Security Requirement

Network Signalling Node en

Deutsche Telekom Group

As of: 30.07.2014

Internal

Table of Contents

1. 3.01Technical Baseline Security for IT/NT Systems (V. 2.0) 4
1.1. Introduction 4
1.2. System hardening 4
1.3. System update 5
1.4. Protecting data and information 6
1.5. Protecting availability and integrity 7
1.6. Authentication and authorization 8
1.7. Protecting sessions 12
1.8. Authentication parameter password 13
1.9. Logging 14

2. 3.03Web Servers (V. 2.0) 16
2.1. Introduction 16
2.2. Platform requirements 16
2.3. Requirements on web server software 16
2.4. Installation requirements 16
2.5. Configuration requirements 17
2.6. HTTPS requirements 21
2.7. Logging 23

3. 3.04Secure Shell (SSH) (V. 2.0) 24
3.1. Introduction 24
3.2. Security requirements 24
3.2.1. General 24
3.2.2. SSH Options 24
3.2.3. Authentication and Accounts 25
3.2.4. Requirements on the SSH client 25

4. 3.21Unix Servers (V. 2.0) 26
4.1. Introduction 26
4.2. Unix systems 26
4.2.1. System hardening 26
4.2.2. System Update 32
4.2.3. Protection of Data and Information 32
4.2.4. Availability and Integrity 33
4.2.5. Authentication and Authorization 35
4.2.6. Authentication Parameter Password 39
4.2.7. Logging 41

5. 3.37Operating Systems (V. 2.0) 43
5.1. Introduction 43
5.2. System Hardening 43
5.3. System Update 46
5.4. Protection of Data and Information 47
5.5. Availability and Integrity 48
5.6. Authentication and Authorization 51

5.7. Authentication Parameter Password 55
5.8. Logging 56
6. 3.42Network Devices (V. 2.0) 59
6.1. Introduction 59
6.2. System hardening 59
6.3. System update 61
6.4. Protecting data and information 62
6.5. Protecting availability and integrity 64
6.6. Authentication and authorization 65
6.7. Protecting sessions 69
6.8. Authentication parameter password 70
6.9. Logging 71

1. 3.01Technical Baseline Security for IT/NT Systems (V. 2.0)

1.1. Introduction

This security requirement has been prepared based on the provisions of the Group Policy on IT/NT Security.

The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions of the Group Policy on IT/NT Security 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.

1.2. System hardening

Req 3.01-1	Unused services and protocols must be deactivated.

After installation of systems and software products typically local or remote reachable services and protocols are active, which are not necessarily needed for operation and functionality of the system. These include also services and protocols which may not be used in Deutsche Telekom Group networks on account of known security vulnerabilities, which can be used to violate confidentiality, availability or integrity of the system . Such services and protocols must be completely disabled on the system. Additionally it is important that a deactivation survives a system reboot.

This kind of system hardening must be done before the system is reachable from the network. Otherwise an attacker has the possibility to attack and maybe compromise the unsecured system.

Motivation: Services and protocols that are not required for system operation increase the potential attack surface and thus the risk of the system being compromised. This risk is further increased by the fact that a security inspection and an appropriate optimisation of the configuration for unused services and protocols will not be done.

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

Req 3.01-2	The reachability of services must be restricted.

Typically services that are enabled in the basic configuration are accessible over all interfaces of the system and can be reached from systems in connected networks. This availability is often not needed or meaningful for system functioning. For this reason, services should only be enabled on interfaces where their usage is required. On interfaces were services are active, the reachability must be limited to legitimate communication peers. This limitation must be realized on the system itself (without measures (e.g. firewall) at network side).

Motivation: Disabling services on interfaces which do not require system accessibility or by limiting the reachability can greatly reduce the potential vulnerabilities offered to an attacker. For example, access to a system via SSH from the Internet is not necessary. If this service could be accessed from the interface connected to the Internet, this would greatly increase the risk of attacks on the service.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.01-3	Unused software must not be installed or must be uninstalled.

During installation of a system often software components will be installed or parts of software will be activated which are not needed for the operation or functionality of the system. This includes also parts of a software, which will be installed as examples but typically not be used (e.g. default web pages, example databases, test data). Such components should not be installed or must be deleted after installation.

Motivation: Vulnerabilities in software of a system offer an attack window for attackers to infiltrate the system. Uninstalling components that are not required can therefore reduce the possibility of a successful compromise of the system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Disruption of availability

Req 3.01-4	Unused functions of the operated software and hardware must be deactivated.

During installation of software and hardware often functions will be activated that are not necessarily needed for operation or function of the system. Functions of software are currently inherent part which could not be deleted or deinstalled individually. Such functions must be deactivated in the configuration of the system permanently.

Beside the functions of the software also hardware functions are active which are not necessary for a system. Functions like unused interfaces must permanently deactivated. Permanent means that they must not be reactivated again after system reboot.

Motivation: The hardware or software of a system often contains functions which are not used and so will be a risk for system security. Such functions give an attacker the possibility to manipulate the system. Furthermore it is possible to get unauthorized access other areas or data of the system. An example is a debugging function in software which can be used for troubleshooting but must not be activated during normal operation. Or a hardware interface that will not be used and so is unsecured an allows possibly unauthorized access to the system.

For this requirement the following threats are relevant:

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

1.3. System update

Req 3.01-5	Software and hardware components that are no longer supported by vendor, producer or developer must not be used.

Only those operating system, middleware and application software and hardware components may be used on a system which are supported by the vendor, the producer, the developer or other contractual partner of Deutsche Telekom AG. Components that have reached end-of-life or end-of-support must not be used. Excluded are components that have a special support contract. This contract must guarantee the correction of vulnerabilities over components lifetime.

Motivation: Hardware and software components that have reached end of life or end of support represent a risk for a system. This means that a vendor does not supply remedial updates or patches for a component should errors or vulnerabilities occur. This means that vulnerabilities cannot be fixed when they occur and could be exploited to compromise the system or to impair its availability.

For this requirement the following threats are relevant:

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

Req 3.01-6	Known vulnerabilities in software and hardware of the system must be fixed or protected.

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. Furthermore it is a ongoing process during the complete life cycle of the system to fix upcoming vulnerabilities promptly. Time demands to correct vulnerabilities will be released by Telekom CERT in their advisories (https://cert.telekom.de).

Motivation: Publication of vulnerabilities increases the risk of successful exploitation by an attacker, especially since the published information usually includes details on how to exploit the vulnerability and tools that make exploitation possible. One example of this is when Web servers are compromised due to not patched vulnerabilities.

For this requirement the following threats are relevant:

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

1.4. Protecting data and information

Req 3.01-7	Data with need of protection must be secured against unauthorized viewing and manipulation during transmission and storage.

Adequate security measures for transmission and storage must be implemented of data with a need for protection that are classified as internal, confidential or strictly confidential. The chosen measure depends on the classification for the data and other factors such as the type of network used during transmission, the storage location for data, etc. Assistance for the classification of data gives the Intranet page http://informationwheel.telekom.de. Furthermore must be guaranteed that confidential and strictly confidential data will not be unprotected during temporary storage (e.g. in web cache, temporary folders).

All authentication data such as user names, passwords, PINs, etc. must be protected against unauthorized viewing and manipulation. This applies equally to permanent storage and transmission. Typical measures taken to protect authentication data or combinations of these are:

For storage:

- Client systems: encryption or obfuscation of authentication data, no persistent storage, limitation of access rights.
- Server systems: hashing of authentication data with PBKDF like scrypt or bcrypt or when this is not possible hashes with salt.

For transmission:

- Usage of cryptographically protected network protocols.
- Challenge response method (Disadvantage of this solution is that passwords needed in clear-text on server. If this solution is feasible depends on the individual threat scenario)

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 the system access this kind of files. An example is the use of checksum or cryptographic methods to validate if e.g. firmware images, patches, drivers or kernel modules are free of manipulations.

For transmission of data with a need of protection it is necessary to use network protocols with insufficient security measures. Examples for these insecure protocols that should not be longer used are: SSLv3, SSHv1, FTP, Telnet, SNMPv1 and 2c. In case of these protocols a newer version without vulnerabilities or a secure alternative must be used.

Motivation: If data with a need of protection will not be 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 when passwords can be read out when they stored in an inadequate secured way (e.g. usage of unsecure hashing algorithms like MD5 or SHA-1) or even without encryption on a system or manipulation of firmware to affected the system integrity.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

Req 3.01-8	Information with need of protection must not be contained in files, outputs or messages that are accessible by unauthorized users.

Information with need of protection must not be accessible in files, outputs or messages of the system by unauthorized users. This includes information relating to the operating system, used middleware or applications such as vendor, product name, product identifier, installed software versions, installed service packs, patches, hot fixes and serial numbers. Examples for system messages which must be free of sensitive data are:

- Comments in downloadable files
- Error and system messages
- Stack traces
- Network protocols
- Login windows and dialogs

Furthermore, details of implementation and information relating, e.g., to backend software/systems, function calls, SQL instructions or structure of database, must not be contained in error messages.
Excluded from this are displays and outputs that can be viewed and retrieved by authorized users who are logged in. In addition, an internal transfer of system internal information for error analysis is allowed in an adequate dimension. In this case the continuative regulations or guidelines (e.g. of data privacy) must be noticed.

Motivation: The information named above can be used by an attacker to prepare specific attacks on a system. In this way an attacker could, for example, use the precise software version to identify vulnerabilities in the product and, in a second step, exploit them.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

1.5. Protecting availability and integrity

Req 3.01-9	The system must be robust against overload situations.

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 application
- Restricting of maximum sessions for a Web application
- Defining the maximum size of a dataset
- Restricting CPU resources per process
- Prioritizing processes
- Limiting of amount or size of transactions of an user or from an IP address in a specific time range

Motivation: An attacker can effect systems availability through targeted exploitation of vulnerabilities in resources ad-

ministration. An attacker can, for example, send bulk queries to a Web server and, by doing so, compromise its availability if the maximum number of permissible Web sessions is not restricted.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-10	If an overload situation cannot be prevented, the system must act in a predictable way.

A system must be built in this way that it can react on a overload situation in a controlled way. However it is possible that a situation happens where the security measures are not longer sufficient.

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.

Motivation: With denial-of-service attacks can an attacker try to overload a system to effect its availability or integrity. Unpredictable acting of the system is then a risk for functionality and data and possibly also for other systems.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-11	The system must be robust against unexpected input.

During transmission of data to a system it is necessary to validate this before processing. This includes all data which are send to the system. Examples for this are user input, values in arrays and content in protocols. 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
- White listing or escaping for inputs outside the values margin

Motivation: An attacker can try to put a system in an unsecure state through targeted manipulation of transmitted data. The object of such an attack is to compromise the usability, availability or integrity of individual services or of the entire system. For instance a unclean memory handling can lead to a buffer overflow that allows an attacker to execute arbitrary code on the effected system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability

1.6. Authentication and authorization

Req 3.01-12	System functions with a need of protection must not be used or accessed without successful authentication and authorization.

The usage of a system functions or access of data classified as internal, confidential or strictly confidential must only be possible unambiguous user identification and successful authentication on basis of the user name and at least one authentication attribute. Excepted from this are functions for public use such as those for a Web server on the Internet, via which information is made available to the public. Examples for functions which require a prior authentication are network services (like SSH, SFTP, Web services), local access via a management console, local usage of operating systems and applications. The following examples are possibilities that could be used for authentication.

- Query user name and password
- Use of cryptographic keys and certificates (e.g. as Smartcard)

This requirement must also be applied to accounts that are only used for communication between systems (M2M).

Motivation: The authentication is necessary to doubtless identify a user because the allocated authorization, and therefore the access on data and services of the system depends on that.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-13	Accounts must be used that allow unambiguous identification of the user.

Users must be identified unambiguously by the system. This can typically be reached by using a unique account per user. So named group accounts, i.e. the use of one account for several persons, must not be used. On exception of this requirement are so named machine accounts. These will be used for authentication and authorization from system 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. In this connection, it has to be guaranteed that this account can't be misused. Possibilities to protect these accounts are:

- Configuring of a Password that fulfils the security requirements and is known by less than possible circle of administrators.
- Configuring the account that only a local use is possible and a interactive login isn't possible.
- Use of a technique for authentication of the specific account with public and private key or certificates.
- Limiting the access over the network for legitimised systems.

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

Motivation: Unambiguous user identification is a prerequisite for assigning a user the rights that he requires to perform his 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 use of services or resources
- Denial of executed activities

Req 3.01-14	Accounts must be protected against unauthorized use by at least one authentication attribute.

The various user 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 keys
- Token
- Passwords
- PINs

This means that authentication based on a parameter that can be spoofed (e.g. phone numbers, public IP addresses or VPN membership) is not permitted. Exceptions are attributes that cannot be faked or spoofed by an attacker. Several of the above options can be combined (dual-factor authentication) to achieve a higher level of security. Whether or not this is suitable and necessary depends on the protection needs of the individual system and its data and must be evaluated for individual cases.

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts that are not protected with a secret authentication attribute can be used 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 use of services or resources
- Denial of executed activities

Req 3.01-15	Accounts with extensive rights must be protected with two authentication attributes.

Accounts, for example used for administration, maintenance and troubleshooting, have extensive rights. Therefore a single protection (e.g. a password), as for normal user accounts with less rights, is not suitable. To get a higher protection level it is necessary to use more than one authentication attribute. For this a combination of an attribute that the user knows and an attribute that the user owns will be used often. This kind of authentication will be named as 2-factor authentication. Examples for 2-factor authentication are:

- Smartcard (e.g. MyCard) with PIN
- Private key with Passphrase
- Secure-ID Token with Password

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts with extensive rights as used for system administration have a higher risk for system's security. An attacker can get extensive rights by compromising such an account to get access to wide parts of the system and stored data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-16	Predefined accounts must be deleted or disabled.

Many systems have default accounts (e.g. guest, ctxsys), some of which are preconfigured with or without known passwords. These standard users must be deleted or disabled. Should this measure not be possible the accounts must be locked for remote login. In any case disabled or locked accounts must configured with a complex password (12 character and more, use of upper/lower case, numbers and special characters). This is necessary to prevent unauthorised use of such a account in case of misconfiguration.

Exceptions to this requirement to delete or disable accounts are accounts that are used only internal on the system involved and that are required for one or more applications on the system to function. Also for this accounts remote access or local login must be forbidden to prevent a abusive use by users of the system.

Motivation: Standard users are typically generally known and can be used by an attacker for targeted brute force and dictionary attacks. Standard user accounts represent a special risk if they do not use a password or only use a standard password that is generally known. Such standard user accounts can easily be exploited by an attacker in order to gain access to the system involved without being authorized to do so.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-17	Predefined authentication attributes must be deleted or disabled.

Normally, authentication attributes such as password or cryptographic keys will be preconfigured from producer, vendor or developer of a system. Such authentication attributes must be changed an own attribute not known by other parties.

Motivation: Authentication attributes like password or cryptographic keys preconfigured from third parties are not trustable. Such authentication attributes can be used to compromise systems or their data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-18	The authorizations for accounts and applications must be reduced to the minimum required for the tasks they have to perform.

Authorizations to a system must be restricted to a level in which a user can only access data and use functions that he needs in the course of his work. Suitable authorizations must also be assigned for access to files that are components of the operating system or of applications or that are generated by the same (e.g. configuration and logging files).

Alongside access to data, execution of applications and components must also take place with rights that are as low as possible. Applications should not be executed with administrator or system rights.

Motivation: If the rights granted to a user on a system are too broad, it could be possible for him to access data and applications which he is not permitted to view or use. This would give him 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 his own authorizations and thus to gain access to files and system components to which he would not have had access with his authorizations under normal circumstances.

For this requirement the following threats are relevant:

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

Req 3.01-19	The system must be connected to a central system for user administration.

Accounts and their rights will be administrated on central identity management systems (e.g. cIAM, WiW, ZAM) in Deutsche Telekom Group. For provisioning of these authorization information must the system provide a central interface (e.g. LDAPs for authorization, Kerberos for authentication, locking information for certificates) or decentralized mechanisms (e.g. public-key authentication) . A central solution for identity management must be preferred.

In areas where a central identity management system is not available a central system such as LDAP, TACACS+ or Radius server for the administration of accounts and their authentication and authorization must be used.

Motivation: Central administration of identity of accounts and their rights means that they only have to be maintained once instead of separately on each system. From the aspect of security, the advantage is that an account and its rights only known on a single central side. This information can be transmitted from a central side to systems (provisioning), central administrated (reconciliation) and central deleted (deprovisioning). This reduces the risk of accounts being forgotten during changing or deletion since they are configured on multiple systems. This could give a user wrong system rights or continued access to a system.

For this requirement the following threats are relevant:

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

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

1.7. Protecting sessions

Req 3.01-20	Sessions must be protected against high jacking.

A function must be implemented for all user sessions to prevent other users assuming or pursuing control of the sessions of an authorized user. Such protection can be achieved by implementing the following measures or a combination of these measures:

- Using an appropriate cryptographic algorithm
- On network level: Use of TCP protocol (with sequence number) and filter lists.
- On transport level: SSL/TLS
- On application level: Negotiation of a random, secret value between sender and recipient (e.g. session ID, sequence number, time stamp)

Motivation: An attacker who gains knowledge about a session which is not protected against assumption of control can continue to participate in such a session and thus gain unauthorized access to the system involved. One example is an attacker who records the session cookie for a Web application during non-encrypted communication. The attacker can then view and take control of the session of the original 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

Req 3.01-21	It must be possible that users can logoff their session.

The system must have a function that allows a signed in user to logout at any time.

Motivation: A user of a system must have the possibility to protect a session and therefore its data against unauthorized access. Therefor a logout function must be available that can be used to end a session. This prevents that the current session will stay open and can be reactivated and used by an unauthorized 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

Req 3.01-22	Session must be terminated after an adjusted period of inactivity for the purpose of use.

It is necessary that a session will be terminated automatically after a specified time of inactivity. For this reason a time-out must be set for sessions. The selected period depends on use and if applicable the physical environment. This means for example that a time-out for an application in an unsecure environment must be shorter (less minutes) than the time out for an application used of operational staff for monitoring tasks used in an protected area (60 minutes and longer).

Motivation: In case of an open or unused session exists the possibility that an unauthorized user can high-jack and use

this session to get access to the effected system and its data. Furthermore open session will allocate system resources.

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

1.8. Authentication parameter password

Req 3.01-23	If a password is used as an authentication attribute, it must have at least 8 characters and contain three of the following categories: upper cases, lower case, numbers and special characters.

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

- Minimum length of 8 characters.
- Comprising at least three of the following categories: upper/lower case letters, numbers and special characters

When a password is assigned, the system must ensure that the password meets these requirements. If a central system will be used for user authentication this function can be forwarded or delegated to this system.

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

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-24	If a password is used as an authentication attribute, users must be enable 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 will be used it is possible 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 use of services or resources
- Denial of executed activities

Req 3.01-25	If a password is used as an authentication attribute, a protection against brute force and dictionary attacks that hinder password guessing must be implemented.

Brute force and dictionary attacks aim to use automated guessing to ascertain passwords for user and machine accounts. Various measures or a combination of this measures can be taken to prevent this.

The most commonly used protection measures are:

- Increasing the delay (e.g. doubling wait times at each attempt) for each renewed password input following an incorrect entry ("tar pit").
- Blocking an account following a specified number of incorrect attempts (typically 5). However has to take in account that this solution needs a process for unlocking and an attacker can force this to deactivate accounts and make them unusable.
- Using CAPTCHA to prevent automated attempts (often used for Web applications).

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: Implementation of one or more of the above measures can prevent successful execution of brute force and dictionary attacks to ascertain passwords.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-26	If a password is used as an authentication attribute, they must be hidden when displayed on screen.

When a user enters a password or the password is otherwise displayed on screen, it must be made unreadable. Typically, the individual characters of the password are replaced by a character such as "*". Under certain circumstances it may be permissible for an individual character to be displayed briefly during input. Such a function is used, for example, on Smartphone's to make input easier. However, the entire password is never output to the display in plaintext.

Motivation: To prevent another person reading a password by chance or intentionally on screen during input, the password must be rendered unreadable during input or display.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

1.9. Logging

Req 3.01-27	Security relevant events must be logged with a precise timestamp and a unique system reference.

Systems must log the occurrence of security-relevant incidents. So that these events can be evaluated and classified, they must be logged together with a unique system reference (e.g., host name, IP or MAC address) and the exact time the incident occurred. Furthermore, legal and data privacy regulations (e.g. time of storage of logging data) must be proved and followed. The following table lists events that are relevant to security and the corresponding data that typically has to be logged by a system. Exceptions are systems for which no or only restricted logging applies. Examples of such systems are customer devices such as Smartphone's or IADs/Homegateways (e.g. Speedport).

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

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

Event	Event data to be logged

Incorrect login attempts	• Account, • No. of failed attempts, • Source (IP address) of remote access
System access with accounts with administrator rights	• Account, • Access timestamp, • Length of session, • Source (IP address) of remote access
Account administration	• Administrator account, • Administered account, • Activity performed (configure, delete, enable and disable)
Change of group membership for accounts	• Administrator account, • Administered account, • Activity performed (group added or removed)
Critical rise in system values such as disk space, CPU load over a longer period	• Value exceeded, • Value reached (Here suitable threshold values must be defined depending on the individual system.)

Logging of additional security-relevant events may be meaningful. This must be verified in individual cases and implemented accordingly where required.

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

Req 3.01-28	Security relevant logging data must be send to an external system direct after their creation.

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.

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:

- Denial of executed activities
- Unnoticeable feasible attacks

2. 3.03Web Servers (V. 2.0)

2.1. Introduction

This security requirement has been prepared based on the provisions of the Group Policy on IT/NT Security.
The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions of the Group Policy on IT/NT Security in units which do not participate in the PSA process. These requirements sh
all 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.

2.2. Platform requirements

2.3. Requirements on web server software

Req 3.01-5	Software and hardware components that are no longer supported by vendor, producer or developer must not be used.

Only those operating system, middleware and application software and hardware components may be used on a system which are supported by the vendor, the producer, the developer or other contractual partner of Deutsche Telekom AG. Components that have reached end-of-life or end-of-support must not be used. Excluded are components that have a special support contract. This contract must guarantee the correction of vulnerabilities over components lifetime.

Motivation: Hardware and software components that have reached end of life or end of support represent a risk for a system. This means that a vendor does not supply remedial updates or patches for a component should errors or vulnerabilities occur. This means that vulnerabilities cannot be fixed when they occur and could be exploited to compromise the system or to impair its availability.

For this requirement the following threats are relevant:

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

Req 3.01-6	Known vulnerabilities in software and hardware of the system must be fixed or protected.

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. Furthermore it is a ongoing process during the complete life cycle of the system to fix upcoming vulnerabilities promptly. Time demands to correct vulnerabilities will be released by Telekom CERT in their advisories (https://cert.telekom.de).

Motivation: Publication of vulnerabilities increases the risk of successful exploitation by an attacker, especially since the published information usually includes details on how to exploit the vulnerability and tools that make exploitation possible. One example of this is when Web servers are compromised due to not patched vulnerabilities.

For this requirement the following threats are relevant:

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

2.4. Installation requirements

Req 3.03-3	All web server processes must not run with system privileges.

If a process is started by a user with system privileges, execution must be transferred to a different user without system privileges after the start.

Motivation: If the web server process runs with administrative privileges, an attacker who obtains control over this process may control the entire system.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

2.5. Configuration requirements

Req 3.01-2	The reachability of services must be restricted.

Typically services that are enabled in the basic configuration are accessible over all interfaces of the system and can be reached from systems in connected networks. This availability is often not needed or meaningful for system functioning. For this reason, services should only be enabled on interfaces where their usage is required. On interfaces were services are active, the reachability must be limited to legitimate communication peers. This limitation must be realized on the system itself (without measures (e.g. firewall) at network side).

Motivation: Disabling services on interfaces which do not require system accessibility or by limiting the reachability can greatly reduce the potential vulnerabilities offered to an attacker. For example, access to a system via SSH from the Internet is not necessary. If this service could be accessed from the interface connected to the Internet, this would greatly increase the risk of attacks on the service.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.01-4	Unused functions of the operated software and hardware must be deactivated.

During installation of software and hardware often functions will be activated that are not necessarily needed for operation or function of the system. Functions of software are currently inherent part which could not be deleted or deinstalled individually. Such functions must be deactivated in the configuration of the system permanently.

Beside the functions of the software also hardware functions are active which are not necessary for a system. Functions like unused interfaces must permanently deactivated. Permanent means that they must not be reactivated again after system reboot.

Motivation: The hardware or software of a system often contains functions which are not used and so will be a risk for system security. Such functions give an attacker the possibility to manipulate the system. Furthermore it is possible to get unauthorized access other areas or data of the system. An example is a debugging function in software which can be used for troubleshooting but must not be activated during normal operation. Or a hardware interface that will not be used and so is unsecured an allows possibly unauthorized access to the system.

For this requirement the following threats are relevant:

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

- Disruption of availability

Req 3.03-4	HTTP methods that are not required must be deactivated.

Standard requests to web servers only use GET and POST. If other methods are required, they must be processed securely.

Motivation: HTTP TRACE could be misused by an attacker. This method allows for debugging and trace analysis of connections between the client and the web server. The Microsoft IIS web server uses the TRACK alias for this method. Other HTTP methods could also be used to obtain information about the server, or they could be directly misused by an attacker.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.03-5	Any add-ons and components that are not required must be deactivated.

All optional add-ons and components of the web server must be deactivated if they are not required. In particular,

- CGI
- Server Side Includes (SSI)
- WebDAV

must be deactivated if they are not required.

Motivation: Each add-on, component or function can have security vulnerabilities.

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

Req 3.03-7	If CGI is used, the CGI directory must not include compilers or interpreters (e.g., PERL interpreter, PHP interpreter/compiler, Tcl interpreter/compiler or operating system shells).

Motivation: Using CGI, the web server communicates with runtime environments which could be misused to execute malicious software.

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

Req 3.03-8	If CGI is used, the CGI directory must not be used for uploads.

Motivation: Using CGI, the web server communicates with runtime environments which could be misused to execute malicious software.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unnoticeable feasible attacks

Req 3.03-9	If Server Side Includes (SSI) are active, the execution of system commands must be deactivated.

Motivation: The Server Side Includes (SSI) technology, which is implemented in most web server products as an additionally loadable module, can potentially be used by attackers. The "exec" function of SSI, in particular, could be used to execute system commands, which represents a risk.

For this requirement the following threats are relevant:

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

Req 3.03-12	Access rights for web server configuration files must only be granted to the owner of the web server processs or a user with system privileges.

Motivation: Configuration files may only be written by the owner of the web server process or a user with system privileges. Otherwise it would be possible for unauthorized users to change the configuration of the web server or to obtain configuration information which could be used for an attack.

Implementation example: Delete "read" and "write" access rights for "others." Only grant "write" access to the user who configures the web server.

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

Req 3.03-13	Default content must be removed.

Default content (examples, help files, documentation, aliases) that is provided with the standard installation must be removed.

Motivation: By using examples, information could be obtained about the installed software (version). Examples can include security vulnerabilities.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.03-15	Directory listings (indexing) must be deactivated.

Motivation: Directory listings provide information on files and directory structure which could be misused.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.03-16	Information about the web server in HTTP headers must be minimized.

The HTTP header must not include information on the version of the web server and the modules/add-ons used.

Motivation: Any information about the web server could allow conclusions to be drawn about security vulnerabilities.

For this requirement the following threats are relevant:

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

Req 3.03-17	Web server information in error pages must be deleted.

Default error pages must be replaced with user-defined error pages.
User-defined error pages must not include version information about the web server and the modules/addons used.
Error messages must not include internal information such as internal server names, error codes, etc.

Motivation: Any information about the web server could allow conclusions to be drawn about security vulnerabilities.

Implementation example: Create own error pages without information about the web server product and version.

For this requirement the following threats are relevant:

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

Req 3.03-18	File type / script mappings that are not required must be deleted.

File type or script mappings that are not required must be deleted, e.g., php, phtml, js, sh, csh, bin, exe, pl, vbe, vbs.

Motivation: Script mappings define how scripts and programs are executed on the server. These mappings could provide a runtime environment for files which were placed by an attacker. This file execution must be blocked.

For this requirement the following threats are relevant:

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

Req 3.03-20	The web server may only deliver files which are meant to be delivered.

Restrictive access rights must be assigned to all files which are directly or indirectly (e.g., via links or in virtual directories) in the web server's document directory. In particular, the web server must not be able to access files which are not meant to be delivered.

Motivation: If additional files or directories are integrated via links or virtual directories into the document directory of the web server, in particular, it is possible that a user can access files via the web server which he should not be allowed to view. This must be prevented through careful configuration.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data

Req 3.01-9	The system must be robust against overload situations.

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 application
- Restricting of maximum sessions for a Web application
- Defining the maximum size of a dataset
- Restricting CPU resources per process
- Prioritizing processes
- Limiting of amount or size of transactions of an user or from an IP address in a specific time range

Motivation: An attacker can effect systems availability through targeted exploitation of vulnerabilities in resources administration. An attacker can, for example, send bulk queries to a Web server and, by doing so, compromise its availability if the maximum number of permissible Web sessions is not restricted.

For this requirement the following threats are relevant:

- Disruption of availability

2.6. HTTPS requirements

Req 3.01-7	Data with need of protection must be secured against unauthorized viewing and manipulation during transmission and storage.

Adequate security measures for transmission and storage must be implemented of data with a need for protection that are classified as internal, confidential or strictly confidential. The chosen measure depends on the classification for the data and other factors such as the type of network used during transmission, the storage location for data, etc. Assistance for the classification of data gives the Intranet page http://informationwheel.telekom.de. Furthermore must be guaranteed that confidential and strictly confidential data will not be unprotected during temporary storage (e.g. in web cache, temporary folders).

All authentication data such as user names, passwords, PINs, etc. must be protected against unauthorized viewing and manipulation. This applies equally to permanent storage and transmission. Typical measures taken to protect authentication data or combinations of these are:

For storage:

- Client systems: encryption or obfuscation of authentication data, no persistent storage, limitation of access rights.
- Server systems: hashing of authentication data with PBKDF like scrypt or bcrypt or when this is not possible hashes with salt.

For transmission:

- Usage of cryptographically protected network protocols.
- Challenge response method (Disadvantage of this solution is that passwords needed in clear-text on server. If this solution is feasible depends on the individual threat scenario)

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 the system access this kind of files. An example is the use of checksum or cryptographic methods to validate if e.g. firmware images, patches, drivers or kernel modules are free of manipulations.

For transmission of data with a need of protection it is necessary to use network protocols with insufficient security measures. Examples for these insecure protocols that should not be longer used are: SSLv3, SSHv1, FTP, Telnet, SNMPv1 and 2c. In case of these protocols a newer version without vulnerabilities or a secure alternative must be used.

Motivation: If data with a need of protection will not be 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 when passwords can be read out when they stored in an inadequate se-

cured way (e.g. usage of unsecure hashing algorithms like MD5 or SHA-1) or even without encryption on a system or manipulation of firmware to affected the system integrity.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

Req 3.03-21	For encryption with HTTPS the TLS protocol must be used.

SSL must be considered outdated and thus may not be activated or must be deactivated, respectively.

Motivation: Particularly SSLv2 has a number of weaknesses that make it impossible to use from a security point of view. TLS is the further development of SSL. It is already established for years so there is no need for further use of SSL.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

Req 3.03-22	The web server must be configured in such a way that the use of the latest version of the TLS protocol is enabled.

In particular, the web server must be configured for the use of TLS 1.2.

Motivation: The latest version of the protocol offers the best possible protection and contains fixes to known vulnerabilities in previous versions of the protocol.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

Req 3.03-24	The TLS configuration must provide that the cipher suite considered most secure is being chosen with highest priority.

A cipher suite contains the definition of four algoritthms. These are used for key exchange, authentication, encryption and as a hash function. General guidelines for the prioritization are

- For the key exchange the Diffie-Hellman method must be preferred because it offers perfect forward secrecy. Cipher suites using the Diffie-Hellman method usually may be identified by the strings DHE or ECDHE. ECDHE has higher priority than DHE.
- For encryption the Advanced Encryption Standard (AES) or Camellia with a key length as big as possible has to be used
- As a hash function SHA-2 has to be used. This function usually may be identified by the string SHA followed by a number (256, 384 or 512). Warning: if the string SHA is not followed by a number this identifies the SHA-1 function which is significantly less secure.

Motivation: When a TLS connection is being established a cipher suite is selected based on the cipher suites available both on client and on server side. In order to ensure a high compatibility to all kinds of client systems the web server must not only allow for the cipher suites considered most secure. To make sure that nevertheless for each client the best possible cipher suite is selected and thus the connection is best protected the configuration must contain an according prioritization.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

2.7. Logging

Req 3.03-25	Access to the webserver must be logged.

The web server log must contain the following information:

- Access timestamp
- Source (IP address)
- Account (if known)
- URL
- Status code of web server response

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

Motivation: For the analysis of security incidents it is very important to have basic information on how the attack has been carried out. Since a webserver represents an external interface certain information about an attack is only available on the webserver, even if the attack is aimed at a downstream system. Thus logging on a web server is mandatory.

For this requirement the following threats are relevant:

- Denial of executed activities
- Unnoticeable feasible attacks

3. 3.04Secure Shell (SSH) (V. 2.0)

3.1. Introduction

This security requirement has been prepared based on the provisions of the Group Policy on IT/NT Security.

The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions of the Group Policy on IT/NT Security 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.

3.2. Security requirements

3.2.1. General

Req 3.04-1	The SSH protocol version 2 must be used exclusively.

Motivation: SSH-2 is the current protocol version, which will be supported in contrast to SSH version 1. SSH protocol version 1 contains design weaknesses. The weaknesses were removed with latest OpenSSH versions, still for secure interaction with other SSH communication partners, version 2 of the protocol must be enforced.

Implementation example: The entry „`Protocol 2`“ in the OpenSSH config file forces the exclusive usage of SSH version 2.

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

Req 3.04-3	The SSH service must be exclusively bound to interfaces and/or IP addresses, which are needed to run/operate the service.

The SSH service on multihomed servers must not listen globally on all interfaces. On network devices this requirement often will be achieved based on ACLs.

Motivation: Exposing the SSH service to not needed IPs / interfaces increases the attack surface. This requirement achieves for devices with internet interfaces that the SSH service is never accessible from the internet.

Implementation example: The following entry in the sshd_config file achieves this requirement:

```
ListenAddress 10.40.2.2
```

(Alternatively you can use an alias name instead of the IP address, which will be resolved via `/etc/hosts`).

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
- Unnoticeable feasible attacks

3.2.2. SSH Options

Req 3.04-12	SSH logins must be logged.

Logins must be logged in a log file. The OpenSSH log level shall be 'verbose'.

Motivation: Only in the verbose-Loglevel the fingerprint of the used SSH-public key is recorded. To facilitate forensic analyses after a security incident and for system auditing, log data detailing accesses must be available.

Implementation example: The entry

```
LogLevel VERBOSE
```

in the OpenSSHD config file achieves this requirement.

For this requirement the following threats are relevant:

- Denial of executed activities
- Unnoticeable feasible attacks

3.2.3. Authentication and Accounts

Req 3.04-17	Host-based authentication (via rhosts/shosts) must not be used.

Motivation: To ensure that access cannot be gained to other systems without authentication once a system has been compromised, a machine-to-machine trust setting must not be used.

Implementation example: The entries

```
RhostsRSAAuthentication no
HostbasedAuthentication no
IgnoreRhosts yes
```

in the OpenSSHD config file achieve this requirement.

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

3.2.4. Requirements on the SSH client

4. 3.21Unix Servers (V. 2.0)

4.1. Introduction

This security requirement has been prepared based on the provisions of the Group Policy on IT/NT Security.

The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions of the Group Policy on IT/NT Security 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.

4.2. Unix systems

4.2.1. System hardening

Req 3.01-1	Unused services and protocols must be deactivated.

After installation of systems and software products typically local or remote reachable services and protocols are active, which are not necessarily needed for operation and functionality of the system. These include also services and protocols which may not be used in Deutsche Telekom Group networks on account of known security vulnerabilities, which can be used to violate confidentiality, availability or integrity of the system . Such services and protocols must be completely disabled on the system. Additionally it is important that a deactivation survives a system reboot.

This kind of system hardening must be done before the system is reachable from the network. Otherwise an attacker has the possibility to attack and maybe compromise the unsecured system.

Motivation: Services and protocols that are not required for system operation increase the potential attack surface and thus the risk of the system being compromised. This risk is further increased by the fact that a security inspection and an appropriate optimisation of the configuration for unused services and protocols will not be done.

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

Req 3.01-2	The reachability of services must be restricted.

Typically services that are enabled in the basic configuration are accessible over all interfaces of the system and can be reached from systems in connected networks. This availability is often not needed or meaningful for system functioning. For this reason, services should only be enabled on interfaces where their usage is required. On interfaces were services are active, the reachability must be limited to legitimate communication peers. This limitation must be realized on the system itself (without measures (e.g. firewall) at network side).

Motivation: Disabling services on interfaces which do not require system accessibility or by limiting the reachability can greatly reduce the potential vulnerabilities offered to an attacker. For example, access to a system via SSH from the Internet is not necessary. If this service could be accessed from the interface connected to the Internet, this would greatly increase the risk of attacks on the service.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.01-3	Unused software must not be installed or must be uninstalled.

During installation of a system often software components will be installed or parts of software will be activated which are not needed for the operation or functionality of the system. This includes also parts of a software, which will be installed as examples but typically not be used (e.g. default web pages, example databases, test data). Such components should not be installed or must be deleted after installation.

Motivation: Vulnerabilities in software of a system offer an attack window for attackers to infiltrate the system. Uninstalling components that are not required can therefore reduce the possibility of a successful compromise of the system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Disruption of availability

Req 3.01-4	Unused functions of the operated software and hardware must be deactivated.

During installation of software and hardware often functions will be activated that are not necessarily needed for operation or function of the system. Functions of software are currently inherent part which could not be deleted or deinstalled individually. Such functions must be deactivated in the configuration of the system permanently.

Beside the functions of the software also hardware functions are active which are not necessary for a system. Functions like unused interfaces must permanently deactivated. Permanent means that they must not be reactivated again after system reboot.

Motivation: The hardware or software of a system often contains functions which are not used and so will be a risk for system security. Such functions give an attacker the possibility to manipulate the system. Furthermore it is possible to get unauthorized access other areas or data of the system. An example is a debugging function in software which can be used for troubleshooting but must not be activated during normal operation. Or a hardware interface that will not be used and so is unsecured an allows possibly unauthorized access to the system.

For this requirement the following threats are relevant:

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

Req 3.21-1	Functions not needed for the operation of a server system must be deactivated.

A server must not perform routing functions; consequently the routing function must be disabled. Additionally the answering routine for broadcast ICMP packages must be disabled. Usually this is already configured correctly out-of-the-box.

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

Implementation example: (Linux)
Set /proc/sys/net/ipv4/ip_forward respectively /proc/sys/net/ipv6/conf/all/forward to „0“.
Set /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts to 1.

For this requirement the following threats are relevant:

- Unauthorized use of services or resources
- Disruption of availability

Req 3.37-1	If services cannot be bind to the minimal required interfaces by configuration, a local packet filter must regulate the accessibility of the service.

Sometimes, software cannot be bind 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

Req 3.37-2	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

Req 3.37-3	Services for administration must be bind to exact one interface.

The administrative services (e.g., SSH, HTTPS, RDP) must be bind to exact one interface. Hence 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 bind to the lowest possible number.

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 and for checking the authenticity of keys and certificates when using cryptographic procedures to secure management traffic.

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

Req 3.37-4	Network based access used for operating system administration must have integrity, 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 exploit vulnerabilities. Information could be gained (configuration settings, access IDs, etc.) to exploit additional security vulnerabilities.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Denial of executed activities

Req 3.37-5	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 share 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: Avoidance of bypassing authorized systems by man-in-the-middle attacks: The automatic configuration of IP addresses enables an attacker to divert traffic or impair communication.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Disruption of availability

Req 3.37-6	Kernel based network functions not needed for the operation as a server 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 are already 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

Req 3.21-3	If available, the function for “rp_filter” (reverse path filter) or the corresponding function of the utilized distribution must be enabled. Likewise, “strict destination multihoming” must be enabled.

Motivation: "rp_filter" filters out incoming packets containing a source address for which the interface over which the packets arrive has no return route. "Strict destination multihoming" (configurable in Linux in the proc file system under “arp_ignore” and "arp_announce”) prevents packets being routed to any address other than that configured at the incoming interface. These two functions shall be taken into special account on all systems with more than one network interface.

Implementation example: Administrators configure rp_filter in Linux by using the switch /proc/sys/net/ipv4/conf/all/rp_filter. This switchcan be set, depending on the distribution, in a system-wide configuration file. The switch for arp_announce andarp_ignore is located under /proc/sys/net/ipv4/conf/all/.

For this requirement the following threats are relevant:

- Disruption of availability
- Unnoticeable feasible attacks

Req 3.37-7	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

Req 3.21-4	Support for Trusted Hosts must be disabled.

Motivation: Trusted host relationships make it possible to use or provide access to various system resources without adequate authentication. Trusted hosts relationships therefore present major attack surface.

Implementation example: The shall be not .rhosts file in the filesystem.

For this requirement the following threats are relevant:

- Unauthorized use of services or resources
- Disruption of availability
- Unnoticeable feasible attacks

Req 3.21-5	Protection from buffer overflows must be enabled.

Some Unix systems offer protection from buffer overflows within the operating system, which is usually enabled by default on most Unix systems. The administrator shall enable this operating system function or leave it enabled.

Motivation: Buffer overflows are major attack vectors and shall be intercepted at the earliest possible point, i.e., at operating system level.

Implementation example: Solaris: Set the following lines in the /etc/system file

```
set noexec_user_stack=1
```

Linux: Set the following lines in the file /etc/sysctl.conf

```
kernel.exec-shield = 1
kernel.randomize_va_space = 1
```

For this requirement the following threats are relevant:

- Unauthorized use of services or resources
- Disruption of availability
- Unnoticeable feasible attacks

Req 3.21-7	Passwords that are stored on the system must be stored in such a way that unauthorized persons cannot access them.

Passwords must be stored only as hashes (bcrypt, scrypt), never in clear text. Files containing password hashes must be protected against unauthorized access.

Motivation: Passwords are in need of protection that only account owners or authorized persons may know and change. This measure is designed to ensure that unauthorized persons cannot gain knowledge of these passwords or have the chance to change them.

Implementation example: For system passwords, the file /etc/shadow shall be used in Linux. For other operating systems, the respectiveequivalent file shall be used, which is only readable for the root and only contains the hashes of the systempasswords.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Denial of executed activities
- Unnoticeable feasible attacks

Req 3.21-8	The path variable of all accounts must not contain the current directory “.”.

System commands have to be started based on a relative or absolute path or can be found in the current PATH variable.

Motivation: Malicious / unwanted code may be executed based on the fact, that the user is e. g. in /temp directory containing malicious codes.

Implementation example: Check: Enter „echo $PATH“ and check the output of the command.

For this requirement the following threats are relevant:

- 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

Req 3.21-9	Configuration files read by processes with root permissions, executable files executed with root permissions (e.g. via cron, "init"-scripts etc.) and directories containing these files must be protected in such a way, that only root users can change these files.

Executable files in system directories must be generally only writeable for root users. Especially this is valid for system directories such as /bin, /sbin, /usr , /root, /etc, /lib, /var. For subdirectories it may be necessary to make them accessible for other users / groups.

Motivation: A file with general write permissions can be misused / changed in such a way, that a root user executes code from an attacker, which leads to a privilege escalation.

Implementation example: All relevant files must be changed with „chmod go-w“ .

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.21-11	All processes must be started with the minimal necessary permissions.

No application, except for cron, syslog and some kernel processes, are allowed to run with UID 0.

Motivation: This requirement is necessary to block (un)intended manipulations of the system. Additionally the possibility of a privilege escalation can be reduced.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized use of services or resources

Req 3.21-13	The mount points for external data storage media which can be mounted by users without root permissions must be assigned the options “nodev” and “nosuid”.

In Linux systems, administrators shall set the options nodev and nosuid in the /etc/fstab for all filesystems, which also have the “user” option.

Motivation: The aim is to effectively prevent an unauthorized extension of privileges for users, who can mount external

data storage mediums. This option prevents users from misusing an external data storage medium with alternative "dev" files and adapted permissions or executable files, which enable an extension of privileges, set by the SUID bit.

For this requirement the following threats are relevant:

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

4.2.2. System Update

Req 3.01-5	Software and hardware components that are no longer supported by vendor, producer or developer must not be used.

Only those operating system, middleware and application software and hardware components may be used on a system which are supported by the vendor, the producer, the developer or other contractual partner of Deutsche Telekom AG. Components that have reached end-of-life or end-of-support must not be used. Excluded are components that have a special support contract. This contract must guarantee the correction of vulnerabilities over components lifetime.

Motivation: Hardware and software components that have reached end of life or end of support represent a risk for a system. This means that a vendor does not supply remedial updates or patches for a component should errors or vulnerabilities occur. This means that vulnerabilities cannot be fixed when they occur and could be exploited to compromise the system or to impair its availability.

For this requirement the following threats are relevant:

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

Req 3.01-6	Known vulnerabilities in software and hardware of the system must be fixed or protected.

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. Furthermore it is a ongoing process during the complete life cycle of the system to fix upcoming vulnerabilities promptly. Time demands to correct vulnerabilities will be released by Telekom CERT in their advisories (https://cert.telekom.de).

Motivation: Publication of vulnerabilities increases the risk of successful exploitation by an attacker, especially since the published information usually includes details on how to exploit the vulnerability and tools that make exploitation possible. One example of this is when Web servers are compromised due to not patched vulnerabilities.

For this requirement the following threats are relevant:

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

4.2.3. Protection of Data and Information

Req 3.01-8	Information with need of protection must not be contained in files, outputs or messages that are accessible by unauthorized users.

Information with need of protection must not be accessible in files, outputs or messages of the system by unauthorized users. This includes information relating to the operating system, used middleware or applications such as vendor, product name, product identifier, installed software versions, installed service packs, patches, hot fixes and serial numbers. Examples for system messages which must be free of sensitive data are:

- Comments in downloadable files
- Error and system messages
- Stack traces
- Network protocols
- Login windows and dialogs

Furthermore, details of implementation and information relating, e.g., to backend software/systems, function calls, SQL instructions or structure of database, must not be contained in error messages.
Excluded from this are displays and outputs that can be viewed and retrieved by authorized users who are logged in. In addition, an internal transfer of system internal information for error analysis is allowed in an adequate dimension. In this case the continuative regulations or guidelines (e.g. of data privacy) must be noticed.

Motivation: The information named above can be used by an attacker to prepare specific attacks on a system. In this way an attacker could, for example, use the precise software version to identify vulnerabilities in the product and, in a second step, exploit them.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.37-8	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

4.2.4. Availability and Integrity

Req 3.01-9	The system must be robust against overload situations.

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 application
- Restricting of maximum sessions for a Web application
- Defining the maximum size of a dataset
- Restricting CPU resources per process
- Prioritizing processes
- Limiting of amount or size of transactions of an user or from an IP address in a specific time range

Motivation: An attacker can effect systems availability through targeted exploitation of vulnerabilities in resources administration. An attacker can, for example, send bulk queries to a Web server and, by doing so, compromise its availability if the maximum number of permissible Web sessions is not restricted.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-10	If an overload situation cannot be prevented, the system must act in a predictable way.

A system must be built in this way that it can react on a overload situation in a controlled way. However it is possible that a situation happens where the security measures are not longer sufficient.

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.

Motivation: With denial-of-service attacks can an attacker try to overload a system to effect its availability or integrity. Unpredictable acting of the system is then a risk for functionality and data and possibly also for other systems.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-11	The system must be robust against unexpected input.

During transmission of data to a system it is necessary to validate this before processing. This includes all data which are send to the system. Examples for this are user input, values in arrays and content in protocols. 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
- White listing or escaping for inputs outside the values margin

Motivation: An attacker can try to put a system in an unsecure state through targeted manipulation of transmitted data. The object of such an attack is to compromise the usability, availability or integrity of individual services or of the entire system. For instance a unclean memory handling can lead to a buffer overflow that allows an attacker to execute arbitrary code on the effected system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability

Req 3.37-9	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

Req 3.37-10	Systems must not process IP packets which source address is not reachable via the incoming interface.

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

Req 3.37-11	The processing of ICMPv4 and ICMPv6 packets which are not required for operation must be disabled.

There are different types of ICMP4 and ICMPv6 that are not used in most networks, but represent a risk. These types must be disabled or filtered and not be answered, send or processed. 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

4.2.5. Authentication and Authorization

Req 3.37-13	If the system is not located in a room with at least protection class "high" (PC3), the BIOS must be secured against unauthorized changes.

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. 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

Req 3.37-14	There must not be a privilege escalation method which allow gaining administrator/root privileges from an user account without anew authentication with two authentication attributes.

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

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

Implementation example: (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

Req 3.37-12	IP packets with unnecessary options or extension headers must not be processed.

IP options and extension headers (e.g., source routing) are only required in exceptional cases. So, all packets with enabled IP options or extension headers must be filtered.

Motivation: Packets with IP options require extended processing. An attacker can exploit this in order to carry out denial-of-service attacks against an affected device.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-12	System functions with a need of protection must not be used or accessed without successful authentication and authorization.

The usage of a system functions or access of data classified as internal, confidential or strictly confidential must only be possible unambiguous user identification and successful authentication on basis of the user name and at least one authentication attribute. Excepted from this are functions for public use such as those for a Web server on the Internet, via which information is made available to the public. Examples for functions which require a prior authentication are network services (like SSH, SFTP, Web services), local access via a management console, local usage of operating systems and applications. The following examples are possibilities that could be used for authentication.

- Query user name and password
- Use of cryptographic keys and certificates (e.g. as Smartcard)

This requirement must also be applied to accounts that are only used for communication between systems (M2M).

Motivation: The authentication is necessary to doubtless identify a user because the allocated authorization, and therefore the access on data and services of the system depends on that.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-14	Accounts must be protected against unauthorized use by at least one authentication attribute.

The various user 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 keys
- Token
- Passwords
- PINs

This means that authentication based on a parameter that can be spoofed (e.g. phone numbers, public IP addresses or VPN membership) is not permitted. Exceptions are attributes that cannot be faked or spoofed by an attacker. Several of the above options can be combined (dual-factor authentication) to achieve a higher level of security. Whether or not this is suitable and necessary depends on the protection needs of the individual system and its data and must be evaluated for individual cases.

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts that are not protected with a secret authentication attribute can be used 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 use of services or resources
- Denial of executed activities

Req 3.01-15	Accounts with extensive rights must be protected with two authentication attributes.

Accounts, for example used for administration, maintenance and troubleshooting, have extensive rights. Therefore a single protection (e.g. a password), as for normal user accounts with less rights, is not suitable. To get a higher protection level it is necessary to use more than one authentication attribute. For this a combination of an attribute that the user knows and an attribute that the user owns will be used often. This kind of authentication will be named as 2-factor authentication. Examples for 2-factor authentication are:

- Smartcard (e.g. MyCard) with PIN
- Private key with Passphrase
- Secure-ID Token with Password

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts with extensive rights as used for system administration have a higher risk for system's security. An attacker can get extensive rights by compromising such an account to get access to wide parts of the system and stored data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-16	Predefined accounts must be deleted or disabled.

Many systems have default accounts (e.g. guest, ctxsys), some of which are preconfigured with or without known passwords. These standard users must be deleted or disabled. Should this measure not be possible the accounts must be locked for remote login. In any case disabled or locked accounts must configured with a complex password (12 character and more, use of upper/lower case, numbers and special characters). This is necessary to prevent unauthorised use of such a account in case of misconfiguration.

Exceptions to this requirement to delete or disable accounts are accounts that are used only internal on the system involved and that are required for one or more applications on the system to function. Also for this accounts remote access or local login must be forbidden to prevent a abusive use by users of the system.

Motivation: Standard users are typically generally known and can be used by an attacker for targeted brute force and dictionary attacks. Standard user accounts represent a special risk if they do not use a password or only use a standard password that is generally known. Such standard user accounts can easily be exploited by an attacker in order to gain access to the system involved without being authorized to do so.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-17	Predefined authentication attributes must be deleted or disabled.

Normally, authentication attributes such as password or cryptographic keys will be preconfigured from producer, vendor or developer of a system. Such authentication attributes must be changed an own attribute not known by other parties.

Motivation: Authentication attributes like password or cryptographic keys preconfigured from third parties are not trustable. Such authentication attributes can be used to compromise systems or their data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-18	The authorizations for accounts and applications must be reduced to the minimum required for the tasks they have to perform.

Authorizations to a system must be restricted to a level in which a user can only access data and use functions that he needs in the course of his work. Suitable authorizations must also be assigned for access to files that are components of the operating system or of applications or that are generated by the same (e.g. configuration and logging files).

Alongside access to data, execution of applications and components must also take place with rights that are as low as possible. Applications should not be executed with administrator or system rights.

Motivation: If the rights granted to a user on a system are too broad, it could be possible for him to access data and applications which he is not permitted to view or use. This would give him 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 his own authorizations and thus to gain access to files and system components to which he would not have had access with his authorizations under normal circumstances.

For this requirement the following threats are relevant:

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

Req 3.01-19	The system must be connected to a central system for user administration.

Accounts and their rights will be administrated on central identity management systems (e.g. cIAM, WiW, ZAM) in Deutsche Telekom Group. For provisioning of these authorization information must the system provide a central interface (e.g. LDAPs for authorization, Kerberos for authentication, locking information for certificates) or decentralized mechanisms (e.g. public-key authentication) . A central solution for identity management must be preferred.

In areas where a central identity management system is not available a central system such as LDAP, TACACS+ or Radius server for the administration of accounts and their authentication and authorization must be used.

Motivation: Central administration of identity of accounts and their rights means that they only have to be maintained once instead of separately on each system. From the aspect of security, the advantage is that an account and its rights only known on a single central side. This information can be transmitted from a central side to systems (provisioning), central administrated (reconciliation) and central deleted (deprovisioning). This reduces the risk of accounts being forgotten during changing or deletion since they are configured on multiple systems. This could give a user wrong system rights or continued access to a system.

For this requirement the following threats are relevant:

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

Req 3.37-15	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.

The passwords needed to login at these accounts must be unique for every machine and account. It is recommended to store them properly, e.g., in a safe or online in a password vault where access is granted only when needed.

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

Req 3.21-16	Each system account must have a unique UID.

Motivation: If accounts other than the root account bear UID 0, this often indicates that a successful attack has been carried out on a UNIX system. In order to prevent misunderstandings, maintain system clarity and take account the requirement, which specifies that functional accounts instead of user-specific accounts shall be used for certain administrative tasks, it is necessary that one Unix account only may have UID 0

For this requirement the following threats are relevant:

- Denial of executed activities

4.2.6. Authentication Parameter Password

Req 3.01-23	If a password is used as an authentication attribute, it must have at least 8 characters and contain

	three of the following categories: upper cases, lower case, numbers and special characters.

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

- Minimum length of 8 characters.
- Comprising at least three of the following categories: upper/lower case letters, numbers and special characters

When a password is assigned, the system must ensure that the password meets these requirements. If a central system will be used for user authentication this function can be forwarded or delegated to this system.

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

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-24	If a password is used as an authentication attribute, users must be enable 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 will be used it is possible 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 use of services or resources
- Denial of executed activities

Req 3.01-25	If a password is used as an authentication attribute, a protection against brute force and dictionary attacks that hinder password guessing must be implemented.

Brute force and dictionary attacks aim to use automated guessing to ascertain passwords for user and machine accounts. Various measures or a combination of this measures can be taken to prevent this.

The most commonly used protection measures are:

- Increasing the delay (e.g. doubling wait times at each attempt) for each renewed password input following an incorrect entry (“tar pit”).
- Blocking an account following a specified number of incorrect attempts (typically 5). However has to take in account that this solution needs a process for unlocking and an attacker can force this to deactivate accounts and make them unusable.
- Using CAPTCHA to prevent automated attempts (often used for Web applications).

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: Implementation of one or more of the above measures can prevent successful execution of brute force and dictionary attacks to ascertain passwords.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-26	If a password is used as an authentication attribute, they must be hidden when displayed on screen.

When a user enters a password or the password is otherwise displayed on screen, it must be made unreadable. Typically, the individual characters of the password are replaced by a character such as "*". Under certain circumstances it may be permissible for an individual character to be displayed briefly during input. Such a function is used, for example, on Smartphone's to make input easier. However, the entire password is never output to the display in plaintext.

Motivation: To prevent another person reading a password by chance or intentionally on screen during input, the password must be rendered unreadable during input or display.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

4.2.7. Logging

Req 3.01-27	Security relevant events must be logged with a precise timestamp and a unique system reference.

Systems must log the occurrence of security-relevant incidents. So that these events can be evaluated and classified, they must be logged together with a unique system reference (e.g., host name, IP or MAC address) and the exact time the incident occurred. Furthermore, legal and data privacy regulations (e.g. time of storage of logging data) must be proved and followed. The following table lists events that are relevant to security and the corresponding data that typically has to be logged by a system. Exceptions are systems for which no or only restricted logging applies. Examples of such systems are customer devices such as Smartphone's or IADs/Homegateways (e.g. Speedport).

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

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

Event	Event data to be logged
Incorrect login attempts	• Account, • No. of failed attempts, • Source (IP address) of remote access
System access with accounts with administrator rights	• Account, • Access timestamp, • Length of session, • Source (IP address) of remote access
Account administration	• Administrator account, • Administered account, • Activity performed (configure, delete, enable and disable)

Change of group membership for accounts	• Administrator account, • Administered account, • Activity performed (group added or removed)
Critical rise in system values such as disk space, CPU load over a longer period	• Value exceeded, • Value reached (Here suitable threshold values must be defined depending on the individual system.)

Logging of additional security-relevant events may be meaningful. This must be verified in individual cases and implemented accordingly where required.

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

Req 3.01-28	Security relevant logging data must be send to an external system direct after their creation.

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.

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:

- Denial of executed activities
- Unnoticeable feasible attacks

5. 3.37Operating Systems (V. 2.0)

5.1. Introduction

This document has been prepared based on the provisions of the Group Policy on IT/NT Security.

The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions of the Group Policy on IT/NT Security 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.

5.2. System Hardening

Req 3.01-1	Unused services and protocols must be deactivated.

After installation of systems and software products typically local or remote reachable services and protocols are active, which are not necessarily needed for operation and functionality of the system. These include also services and protocols which may not be used in Deutsche Telekom Group networks on account of known security vulnerabilities, which can be used to violate confidentiality, availability or integrity of the system . Such services and protocols must be completely disabled on the system. Additionally it is important that a deactivation survives a system reboot.

This kind of system hardening must be done before the system is reachable from the network. Otherwise an attacker has the possibility to attack and maybe compromise the unsecured system.

Motivation: Services and protocols that are not required for system operation increase the potential attack surface and thus the risk of the system being compromised. This risk is further increased by the fact that a security inspection and an appropriate optimisation of the configuration for unused services and protocols will not be done.

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

Req 3.01-2	The reachability of services must be restricted.

Typically services that are enabled in the basic configuration are accessible over all interfaces of the system and can be reached from systems in connected networks. This availability is often not needed or meaningful for system functioning. For this reason, services should only be enabled on interfaces where their usage is required. On interfaces were services are active, the reachability must be limited to legitimate communication peers. This limitation must be realized on the system itself (without measures (e.g. firewall) at network side).

Motivation: Disabling services on interfaces which do not require system accessibility or by limiting the reachability can greatly reduce the potential vulnerabilities offered to an attacker. For example, access to a system via SSH from the Internet is not necessary. If this service could be accessed from the interface connected to the Internet, this would greatly increase the risk of attacks on the service.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.37-1	If services cannot be bind to the minimal required interfaces by configuration, a local packet filter

	must regulate the accessibility of the service.

Sometimes, software cannot be bind 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

Req 3.37-2	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

Req 3.37-3	Services for administration must be bind to exact one interface.

The administrative services (e.g., SSH, HTTPS, RDP) must be bind to exact one interface. Hence 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 bind to the lowest possible number.

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 and for checking the authenticity of keys and certificates when using cryptographic procedures to secure management traffic.

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

Req 3.37-4	Network based access used for operating system administration must have integrity, 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 exploit vulnerabilities. Information could be gained (configuration settings, access IDs, etc.) to exploit additional security vulnerabilities.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Denial of executed activities

Req 3.01-3	Unused software must not be installed or must be uninstalled.

During installation of a system often software components will be installed or parts of software will be activated which are not needed for the operation or functionality of the system. This includes also parts of a software, which will be installed as examples but typically not be used (e.g. default web pages, example databases, test data). Such components should not be installed or must be deleted after installation.

Motivation: Vulnerabilities in software of a system offer an attack window for attackers to infiltrate the system. Uninstalling components that are not required can therefore reduce the possibility of a successful compromise of the system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Disruption of availability

Req 3.01-4	Unused functions of the operated software and hardware must be deactivated.

During installation of software and hardware often functions will be activated that are not necessarily needed for operation or function of the system. Functions of software are currently inherent part which could not be deleted or deinstalled individually. Such functions must be deactivated in the configuration of the system permanently.

Beside the functions of the software also hardware functions are active which are not necessary for a system. Functions like unused interfaces must permanently deactivated. Permanent means that they must not be reactivated again after system reboot.

Motivation: The hardware or software of a system often contains functions which are not used and so will be a risk for system security. Such functions give an attacker the possibility to manipulate the system. Furthermore it is possible to get unauthorized access other areas or data of the system. An example is a debugging function in software which can be used for troubleshooting but must not be activated during normal operation. Or a hardware interface that will not be used and so is unsecured an allows possibly unauthorized access to the system.

For this requirement the following threats are relevant:

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

Req 3.37-5	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 share 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: Avoidance of bypassing authorized systems by man-in-the-middle attacks: The automatic configuration of IP addresses enables an attacker to divert traffic or impair communication.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data

- Disruption of availability

Req 3.37-6	Kernel based network functions not needed for the operation as a server 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 are already 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

Req 3.37-7	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

5.3. System Update

Req 3.01-5	Software and hardware components that are no longer supported by vendor, producer or developer must not be used.

Only those operating system, middleware and application software and hardware components may be used on a system which are supported by the vendor, the producer, the developer or other contractual partner of Deutsche Telekom AG. Components that have reached end-of-life or end-of-support must not be used. Excluded are components that have a special support contract. This contract must guarantee the correction of vulnerabilities over components lifetime.

Motivation: Hardware and software components that have reached end of life or end of support represent a risk for a system. This means that a vendor does not supply remedial updates or patches for a component should errors or vulnerabilities occur. This means that vulnerabilities cannot be fixed when they occur and could be exploited to compromise the system or to impair its availability.

For this requirement the following threats are relevant:

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

Req 3.01-6	Known vulnerabilities in software and hardware of the system must be fixed or protected.

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. Furthermore it is a ongoing process during the complete life cycle of the system to fix upcoming vulnerabilities promptly. Time demands to correct vulnerabilities will be released by Telekom CERT in their advisories (

https://cert.telekom.de).

Motivation: Publication of vulnerabilities increases the risk of successful exploitation by an attacker, especially since the published information usually includes details on how to exploit the vulnerability and tools that make exploitation possible. One example of this is when Web servers are compromised due to not patched vulnerabilities.

For this requirement the following threats are relevant:

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

5.4. Protection of Data and Information

Req 3.01-7	Data with need of protection must be secured against unauthorized viewing and manipulation during transmission and storage.

Adequate security measures for transmission and storage must be implemented of data with a need for protection that are classified as internal, confidential or strictly confidential. The chosen measure depends on the classification for the data and other factors such as the type of network used during transmission, the storage location for data, etc. Assistance for the classification of data gives the Intranet page http://informationwheel.telekom.de. Furthermore must be guaranteed that confidential and strictly confidential data will not be unprotected during temporary storage (e.g. in web cache, temporary folders).

All authentication data such as user names, passwords, PINs, etc. must be protected against unauthorized viewing and manipulation. This applies equally to permanent storage and transmission. Typical measures taken to protect authentication data or combinations of these are:

For storage:

- Client systems: encryption or obfuscation of authentication data, no persistent storage, limitation of access rights.
- Server systems: hashing of authentication data with PBKDF like scrypt or bcrypt or when this is not possible hashes with salt.

For transmission:

- Usage of cryptographically protected network protocols.
- Challenge response method (Disadvantage of this solution is that passwords needed in clear-text on server. If this solution is feasible depends on the individual threat scenario)

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 the system access this kind of files. An example is the use of checksum or cryptographic methods to validate if e.g. firmware images, patches, drivers or kernel modules are free of manipulations.

For transmission of data with a need of protection it is necessary to use network protocols with insufficient security measures. Examples for these insecure protocols that should not be longer used are: SSLv3, SSHv1, FTP, Telnet, SNMPv1 and 2c. In case of these protocols a newer version without vulnerabilities or a secure alternative must be used.

Motivation: If data with a need of protection will not be 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 when passwords can be read out when they stored in an inadequate secured way (e.g. usage of unsecure hashing algorithms like MD5 or SHA-1) or even without encryption on a system or manipulation of firmware to affected the system integrity.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data

- Unauthorized modification of data

Req 3.37-8	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

Req 3.01-8	Information with need of protection must not be contained in files, outputs or messages that are accessible by unauthorized users.

Information with need of protection must not be accessible in files, outputs or messages of the system by unauthorized users. This includes information relating to the operating system, used middleware or applications such as vendor, product name, product identifier, installed software versions, installed service packs, patches, hot fixes and serial numbers. Examples for system messages which must be free of sensitive data are:

- Comments in downloadable files
- Error and system messages
- Stack traces
- Network protocols
- Login windows and dialogs

Furthermore, details of implementation and information relating, e.g., to backend software/systems, function calls, SQL instructions or structure of database, must not be contained in error messages.
Excluded from this are displays and outputs that can be viewed and retrieved by authorized users who are logged in. In addition, an internal transfer of system internal information for error analysis is allowed in an adequate dimension. In this case the continuative regulations or guidelines (e.g. of data privacy) must be noticed.

Motivation: The information named above can be used by an attacker to prepare specific attacks on a system. In this way an attacker could, for example, use the precise software version to identify vulnerabilities in the product and, in a second step, exploit them.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

5.5. Availability and Integrity

Req 3.01-9	The system must be robust against overload situations.

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 application
- Restricting of maximum sessions for a Web application
- Defining the maximum size of a dataset
- Restricting CPU resources per process
- Prioritizing processes
- Limiting of amount or size of transactions of an user or from an IP address in a specific time range

Motivation: An attacker can effect systems availability through targeted exploitation of vulnerabilities in resources administration. An attacker can, for example, send bulk queries to a Web server and, by doing so, compromise its availability if the maximum number of permissible Web sessions is not restricted.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-10	If an overload situation cannot be prevented, the system must act in a predictable way.

A system must be built in this way that it can react on a overload situation in a controlled way. However it is possible that a situation happens where the security measures are not longer sufficient.

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.

Motivation: With denial-of-service attacks can an attacker try to overload a system to effect its availability or integrity. Unpredictable acting of the system is then a risk for functionality and data and possibly also for other systems.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-11	The system must be robust against unexpected input.

During transmission of data to a system it is necessary to validate this before processing. This includes all data which are send to the system. Examples for this are user input, values in arrays and content in protocols. 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
- White listing or escaping for inputs outside the values margin

Motivation: An attacker can try to put a system in an unsecure state through targeted manipulation of transmitted data. The object of such an attack is to compromise the usability, availability or integrity of individual services or of the entire system. For instance a unclean memory handling can lead to a buffer overflow that allows an attacker to execute arbitrary code on the effected system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability

Req 3.37-9	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

Req 3.37-10	Systems must not process IP packets which source address is not reachable via the incoming interface.

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

Req 3.37-11	The processing of ICMPv4 and ICMPv6 packets which are not required for operation must be disabled.

There are different types of ICMP4 and ICMPv6 that are not used in most networks, but represent a risk. These types must be disabled or filtered and not be answered, send or processed. 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

Req 3.37-12	IP packets with unnecessary options or extension headers must not be processed.

IP options and extension headers (e.g., source routing) are only required in exceptional cases. So, all packets with enabled IP options or extension headers must be filtered.

Motivation: Packets with IP options require extended processing. An attacker can exploit this in order to carry out denial-of-service attacks against an affected device.

For this requirement the following threats are relevant:

- Disruption of availability

5.6. Authentication and Authorization

Req 3.01-12	System functions with a need of protection must not be used or accessed without successful authentication and authorization.

The usage of a system functions or access of data classified as internal, confidential or strictly confidential must only be possible unambiguous user identification and successful authentication on basis of the user name and at least one authentication attribute. Excepted from this are functions for public use such as those for a Web server on the Internet, via which information is made available to the public. Examples for functions which require a prior authentication are network services (like SSH, SFTP, Web services), local access via a management console, local usage of operating systems and applications. The following examples are possibilities that could be used for authentication.

- Query user name and password
- Use of cryptographic keys and certificates (e.g. as Smartcard)

This requirement must also be applied to accounts that are only used for communication between systems (M2M).

Motivation: The authentication is necessary to doubtless identify a user because the allocated authorization, and therefore the access on data and services of the system depends on that.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.37-13	If the system is not located in a room with at least protection class "high" (PC3), the BIOS must be secured against unauthorized changes.

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. 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

Req 3.01-13	Accounts must be used that allow unambiguous identification of the user.

Users must be identified unambiguously by the system. This can typically be reached by using a unique account per user. So named group accounts, i.e. the use of one account for several persons, must not be used. On exception of this requirement are so named machine accounts. These will be used for authentication and authorization from system 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. In this connection, it has to be guaranteed that this account can't be misused. Possibilities to protect these accounts are:

- Configuring of a Password that fulfils the security requirements and is known by less than possible circle of administrators.
- Configuring the account that only a local use is possible and a interactive login isn't possible.
- Use of a technique for authentication of the specific account with public and private key or certificates.
- Limiting the access over the network for legitimised systems.

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

Motivation: Unambiguous user identification is a prerequisite for assigning a user the rights that he requires to perform his 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 use of services or resources
- Denial of executed activities

Req 3.01-14	Accounts must be protected against unauthorized use by at least one authentication attribute.

The various user 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 keys
- Token
- Passwords
- PINs

This means that authentication based on a parameter that can be spoofed (e.g. phone numbers, public IP addresses or VPN membership) is not permitted. Exceptions are attributes that cannot be faked or spoofed by an attacker. Several of the above options can be combined (dual-factor authentication) to achieve a higher level of security. Whether or not this is suitable and necessary depends on the protection needs of the individual system and its data and must be evaluated for individual cases.

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts that are not protected with a secret authentication attribute can be used 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 use of services or resources
- Denial of executed activities

Req 3.01-15	Accounts with extensive rights must be protected with two authentication attributes.

Accounts, for example used for administration, maintenance and troubleshooting, have extensive rights. Therefore a single protection (e.g. a password), as for normal user accounts with less rights, is not suitable. To get a higher protection level it is necessary to use more than one authentication attribute. For this a combination of an attribute that the user knows and an attribute that the user owns will be used often. This kind of authentication will be named as 2-factor authentication. Examples for 2-factor authentication are:

- Smartcard (e.g. MyCard) with PIN
- Private key with Passphrase
- Secure-ID Token with Password

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts with extensive rights as used for system administration have a higher risk for system's security. An attacker can get extensive rights by compromising such an account to get access to wide parts of the system and stored data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.37-14	There must not be a privilege escalation method which allow gaining administrator/root privileges from an user account without anew authentication with two authentication attributes.

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

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

Implementation example: (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

Req 3.01-16	Predefined accounts must be deleted or disabled.

Many systems have default accounts (e.g. guest, ctxsys), some of which are preconfigured with or without known passwords. These standard users must be deleted or disabled. Should this measure not be possible the accounts must be locked for remote login. In any case disabled or locked accounts must configured with a complex password (12 character and more, use of upper/lower case, numbers and special characters). This is necessary to prevent unauthorised use of such a account in case of misconfiguration.

Exceptions to this requirement to delete or disable accounts are accounts that are used only internal on the system involved and that are required for one or more applications on the system to function. Also for this accounts remote access or local login must be forbidden to prevent a abusive use by users of the system.

Motivation: Standard users are typically generally known and can be used by an attacker for targeted brute force and dictionary attacks. Standard user accounts represent a special risk if they do not use a password or only use a stand-

ard password that is generally known. Such standard user accounts can easily be exploited by an attacker in order to gain access to the system involved without being authorized to do so.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-17	Predefined authentication attributes must be deleted or disabled.

Normally, authentication attributes such as password or cryptographic keys will be preconfigured from producer, vendor or developer of a system. Such authentication attributes must be changed an own attribute not known by other parties.

Motivation: Authentication attributes like password or cryptographic keys preconfigured from third parties are not trustable. Such authentication attributes can be used to compromise systems or their data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-18	The authorizations for accounts and applications must be reduced to the minimum required for the tasks they have to perform.

Authorizations to a system must be restricted to a level in which a user can only access data and use functions that he needs in the course of his work. Suitable authorizations must also be assigned for access to files that are components of the operating system or of applications or that are generated by the same (e.g. configuration and logging files).

Alongside access to data, execution of applications and components must also take place with rights that are as low as possible. Applications should not be executed with administrator or system rights.

Motivation: If the rights granted to a user on a system are too broad, it could be possible for him to access data and applications which he is not permitted to view or use. This would give him 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 his own authorizations and thus to gain access to files and system components to which he would not have had access with his authorizations under normal circumstances.

For this requirement the following threats are relevant:

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

Req 3.01-19	The system must be connected to a central system for user administration.

Accounts and their rights will be administrated on central identity management systems (e.g. cIAM, WiW, ZAM) in Deutsche Telekom Group. For provisioning of these authorization information must the system provide a central interface (e.g. LDAPs for authorization, Kerberos for authentication, locking information for certificates) or decentralized mechanisms (e.g. public-key authentication) . A central solution for identity management must be preferred.

In areas where a central identity management system is not available a central system such as LDAP, TACACS+ or Radius server for the administration of accounts and their authentication and authorization must be used.

Motivation: Central administration of identity of accounts and their rights means that they only have to be maintained once instead of separately on each system. From the aspect of security, the advantage is that an account and its rights only known on a single central side. This information can be transmitted from a central side to systems (provisioning),

central administrated (reconciliation) and central deleted (deprovisioning). This reduces the risk of accounts being forgotten during changing or deletion since they are configured on multiple systems. This could give a user wrong system rights or continued access to a system.

For this requirement the following threats are relevant:

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

Req 3.37-15	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.

The passwords needed to login at these accounts must be unique for every machine and account. It is recommended to store them properly, e.g., in a safe or online in a password vault where access is granted only when needed.

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

5.7. Authentication Parameter Password

Req 3.01-23	If a password is used as an authentication attribute, it must have at least 8 characters and contain three of the following categories: upper cases, lower case, numbers and special characters.

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

- Minimum length of 8 characters.
- Comprising at least three of the following categories: upper/lower case letters, numbers and special characters

When a password is assigned, the system must ensure that the password meets these requirements. If a central system will be used for user authentication this function can be forwarded or delegated to this system.

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

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-24	If a password is used as an authentication attribute, users must be enable 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 will be used it is possible 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 use of services or resources
- Denial of executed activities

Req 3.01-25	If a password is used as an authentication attribute, a protection against brute force and dictionary attacks that hinder password guessing must be implemented.

Brute force and dictionary attacks aim to use automated guessing to ascertain passwords for user and machine accounts. Various measures or a combination of this measures can be taken to prevent this.

The most commonly used protection measures are:

- Increasing the delay (e.g. doubling wait times at each attempt) for each renewed password input following an incorrect entry ("tar pit").
- Blocking an account following a specified number of incorrect attempts (typically 5). However has to take in account that this solution needs a process for unlocking and an attacker can force this to deactivate accounts and make them unusable.
- Using CAPTCHA to prevent automated attempts (often used for Web applications).

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: Implementation of one or more of the above measures can prevent successful execution of brute force and dictionary attacks to ascertain passwords.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-26	If a password is used as an authentication attribute, they must be hidden when displayed on screen.

When a user enters a password or the password is otherwise displayed on screen, it must be made unreadable. Typically, the individual characters of the password are replaced by a character such as "*". Under certain circumstances it may be permissible for an individual character to be displayed briefly during input. Such a function is used, for example, on Smartphone's to make input easier. However, the entire password is never output to the display in plaintext.

Motivation: To prevent another person reading a password by chance or intentionally on screen during input, the password must be rendered unreadable during input or display.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

5.8. Logging

Req 3.01-27	Security relevant events must be logged with a precise timestamp and a unique system reference.

Systems must log the occurrence of security-relevant incidents. So that these events can be evaluated and classified,

they must be logged together with a unique system reference (e.g., host name, IP or MAC address) and the exact time the incident occurred. Furthermore, legal and data privacy regulations (e.g. time of storage of logging data) must be proved and followed. The following table lists events that are relevant to security and the corresponding data that typically has to be logged by a system. Exceptions are systems for which no or only restricted logging applies. Examples of such systems are customer devices such as Smartphone's or IADs/Homegateways (e.g. Speedport).

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

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

Event	Event data to be logged
Incorrect login attempts	• Account, • No. of failed attempts, • Source (IP address) of remote access
System access with accounts with administrator rights	• Account, • Access timestamp, • Length of session, • Source (IP address) of remote access
Account administration	• Administrator account, • Administered account, • Activity performed (configure, delete, enable and disable)
Change of group membership for accounts	• Administrator account, • Administered account, • Activity performed (group added or removed)
Critical rise in system values such as disk space, CPU load over a longer period	• Value exceeded, • Value reached (Here suitable threshold values must be defined depending on the individual system.)

Logging of additional security-relevant events may be meaningful. This must be verified in individual cases and implemented accordingly where required.

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

Req 3.01-28	Security relevant logging data must be send to an external system direct after their creation.

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.

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:

- Denial of executed activities
- Unnoticeable feasible attacks

6. 3.42Network Devices (V. 2.0)

6.1. Introduction

This security requirement has been prepared based on the provisions of the Group Policy on IT/NT Security.

The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for provisions of the Group Policy on IT/NT Security 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.

6.2. System hardening

Req 3.42-1	Any services and protocols that are not secure and not used must be disabled.

Many network devices offer services which may not be used in the Deutsche Telekom Group's networks on account of known security vulnerabilities such as non-encrypted transmission or inadequate authentication, etc. These services must be completely disabled. The services not to be used are:

- FTP
- TFTP
- Telnet
- rlogin, RCP, RSH
- HTTP
- SNMPv1 and v2
- SSHv1
- TCP/UDP Small Servers (Echo, Chargen, Discard und Daytime)
- Finger
- BOOTP server
- Discovery protocols (CDP, LLDP)
- IP Identification Service (Identd)
- PAD
- MOP

As an alternative to disabling the HTTP service, it is also possible for this service to remain active for reasons of user friendliness. In this case, however, queries to the web service may not be answered directly on this port but must be diverted to a port on which the encrypted HTTPS protocol is used.
Discovery protocols such as the Cisco Discovery Protocol (CDP) or the Link Layer Discovery Protocol (LLDP) must be completely disabled. These protocols may be used in well-founded, exceptional cases. However, it must be ensured that the protocols are only active on internal links. Discovery protocols must be disabled on interfaces to customers or devices.

Should additional services be available on a network devices, a check should be carried out in each case to establish whether the services are necessary for the operation of the network devices. Otherwise these services shall be disabled.

Motivation: The protocols named display various security vulnerabilities. A large proportion of the protocol messages is transmitted in plain text, for example. An attacker who is able to record such communication is then able to obtain confidential data such as user names and passwords. Another vulnerability inherent in the aforementioned services is the susceptibility to denial-of-service attacks (DoS). These can be used by attackers to compromise network device availability.

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

Req 3.42-3	Management traffic must be separated from signalling and commercial traffic.

Data traffic for the management of a network device must be physically or logically separated from any other traffic. Physical separation is, for example, a separate dedicated interface via which the network device is connected to a separate management network. Logical separation can be achieved via VPNs or VLANs. In this case, traffic is transmitted within the same physical network as other traffic but is logically separate. Thus direct access to the management of the network device from the production network i.e., possible access by customers, is therefore prevented.

Motivation: The management services and traffic are an attractive target for attackers. By recording management traffic, an attacker may obtain important information which can be used to prepare and carry out attacks. Direct accessibility of the management services by customers or from within the Internet increases the risk of a successful attack against a network device. Since system administration with high-level access rights normally takes place via such services, an attacker might compromise the entire network device via this and, in doing so, gain unauthorized access to the network device and any networks connected to it.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Disruption of availability

Req 3.42-4	The accessibility of management services must be restricted to legitimate systems.

Access can be restricted, for example, through filters, access lists or a local firewall. The restriction must be as strict as posible.This means to host or network adresses to achive that the managment services can only be reached from legitimated systems.

Motivation: Management services enable access to network devices in order to perform operational tasks. 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.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Disruption of availability

Req 3.42-5	Management services must be permanently connected to an address.

The management services (e.g., SSH, HTTPS or SNMP) that are active on a network device must be permanently connected to an address of the network device. Hence the required separation of management traffic from control and user traffic is the appropriate adress from the management address range. This ensures that the relevant traffic always comes from a fixed sender address and on the other side the management servicve can be reached under the same address at any time.

Motivation: Without the implementation of this measure, it cannot be clearly foreseen with which sender address packets of the management services of the network device are sent out or on which address management services are reachable. This causes a number of disadvantages. Thus recognition of attacks in logging and monitoring and the analysis of the data arising therefrom is made much more difficult. In addition, a permanent sender address is important for implementing filters and firewall rules and for checking the authenticity of keys and certificates when using cryptographic procedures to secure management services and traffic.

For this requirement the following threats are relevant:

- Denial of executed activities
- Unnoticeable feasible attacks

Req 3.01-3	Unused software must not be installed or must be uninstalled.

During installation of a system often software components will be installed or parts of software will be activated which are not needed for the operation or functionality of the system. This includes also parts of a software, which will be installed as examples but typically not be used (e.g. default web pages, example databases, test data). Such components should not be installed or must be deleted after installation.

Motivation: Vulnerabilities in software of a system offer an attack window for attackers to infiltrate the system. Uninstalling components that are not required can therefore reduce the possibility of a successful compromise of the system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Disruption of availability

Req 3.01-4	Unused functions of the operated software and hardware must be deactivated.

During installation of software and hardware often functions will be activated that are not necessarily needed for operation or function of the system. Functions of software are currently inherent part which could not be deleted or deinstalled individually. Such functions must be deactivated in the configuration of the system permanently.

Beside the functions of the software also hardware functions are active which are not necessary for a system. Functions like unused interfaces must permanently deactivated. Permanent means that they must not be reactivated again after system reboot.

Motivation: The hardware or software of a system often contains functions which are not used and so will be a risk for system security. Such functions give an attacker the possibility to manipulate the system. Furthermore it is possible to get unauthorized access other areas or data of the system. An example is a debugging function in software which can be used for troubleshooting but must not be activated during normal operation. Or a hardware interface that will not be used and so is unsecured an allows possibly unauthorized access to the system.

For this requirement the following threats are relevant:

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

Req 3.42-6	Unused interfaces must be disabled.

Unused interfaces of a network device shall be disabled. It must be assured that interfaces remain inactive after a reboot.

Motivation: Unused interfaces are usually not taken into account in the configuration process of a network device. As a result, these interfaces are operated with the manufacturer's default configuration. This may enable an attacker who has direct physical access to such a network device to gain unauthorized access to the system or to networks connected to it.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Disruption of availability

6.3. System update

Req 3.01-5	Software and hardware components that are no longer supported by vendor, producer or developer must not be used.

Only those operating system, middleware and application software and hardware components may be used on a system which are supported by the vendor, the producer, the developer or other contractual partner of Deutsche Telekom AG. Components that have reached end-of-life or end-of-support must not be used. Excluded are components that have a special support contract. This contract must guarantee the correction of vulnerabilities over components lifetime.

Motivation: Hardware and software components that have reached end of life or end of support represent a risk for a system. This means that a vendor does not supply remedial updates or patches for a component should errors or vulnerabilities occur. This means that vulnerabilities cannot be fixed when they occur and could be exploited to compromise the system or to impair its availability.

For this requirement the following threats are relevant:

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

Req 3.01-6	Known vulnerabilities in software and hardware of the system must be fixed or protected.

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. Furthermore it is a ongoing process during the complete life cycle of the system to fix upcoming vulnerabilities promptly. Time demands to correct vulnerabilities will be released by Telekom CERT in their advisories (https://cert.telekom.de).

Motivation: Publication of vulnerabilities increases the risk of successful exploitation by an attacker, especially since the published information usually includes details on how to exploit the vulnerability and tools that make exploitation possible. One example of this is when Web servers are compromised due to not patched vulnerabilities.

For this requirement the following threats are relevant:

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

6.4. Protecting data and information

Req 3.01-7	Data with need of protection must be secured against unauthorized viewing and manipulation during transmission and storage.

Adequate security measures for transmission and storage must be implemented of data with a need for protection that are classified as internal, confidential or strictly confidential. The chosen measure depends on the classification for the data and other factors such as the type of network used during transmission, the storage location for data, etc. Assistance for the classification of data gives the Intranet page http://informationwheel.telekom.de. Furthermore must be guaranteed that confidential and strictly confidential data will not be unprotected during temporary storage (e.g. in web cache, temporary folders).

All authentication data such as user names, passwords, PINs, etc. must be protected against unauthorized viewing and manipulation. This applies equally to permanent storage and transmission. Typical measures taken to protect authentication data or combinations of these are:

For storage:

- Client systems: encryption or obfuscation of authentication data, no persistent storage, limitation of access rights.

- Server systems: hashing of authentication data with PBKDF like scrypt or bcrypt or when this is not possible hashes with salt.

For transmission:

- Usage of cryptographically protected network protocols.
- Challenge response method (Disadvantage of this solution is that passwords needed in clear-text on server. If this solution is feasible depends on the individual threat scenario)

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 the system access this kind of files. An example is the use of checksum or cryptographic methods to validate if e.g. firmware images, patches, drivers or kernel modules are free of manipulations.

For transmission of data with a need of protection it is necessary to use network protocols with insufficient security measures. Examples for these insecure protocols that should not be longer used are: SSLv3, SSHv1, FTP, Telnet, SNMPv1 and 2c. In case of these protocols a newer version without vulnerabilities or a secure alternative must be used.

Motivation: If data with a need of protection will not be 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 when passwords can be read out when they stored in an inadequate secured way (e.g. usage of unsecure hashing algorithms like MD5 or SHA-1) or even without encryption on a system or manipulation of firmware to affected the system integrity.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data

Req 3.42-7	Encrypted protocols must be used for management.

Access to management services may only take place by means of secure protocols (e.g., SSHv2, HTTPS or SNMPv3). This is necessary because when accessing management services of a network device, data requiring protection such as user names, passwords or configuration data is transmitted. In addition, the use of encrypted protocols is also necessary for the transmission of new operating system versions and for updates and patches, etc.

Motivation: When plain text protocols such as Telnet, HTTP, FTP, TFTP or SNMP (version 1 and 2) are used for the management of network devices, an attacker record and manipulate data or posibbly highjack the current session. In his next step, he can then use the information gained for attacks on the network device.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Unauthorized modification of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

Req 3.01-8	Information with need of protection must not be contained in files, outputs or messages that are accessible by unauthorized users.

Information with need of protection must not be accessible in files, outputs or messages of the system by unauthorized users. This includes information relating to the operating system, used middleware or applications such as vendor, product name, product identifier, installed software versions, installed service packs, patches, hot fixes and serial numbers. Examples for system messages which must be free of sensitive data are:

- Comments in downloadable files
- Error and system messages
- Stack traces
- Network protocols

- Login windows and dialogs

Furthermore, details of implementation and information relating, e.g., to backend software/systems, function calls, SQL instructions or structure of database, must not be contained in error messages.
Excluded from this are displays and outputs that can be viewed and retrieved by authorized users who are logged in. In addition, an internal transfer of system internal information for error analysis is allowed in an adequate dimension. In this case the continuative regulations or guidelines (e.g. of data privacy) must be noticed.

Motivation: The information named above can be used by an attacker to prepare specific attacks on a system. In this way an attacker could, for example, use the precise software version to identify vulnerabilities in the product and, in a second step, exploit them.

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

6.5. Protecting availability and integrity

Req 3.01-9	The system must be robust against overload situations.

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 application
- Restricting of maximum sessions for a Web application
- Defining the maximum size of a dataset
- Restricting CPU resources per process
- Prioritizing processes
- Limiting of amount or size of transactions of an user or from an IP address in a specific time range

Motivation: An attacker can effect systems availability through targeted exploitation of vulnerabilities in resources administration. An attacker can, for example, send bulk queries to a Web server and, by doing so, compromise its availability if the maximum number of permissible Web sessions is not restricted.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-10	If an overload situation cannot be prevented, the system must act in a predictable way.

A system must be built in this way that it can react on a overload situation in a controlled way. However it is possible that a situation happens where the security measures are not longer sufficient.

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.

Motivation: With denial-of-service attacks can an attacker try to overload a system to effect its availability or integrity. Unpredictable acting of the system is then a risk for functionality and data and possibly also for other systems.

For this requirement the following threats are relevant:

- Disruption of availability

Req 3.01-11	The system must be robust against unexpected input.

During transmission of data to a system it is necessary to validate this before processing. This includes all data which

are send to the system. Examples for this are user input, values in arrays and content in protocols. 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
- White listing or escaping for inputs outside the values margin

Motivation: An attacker can try to put a system in an unsecure state through targeted manipulation of transmitted data. The object of such an attack is to compromise the usability, availability or integrity of individual services or of the entire system. For instance a unclean memory handling can lead to a buffer overflow that allows an attacker to execute arbitrary code on the effected system.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Disruption of availability

Req 3.42-15	Manipulated packets, that are sent to an address of the network device, must not lead to an impairment of availability.

A network device shall be not effected in its availability or robustness by packets that are manipulated or differing the norm. This means that appropriate packets must be detected as invalid and be discarded. The process shall not be effect the performance of the network device. This robustness must be just as effective for a great mass of invalid packets as for individual or a small number of packets.

Examples of such packets are:

- Mass-produced TCP packets with a set SYN flag to produce half-open TCP connections (SYN flooding attack)
- Packets with the same IP sender address and IP recipient address (Land attack)
- Mass-produced ICMP packets with the broadcast address of a network as target address (Smurf attack)
- Fragmented IP packets with overlapping offset fields (Teardrop attack)
- ICMP packets that are larger than the maximum permitted size (65,535 Bytes) of IPv4 packets (Ping-of-death attack)

Sometimes the relevant behaviour of the network device must be configured. In other cases, the behaviour of the network device may only be verified by the relevant tests.

Motivation: An attacker can use tampered packets to perform so-called denial-of-service attacks, in order to impair the availability of the network device as a whole or in part. Sometimes it only requires individual packets, or a few of them, to make a vulnerable network device crash.

For this requirement the following threats are relevant:

- Disruption of availability

6.6. Authentication and authorization

Req 3.01-12	System functions with a need of protection must not be used or accessed without successful authentication and authorization.

The usage of a system functions or access of data classified as internal, confidential or strictly confidential must only be possible unambiguous user identification and successful authentication on basis of the user name and at least one authentication attribute. Excepted from this are functions for public use such as those for a Web server on the Internet,

via which information is made available to the public. Examples for functions which require a prior authentication are network services (like SSH, SFTP, Web services), local access via a management console, local usage of operating systems and applications. The following examples are possibilities that could be used for authentication.

- Query user name and password
- Use of cryptographic keys and certificates (e.g. as Smartcard)

This requirement must also be applied to accounts that are only used for communication between systems (M2M).

Motivation: The authentication is necessary to doubtless identify a user because the allocated authorization, and therefore the access on data and services of the system depends on that.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-13	Accounts must be used that allow unambiguous identification of the user.

Users must be identified unambiguously by the system. This can typically be reached by using a unique account per user. So named group accounts, i.e. the use of one account for several persons, must not be used. On exception of this requirement are so named machine accounts. These will be used for authentication and authorization from system 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. In this connection, it has to be guaranteed that this account can't be misused. Possibilities to protect these accounts are:

- Configuring of a Password that fulfils the security requirements and is known by less than possible circle of administrators.
- Configuring the account that only a local use is possible and a interactive login isn't possible.
- Use of a technique for authentication of the specific account with public and private key or certificates.
- Limiting the access over the network for legitimised systems.

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

Motivation: Unambiguous user identification is a prerequisite for assigning a user the rights that he requires to perform his 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 use of services or resources
- Denial of executed activities

Req 3.01-14	Accounts must be protected against unauthorized use by at least one authentication attribute.

The various user 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 keys
- Token
- Passwords
- PINs

This means that authentication based on a parameter that can be spoofed (e.g. phone numbers, public IP addresses or VPN membership) is not permitted. Exceptions are attributes that cannot be faked or spoofed by an attacker. Sever-

al of the above options can be combined (dual-factor authentication) to achieve a higher level of security. Whether or not this is suitable and necessary depends on the protection needs of the individual system and its data and must be evaluated for individual cases.

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts that are not protected with a secret authentication attribute can be used 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 use of services or resources
- Denial of executed activities

Req 3.01-15	Accounts with extensive rights must be protected with two authentication attributes.

Accounts, for example used for administration, maintenance and troubleshooting, have extensive rights. Therefore a single protection (e.g. a password), as for normal user accounts with less rights, is not suitable. To get a higher protection level it is necessary to use more than one authentication attribute. For this a combination of an attribute that the user knows and an attribute that the user owns will be used often. This kind of authentication will be named as 2-factor authentication. Examples for 2-factor authentication are:

- Smartcard (e.g. MyCard) with PIN
- Private key with Passphrase
- Secure-ID Token with Password

In areas of Deutsche Telekom AG where the MyCard is available this solution should be preferred for internal projects and systems.

Motivation: Accounts with extensive rights as used for system administration have a higher risk for system's security. An attacker can get extensive rights by compromising such an account to get access to wide parts of the system and stored data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-16	Predefined accounts must be deleted or disabled.

Many systems have default accounts (e.g. guest, ctxsys), some of which are preconfigured with or without known passwords. These standard users must be deleted or disabled. Should this measure not be possible the accounts must be locked for remote login. In any case disabled or locked accounts must configured with a complex password (12 character and more, use of upper/lower case, numbers and special characters). This is necessary to prevent unauthorised use of such a account in case of misconfiguration.

Exceptions to this requirement to delete or disable accounts are accounts that are used only internal on the system involved and that are required for one or more applications on the system to function. Also for this accounts remote access or local login must be forbidden to prevent a abusive use by users of the system.

Motivation: Standard users are typically generally known and can be used by an attacker for targeted brute force and dictionary attacks. Standard user accounts represent a special risk if they do not use a password or only use a standard password that is generally known. Such standard user accounts can easily be exploited by an attacker in order to gain access to the system involved without being authorized to do so.

For this requirement the following threats are relevant:

- Unauthorized access to the system

- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-17	Predefined authentication attributes must be deleted or disabled.

Normally, authentication attributes such as password or cryptographic keys will be preconfigured from producer, vendor or developer of a system. Such authentication attributes must be changed an own attribute not known by other parties.

Motivation: Authentication attributes like password or cryptographic keys preconfigured from third parties are not trustable. Such authentication attributes can be used to compromise systems or their data.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-18	The authorizations for accounts and applications must be reduced to the minimum required for the tasks they have to perform.

Authorizations to a system must be restricted to a level in which a user can only access data and use functions that he needs in the course of his work. Suitable authorizations must also be assigned for access to files that are components of the operating system or of applications or that are generated by the same (e.g. configuration and logging files).

Alongside access to data, execution of applications and components must also take place with rights that are as low as possible. Applications should not be executed with administrator or system rights.

Motivation: If the rights granted to a user on a system are too broad, it could be possible for him to access data and applications which he is not permitted to view or use. This would give him 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 his own authorizations and thus to gain access to files and system components to which he would not have had access with his authorizations under normal circumstances.

For this requirement the following threats are relevant:

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

Req 3.01-19	The system must be connected to a central system for user administration.

Accounts and their rights will be administrated on central identity management systems (e.g. cIAM, WiW, ZAM) in Deutsche Telekom Group. For provisioning of these authorization information must the system provide a central interface (e.g. LDAPs for authorization, Kerberos for authentication, locking information for certificates) or decentralized mechanisms (e.g. public-key authentication) . A central solution for identity management must be preferred.

In areas where a central identity management system is not available a central system such as LDAP, TACACS+ or Radius server for the administration of accounts and their authentication and authorization must be used.

Motivation: Central administration of identity of accounts and their rights means that they only have to be maintained once instead of separately on each system. From the aspect of security, the advantage is that an account and its rights only known on a single central side. This information can be transmitted from a central side to systems (provisioning), central administrated (reconciliation) and central deleted (deprovisioning). This reduces the risk of accounts being forgotten during changing or deletion since they are configured on multiple systems. This could give a user wrong system rights or continued access to a system.

For this requirement the following threats are relevant:

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

6.7. Protecting sessions

Req 3.01-20	Sessions must be protected against high jacking.

A function must be implemented for all user sessions to prevent other users assuming or pursuing control of the sessions of an authorized user. Such protection can be achieved by implementing the following measures or a combination of these measures:

- Using an appropriate cryptographic algorithm
- On network level: Use of TCP protocol (with sequence number) and filter lists.
- On transport level: SSL/TLS
- On application level: Negotiation of a random, secret value between sender and recipient (e.g. session ID, sequence number, time stamp)

Motivation: An attacker who gains knowledge about a session which is not protected against assumption of control can continue to participate in such a session and thus gain unauthorized access to the system involved. One example is an attacker who records the session cookie for a Web application during non-encrypted communication. The attacker can then view and take control of the session of the original 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

Req 3.01-21	It must be possible that users can logoff their session.

The system must have a function that allows a signed in user to logout at any time.

Motivation: A user of a system must have the possibility to protect a session and therefore its data against unauthorized access. Therefor a logout function must be available that can be used to end a session. This prevents that the current session will stay open and can be reactivated and used by an unauthorized 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

Req 3.01-22	Session must be terminated after an adjusted period of inactivity for the purpose of use.

It is necessary that a session will be terminated automatically after a specified time of inactivity. For this reason a time-out must be set for sessions. The selected period depends on use and if applicable the physical environment. This means for example that a time-out for an application in an unsecure environment must be shorter (less minutes) than the time out for an application used of operational staff for monitoring tasks used in an protected area (60 minutes and

longer).

Motivation: In case of an open or unused session exists the possibility that an unauthorized user can high-jack and use this session to get access to the effected system and its data. Furthermore open session will allocate system resources.

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

6.8. Authentication parameter password

Req 3.01-23	If a password is used as an authentication attribute, it must have at least 8 characters and contain three of the following categories: upper cases, lower case, numbers and special characters.

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

- Minimum length of 8 characters.
- Comprising at least three of the following categories: upper/lower case letters, numbers and special characters

When a password is assigned, the system must ensure that the password meets these requirements. If a central system will be used for user authentication this function can be forwarded or delegated to this system.

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

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-24	If a password is used as an authentication attribute, users must be enable 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 will be used it is possible 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 use of services or resources
- Denial of executed activities

Req 3.01-25	If a password is used as an authentication attribute, a protection against brute force and dictionary attacks that hinder password guessing must be implemented.

Brute force and dictionary attacks aim to use automated guessing to ascertain passwords for user and machine accounts. Various measures or a combination of this measures can be taken to prevent this.

The most commonly used protection measures are:

- Increasing the delay (e.g. doubling wait times at each attempt) for each renewed password input following an incorrect entry ("tar pit").
- Blocking an account following a specified number of incorrect attempts (typically 5). However has to take in account that this solution needs a process for unlocking and an attacker can force this to deactivate accounts and make them unusable.
- Using CAPTCHA to prevent automated attempts (often used for Web applications).

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: Implementation of one or more of the above measures can prevent successful execution of brute force and dictionary attacks to ascertain passwords.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

Req 3.01-26	If a password is used as an authentication attribute, they must be hidden when displayed on screen.

When a user enters a password or the password is otherwise displayed on screen, it must be made unreadable. Typically, the individual characters of the password are replaced by a character such as "*". Under certain circumstances it may be permissible for an individual character to be displayed briefly during input. Such a function is used, for example, on Smartphone's to make input easier. However, the entire password is never output to the display in plaintext.

Motivation: To prevent another person reading a password by chance or intentionally on screen during input, the password must be rendered unreadable during input or display.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized use of services or resources
- Denial of executed activities

6.9. Logging

Req 3.42-17	Security-relevant events must be logged with a precise time stamp and a unique system reference.

Network devices must log the occurrence of security-relevant events. So that these events can be evaluated and classified, they must be logged together with a unique system reference (e.g., host name, IP or MAC address) and the exact time the event occurred.

When logging, the applicable statutory, collective agreement and operating provisions must be taken into account; these include the statement that the logging of events may only take place for the intended purpose. Logging events in order to exploit these for the control of employees' work is not permitted.

The following security-relevant events must be logged by a network device:

Event	Event data to be logged

Failed login attempts	• Account, • No. of failed attempts, • Source (IP address) of remote access
Changes to configuration	• Change made, • User
Reboot/shutdown/crash	• Action performed (reboot, shutdown, etc.), • User (for intentional actions)
Change to the status of interfaces (e.g., shutdown)	• Interface name and type, • Status (shutdown, missing link, etc.´)
Critical rise in system values of memory or CPU load over a longer period	• Value exceeded, • Value reached (Here suitable threshold values must be defined depending on the individual system.)

Suitable thresholds must be defined depending on the system type and hardware used. Logging of additional security-relevant events may be appropriate. This is to be verified in individual cases and implemented accordingly where required.

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 action can be taken against attackers.

For this requirement the following threats are relevant:

- Denial of executed activities
- Unnoticeable feasible attacks

Req 3.01-28	Security relevant logging data must be send to an external system direct after their creation.

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.

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:

- Denial of executed activities
- Unnoticeable feasible attacks