Security requirement

Operating Systems

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
	3.37	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
Generic security requirements on operating systems for servers.

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

Table of Contents

1.	Introduction	4
2.	System Hardening	5
2.1.	Administrative Access	8
3.	System Update	10
4.	Protection of Data and Information	14
5.	Availability and Integrity	16
5.1.	TCP/IP Stack	16
6.	Authentication and Authorization	19
7.	Logging	26

1. Introduction

This 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 in units which do not participate in the PSA process. These requirements shall be taken into account from the very beginning, including during the planning and decision-making processes. When implementing these security requirements, the precedence of national, international and supranational law shall be observed.

2. System Hardening

Req 1	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 2	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 3	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 4	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 5	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 6	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 7	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 8	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

Req 9	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

2.1. Administrative Access

Req 10	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 11	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 12	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 13	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

3. System Update

Req 14	Software and hardware of the system must be covered by security vulnerability support from the supplier.

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

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

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

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

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

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

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

ID: 3.01-1/8.0

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

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

- Firmware and microcode for hardware components
- Operating systems
- Software Libraries
- Application Software

- Pre-integrated application solutions, such as software appliances or containers

as well as other software that may be used.

Trusted Sources

Trusted sources are generally considered to be:

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

Integrity Check

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

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

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

Extended integrity checking when pulling software from public registries

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

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

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

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

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

- Examination using an SCA (Software Composition Analysis) tool or dependency scanner (e.g. OWASP Dependency Check, Snyk)

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

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

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

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

ID: 3.01-2/8.0

Req 16	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 17	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

Req 18	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

4. Protection of Data and Information

Req 19	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

Req 20	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 21	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

5. Availability and Integrity

Req 22	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 23	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

5.1. TCP/IP Stack

Req 24	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 25	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 26	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 (clusters, load balancers) this 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.

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

ID: 3.37-26/8.0

Req 27	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

6. Authentication and Authorization

Req 28	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 29	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 30	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 31	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 32	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 33	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 34	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 35	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 addi-

tion, 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

Req 36	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 37	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 38	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

7. Logging

Req 39	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 40	Security relevant events must be logged with a precise timestamp and a unique system reference.

Systems must log the occurrence of security-relevant incidents. In order to allow that these events can be evaluated and classified, they must be logged together with a unique system reference (e.g., hostname, IP or MAC address) and the exact time the incident occurred ("Timestamp").

The Timestamp of a logged event must contain at least the following information:

- date of the event (Year, Month, Day)
- time of the event (Hours, Minutes, Seconds)
- Timezone, those information belongs to

When logging, the applicable legal and operational regulations must be observed. The latter also include agreements that have been made with the company's social partners. Following these regulations logging of events is only allowed for a defined use case. Logging of events for doing a work control of employees is not allowed.

Typical events that reasonable should be logged in many cases are:

Event	Event data to be logged
Incorrect login attempts	- User account - Source (IP adress) of remote access - Number of failed attempts
Logins, without logins granted by an authentication server, e.g., domain comtroller	- User account - Source (IP adress) of remote access - Used login method (RDP, SSH, VNC, ...)

Successful access to privileged accounts	- User account - Source (IP adress) of remote access
Administration of accounts, roles and groups	- Administrator account - Administered user account - Activity performed (add, delete, change, enable, disable)
Shell commands executed in userspace with and without privileged permissions	- User account - Command - If possible: Return value
Critical rise in system values such as disk space, CPU load	- Name of critical value exceeded - Timestamp (begin, end) - Measured data

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.

Logging of the console commands is necessary because attackers could gain access to a shell on a system via exploits. Shell commands used in this way would otherwise not be recorded.

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

ID: 3.37-40/8.0

Req 41	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 recon-

cile 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 42	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 43	For security-relevant logging data that is forwarded to the separate log server, compliance with the applicable retention and deletion periods must be ensured.

The following basic rules must be taken into account:

- security-related logging data must be retained for a period of 90 days on the separate log server.
- after 90 days, stored logging data must be deleted immediately on the separate log server.

Deviances

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

Log server under the responsibility of a third party

If the selected separate log server is not within the same operational responsibility as the source system of the loggin

data, it must be ensured that the responsible operator of the log server is aware of the valid parameters for the logging data to be received and that they are adhered to in accordance with the regulations mentioned here.

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

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

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

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

ID: 3.01-36/8.0

Req 44	Security-relevant logging data must be provided to a security incident detection and response solution in near-real-time.

An acceptable detection and response solution for the purposes of this requirement must meet the following criteria:

- There is a continuous analysis and correlation of the logging data provided by connected IT/NT systems.
- The identification of threatening activities and ongoing attacks takes place on the basis of a continuously updated detection pattern database
- Integrated alerting processes trigger the initiation of countermeasures in the event of a find.

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

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

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

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

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

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

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

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

ID: 3.01-37/8.0