and read access to sensitive information must be granted as restrictive as possible, e.g. using ACL.

Motivation: For example, if a web browser with a not yet fixed vulnerability is launched with administrative rights and a manipulated web site is visited, the latter might exploit the vulnerability without the user's interaction and install malware onto the workstation. If a user is working with advanced rights on the workstation, it increases the risk of impact a programm executed within the user's rights context can have.

Implementation example: An account with advanced rights e.g. as member of the group "administrators" or "power users" must not be used for daily business. A user must use an account with restricted rights for his daily business on the workstation.

Sensitive files and directories like %systemroot% or %programfiles% as well as important parts of the registry must also be protected against access through the user. Those are neither allowed to write nor edit or delete these files or in these directories. That is why the predefined Windows ACL (Access control lists) must not be changed.

Installed applications can be checked for rights respectively functionalities using Microsoft Attack Surface Analyzer (https://www.microsoft.com/en-us/download/details.aspx?id=24487).

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

ID: 3.19-45/6.0

Req 46	Safety critical settings of the workstation or its installed applications must not be editable by the user.

The control panels or access possibilities used to change safety-critical settings of operating systems or applications on the workstation must be disabled for the logged-in user and/or not operational by the user through further technical measures. This applies equally to other important mechanisms (registry, API call etc.).

Motivation: For instance by changing the Internet Explorer's settings for the functionality "Trusted sites", a manipulated

web site with malicious code can be classified as trustworthy, which then might perform manipulations on the local workstation. By that, users could disable established security settings unknowingly and thus open a potential security leak for malware.

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
- Denial of executed activities

ID: 3.19-46/6.0

Req 47	A domain account of the Active Directory of group Deutsche Telekom AG must be used for administration and support of the workstation.

A required administration process and/or troubleshooting if needed must not be made through a local administration account. The domain account used for that matter must not be member of the domain user accounts used for daily business. Furthermore no uniform or derivable master password must be used for all workstation.

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

ID: 3.19-47/6.0

Req 48	If a password is used as an authentication attribute, it must have at least 12 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

A system may only accept passwords that comply with the following complexity rules:
- Minimum length of 12 characters.
- Comprising at least three of the following four character categories:
  - lower-case letters
  - upper-case letters
  - digits
  - special characters

The usable maximum length of passwords shall not be limited to less then 25 characters. This will provide more freedom to End Users when composing individual memorizable passwords and helps to prevent undesired behavior in password handling.

When a password is assigned, the system must ensure that the password meets these policies. This must be preferably enforced by technical measures; if such cannot be implemented, organizational measures must be established.
If a central system is used for user authentication [see also Root Security Requirements Document[i] "3.69 IAM (Identity Access Management) - Framework"], it is valid to forward or delegate this task to that central system.

Permissible deviation in the password minimum length
Under security-related criteria, conditions can potentially be identified for a system that enable the minimum password length to be reduced:
- It is generally permissible to reduce the minimum password length for systems that use additional independent authentication attributes within the authentication process in addition to the password (implementation of 2-Factor or Multi-Factor Authentication).
- Any reduction in the minimum password length must be assessed individually by a technical security advisor

(e. g. a PSM from Telekom Security) and confirmed as permissible. In the assessment, the surrounding technical, organizational and legal framework parameters must be taken into account, as well as the system-specific protection requirements and the potential amount of damage in the event of security incidents.

- The absolute minimum value of 8 characters length for passwords must not be undercut.

Motivation: Passwords with the above complexity offer contemporary robustness against attacks coupled with acceptable user friendliness. Passwords with this level of complexity have proven their efficiency in practice. Trivial and short passwords are susceptible to brute force and dictionary attacks and are therefore easy for attackers to determine. Once a password has been ascertained it can be used by an attacker for unauthorized access to the system and the data on 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
- Denial of executed activities

ID: 3.01-26/8.0

Req 49	If a password is used as an authentication attribute, it must be changed after 12 months at the latest.

The maximum permitted usage period for passwords is 12 months.
If a password reaches the maximum permitted usage period, it must be changed.

For this purpose, the system must automatically inform the user about the expired usage period the next time he logs on to the system and immediately guide him through a dialog to change the password. Access to the system must no longer be permitted without a successfully completed password change.
For technical user accounts (M2M or Machine-2-Machine), which are used for the authentication and authorization of systems among themselves or by applications on a system, automated solutions must also be implemented to comply with the permitted usage period for passwords.

Alternatively, if such an automatic mapping of the process for changing the password cannot be implemented, an effective organizational measure must be applied instead, wich ensures a binding manual password change at the end of the permissible period of use.

Motivation: Unlike more modern authentication attributes, passwords are easier to attack. Without specific measures for reliable, technically automated detection of compromises, the risk of a password being discovered or broken by an attacker can increase considerably over time.

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
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-30/8.0

Req 50	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 51	If a password is used as an authentication attribute for technical accounts, it must have at least 30 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

Technical user accounts are characterized by the fact that they are not used by people. Instead, they are used to authenticate and authorize systems to each other or applications on a system.

A system must only use passwords for technical user accounts that meet the following complexity:

- Minimum length of 30 characters
- Comprising at least three of the following four character categories:
    - lower-case letters
    - upper-case letters
    - digits
    - special characters

Motivation: Due to their use in machine-to-machine (M2M) communication scenarios, technical user accounts are often equipped with privileges that can be of high interest to an attacker to compromise infrastructures. Without mechanisms of extensive compromise detection, the risk of a password being determined or broken by an attacker can increase significantly over time. A significant increase in password length counteracts these risks and can also be implemented particularly easily in M2M scenarios, since handling a very long password is not a particular challenge for a machine (as opposed to a person).

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: 3.01-27/8.0

Req 52	If a password is used as an authentication attribute, the reuse of previous passwords must be prevented.

A history of the previously used passwords must be recorded for each user account. When a password change is initiated for a user account, the new password must be compared with this password history. If the reuse of a password is detected, the password change must be rejected. This validation process must be implemented in the system on the basis of technical measures. If a central IAM system is used for user authentication, the implementation can be forwarded to the central IAM system or outsourced there [see also Root Security Requirements Document[i] "3.69 IAM (Identity Access Management) - Framework"].

In general, the password history should ensure that a password that has already been used can never be used again.

However, due to technical limitations, a password history cannot be recorded indefinitely in many IT/NT products. In this case, the following basic rules must be observed:

- a password that has already been used must not be reusable for a period of at least 60 days (measured from the point in time at which the affected password was replaced by another)
- in systems in which the period of at least 60 days cannot be implemented, the longest possible period must be configured. In addition, it must be confirmed by a Project Security Manager (PSM) that the configured period is still sufficient in the overall context of the system with regard to the security requirement.

Annotation:
Some IT/NT products do not offer any technical configuration parameters with which the password history can be linked directly to a time period, but only allow the definition of the number of passwords to be recorded. In such cases, the time period can alternatively be ensured by linking the following, usually generally available configuration parameters. Within the resulting policy, a user can only change his password once a day and, due to the number of passwords recorded, can reuse an old password effectively after 60 days at the earliest.

- Minimum Password Age: 1 day
- Password History: Record of the last 60 passwords used

With this implementation variant, it should be noted that the minimum age for the password should not be more than one day in order not to inappropriately restrict the user with regard to the fundamental need to be able to change the password independently at any time.

Motivation: Users prefer passwords that are easy to remember and often use them repeatedly over long periods of time when the system allows. From the user's point of view, the behavior is understandable, but effectively leads to a considerable reduction in the protective effect of this authentication parameter. With adequate knowledge of the user or information obtained from previous system compromises, an attacker can gain access to supposedly protected user accounts. Particularly in situations in which new initial passwords are assigned centrally as part of an acute risk treatment, but users change them immediately to a previous password for the sake of simplicity, there is a high risk that an attacker will resume illegal access. It is therefore important to prevent users from reusing old passwords.

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

8. Logging

Req 53	Security relevant Windows operationsystem events must be logged with a precise timestamp and a unique system reference.

Windows operating systems events must log the occurrence of security-relevant incidents. So that these events can be evaluated and classified, they must be logged together with a unique system reference (e.g., hostname, IP or MAC address) and the exact time the incident occurred.

Logging must be done considering the currently valid legal, wage and company regulations. This regulations state among others that logging of events can be done only earmarked. Logging of events for doing a work control of employees is not allowed.

Typical events are:

Event	Event data to be logged
Successful log on / use from accounts	- Account - Source (IP-Adress) in case of remote access - LogonType - Type of log-on (process and log-on type, both contained in event ID 4624) - Authorization for log-on (event ID 4672) - Session ID
Log off from accounts	- Session ID - Account
Events of the Antivirussystem	- Update of signatures - Activating/deactivating of the antivirussystem - Identification of threats - Reaction of threats
Creation of autostart functions as far as possible such as (services, scheduled tasks, autostart in the registry (Run, RunOnce))	- Affected Event IDs (not a complete list): • Scheduled TasksSecurity.evtx: 4624, 4672,4698, 4702, 4699, 4700, 4701Microsoft-Windows-Task-Scheduler%4Operational.evtx: 106, 140, 141, 200, 201 • ServicesSecurity.evtx: 4624 Logon Type 3, 4697system.evtx: 7034, 7035, 7036, 7040, 745 • WMISecurity.evtx: 4624 Logon Type 3, 4672Microsoft-Windows-WMI-Activity%4Operational.evtx: 5857, 5860,5861 - Basically, the following information should be included in the logs for autostart services • Started Programm • Starting conditions • Rights of the process
Log manipulation	- Erase of the logfiles (event id 1102)

Use of remote commands; incoming and outgoing	- WMI Remoting • Security.evtx: 4624, 4672 • Microsoft-Windows-WMI-Activity%4Operational.evtx: 5857, 5860, 5861 - PowerShell Remoting • Security.evtx: 4624, 4672 • Microsoft-Windows-PowerShell%4Operational.evtx: 4103, 4104, 53504 • Windows PowerShell.evtx: 400, 403, 800 • Microsoft-Windows-WinRM$4Operational.evtx: 91, 168
Execution of commands / start of processes as covered by Defender ATP, among others	- Starting conditions - Rights of Process - If available: Hash about the executed binary - If available: Network connections established by process

Motivation: Logging security-relevant events is a basic requirement for detecting ongoing attacks as well as attacks that have already occurred. This is the only way in which suitable measures can be taken to maintain or restore system security. Furthermore, the logging data is used as evidence so that legal steps can be taken against attackers.

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

ID: 3.19-53/6.0

Req 54	Security-relevant logging data must be provided to a security incident detection and response solution in near-real-time.

An acceptable detection and response solution for the purposes of this requirement must meet the following criteria:
- There is a continuous analysis and correlation of the logging data provided by connected IT/NT systems.
- The identification of threatening activities and ongoing attacks takes place on the basis of a continuously updated detection pattern database
- Integrated alerting processes trigger the initiation of countermeasures in the event of a find.

Such a solution is not the same as a log server in terms of its function. Accordingly, the provision to a detection and response solution must always be in addition to an existing forwarding to a log server. An exception exists if the log server used provides the logging data to a detection and response solution immediately after receipt. In this case, it is sufficient for the system to forward the logging data only to the log server.
The logging data must be provided to the detection and response solution within 5 minutes ("near-real-time") from the time it is generated in the system.

The specific technical interfaces and protocols to be supported for deployment are determined by the detection and response solution available in the system's operating environment as well as the technical basis of the system itself.

Motivation: IT/NT systems are constantly exposed to security threats and attacks. By means of a central analysis and the correlation of recorded system events of entire IT landscapes in almost real time, current threats and ongoing attacks can be detected even beyond system boundaries and appropriate countermeasures can be initiated promptly. The prompt delivery of logging data to the detection and response solution is essential for its effectiveness.

Implementation example: **Example 1:** A system based on a classic server collects security-relevant logging data in the operating system's local syslog or Windows Event Log. Logging data is continuously forwarded to a Security Information and Event Management (SIEM) for further analysis, intrusion detection, and alarm handling.

Example 2: An Endpoint Detection and Response (EDR) agent is running on a system. The EDR agent continuously forwards the security-related logging data it collects on the system to the central backend of the EDR solution, where further analysis, intrusion detection, and alarm handling takes place

Example 3: Systems designed on the basis of cloud-native functions generate logging data directly from these functions using the cloud's own logging and monitoring infrastructure. In such a system, the generation of security-relevant logging data is specifically activated in the corresponding configuration options of the cloud-native functions used. The resulting logging data is continuously transferred to a SIEM (Security Information and Event Management) for further analysis, intrusion detection and alarm handling. Some cloud providers already provide some attack pattern detection in the cloud's own monitoring infrastructure and generate specific cloud alerts from it. These are also transferred to the SIEM.

Example 4: A system is based on a SaaS ("Software as a Service") application from a cloud provider. Typically, in the SaaS model, the cloud provider is responsible for all operational aspects of the application, including security management. The cloud customer itself has no influence on the underlying technical infrastructure or the operation of the application. Based on contractual agreements, recognized certificates, and documented self-disclosures from the cloud provider, it is formally verified that the cloud provider has implemented a security incident detection and response solution. Some cloud providers also generate dedicated logging data about security-relevant activities within the separate customer tenants of their SaaS application (e.g. changes to configuration settings, permission assignments or user accounts within the tenants) and make them available to cloud customers. These so-called audit logs are continuously transferred to a SIEM (Security Information and Event Management) for further analysis, attack detection and alarm handling.

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

ID: 3.01-37/8.0

Req 55	The system clock must be synchronized to an accurate reference time (Time Standard).

A time reference source must be used which provides a time signal based on the Coordinated Universal Time ("UTC" = "**U**niversal **T**ime **C**oordinated").

Please Note: The UTC-synchronized system time may be transformed to local time using a corresponding timezone configuration setup for any output of time information, as long as this timezone adjustment is fully accountable.

Systems belonging to the same security domain must synchronize to one and the same time reference source.

Motivation: Reference time synchronization may be a technical prerequisite for many time-dependent mechanisms, for example: Validation of Certificates; Authentication. It is also much-needed to generate exact timestamps for logged events, since without the often required time-related correlation in case of a Security Incident or during a Problem Analysis cannot be achieved.

Implementation example: some valid time reference sources:

- trustworthy NTP ("**N**etwork**T**ime**P**rotocol") Server on the IP network
- DCF77 radio signal received via a physically connected receiver
- GPS radio signal received via a physically connected receiver

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

ID: 3.01-32/8.0

Req 56	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.

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.
  (*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.*)
- After 90 days, stored logging data must be deleted immediately.

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 (PSM) / Data Privacy Advisor (DPA) 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 e-mail 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: 3.01-34/8.0

Req 57	Security-relevant logging data must be forwarded to a separate log server immediately after it has been generated.

Logging data must be forwarded to a separate log server immediately after it has been generated. Standardized proto-

cols such as Syslog, SNMPv3 should be preferred.

Motivation: If logging data is only stored locally, it can be manipulated by an attacker who succeeds in compromising the system in order to conceal his attack and any manipulation he has performed on the system. This is the reason why the forwarding must be done immediately after the event occurred.

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

ID: 3.01-35/8.0

Req 58	For security-relevant logging data that is forwarded to the separate log server, compliance with the applicable retention and deletion periods must be ensured.

The following basic rules must be taken into account:
- security-related logging data must be retained for a period of 90 days on the separate log server.
- after 90 days, stored logging data must be deleted immediately on the separate log server.

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 (PSM) / Data Privacy Advisor (DSB) or are specified by them.

Log server under the responsibility of a third party
If the selected separate log server is not within the same operational responsibility as the source system of the loggin data, it must be ensured that the responsible operator of the log server is aware of the valid parameters for the logging data to be received and that they are adhered to in accordance with the regulations mentioned here.

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 forwarded security-relevant logging data from an exemplary telecommunications system are implemented on the separate log server:
- 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 e-mail 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: 3.01-36/8.0

Req 59	The workstation must be configured, controlled through central guide lines and the observance of those must be monitored.

Motivation: By using central guide lines multiple workstations affiliated to the same domain can be configured equally. Any violation of these guide lines through attackers or rather malware can be detected by appropriate logging mechanisms to take countermeasures.

Implementation example: Several program or security settings can be applied by using group policies or group policy objects (GPOs) similar to a registry database. Therfore group policies must be configured applicably to every operating system or use case. An overview of any available GPOs for Windows systems can be found at https://www.microsoft.com/en-us/download/details.aspx?id=25250 or at http://gpsearch.azurewebsites.net/.

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

ID: 3.19-59/6.0