Security requirement

Windows Servers

Deutsche Telekom Group

Version	8.0
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
	3.15	Security requirement
Version	**State**	**Status**
8.0	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
Windows Servers

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

Table of Contents

1.	Introduction	4
2.	System hardening	5
2.1.	Installation	5
2.2.	Configuration	8
2.3.	Network	11
3.	System update	15
4.	Protecting availability and integrity	17
5.	Session Protection	20
6.	Protecting availability and integrity	22
7.	Authentication and authorization	25
8.	Authentication parameter password	33
9.	Logging	38
10.	Technical Baseline Security for IT/NT Systems	42
10.1.	Logging	42
11.	Operating Systems	44
11.1.	System Hardening	44
11.2.	System Update	44

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.

2. System hardening

2.1. Installation

Req 1	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 2	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 3	The Windows Server Core installation must be used.

Windows Server Core supports various server roles that can be viewed at following link. If technically possible, Windows Server should be installed with the listed server roles as a Windows Server Core:

- https://learn.microsoft.com/en-us/windows-server/administration/server-core/what-is-server-core

Motivation: Windows Server Core is an installation option in Windows Server, which significantly reduces attack surface, since it installs no graphic user interface and only the most urgently required system files and services.

Implementation example: When installing Windows Server using the Installation Wizard, you can choose between the installation options "Server Core" and "Server with Desktop Experience".

Server Core does not have the default graphical user interface (Desktop Experience) installed. In the Server Core variant, the server is managed via the command line using PowerShell, the Server Configuration Tool (SConfig) or using remote methods.

The installation wizard lists the following installation options. In this list, editions without a desktop experience represent the Server Core installation options:

- Windows Server Standard
- Windows Server Standard with Desktop Experience
- Windows Server Datacenter
- Windows Server Datacenter with Desktop Experience

For Server Core, the options must be selected without the addition of "with Desktop Experience"

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

ID: 3.15-3/8.0

Req 4	The Windows Server operating system must be installed with the minimum of components, server roles and features required for operation.

With regard to Windows Server this refers to the functions and roles which can be managed through the server manager. Server roles provide basic services, and features expand the server installation to include additional functions. In many cases, server roles depend on features or optionally supplement them with additional attributes.

More information on the individual server roles:

- Roles, Role Services, and Features included in Windows Server & Server Core
- Roles, Role Services, and Features not in Windows Server Core

Motivation: Every Windows Server component offers an attack surface – components that are not used and therefore not installed minimize this attack surface. This increases security because only the applications that are actually needed are installed on the server.

Implementation example: See the descriptions in the Microsoft article "Install or Uninstall Roles, Role Services, or Features":

- Install roles, role services, and features by using the add Roles and Features Wizard
- Install roles, role services, and features by using Windows PowerShell cmdlets
- Remove roles, role services, and features by using the remove Roles and Features Wizard
- Remove roles, role services, and features by using Windows PowerShell cmdlets

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: 3.15-4/8.0

2.2. Configuration

Req 5	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 6	Features that are not required in the software and hardware used must be deactivated.

During the initial installation of software, features may have been activated by default that are not necessary for the operation and functionality of the specific system. Features are usually an integral part of the software that cannot be deleted or uninstalled individually.

Such features must be disabled immediately after the initial installation through the software's configuration settings, so that they remain permanently disabled even after the system is rebooted.

Even before delivery or during initial commissioning, features may have been activated by default in the hardware that are not required for the purpose of the specific system. Such functions, for example unnecessary interfaces, must also be permanently deactivated immediately after initial commissioning.

Motivation: A system's hardware or software often contains enabled features that are not being used. Such features can be an unnecessary target for manipulation. Furthermore, there is a potential that unauthorized access to areas or data of the system can be created.

Implementation example: [Example 1]
Deactivation of debugging functions in the software that are used in the event of fault analysis, but do not have to be active during normal operation.

[Example 2]
Disabling unused network interfaces of a server.

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: 3.01-4/8.0

Req 7	The privileges for processes, services and applications must be reduced to the minimum required for the tasks they have to perform.

Privileges to processes and services must be restricted to a level in which they have only access to system ressources really needed. Suitable access restrictions must also be assigned for files that are components of the operating system or of applications or that are generated by the same (e.g. configuration and logging files).

The execution of applications and parts of them must also take place with privileges that are as low as possible. Applications should not be executed with administrator or system rights. In particular, mechanisms have to be avoided that allow processes to gain increased privileges during run-time.

Motivation: If the privileges granted to a process on a system are too broad, it could be possible to access data and parts of other services/applications for which viewing or the use is not permitted. This would give the opportunity to disclose or modify confidential data and to manipulate system files. Applications with rights that are too broad can be used by a user to extend the own authorizations and thus to gain access to files and system components to which the user would not have had access with the authorizations under normal circumstances.

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

ID: 3.37-6/8.0

Req 8	The automatic launch of applications on removable media must be deactivated.

Removable media such as CD-, DVD-, USB-Sticks or USB-Storage drives shall not automatically start any applications they contain.

Motivation: Automatic application launch could inadvertently launch malware.

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

ID: 3.37-9/8.0

Req 9	The Windows server system must be configured in accordance with the Baseline Server Hardening by Microsoft or "Security Best Practices".

General security instructions from Microsoft can be found at https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-security-configuration-framework/windows-security-baselines & https://techcommunity.microsoft.com/t5/microsoft-security-baselines/bg-p/Microsoft-Security-Baselines

The Security Benchmarks of the Centers for Internet Security (CIS) can also be used: https://www.cisecurity.org/benchmark/microsoft_windows_server/

Motivation: The application of Baseline Server Hardening by the manufacturer as well as of "security best practices" to reduce vulnerabilities by restrictive configuration settings.

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: 3.15-9/8.0

Req 10	Administrative templates (ADMX) must be used to set security settings

ADMX files are XML-based files that provide registry-based settings to the Group Policy Editor. They enable to choose the particular Group Policy settings to implement.

Motivation: Manual configurations of Windows Server via GUI, etc. can lead to errors, especially in larger system networks. The configuration via administrative templates to be imported has the advantage that the identical configuration is rolled out across all servers.

In addition, configuration via ADMX templates enables a high level of automation, which includes automated testing & auditing using CIS-benchmarks, Taste-OS, etc.

Implementation example: Windows Server 2012R2: https://www.microsoft.com/de-de/download/details.aspx?id=41193
Windows Server 2016 (includes 2012R2): https://www.microsoft.com/de-DE/download/details.aspx?id=53430
Windows Server 2019 (includes 2012R2 & 2016): https://www.microsoft.com/en-us/download/details.aspx?id=57576
Windows Server 2022 (includes 2012R2, 2016 & 2019): https://www.microsoft.com/en-us/download/details.aspx?id=104003

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

ID: 3.15-10/8.0

Req 11	Backup servers and Internet-exposed servers cannot be members of an Active Directory domain.

If the Active Directory is taken over, the attackers will attempt to encrypt backups accessible via the Active Directory (ransomware attacks). In addition, previously copied or extracted data from backups can be used to publish information for the purpose of blackmail (double extortion). Therefore, backup servers must be operated independently of an Active Directory domain.

Especially for Active Directory Domain Controller backup servers at domain and forest level, the backup servers must be operated independently of domain membership.

Motivation: Active directories and domain controllers are particularly the focus of attacks. An attempt is made to take over the entire Active Directory. If Windows servers exposed on the Internet are members of an Active Directory domain, they are a possible entry point for penetrating the Active Directory.

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: 3.15-11/8.0

Req 12	NTLM may not be used as an authentication protocol.

NTLM does not provide mechanisms to encrypt the authentication session, which compromises the confidentiality of the data being transmitted. Because of this weakness, it is necessary to use secure authentication protocols such as Kerberos, which offers stronger encryption and authentication methods.

Motivation: NTLM was replaced by Kerberos as an authentication protocol in 2000. Despite its widespread use, NTLM shows significant security vulnerabilities that make it vulnerable to cyberattacks. The vulnerability of NTLM results primarily from its design, which is based on challenge-response mechanisms, without ensuring sufficient verification of the communication partners. Attackers can use various techniques such as pass-the-hash, man-in-the-middle attacks, or brute force attacks to exploit NTLM authentication processes.

An example of cyberattacks on NTLM is the PetitPotam attack, which exploits vulnerabilities in the NTLM protocol to gain control of domain controllers on Windows networks. By exploiting this vulnerability in combination with AD-CS relay attacks, attackers can cause complete domain compromise. Microsoft released patches to fix PetitPotam, however, challenges remain as the patches don't solve all the issues and new authentication enforcement methods have emerged, such as DFSCoerce, which continue to pose security risks.

Implementation example: In the path "Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Security Options" configure the policies as follows:

Setting	Value
Network security: Restrict NTLM: Incoming NTLM traffic	Deny all accounts
Network security: Restrict NTLM: NTLM authentication in this domain	Deny all
Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers	Deny all

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

ID: 3.15-12/8.0

2.3. Network

Req 13	The administration of the operating system must be done via a network interface which is independent from the production network.

Administrative access to a server must not be done via an interface which provides productive services. Access must be limited to legitimate systems. The administration of applications can also be done using this network interface.

The restriction can be done with, e.g., filter mechanisms, local access lists or a packet filter. This limitation has to be done as restrictive as possible, i.e., limit to single IP addresses or at least small IP ranges.

Motivation: In the event of a successful attack, an attacker may gain access to confidential information or even to the entire system. By restricting the accessibility to legitimate systems, the group of potential attackers can be reduced, and thus also the likeliness of a successful attack. Furthermore, systems must be manageable even in the case the customer network is down.

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

ID: 3.37-10/8.0

Req 14	Administrative services and accesses must be bound to only those interfaces that have been set up to administer.

The administrative services (interactive by persons or machine-to-machine, e.g., SSH, HTTPS, RDP) must be bound to the appropriate interface(s). Due to the separation of management traffic from user traffic, this is the IP address in the management network. If the system - or parts of it - is managed by more than one interface, the management services have to be bound to the lowest possible number.

Depending on the respective service, the access to these services must be restricted to a few, trustworthy, necessary target or source addresses.

Motivation: This ensures that it can be clearly foreseen under which address these management services are reachable. In addition, a unique address is important for implementing filters and firewall rules that restrict access to these management services to legitimate addresses.

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

ID: 3.37-11/8.0

Req 15	Network based access used for operating system administration must have integrity protection, be encrypted and securely authenticated.

Access is only permitted by using secure protocols (e.g., SSHv2, HTTPS, SNMPv3). The administrator must ensure that any network connection between his workstation or a management system and the operating system to be administrated is securely authenticated, encrypted and protected against tampering.

Motivation: If the administrator transmits changes to the configuration settings via unencrypted or unsecure connections, there is a risk that unauthorized parties gather system information (configuration settings, access IDs, etc.) to exploit security vulnerabilities.

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

ID: 3.37-12/8.0

Req 16	The accessibility of activated services must be restricted.

In principle, a service provided must be completely deactivated on all interfaces of the system through which accessibility of the service is not required for the proper operation of the system. The deactivation is primarily to be implemented by a corresponding configuration of the service or operating system. In cases where the available configuration options do not allow deactivation on individual interfaces, a local filter ("Host Firewall") may instead be used on the system to block access to the service via unnecessary interfaces.

The accessibility of a service via the required interfaces must also be restricted to legitimate communication partners. The restriction must be implemented by a corresponding configuration of the service or operating system or by means of a local filter ("Host Firewall"). Alternatively, this task may be outsourced to a network-side filter element, provided

that the system is located in a separate network segment and communication with this segment is only possible via the network-side filter element.

Motivation: By deactivating services on interfaces through which accessibility is not necessary, as well as by restricting possible communication partners, the attack surface offered by a system can be greatly reduced.

Implementation example: An SNMP service used to monitor a system is enabled exclusively on the dedicated management network interface of the system. A firewall also regulates that only the legitimate monitoring system of the infrastructure environment can reach this service.

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: 3.01-6/8.0

Req 17	If services cannot be bound to the minimal required interfaces by configuration, a local (packet) filter must regulate the accessibility of the service.

Sometimes, software cannot be bound to dedicated interfaces. A local packet filter or TCP-wrapper can ensure this.

Motivation: Packet filters offer effective protection in order to prevent misuse services from other networks.

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: 3.37-3/8.0

Req 18	The Windows Server Firewall must be configured to filter out ports that are not required provided there is no alternative firewall for the system.

The Windows Firewall filters TCP/IP data traffic on the basis of various TCP/IP attributes (e.g., TCP port, source and destination address), thus preventing unwanted data traffic. The Windows Server Firewall does not need to be used if the Windows Server is operated in an Internet Protocol (IP) subnet that is protected by a firewall anyway or if a third-party manufacturer's software firewall is used which applies the same filter rules.

Motivation: A component that is jointly used by the subnet servers can filter TCP/IP data traffic. This is possible, for example, if multiple file and print servers of an Internet Protocol (IP) subnet are united in a network which is available for the same workstation system services. It is also permissible for the Windows Firewall to filter TCP/UDP ports and an additional component to filter subnets that are permitted to set up connections.

The use of filter rules effectively prevents unwanted data traffic and thus makes it more difficult for any malicious software to spread in an uncontrolled manner.

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: 3.15-18/8.0

Req 19	The IPv4 and IPv6 addresses of all interfaces of a server must be configured statically.

IP addresses providing services must not be changed on external influence, even in the case of an enforced reboot. An automatic assignment of IP addresses, e.g., using DHCPv4/v6 or IPv6 auto-configuration is permitted only in the case when deactivated after initial allocation or secured otherwise. IPv6 router advertisements must be ignored.

It is recommended to form the host part of the IPv6 addresses randomly. Due to the very large address space of IPv6, this way it is very time-consuming for an attacker to use scans to discover systems.

Motivation: An attacker on the same network segment can send DHCP / RA packets to redirect network traffic to its own server (Man-In-The-Middle) and so listen to all communications.

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

ID: 3.37-24/8.0

Req 20	Kernel based network functions not needed for the operation as a server must be deactivated.

If IPv6 is not required, it must be deactivated. Routing functions are not needed on a server; consequently the routing function must be disabled. Additionally the answering routine for broadcast ICMP packages must be disabled. Usually this and other network features should be configured correctly out-of-the-box.

Motivation: The routing functions enable misuse scenarios, meaning that an attack can route malicious packets through the server to connected networks.

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

ID: 3.37-7/8.0

Req 21	The TCP/IP protocol must be installed and used to the exclusion of all others.

Administrators should uninstall or deactivate alternative network protocols.

Motivation: Alternative network protocols are less common and are usually not developed further. Some of them also represent a risk, e.g., NetBIOS facilitates the search for potential vulnerabilities and can be used for attacks. Also, a NetBIOS that is activated over TCP/IP, in particular in distributed environments, can lead to a marked reduction in response times. TCP/IP is the standard network protocol of the Windows Server operating system. Web applications do not require the NetBIOS enhancement of TCP/IP.

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

ID: 3.15-21/8.0

3. System update

Req 22	The operating system must have an Endpoint Detection and Response (EDR) solution.

An up-to-date Endpoint Detection and Response (EDR) solution must be used on the operating system. An EDR solution collects security-relevant activity data from processes and evaluates them centrally. Alarms can be generated from malicious behavior of processes or by a specific signature (like classic virus scanners). In addition, it is possible to react directly to suspicious program behavior via a central console. For example, the operating system or rather the client can be isolated or the malicious process terminated. Furthermore, an overview of the vulnerabilities of all operatings systems 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.

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

ID: 3.37-18/8.0

Req 23	Known vulnerabilities in software and hardware of a system must be fixed or protected against misuse.

Prior to installation of a software or hardware component, users must check whether any vulnerability has been discovered and published for the version they are installing. Any component that proves to have a vulnerability must not be installed or used. Excepted from this rule are components for which the vendor has already provided a measure to remedy the vulnerability, e.g. a patch, update or workaround. In this case, the additional measure must be implemented on the system.

Hint: For the vulnerability management, a continuous process during the complete life cycle of the system is needed to fix upcoming vulnerabilities promptly.

Motivation: Publication of vulnerabilities increases the risk of successful exploitation by an attacker. The likelihood raises because of publication of detailed information and tools that help to exploit the vulnerability.

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

ID: 3.37-16/8.0

Req 24	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 25	A Windows server operating system must be migrated before the end of the Extended Support on an operating system in Mainstream Support.

Motivation: Windows Server operating systems contain design errors, critical errors and security gaps. Such gaps in older Windows Server operating systems are known and can be used with the intention of planting viruses or similar malicious software into the server or to misuse the server.

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: 3.15-25/8.0

4. Protecting availability and integrity

Req 26	If the system is not located in a room with at least protection class "high" (PC3), the BIOS and, if available, other options for local management must be secured against unauthorized access.

The Protection Classes (PC) are defined in the Annex 1, "Physical Security of Buildings", of the Group Policy on "Physical Security". Typically, Datacenters are compliant to the requirements of PC3.

Servers operated in public or customer areas must be especially protected against unauthorized access and changes: The BIOS settings must be protected against export and tampering. Are further access options to the system configuration possible, e.g. by Intel AMT, iLO, LOM, and others, these must be protected as well. In case passwords are used, these must be exclusive to the individual server and must not allow conclusions to be drawn about a distinguishing feature of the server.

The BIOS must be configured in such a way that only the designated operating system can be started with it from the designated partition.

Motivation: Motivation: Changing BIOS settings can facilitate attacks. Since, for example, local rooms with technical installations seldom offer access protection to the servers, attackers could change the startup sequence of data storage media when the server is started in the BIOS without the password protection described. This would make it possible to start an alternative operating system which circumvents the security mechanisms of the implemented 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

ID: 3.37-20/8.0

Req 27	If the system is not located in a room with at least protection class "high" (PC3), used data storages must be fully encrypted.

The Protection Classes (PC) are defined in the Annex 1, "Physical Security of Buildings", of the Group Policy on "Physical Security". Typically, Datacenters are compliant to the requirements of PC3.

Data storages are all disks and flash memory in the systems.

Motivation: Access to devices which are operated outside of data centers with protected access is relatively easy. Physical data storage media can be easily stolen as a result.

Implementation example: On Windows Server 2008, the administrator can activate drive encryption using BitLocker and key storage on a TPM.

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

ID: 3.37-21/8.0

Req 28	Outputs and messages must not disclose information on internal structures of the system.

Information about the internal structures of a system, including the components used there, and corresponding implementation details are generally considered to be in need of protection.

In general, this concerns information on

- Product names and product identifiers of implemented system components

- Operating systems, middleware, backend software, software libraries and internal applications as well as their software versions
- installed service packs, patches, hotfixes
- Serial numbers of components as well as stored product licenses
- Database Structures

Typical examples of outputs and messages in which disclosure of such system information can potentially occur:

- Login windows and dialogs
- Error messages
- Status messages
- Banners of active network services
- System logs and log files
- Debug logs, stack traces

As far as it is technically feasible without impairing the function and operation of the system, the output of affected system information must always be deactivated.

Access to affected system information must only be possible for authorized users of the system. As a rule, this circle of authorized users is to be limited to administrators and operators of the system. Access for authorized monitoring and inventory systems within the operating environment is also permitted.

A permissible exception to these restrictions exists for specific individual system information, the disclosure of which is technically mandatory for the intended function of the system in conjunction with third-party systems; For example, the presentation of supported protocols and their versions during the initial parameter negotiation in session setups between a client and a server.

Motivation: Information about the internal structures of a system can be used by an attacker to prepare attacks on the system extremely effective. For example, an attacker can derive any known vulnerabilities of a product from the software version in order to exploit them specifically during the attack on the system.

Implementation example: [Example 1]
Deactivation of the display of the product name and the installed version of a Web server in its delivered error web pages.

[Example 2]
Removal of the product name and the corresponding version string from the login banner of a deployed SSH server.

For this requirement the following threats are relevant:

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

ID: 3.01-9/8.0

Req 29	All partitions must be operated with the New Technology File System (NTFS).

For compatibility reasons, Windows Server offers the option to use alternative file systems. Administrators shall only use the NTFS. Even a data partition assigned to a Windows Failover Cluster in the SAN shall only be operated with NTFS by the administrators. It can ReFS (Resilient File System) can be used if this reflects the semantics of NTFS.

Motivation: NTFS is the only system to offer security settings such as authorizations and monitoring settings.

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: 3.15-29/8.0

Req 30	Data with need of protection must be protected against unauthorized viewing and manipulation during transmission and storage.

Adequate security measures for transmission and storage of data worth of protection must be implemented. Authentication data such as passwords, PINs, certificates and krytographic keys must be protected against unauthorized viewing and manipulation. This applies equally to storage and transmission. Furthermore must be guaranteed that confidential data will not be unprotected during temporary storage (e.g. in a temp-folder).

Files of a system that are needed for the functionality must also be protected against manipulation. This is necessary because system's integrity can be damaged when unauthorized changes to this kind of files is possible. An example is the use of cryptographic methods (signatures, hashes) to validate if, e.g., firmware images, patches, drivers or kernel modules are free of manipulations.

Write access to executable files that are executed with system priviliges must be restricted to users with system priviliges. Other user groups must not be able to modify these files. This is highlighted for start/stop/log rotation scripts.

For transmission of data with a need of protection network protocols that are insecure due to sufficient security measures shall not be used. Examples are: SSLv3, SSHv1, FTP, Telnet, SNMPv1 and 2c. In case of these protocols a newer version or a more secure alternative must be used.

Motivation: If data with a need of protection is not secured, an attacker could record or manipulate the data during transmission over a network. An example is the recording of user names and passwords during system administration with the telnet clear-text protocol. Storing data on a system without adequate protection may mean that unauthorized users can copy or modify it. One example is passwords stored without proper encryption on a system.

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

ID: 3.37-19/8.0

5. Session Protection

Req 31	Sessions must be protected against unauthorized takeover ("session hijacking").

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

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

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

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

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

ID: 3.01-16/8.0

Req 32	The system must allow users to log out of their current session.

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

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

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

ID: 3.01-17/8.0

Req 33	Sessions must be automatically terminated after a period of inactivity adapted to the intended use.

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

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

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

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

ID: 3.01-18/8.0

6. Protecting availability and integrity

Req 34	The system must be robust against overload situations and must act in a predictable way.

A system must provide security measures to deal with overload situations. In particular, partial or complete impairment of system availability must be avoided. Potential protective measures include:

- Restricting of available RAM per process / application
- Restricting CPU resources per process
- Prioritizing processes

If an overload situation cannot be prevented, the system must act in a predictable way. In such case it must be ensured that the system cannot reach an undefined and thus potentially insecure state. In an extreme case this means that a controlled system shutdown is preferable to uncontrolled failure of the security functions and thus loss of system protection.

Note: A system cannot defend itself against attacks with high data volume, also named distributed denial of service attacks. To defend this kind of attacks an (external) network based solution is necessary.

Motivation: Attackers try to force an overload situation on a system with denial of service attacks. If such an attack is successful, the availability can be compromised and integrity of system can be influenced.

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

ID: 3.37-22/8.0

Req 35	The TCP/IP stack must be implemented and configured in accordance with current knowledge to prevent attacks against the system and its network connections.

Typical attacks against the TCP / IP stack must not jeopardize the stability and integrity of the system. This includes attacks such as SYN attacks, hijacking of TCP connections, Ping of Death. Corresponding kernel features and parameters have to be set appropriately for the intended system use, as certainly respected for the operating system parameters. This also includes dropping of IP packets with unnecessary options or extension headers, e.g., source routing. Normally, no IP options or extension headers are needed.

In general, the system must be robust against incorrect and unexpected data packets on the network interface. The following typical implementation mistakes must not be done:

- No validation on the lengths of transferred data
- Incorrect assumptions about data formats
- No validation that received data complies with the specification
- Insufficient handling of protocol errors in received data
- Insufficient restriction on recursion when parsing complex data formats

Motivation: Avoid potential denial of service attacks against the TCP / IP stack by exploiting vulnerabilities or forcing a resource-intensive processing of IP packets.

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

ID: 3.37-25/8.0

Req 36	Growing (dynamic) content must not influence system functions.

Growing log data and uploads must not influence system functions.

Motivation: A filled up filesystem could stop the system from operations.

Implementation example: Usage of dedicated filesystems, separated from main system functions, or quotas, or at least a file system monitoring.

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

ID: 3.37-23/8.0

Req 37	Systems must not process IP packets whose source address is not reachable via the incoming interface.

The TCP / IP stack default behaviour is that IP packets can be accepted on an interface that is not the destination of the IP packet. For that it is only necessary that a route to the sender's source address exists. In certain cases (cluster, load balancers) which is an intentional effect, but not the default.

It is necessary to ensure there are no unneeded default routes which is typically the case for internal systems.

Motivation: In such a case the IP packet comes from an untrusted source (spoofed address) or a routing error exists in the network. In both cases the packet has to be dropped.

Implementation example: Use of "Reverse Path Filter" (RPF) which provides this feature.

Microsoft's Version of "Reverse Path Filtering", named "weak host send behavior" and "weak host receive behavior":
https://docs.microsoft.com/en-us/previous-versions/technet-magazine/cc137807(v=msdn.10)

"
FIGURE 4 COMMANDS TO CONFIGURE STRONG AND WEAK SEND AND RECEIVE BEHAVIOR
[...]
netsh interface ipv4 set interface [InterfaceNameOrIndex] weakhostsend=enabled
netsh interface ipv4 set interface [InterfaceNameOrIndex] weakhostreceive=enabled
netsh interface ipv6 set interface [InterfaceNameOrIndex] weakhostsend=enabled
netsh interface ipv6 set interface [InterfaceNameOrIndex] weakhostreceive=enabled
"

ID: 3.15-37/8.0

Req 38	The processing of ICMPv4 and ICMPv6 packets which are not required for operation must be disabled.

There are different types of ICMPv4 and ICMPv6 that are not used in most networks, but represent a risk. These types must be disabled or filtered and not be answered, sent or processed.

By contrast, the following ICMP types are permitted and may be used:

- Echo Request [Type 8 (v4), Type 128 (v6)]
- Echo Reply [Type 0 (v4), Type 129 (v6)]
- Destination Unreachable [Type 3 (v4), Type 1 (v6)]
- Time Exceeded [Type 11 (v4), Type 3 (v6)]
- Parameter Problem [Type 12 (v4), Type 4 (v6)]
- Packet Too Big [Type 2 (only v6)]
- Neighbor Solicitation [Type 135 (only v6)]
- Neighbor Advertisement [Type 136 (only v6)]

It is possible that other types will be necessary. This should be checked in each individual case. The ICMPv4 types "Timestamp Reply (14)," "Netmask Reply (18)," "Information Reply (16)" and "Redirect (5)" and ICMPv6 types "Router Solicitation" (133), "Router Advertisement" (134) und "Redirect" (137) must not be responded to or processed under any circumstances.

Motivation: ICMPv4 and v6 packets can be used by an attacker to request specific information which can be helpful for planning further attacks. In addition, it may be possible to influence the availability of systems.

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

ID: 3.37-27/8.0

7. Authentication and authorization

Req 39	An authentication method must be used that allows unambiguous identification of the user.

Users must be identified unambiguously by the system. This can typically be reached by using a unique user account per user.

Role-based accounts (functional account) can be used, in which individual authentication features (e.g. certificates, cryptographic keys) are neccessary.
Intended use of this is, for example, the administration account for a database or a web server, where usually several administrators need access to this account.

So-called group accounts where authentication is done by a secret known by all involved users, e.g., a password, must not be used.

So-called machine accounts will be used for identification and authorization from systems to each other or for applications on a system and can't be assigned to a single person. Such accounts must be assigned on a per system or per application basis. It has to be guaranteed that these accounts can't be misused. Possibilities to protect them are:

- Configuration of a password - fulfilling well-known security requirements - that is known by only the administrators with a need-to-know.
- Configuring the account that only a local use is possible and an interactive login isn't possible.
- Use of a technique for authentication of the specific user account with public and private key or certificates.
- Limiting the access for legitimized systems.

Additional solutions must be checked on their usability per individual case.

Motivation: Unambiguous user identification is mandatory to assign a user rights that are necessary to perform the required tasks on the system. This is the only way to adequately control access to data and services and to prevent misuse.

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

ID: 3.37-28/8.0

Req 40	Accounts must be protected against unauthorized use by at least one authentication attribute.

The various user, role and machine accounts on a system must be protected from misuse. To this end, an authentication attribute is typically used, which, when combined with the user name, enables unambiguous authentication and identification of the authorized user.

Authentication attributes include: Cryptographic key, Token, Password, PIN.

This means that authentication based on a parameter that can be spoofed is not permitted. Exceptions are attributes that cannot be faked or spoofed by an attacker. Two of the above options can be combined (2-factor authentication) to achieve a higher level of security.

In companies of Deutsche Telekom group where the MyCard or a comparable smartcard is available this solution should be preferred.

Motivation: Accounts that are not protected can be used by an attacker to gain unauthorized access to a system, the data and applications stored on it.

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: 3.37-29/8.0

Req 41	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 42	User accounts must be protected with at least one authentication attribute.

All user accounts in a system must be protected against unauthorized use.

For this purpose, the user account must be secured with an authentication attribute that enables the accessing user to be unambiguously authenticated. Common authentication attributes are e.g.:

- passwords, passphrases, PINs (factor KNOWLEDGE: "something that only the legitimate user knows")
- cryptographic keys, tokens, smart cards, OTP (factor OWNERSHIP: "something that only the legitimate user has")
- biometric features such as fingerprints or hand geometry (factor INHERENCE: "something that only the legitimate user is")

The authentication of users by means of an authentication attribute that can be faked or spoofed by an attacker (e.g. telephone numbers, IP addresses, VPN affiliation) is generally not permitted.

In companies of Deutsche Telekom group where the MyCard or a comparable smartcard is available this should be a preferred authentication attribute.

If the system and the application scenario support it, multiple independent authentication attributes should be combined if possible in order to achieve an additional increase in security (so-called MFA or Multi-Factor-Authentication).

Motivation: User accounts that are not protected by appropriate authentication attributes can be abused by an attacker to gain unauthorized access to a system and the data and applications stored 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: 3.01-20/8.0

Req 43	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 44	User accounts must ensure the unique identification of the user.

Users must be identified unambiguously by the system.

This can typically be reached by using a unique user account per user.

So-called group accounts, which are characterized by the fact that they are used jointly by several people, must not be used. This also applies without restriction to privileged user accounts. Most systems initially have only a single user account with administrative privileges after the basic installation. If the system is to be administered by several persons, each of these persons must use a personal, individual user account to which appropriate administrative authorizations or roles are assigned

A special feature are so named technical user accounts. These are used for the authentication and authorization of systems among themselves or of applications on a system and can therefore not be assigned to a specific person. Such user accounts must be assigned on a per system or per application basis. In this connection, it has to be ensured that these user accounts can't be misused.
Ways to prevent misuse of such user accounts by individuals include:

- Configuration of a password that meets the security requirements and is known to as few administrators as possible.
- Configuring the user account that only a local use is possible and a interactive login isn't possible.
- Use of a technique for authentication of the specific user account with public and private key or certificates.
- Limiting the access over the network to legitimate systems.

Additional solution must be checked on their usability per individual case.

Motivation: Unambiguous user identification is mandatory to assign a user permissions that are necessary to perform the required tasks on the system. This is the only way to adequately control access to system data and services and to prevent misuse. Furthermore, it makes it possible to log activities and actions on a system and to assign them to individual users.

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: 3.01-22/8.0

Req 45	There must be no privilege escalation method which allows gaining administrator/root privileges from a user account without a sufficiently strong, renewed authentication.

Privilege escalation methodes like su, sudo or runas include always the risk that more permissions are gained than needed. The number of exploits in such mechanisms shows the complexity and vulnerability of these solutions which therefore cannot be trusted.

The login into a privileged account must always be done with two authentication attributes.

Motivation: If an attacker compromises an account which has permission to a privilege escalation, it may be possible that the attacker gets access to wide parts of the system and stored data.

Implementation example: Disable insecure privilege escalation methods so that users are required to (re-)login directly into the account with the needed permissions.

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

ID: 3.37-31/8.0

Req 46	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 47	The privileges of accounts must be reduced to the minimum required for the tasks they have to perform.

Privileges of accounts must be restricted to a level in which a user can only access data and use functions that are needed by the user role. Suitable privileges must also be assigned to files that are components of the operating system or pplications.

Motivation: If the privileges granted to a user on a system are too broad, it could be possible to access data and applications for which viewing or the use is not permitted in this user role. This would give the opportunity to disclose or modify confidential data and to manipulate system files.

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

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

ID: 3.37-33/8.0

Req 48	Active Directory accounts must be used for technical administration of domain members.

Windows Server operating systems are usually members of an Active Directory domain that enables central administration of user accounts. The administrator can grant a user account in the Active Directory, for example, the authority to manage the server completely through membership in the local administrator group on the server that is to be managed.

Motivation: The use of accounts of the central directory services makes it possible to map an account to an individual and to implement security policies (e.g., smartcard login required) for these centrally managed accounts.

Implementation example: *An employee who wants to manage file releases in a network of several file and print servers on a regular basis is assigned an account in the Active Directory, which is a member of the local group of printer administrators on all servers in the network.*

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

ID: 3.15-48/8.0

Req 49	The number of local accounts needed for operation must be minimized.

All operating systems have high security requirements regarding local accounts. The administrator must ensure that all unused local accounts are deactivated.

For non-preventable (system) accounts where a login in certain cases must also be possible with a password (root, local administrator), these passwords must be selected for each machine and each account individually and with no apparent formation rule.

Motivation: Local accounts are additional points of attack which can be used by attackers or unauthorized individuals. This requires that only absolutely necessary local accounts required for operation exist in the operating system.

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

ID: 3.37-34/8.0

Req 50	Predefined user accounts that are not required must be deleted or at least disabled.

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

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

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

Motivation: User accounts that are predefined by default in a product are typically common knowledge and can be tar-

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

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

ID: 3.01-7/8.0

Req 51	In operating system installations that are members of an Active Directory domain, the first local administrator account of the operating system must be blocked.

All Windows Server operating systems come with an initial local administrator account, whose technical security ID (SID) is identical for all Windows Server operating systems. The owner shall deactivate this in order to effectively prevent any use of the account.

Note: If the operation system can no longer be deactivated by means of a domain account, the administrator would have to interrupt server operation and start the operating system in safe mode. This local account is automatically reactivated in this process.

Motivation: The initial local administrator account of a Windows Server represents a vulnerability since it has the same technical security ID (SID) in all systems. It is not automatically blocked if an attacker attempts to guess the password through recurrent login attempts.

Implementation example: *The administrator can deactivate the local administrator account by means of a group policy, for example.*

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: 3.15-51/8.0

Req 52	The predefined local guest account must be blocked.

All Windows Server operating systems come with a predefined local and deactivated guest account. The administrator shall deactivate this account in order to effectively prevent any use of the account.

Motivation: The predefined local guest account represents a vulnerability, since this identity can be used to obtain information from the operating system that would make it easier to launch attacks.

Implementation example: *The administrator can deactivate the predefined guest account by means of a group policy, for example.*

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: 3.15-52/8.0

Req 53	If passwords are used for authentication, precautions must be taken to minimize the risks involved by using passwords.

If passwords are used, the following criteria must be met:

(1) A system may only accept passwords that comply with the following complexity:

- Minimum length of 12 characters.
- Comprising at least three of these four categories: upper case letters, lower case letters, numbers, special characters

When a password is assigned, the system must ensure that it meets these minimal requirements.

(2) Passwords must not be shown in clear-text.

(3) Passwords must only be stored in such a way that

- only administrators can access files containing passwords and
- according to the state-of-technology, it would not be possible without disproportionate expense to convert it back from the stored value (usage of as safe recognized hash algorithms, multiple hashing, e.g. bcrypt, "salting" of hashes to protect against rainbow table attacks).

This also reduces the risk of dictionary and brute force attacks against the content of a password file.

Explicitly NOT PERMITTED is

- to store passwords in cleartext
- to store passwords in any format which can be backwards calculated, e. g. base64

(4) To prevent password guessing at login, various measures, or a combination thereof can be implemented, e.g.:

- Rising delay before next login is possible after a failed attempt ("tar pit", e.g. by doubling the waiting time or by using the Fibonacci numbers 2,3,5,8,13,21,34, ... as value for the next longer time period to wait).
- Locking the account after a predetermined number of failed attempts (typically 5). For an attacker not to selectively terminate accounts to make them unusable, this blocking should be limited by time (typically 1h). In addition, a process for unlocking is required.

If a central system is being used for authentication, this function can be forwarded or delegated to this system.

Motivation: Passwords with the required complexity provide a high robustness against attacks while at the same time acceptable usability. Trivial, too short or poorly stored passwords are vulnerable to brute force and dictionary attacks, and could be determined by an attacker. 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

ID: 3.37-35/8.0

8. Authentication parameter password

Req 54	If passwords are used as an authentication attribute, they must not be displayed in plain text during input.

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

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

Exceptions:

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

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

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

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

ID: 3.01-31/8.0

Req 55	If a password is used as an authentication attribute, users must be able to independently change the password anytime.

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

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

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

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-29/8.0

Req 56	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 57	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

ID: 3.01-28/8.0

Req 58	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 59	If passwords are used as an authentication attribute, those must be stored using a suitable and approved "Password Hashing" method to protect against offline-attacks like brute force or dictionary attacks.

This requirement relates to the storage of passwords in all types of user databases, as used in this system, in order to authenticate incoming access (local or remote) by users or other systems.

If an attacker obtains the copy of a user database of the system, he is able to bring it into a fully independent environment and utilize automatized dictionary or brute force attacks to determine contained passwords. Specialized tools in combination with high computing power allow for high cracking rates in a relatively short period of time, if protective measures are insufficient. Due to the independency from the source system, such an offline attack happens unnoticed.

The following countermeasure must be implemented, since this ensures best possible protection against offline attacks:

- passwords must be stored using a cryptographic one-way function (known as "Password Hashing") that is recognized as effective for this purpose and is classified as secure at the current state of the art.

Please Note: valid password hashing algorithms are described in Security Requirement Catalog "3.50 Cryptographic Algorithms and Security Protocols".

Explicitly NOT PERMISSIBLE is:

- to store passwords in cleartext
- to store passwords in any format which can be directly backcalculated
- to store passwords using reversible encryption

Please Note:
In this context, "directly backcalculatable formats" refers to those that simply encode the password, without involving a secret key in the transformation process. Since the password will no longer show up as original cleartext after it has been processed, those formats may easily be mistaken to provide confidentiality. Effectively, they do not offer any protection. The enconding is fixed and therefore an attacker can easily make use of it to compute the original cleartext password from the encoded string.
Examples for directly backcalculatable formats are: "base64", "rot13"
"Reversible" are all encryption methods which, using the appropriate key, enable encrypted content to be transformed back into the original content. Accordingly, with reversible encryption there is always the challenge of keeping the key secure and protecting it from unauthorized access. Reversibility is a required fundamental property in many areas of encryption applications, e.g. for transferring confidential messages, but it is counterproductive for storing passwords: a stored password must remain comparable by means of technical methods, but it must no longer be possible to convert it back into plain text in order to protect it as well as possible from unauthorized viewing.
Examples for reversible encryption are: "AES", "CHACHA20", "3DES", "RSA"

Motivation: Without protective measures, an attacker in possession of a user database copy is able to determine masses of contained passwords in short time by merely trying out character string combinations or making use of dic-

tionaries. Passwords stored in cleartext or any backcalculatable format are fully defenseless to an offline attack. 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
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-24/8.0

Req 60	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

9. Logging

Req 61	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 62	Security relevant Windows Server events must be logged with a precise timestamp and a unique system reference.

Windows Server must log the occurrence of security-relevant incidents. In order to enable the evaluation and classification of the occurring events, they must be logged together with a unique system designation (e.g. host name, IP or MAC address) and the exact time of occurrence.

When logging, the applicable valid legal, wage and company regulations must be observed. These regulations state, among other things, that the logging of events may only be carried out for a specific purpose. Logging for the purpose of work control of employees is not permitted.

Typical events are:

Event	Event data to be logged
Successful log on / use of all accounts (in the case of domain controllers only privileged accounts)	- Account - Source (IP-Address) 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

Events of Antivirussystem	- Update of signatures - Activating/deactivating of the antivirus-protection - 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))	- 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 - PowerShell Remoting - PSExec
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
Logoff of an account	- Session ID - Account

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: 3.15-62/8.0

Req 63	Security relevant logging data must be sent to a remote system directly after their creation.

Security relevant logging data must be forwarded to an external system in appropriate logging files as well as being stored locally. Standard protocols like Syslog, SNMPv3 must be preferred. The transfer should be secured, i.e. encrypted and authenticated, when data with need of protection is transmitted. To enable testing for consistency and completeness, sequence numbers are to be used and, if feasible, TCP instead of UDP.

Hint: The receiver of the data must be informed, how the data provided should be assessed.

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:
- Unnoticeable feasible attacks

ID: 3.37-42/8.0

Req 64	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 65	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 con-

nection 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

10. Technical Baseline Security for IT/NT Systems

10.1. Logging

Req 66	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

11. Operating Systems

11.1. System Hardening

Req 67	System kernel hardening measures must be enabled.

Implement kernel hardening techniques: These methods modify kernel configuration or behavior to reduce the attack surface and increase resistance to attacks. To do this, unnecessary kernel functions, modules, or services must be disabled or removed. Security features such as address space layout randomization (ASLR), stack shattering protection (SSP), or data execution prevention (DEP) must be active.

Motivation: In particular, measures such as ASLR protect the system from the execution of malicious code during so-called buffer overflows.

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

ID: 3.37-8/8.0

11.2. System Update

Req 68	If needed, active software licenses must be installed to ensure security updates.

Motivation: Some operating system vendors license their products and require the purchase of licenses. These allow access to (security) updates. Without update options, a system may not be operated.

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

ID: 3.37-17/8.0