Security requirement

Hyper-V Servers

Deutsche Telekom Group

Version	6.0
Date	Dec 1, 2023
Status	Released

Publication Details

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

Reuterstrasse 65, 53315 Bonn
Germany

File name	Document number	Document type
	3.49	Security requirement
Version	**State**	**Status**
6.0	Dec 1, 2023	Released
Contact	**Validity**	**Released by**
Telekom Security psa.telekom.de	Dec 1, 2023 - Nov 30, 2028	Stefan Pütz, Leiter SEC-T-TST

Summary
Hyper-V Servers

Table of Contents

1. Introduction

This security document has been prepared based on the general security policies of the group.

The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for units which do not participate in the PSA process. These requirements shall be taken into account from the very beginning, including during the planning and decision-making processes.

When implementing these security requirements, the precedence of national, international and supranational law shall be observed.

2. Dependencies

Most of the requirements described in this document result from the general security requirements for virtualization solutions, which are specified in the Security Requirement “3.35 Virtualization Solutions”. In addition, the requirements from the Security Requirement 3.37 on Operating Systems and, in general, all requirements specified in the document entitled “3.1 Technical Baseline Security for IT/NT Systems” ans 3.14 "Architectures for network services and data centers" shall be observed for virtual machines (guests). The requirements in these documents are binding for Hyper-V servers.

3. System hardening

Req 1	Unnecessary services must be disabled.

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-5/7.0

Req 2	The accessibility of activated services must be restricted.

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-6/7.0

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

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-3/7.0

Req 4	The software used must be obtained from trusted sources and checked for integrity.

The software used on the system must be obtained from trusted sources and checked for integrity before installation.

This requirement applies to all types of software:
- Firmware and microcode for hardware components
- Operating systems
- Software Libraries
- Application Software
- Pre-integrated application solutions, such as software appliances or containers

as well as other software that may be used.

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

Integrity Check
The integrity check is intended to ensure that the received software is free of manipulation and malware infection. If available, the mechanisms implemented by the supplier must be used for checking.
Valid mechanisms are:
- physical seals or permanently applied certificates of authenticity (if the software is provided on physical media)

- comparison of cryptographic hash values (e.g. SHA256, SHA512) of the received software against target values, which the supplier provides separately
- verification of cryptographic signatures (e.g. GPG, certificates) with which the supplier provides its software

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

Extended integrity checking when pulling software from public registries

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

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

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

Suitable methods for an extended integrity check can be, for example:

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

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

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

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

For this requirement the following threats are relevant:

- Unauthorized modification of data
- Unnoticeable feasible attacks

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-2/7.0

Req 5	The Windows Server Core installation mode must be used.

Motivation: Windows Server Core is an installation option in Windows Server which significantly reduces the area of attack, since it installs no graphical user interface (GUI) and only the most urgently required system files and services. Another advantage is the usually lower number of updates/patches and the associated downtime.

Implementation example: The Windows Server Core operating mode must be selected right at the start during installation Windows Server 2008 or Windows Server 2008 R2, since it cannot subsequently be changed. With Windows Server 2012 or later it is possible to change from core to full installation afterwards.

Alternatively, the installation from Windows Server 2012 can be done in "Graphical management tools and infrastructure" (Minimal Server Interface) mode. If it is foreseeable, that no local adjustments are required, you can change to the favorite manufacturer core mode.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-5/6.0

Req 6	Devices, software and hardware components that are not needed by the virtual machine must be disabled on the virtualization server.

In order to minimize the attack surface of the virtual machines, the virtual hardware or devices (e.g. virtual network cards, drivers) that are not needed for operation shall be disabled or deleted for the virtual machines on the virtual host system.

Motivation: Each additional device offers additional points of attack on the virtual system.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-10/6.0

Req 7	The Hyper-V host must be configured and must only have the necessary functions for the request.

Motivation: In order to minimize the influence that Hyper-V system devices have on the virtual machines, these devices or software components shall be disabled. Moreover, unused hardware functions and the associated installed drivers constitute an increased area of attack.

Implementation example: Deactivation of the CD/DVD drive function on the Hyper-V system in the device manager or deactivation of not used virtual network cards.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-7/6.0

Req 8	The Hyper-V system must not be installed with a server role other than Hyper-V or an additional application.

During the installation of Windows Server 2008 it is technically possible to install additional server roles or additional applications or administration tools such as "virtual machine manager" and execute them in the parent partition of Hyper-V system.

Motivation: Each network or storage access (I/O access) is routed through the parent partition. If the parent partition must execute other services, the virtual machines do not have access to the necessary resources. Since the parent partition in the Hyper-V system has a privileged position and controls via the VMBus the entire incoming and outgoing data traffic of all virtual machines operated on the system, running another application constitutes an additional area of attack that can be compromised on the virtual machines.

Implementation example: When installing Windows Server and Hyper-V select no other server role other than the server role Hyper-V and do not install any other application (exceptions for operational reasons can be software agents for software distribution, performance monitoring, asset management, anti malware, etc.)

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-8/6.0

Req 9	The “Security Best Practices” and the Microsoft hardening policy for Hyper-V servers must be implemented.

If hardening baselines from internal security organizations of the Deutsche Telekom Group are available, should the Hyper-V systems are hardened according to these specifications. If appropriate hardening baselines should not be present, a corresponding hardening concept based on the "Security Best Practices" is to be made.

Motivation: Implementing the best practice settings from manufacturers or other organisations that are adapted to Deutsche Telekom’s need will result in security settings that are based on years of practical experience. Settings are also made that may not have been considered as part of an individual project since they do not have any obvious or active relevance in the project.

Implementation example: Compliance with the Microsoft TechNet configuration notes and instructions at http://technet.microsoft.com/de-de/library/dd283088(WS.10).aspx, in the Microsoft TechNet wiki at http://social.technet.microsoft.com/wiki/contents/articles/151.aspx, the applicable settings of the Microsoft Windows Server 2008 Hyper-V Common Criteria Guides at http://www.microsoft.com/download/en/details.aspx?id=14252, the Hyper-V Security Guides at http://technet.microsoft.com/en-us/library/dd569113.aspx and the activation of Specialized Security, Limited Functionality (SSLF).

Tools like the Microsoft Security Compliance Manager (http://technet.microsoft.com/enus/library/cc677002.aspx), the Microsoft Best Practice Analyzer (http://technet.microsoft.com/dede/library/dd759260.aspx), the Microsoft Baseline Security Analyzer (http://technet.microsoft.com/enus/security/cc184924) etc., can further provide information on the security of the Hyper-V-System.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-9/6.0

Req 10	The integrity of the virtualization software (hypervisor) must be checked before execution.

The hypervisor is a critical component and should be protected from manipulation. Regular checks at system reboot helps to detect changes and manipulation before they can take place. The usage of hardware supported technologies should be checked.
To be sure that the integrity of the software is certified before execution (installation), the installation media should be downloaded only from trustworthy sources (the manufacture website) and be checked by hashes (e.g. MD5, SHA256).

Motivation: Manipulations of the virtualization software affect all virtual machines running on this layer. Therefore, the hypervisor should be regularly checked and specially protected.

Implementation example: Such checks (file integrity monitoring) could be done with appropriate tools such as OSSEC or with support of hardware solutions such as Trusted Platform Module (TPM). The support of the hypervisor for the corresponding solution should be confirmed in advance by the manufacturer.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-9/6.0

Req 11	The Hyper-V system must be protected against tampering and to preserve the integrity before and during execution.

Motivation: Tampering with the virtualization layer (Hyper Jacking, Rogue Hypervisor and SubVirt Virtual Machine Based Root Kit) and with the parent partition has an impact on all virtual machines run on the Hyper-V system.

Implementation example: Setting restrictive access permissions to the aforementioned directories in the Hyper-V Attack Surface Reference Workbook using ACLs (http://download.microsoft.com/download/8/2/9/829bee7b-821b-4c4c-8297-13762aa5c3e4/Windows%20Server%202008%20Hyper-V%20Attack%20Surface%20Reference.xlsx). An analysis of the server logs (monitoring) via System Center Operations Manager can also support the integrity of the

Hyper-V system (http://technet.microsoft.com/enus/library/ff621102.aspx, chapter “Monitor Hyper-V health and performance”).

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-11/6.0

Req 12	The communication channel from virtual machine to the Hyper-V host and conversely must be restricted.

Hyper-V provides the "Integration Services" for the virtual machine specially equipped drivers for synthetic I / O devices on VMBus can achieve so the hardware access by the virtual machine a performance comparable to physical servers. Other guest operating systems that do not have Enlightenment for the Hyper-V and the VMBus, communicate with the emulated devices on Hyper-V system.

Motivation: The direct interface between virtual machines and the virtualization server is particularly critical, since it is difficult to control. It offers attackers additional points of attack against the virtualization server and shall therefore be restricted.
This interface also allows hypervisor administrators to access the guest machines, even if they do not have administrator permissions on the guest system. Additionally this interface might be misused by attackers. Actions establisehd through this interface are not noticed within basic logging functions of the guest machines. Guest machines should be adminsitrated similar to bare metal machines.

Implementation example: There should only be virtualized guest operating systems on Hyper-V system, which are listed under http://technet.microsoft.com/en-us/library/cc794868%28WS.10%29.aspx be offered or certified by Microsoft for the guest additions (Integration Services). Additionally, all security updates are installed quickly to fix the security holes in VMBus.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-12/6.0

Req 13	Virtualization solutions must be protected in such a way, that they do not circumvent any security mechanisms.

In virtualization solutions security mechanisms shall not be circumvented.

Other examples or keywords for this requirement are: local firewall, IDS/IPS, proxy restrictions, local ACLs, 'enforce to safe data encrypted on a external device'.

Motivation: Increase security by using all the existing security interfaces without bypassing them by shortening the *communication paths.*

Implementation example: This means, for example, firewalls that exist between two networks, however, could be circumvented by a direct communication between virtual machines on local interfaces.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-12/6.0

4. System update

Req 14	The Hyper-V system must be operated within the support service offered by the manufacturer.

The Hyper-V system should be operated within mainstream support and must not be operated beyond the end of extended support. A Hyper-V system, which is covered by extended support, must no longer be utilized for newly designed production systems.

Motivation: Older Hyper-V systems provide a greater area of attack due to new attack possibilities and technical vulnerabilities since the vendor no longer provides any patches/bugfixes which resolve a potential problem.

Implementation example: For Hyper-V Server 2008 R2, mainstream support ends on January 14, 2014. Up to the end of extended support on January 08, 2019, the Hyper-V system must be migrated to a product in mainstream support.

See: http://support.microsoft.com/lifecycle/?ln=us.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-14/6.0

Req 15	Known vulnerabilities in the software or hardware of the system must be fixed or protected against misuse.

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

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

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-10/7.0

Req 16	Anti-malware software must be operated on the Hyper-V system.

Motivation: Malicious code compromises the Hyper-V system and jeopardizes the security as well as the stability of the entire system. An installed anti-malware program can further reduce this area of attack.

Implementation example: Installation of a feasable anti malware software on a Hyper-V system (host).

To ensure that a Hyper-V system can be operated smoothly with installed anti-malware software, the following directories must be excluded from the virus scan (real time and scheduled):

- Virtual Machine configuration files
    - Default “C:\ProgramData\Microsoft\Windows\Hyper-V”
- Virtual Machine VHD files
    - Default “C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks”
    - Default by a Cluster ”C:\ClusterStorage"
- Snapshot files
    - Default “C:\ProgramData\ProgramData\Microsoft\Windows\Hyper-V\Snapshots”
- Virtual Machine prozesses
    - Vmms.exe
    - Vmwp.exe

Note: With Microsoft Forefront Endpoint Protection (FEP) it is very easy to resort to a predefined policy.

See: http://technet.microsoft.com/en-us/library/gg412475.aspx.

Note: If the default directory paths during the installation or in the subsequent configuration are changed, the one selected in the anti-malware software must be adapted accordingly.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-16/6.0

Req 17	Switched-off virtual machines must be provided with the latest version of the operating system, the installed applications and the latest anti-malware pattern.

Virtual machines which are offline cannot be updated using automatic update mechanisms such as Windows Update or Windows Software Update Services (WSUS).

Motivation: A virtual machine that has been switched off for a long period may be susceptible to attacks due to the virtual machines not being up-to-date, and thus may compromise other virtual machines, network drives, etc.

Implementation example: Microsoft's "Offline Virtual Machine Servicing Tool" integrates System Center Virtual Machine Manager (SCVMM) with the Windows Server Update Services or the ConfigMgr patching infrastructure. It automatically launches a software update cycle for switched-off virtual machines, which are also again set offline following the update.

For this requirement the following threats are relevant:

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-17/6.0

5. Protecting data and information

Req 18	The system partition of virtual machines and copies must be encrypted.

Virtual machines are often stored in files on the virtualization server or SAN. It shall be ensured that no shadow copies or snapshots exist that are not encrypted. The encryption of the data partition of a virtual machine is defined by the data security level stored on the partition.

Motivation: It must be prevented that data can still be read form copies.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-19/6.0

Req 19	The system and administration partition of a virtual machine including all files (copys, snap shots, etc.) must be encrypted.

A virtual machine consists of a series of files, including the VHD/VDMK files and VHD/VDMK configuration files. The content of the virtual RAM is also stored in the VHD files – depending on the scenario – when storing the virtual machines. This means that need of protection content such as passwords or hashes, which are normally maintained encrypted on the hard disk, may be stored in unencrypted format in the files for the virtual machine (disk as state information).

Motivation: Compromising the files of the virtual machines may provide a potential attacker with knowledge of need of protection information by gaining access to unencrypted data from the virtual RAM or being able to extract need of protection data on the virtual hard disk. Encryption ensures that attackers cannot circumvent this kind of encryption by booting the system from another operating system or using specific attack tools.

Implementation example: Dividing up the virtual machine into a system or administration partition and data partition. Encryption of the system or administration partition in the virtual machine using Microsoft tools or 3rd Party Sofware liek Tren Micro SecureCloud, Veeam Backup& Replication, Altaro Hyper-V Backup, etc..

Note: Important! Encrypting File System (EFS) cannot be used to encrypt folders in which files from the virtual machine are stored. Hyper-V does not support the usage of storage media if EFS has been used to encrypt the VHD/VDMK file.

Note: The encryption of the data partition will be defined by the classification in accordance with the Group Policy IT-/NT-Security and must be transposed separately. This is not part of this security requirement.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-19/6.0

Req 20	Access to virtual machines and the associated files must be regulated by means of Access Control Lists (ACLs).

A virtual machine consists of a set of files. These include the virtual disks in which the entire software environment resides – in other words the guest operating system, the applications that are installed and run on that system as well as the associated data. Added to which are other configuration files that define the basic characteristics and resources of a virtual machine. These include, for instance, the number of virtual processors, the RAM capacity as well as the associated input/output options for the virtual machine. The content of the virtual RAM is also stored in the VHD files – depending on the scenario – when storing the virtual machines. This means that need of protection content such as passwords or hashes, which are normally maintained encrypted on the hard disk, may be stored in unencrypted format in the files for the virtual machine (disk as state information).

Motivation: Compromising the files of the virtual machines may provide a potential attacker with knowledge of need of protection information by gaining access to unencrypted data from the virtual RAM or being able to extract need of protection data on the virtual hard disk.

Implementation example: Access Control Lists (ACLs) can be used to restrict access to the respective files on the virtual machine to authorized users only.

Note: In order to ensure secure separation, the ACLs should be integrated in the software switch. Where this does not occur, two virtual machines that are installed on the same physical server may not be fully isolated from each other. If someone has control of virtual machine 1, this person could also access virtual machine 2 and possibly steal data.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-20/6.0

Req 21	The virtualization software must prevent virtual machines sending data if its layer-2 address (MAC) changes and prevent virtual machines being able to receive data with faked layer-2 addresses (MAC).

Motivation: This prevents a virtual machine obtaining an IP address via DHCP, for example, which has firewall activation permissions, without having authorization to do so.

Implementation example: A cluster virtual machine for higher availability does not have to be compliant with this requirement.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-21/6.0

Req 22	The Hyper-V System must prevent virtual machines from receiving packets if its layer-2 address (MAC) changes.

In order to prevent packets from a virtual machine being read without authorization, the receipt of packets by changing the MAC address on the virtual machine shall be prevented. This is especially true for installations with several connected Hyper-V systems. A single Hyper-V system has implemented security measures at the double MAC address assignment or prevent a takeover by another virtual machine, this security measure does not work over several Hyper-V systems.

Motivation: This prevents a virtual machine assuming the MAC address of another virtual machine and receiving network traffic without authorization to do so.

Implementation example: Using System Center Virtual Machine Manager 2008 or higher (SCVMM). SCVMM manages a separate global pool of MAC addresses for distribution to virtual machines, irrespective of the host system (Hyper-V, VMWare, etc.). By default, in the range of 00-1D-D8-B7-1C-00 to 00-1D-D8-F4-1F-FF a total of 3,998,719 MAC addresses are available. Web application firewalls such as the Microsoft Forefront Threat Management Gateway (http://www.microsoft.com/germany/forefront/edgesecurity/tmg/default.mspx) may be useful as an additional security authority.
Deactivating „Spoofing of MAC-Adresses“-Feature in Hyper V-Managers function "Network Adapter Settings"and deactivating „AllowMacSpoofing“ Feature via Windows Management Integration (WMI) and activating„ARP Spoofing Prevention“-Feature is advised too.

Note: For smaller environments can perform the allocation of static MAC addresses assigned to the virtual machine to the desired result.

Note: Hyper-V systems not mutually adjust the settings automatically. This also means that they do not match their active MAC address ranges. In other words, two Hyper-V systems may well use the same address range. Then, when an external network is connected, error messages are generated in the network. It is therefore advisable to check the address ranges for the installation of Hyper-V and corrected if necessary.

Note: The Hyper-V system does not change the MAC address of the virtual machine during operation. This also applies for a live migration. In a dynamic allocation of the MAC address of the virtual machine gets at every restart a different MAC address assigned. In a Linux system, e.g. created by default on every reboot a new network interface.

Note: In special usecases, for instance when using an NLB (Network Load Balancer)-Cluster in Hyper-V R2 environments, Problems may occur because NLB works with MAC Spoofing to distribute load to the cluster-knots. In this case „Spoofing of MAC-Adresses“-Feature could be re-activated if changes of configuration settings of the Hyper-V Systems are monitored with tools like Microsoft Systems Operation Manager. Aktivating „Spoofing of MAC-Adresses shold be used rarely, because it also opens the system for Unicast-Pakets caused by Flooding-Attacks.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-22/6.0

Req 23	The virtualization software must prevent virtual machines eavesdropping on packets in the network (promiscuous mode).

To prevent a virtual machine eavesdropping on network traffic, promiscuous mode shall be permanently disabled on all virtual machines. This setting should be centrally administered in the virtualization software of the host, to prevent a change by an administrative user of the virtual machine.

Motivation: This prevents a virtual machine eavesdropping on third-party network traffic.

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-20/6.0

Req 24	The Hyper-V system must prevent virtual machines eavesdropping on packets in the network (promiscuous mode).

To prevent a virtual machine eavesdropping on network traffic, promiscuous mode shall be permanently disabled on all virtual machines.

Motivation: This prevents a virtual machine eavesdropping on third-party network traffic.

Implementation example: Hyper-V does not support promiscuous mode of the virtual network adapters. Each virtual network adapter should be connected to a designated network and virtual LAN (VLAN) to isolate network traffic as far apart as possible.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-24/6.0

Req 25	An effective change of the VLAN ID is not allowed to be done by the virtual machine.

The virtualization software must assign a dedicated VLAN for each virtual machine (-group). The configuration of the VLAN must be done by the virtualization server.

This requirement does not apply if each network interface of a virtual machine (-group) is assigned a exclusive physical network interface.

Motivation: Virtual machines could gain unauthorized access to networks that they cannot normally reach.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-22/6.0

Req 26	The Hyper-V system must prevent VLAN hopping by virtual machines.

The Hyper-V system shall assign each virtual machine a dedicated VLAN. This requirement does not apply if each network interface of a virtual machine is assigned its own physical network interface.

Motivation: Virtual machines could gain unauthorized access to networks that they cannot normally reach.

Implementation example: The virtual network adapters for each virtual machine should be on a designated network and virtual LAN (VLAN) are bound to isolate network traffic from each other.

See also: http://technet.microsoft.com/en-us/library/ee236499.aspx

Note: Each virtual network adapter offers integrated virtual networks to (VLANs), you can order a unique VLAN channel will be assigned.

Each virtual machine can be configured with up to 12 virtual network adapters—8 can be the “network adapter” type and 4 can be the “legacy network adapter” type.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-26/6.0

Req 27 Outputs and messages must not disclose information on internal structures of the system.

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

In general, this concerns information on
- Product names and product identifiers of implemented system components
- Operating systems, middleware, backend software, software libraries and internal applications as well as their software versions
- installed service packs, patches, hotfixes
- Serial numbers of components as well as stored product licenses
- Database Structures

Typical examples of outputs and messages in which disclosure of such system information can potentially occur:
- Login windows and dialogs
- Error messages
- Status messages
- Banners of active network services
- System logs and log files
- Debug logs, stack traces

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

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

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-9/7.0

Req 28	Stored data in need of protection must be protected against unauthorized access, modification and deletion.

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

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

Basically, access to data in need of protection in a system must be fully regulated on the basis of technically implemented authorization assignments and controls.

If such technical access control alone is no longer sufficient to ensure the necessary protection requirements of stored data, or if its effectiveness cannot be consistently ensured, additional cryptographic methods (e.g. encryption, signing, hashing) must be implemented. Cryptographic methods used in the storage of data must be suitable for this purpose and must have no known vulnerabilities.

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-14/7.0

Req 29	Data in need of protection must be protected against unauthorized access and modification during transmission.

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

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

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

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

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

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

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

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

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

have been manipulated during transmission.

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-15/7.0

Req 30	Virtual machines that contain sensitive data must be securely deleted at deactivation.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-37/6.0

6. Protecting availability and integrity

Req 31	The system must be protected against overload situations.

A system must have protective mechanisms that prevent overload situations as far as possible.
In particular, a partial or complete impairment of the availability of the system must be avoided.

Examples of possible protective measures are:

- Limiting the amount of memory (RAM) available per application
- Limiting the maximum sessions of a web application
- Limiting the maximum size of a dataset
- Limiting CPU resources per process
- Prioritizing processes
- Limiting the number or size of transactions by a user or from an IP address over time

Note:
A system can usually not protect itself against network-based attacks with extremely high data or packet rates, the so-called "Distributed Denial of Service" (DDoS) attacks. To defend against DDoS attacks, an upstream solution in the network layer is required.

Motivation: Attackers can try to use up the resources of a system with targeted resource-intensive or large-volume requests, so that the system can no longer fulfill its regular tasks or intended task volumes and the availability of the services offered is effectively disrupted. Limiting the maximum resources that can be used per request made to the system is a fundamental measure to reduce the impact of such denial-of-service (DoS) attacks.

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-12/7.0

Req 32	In overload situations, the system must behave in a predictable manner.

Even comprehensive native protections may not be able to prevent a system from becoming overloaded in extreme situations.

It must therefore be ensured that, in overload situations, the system does not switch to a state that overrides security-relevant functions or properties of the system. Performance losses (e.g. the reduction of the throughput of legitimate network packets or the number of answered server requests per period) are usually unavoidable in overload situations, but the regular functional behavior of the system must be fundamentally preserved.

In extreme cases, this can mean that a controlled shutdown of the system is more acceptable than continued operation in the event of uncontrolled failure of the security functions and thus the loss of system protection.

Motivation: By means of a denial-of-service attack, an attacker can try to overload a system in a targeted manner. If such a system then reacts unpredictably or fails its regular behavior, especially with regard to its security functions, this can open up an extended attack surface for the attacker on functions and data of the system and potentially endanger other linked systems.

Implementation example: A firewall that discards its filter rules in overload situations and forwards all packets without checking would not meet the requirement. In this case, blocking all packets by shutting down the firewall would be more acceptable than failing their regular task of protecting downstream systems.

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-13/7.0

Req 33 The system must be implemented robustly against unexpected inputs.

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-11/7.0

Req 34 If an service level agreement (SLA) exists for a system, a violation of the SLA by overbooking of resources must be prevented.

The virtualization server must prevent, that virtual machines could overbook resources and affect other virtual machines to fall below the threshold defined in the SLA. All resources (CPU, RAM, hard drive, network) shall therefore only be assigned statically (within limits) to the virtual machines. Changes to bookings should only be made following an administrator review or up to a threshold value that prevents disadvantages to other instances.

Motivation: A virtual machine that is compromised or contains a malfunction or administration error shall not prevent other virtual machines fulfilling their respective functions.

Implementation example: A limitation of the maximum transmitted data over an defined interface can have such a desired effect.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-17/6.0

Req 35	A Hyper-V system must prevent virtual machines from overbooking resources, which would impair the Hyper-V system itself and other virtual machines.

All resources (virtual processors, I/O access, RAM, hard disk size, network, etc.) may only be assigned statically or with limited size to the virtual machines (server resource sizing). Changes to bookings should only be made following an administrator review or up to a threshold value that prevents disadvantages to other instances. In uncritically classified Hyper-V systems should be prevented, the resource can be oversubscribed.

Motivation: Compromising (hacking) or a malfunction must not prevent virtual machines or the Hyper-V system itself from fulfilling its function or sufficient resources for error-free operation being made available.

Implementation example: Two virtual CPUs should always be reserved as a minimum for the parent partition. The RAM and virtual hard disks (VHDs) can also be assigned dynamically (Dynamic Memory & Dynamic VHDs) to the virtual machines up to a fixed limit. Depending on the application, at least 2 to 4 GB RAM should be reserved permanently for the parent partition; this shall be stipulated in the Hyper-V system by means of a registry entry.

- Launch regedit.exe on the Hyper-V system and navigate to the following key: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Virtualization
- Create a new DWORD-type key and state as name “MemoryReserve” (with Hyper-V Server 2008 R2 SP1) or “RootMemoryReserve” (with Hyper-V Server 2008 R2 without service pack)
- The value to be reserved for the parent memory must now be entered in this key, e.g. 2048 MB (2 GB) RAM. You can then set the value to the static amount of memory that you want to reserve for the parent.
- Save and close regedit
- Reboot Hyper-V server

Note: Individual applications are only supported with a 1:1 allocation (1 complete physical process core) by their respective vendors (may apply to SAP, for instance). This must be taken into account during resource allocation.

Note: Reserving RAM must be taken into account particularly with cluster nodes to ensure that this is not overloaded in the case of a failover. For instance with a two-node cluster, a 50 percent reserve shall be envisaged, while the reserve to be earmarked is reduced to 33.33 percent with an additional cluster node. This also applies similarly to the CPU cores.

Note: Regardless of whether Dynamic Memory is involved or not – “overbooking” the RAM is not possible on a Hyper-V system! If the physical RAM of the Hyper-V system is fully booked and another virtual machine tries to boot up, the associated boot process fails without any clear message. This shall be taken into account particularly with live-migration scenarios as significant modifications to the resource allocation occur with such scenarios.

Note: The capacity limits can also be exceeded in a storage area network (SAN) where the Dynamic Disk is used. The data can also be fragmented faster with the use of Dynamic Disk. Therefore it is recommended solely configuring Fixed Disk.

Note: The swap file (pagefile) is one of the key elements of Windows memory management. The following rule applies to “general” server systems: Pagefile equals one and a half times RAM. With Hyper-V systems which tend to feature large amounts of RAM, with most of that RAM nonetheless never being swapped out since it is assigned to the virtual machines, the configuration is only required for debugging purposes. Since the host itself normally manages only between 2 and 4 GB RAM, you can use this figure as the basis for the host pagefile if disk space is scarce. Since however many Hyper-V systems are fitted with local hard disks for the host operating system and the virtual machines then store data in the SAN, the system specification can be adopted. Many administrators also move the swap file to a separate drive – this must then however be a dedicated physical hard disk, and on no account just a single partition or even a single drive on which there is another I/O load.

See: http://www.server-talk.eu/2011/01/31/wie-viel-virtual-memory-braucht-ein-hyper-v-host/

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-35/6.0

7. Authentication and authorization

Req 36	The use of system functions that require protection as well as access to internal or confidential data must not be possible without prior authentication and authorization.

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

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

Examples of features that require prior authentication include:

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

Examples of authentication features that can be used:

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-19/7.0

Req 37	User accounts must ensure the unique identification of the user.

Users must be identified unambiguously by the system.

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

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

A special feature are so named technical user accounts. These are used for the authentication and authorization of systems among themselves or of applications on a system and can therefore not be assigned to a specific person.

Such user accounts must be assigned on a per system or per application basis. In this connection, it has to be ensured that these user accounts can’t be misused.
Ways to prevent misuse of such user accounts by individuals include:

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-22/7.0

Req 38	User accounts must be protected with at least one authentication attribute.

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

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

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

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

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

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

Motivation: User accounts that are not protected by appropriate authentication attributes can be abused by an attacker to gain unauthorized access to a system and the data and applications stored on it.

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

- Denial of executed activities

For this requirement the following warranty objectives are relevant:

ID: 3.01-20/7.0

Req 39	Privileged user accounts must be protected with at least two authentication attributes from different factors.

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

Examples of privileged user accounts:

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

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

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

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

Motivation: Privileged user accounts represent an increased risk to the security of a system. If an attacker successfully compromises such a user account, he receives extensive authorizations with which he can bring the system or system parts under his control, disrupt system functions, view/manipulate processed data or influence business-critical processes. The combination of multiple authentication attributes of different types significantly minimizes the risk of a user account being compromised.

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

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

For this requirement the following threats are relevant:

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-21/7.0

Req 40 Predefined user accounts that are not required must be deleted or at least disabled.

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-7/7.0

Req 41 Predefined authentication attributes must be changed.

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-8/7.0

Req 42 The permissions for users and applications must be limited to the extent necessary to fulfill their tasks.

The permissions on a system must be restricted to such an extent that a user can only access data and use functions that he needs in the context of his work. Appropriate permissions must also be assigned for access to files that are part of the operating system or applications or that are generated by the same (e.g. configuration and logging files).

In addition to access to data, applications and their components must also be executed with the lowest possible permissions. Applications should not be run with administrator or system privileges.

Motivation: If a user is granted too far-reaching permissions on a system, he can access data and applications to an extent that is not necessary for the fulfillment of the assigned tasks. This creates an unnecessarily increased risk in the event of abuse, in particular if the user or his user account is compromised by an attacker.
Applications with too far-reaching permissions can be misused by an attacker to gain or expand unauthorized access to sensitive data and system areas.

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-23/7.0

Req 43 The administration of Hyper-V systems and all components must be separated from each other by different user roles.

In operational systems the administrator roles for a Hyper-V system shall be strictly separated from the administrator roles for virtual machines. Administrators of virtual machines shall not under any circumstances include the functions of the administrator for Hyper-V systems, or vice versa (least privileges).

Motivation: Through an available role-based security settings in a Hyper-V environment, a secure control of permitted operations can be guaranteed in the virtualized environment. This implements a separation of duties (SOD), as conditions can also be set in addition to the parameters for specifying a role in greater detail.

Implementation example: A user role defines the transactions (summarized in a profile) that are permitted for a selected group of objects (defined by the user role area). In this way, various roles in the category “Delegated administrator” can be created, e.g., for a general administrator that administers all transactions at a location, a specialized administrator to manage the library servers, as well as an advanced user who needs to set up complex virtual environments in a test laboratory. Self-service user roles can also be used to execute a defined series of transactions for one’s own virtual machines.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-43/6.0

Req 44	The operating system administrator of a Hyper-V system must be assigned to another role than the Hyper-V administrator.

An administrator of the operating system of a Hyper-V system shall be strictly separated from the functions of an administrator for the Hyper-V role (least privileges).

Motivation: Hyper-V constitutes a separate application within the server roles of a Windows Server operating system. To ensure secure separation of duties, the administrative activities to operate the server shall be separated from the administrative activities of the assigned server role. The administrator of the Hyper-V system has, depending on the configuration, administrator permissions to other Windows Server operating systems that do not carry out the Hyper-V role.

Implementation example: During installation, separate administrator groups must be set up to administer the operating system and to administer the Hyper-V role, and suitable users assigned. This can be stipulated by means of policies, for instance, in the Active Directory Service or by using the Authorization Manager (AzMan), a snap-in for the Microsoft Management Console (MMC), which is assigned to selected users and groups that belong to the Hyper-V role.

Note: A list of the 33 different functions that can be assigned to various other roles in addition to the three standard operations can be found in chapter 2 ‘Delegating Virtual Machine Management’. In the document “Hyper-V Security Guide” (http://www.microsoft.com/download/en/details.aspx?id=16650).

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-44/6.0

Req 45	Access by users with the role ‘Hyper-V administrator’ to virtual machines and the associated files must be restricted.

It is not that easy for a user with insufficient access permissions to access a virtual disk and to simply copy it. Nonetheless, with Microsoft’s Hyper-V each virtual machine runs in the context of a special process, the Virtual Machine Worker Process (vmwp.exe). This in turn is started via the NETWORK service account and thus users with the role ‘Hyper-V administrator’ can access resources in the file system which map a VM. This concerns in particular the scenario where various administrators support different virtual machines on a physical Hyper-V system.

Motivation: Each user who can access the Hyper-V Manager can create snapshots of a virtual machine. That is also possible even if the user does not actually have access to the suitably protected objects in the file system. If such a user subsequently stores a snapshot of a VM in an area to which the user has access, then the system is rapidly compromised.

Implementation example: Access to the respective virtual machines, folders and storage locations can be restricted to the respective users by means of Access Control Lists (ACLs). The virtual machines should also be encrypted.

Note: A structured system of subdirectories allows access to be configured more flexibly through ACLs

Example:
W:\Virtualization Resources\Project A\Virtual Machines
W:\Virtualization Resources\Project A\Virtual Hard Disks
W:\Virtualization Resources\Project A\Virtual Floppy Disks
W:\Virtualization Resources\Project A\ ISO-Dateien
W:\Virtualization Resources\Project B\Virtual Machines
W:\Virtualization Resources\Project B\Virtual Hard Disks

W:\Virtualization Resources\Project B\Virtual Floppy Disks
W:\Virtualization Resources\Project B\ISO-Dateien
W:\Virtualization Resources\Project C\Virtual Machines
W:\Virtualization Resources\Project C\Virtual Hard Disks
W:\Virtualization Resources\Project C\Virtual Floppy Disks
W:\Virtualization Resources\Project C\ISO-Dateien

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-45/6.0

Req 46	The management software must support a administration separation for the virtual machines and the network.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-36/6.0

Req 47	The System Center Virtual Machine Manager in a current version (SCVMM) must be used for administrative tasks in the Hyper-V system environment.

SCVMM is at present the “highest-quality” administration tool for Hyper-V system environments.

Motivation: The versatile Hyper-V administrative tools are not fully harmonized; modifying the configuration may lead to inconsistency of the overall system if another administration tool has not received the modification that has been made.

Implementation example: If a Hyper-V administration tool has not received a modification as a result of using another tool, the SCVMM features harmonization options to prevent an inconsistency.

Note: The "Virtual Machine Manager" or "System Center Virtual Machine Manager" must be installed on a separate physical server and not in the parent partition of a Hyper-V system. Microsoft does not support the installation of applications in the parent partition. See: http://technet.microsoft.com/en-us/library/jj136794.aspx

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-47/6.0

Req 48	A Hyper-V system must belong to a individual Active Directory domain.

Since a Windows cluster necessitates that the cluster nodes are members of an Active Directory domain, the Hyper-V servers tend to be included directly in the production domain. It is however recommended to separate the Hyper-V systems from the production environment.

Motivation: Access to this domain can be effectively controlled since a very small number of logon accounts are required to administer the Hyper-V systems.

Implementation example: A separate Active Directory domain is created for the Hyper-V systems; this domain consists of physical domain controllers and the host cluster servers. The actual production Active Directory can be operated on a virtual machine.

Note: When using the System Center Virtual Machine Manager (SCVMM), make sure that it is installed in this scenario in the Hyper-V domain, as a Hyper-V cluster, the SCVMM only in his own or a familiar Manage domain.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-48/6.0

8. Protecting sessions

Req 49	The system must allow users to log out of their current session.

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-17/7.0

Req 50	Sessions must be automatically terminated after a period of inactivity adapted to the intended use.

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-18/7.0

9. Authentication parameter password

Req 51	If passwords are used as an authentication attribute, those must be stored using a suitable and approved "Password Hashing" method to protect against offline-attacks like brute force or dictionary attacks.

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

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

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

- passwords must be stored using a cryptographic one-way function ("Password Hashing") which is suitable for that purpose and verifiably secure as matters stand

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

Explicitly NOT PERMISSIBLE is:

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

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

Motivation: Without protective measures, an attacker in possession of a user database copy is able to determine masses of contained passwords in short time by merely trying out character string combinations or making use of dictionaries. Passwords stored in cleartext or any backcalculatable format are fully defenseless to an offline attack. Once a password has been ascertained it can be used by an attacker for unauthorized access to the system and the data on it.

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

- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

For this requirement the following warranty objectives are relevant:

ID: 3.01-24/7.0

Req 52	If a password is used as an authentication attribute, a protection against online attacks like brute force and dictionary attacks that hinder password guessing must be implemented.

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

To prevent this, a countermeasure or a combination of countermeasures from the following list must be implemented:

  - technical enforcement of a waiting period after a login failed, right before another login attempt will be granted. The waiting period shall increase significantly with any further successive failed login attempt (for example, by doubling the waiting time after each failed attempt)
  - automatic disabling of the user account after a defined quantity of successive failed login attempts (usually 5). However, it has to be taken into account that this solution needs a process for unlocking user accounts and an attacker can abuse this to deactivate accounts and make them temporarily unusable
  - Using CAPTCHA ("**C**ompletely**A**utomated**P**ublic**T**uring test to tell**C**omputers and**H**umans**A**part") to prevent automated login attempts by machines ("robots" or "bots") as much as possible. A CAPTCHA is a small task that is usually based on graphical or acoustic elements and is difficult to solve by a machine. It must be taken into account that CAPTCHA are usually not barrier-free.

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-25/7.0

Req 53	If a password is used as an authentication attribute, it must have at least 12 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

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

  - Minimum length of 12 characters.
  - Comprising at least three of the following four character categories:
    - lower-case letters
    - upper-case letters

    - digits
    - special characters

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

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

Permissible deviation in the password minimum length
Under suitable security-related criteria, conditions can potentially be identified for a system that enable the minimum password length to be reduced:

- It is generally permissible to reduce the minimum password length for systems that use additional independent authentication attributes within the authentication process in addition to the password (implementation of 2-Factor or Multi-Factor Authentication).
- Any reduction in the minimum password length must be assessed individually by a suitable technical security advisor (e. g. a PSM from Telekom Security) and confirmed as permissible. In the assessment, the surrounding technical, organizational and legal framework parameters must be taken into account, as well as the system-specific protection requirements and the potential amount of damage in the event of security incidents.
- The absolute minimum value of 8 characters length for passwords must not be undercut.

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

For this requirement the following threats are relevant:

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-26/7.0

Req 54	If a password is used as an authentication attribute for technical accounts, it must have at least 30 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

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

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

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

- upper-case letters
- digits
- special characters

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

For this requirement the following threats are relevant:

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-27/7.0

Req 55	If a password is used as an authentication attribute, the reuse of previous passwords must be prevented.

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

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

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

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

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

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

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

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

Implementation example: [Example 1]
Linux System

set entry in /etc/login.defs

```
PASS_MIN_DAYS 1
```

and additionaly set entries in PAM Konfiguration

```
password requisite pam_pwquality.so try_first_pass local_users_only enforce-for-root retry=3 remember=60
password sufficient pam_unix.so sha512 shadow try_first_pass use_authtok remember=60
```

[Example 2]
Windows System

set entries in GPO

Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy\Minimum password age = **1**

Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy\Enforce password history = **24** (technical maximum)

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-28/7.0

Req 56	If a password is used as an authentication attribute, users must be able to independently change the password anytime.

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-29/7.0

Req 57	If a password is used as an authentication attribute, it must be changed after 12 months at the latest.

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

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-30/7.0

Req 58	If passwords are used as an authentication attribute, they must not be displayed in plain text during input.

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

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

Exceptions:

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

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

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-31/7.0

10. Logging

Req 59	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 ("Timestamp").

Exceptions of this requirement are systems for which logging cannot be implemented because of building techniques, use case or operation area. Examples for these kind of systems are customer devices such as Smartphones or IADs/ home gateways (e.g. Speedport).

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

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

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

In addition - as for any data that is processed by a system - an appropriate protection requirement must also be taken into account and implemented for logging data; this applies to storage, transmission and access. In particular, if the logging data contains real data, the same protection requirements must be taken into account that is also used for the regular processing of this real data within the source system.

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

Event	**Event data to be logged**
Incorrect login attempts	• User account, • Number of failed attempts, • Source (IP address, client ID / client name) of remote access
System access from user accounts with administrator permissions	• User account, • Access timestamp, • Length of session, • Source (IP address) of remote access
Account administration	• Administrator account, • Administered user account, • Activity performed (configure, delete, enable and disable)
Change of group membership for accounts	• Administrator account, • Administered user 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. Logging data could be used as evidence to take legal steps against attackers.

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-33/7.0

Req 60	The changeable configuration parameters which are necessary for the operation and safety must be monitored periodically.

In order to identify unauthorized changes to a configuration of a virtual machine or virtual host promptly, the configuration settings for the virtual environment (e.g. network settings, device configurations) must be monitored. This means for example, a unauthorized change in network settings for a VM or the unauthorized change of services used by the virtualization hosts as NTP.

Motivation: Ongoing monitoring of the configuration makes it possible to identify configuration errors and manipulation. Such monitoring can be carried out with suitable tools such as Veeam Reporter or Tripwire.

Implementation example: Such monitoring can be carried out with the support of appropriate management tools such as 'Veeam Reporter'.

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

For this requirement the following warranty objectives are relevant:

ID: 3.35-38/6.0

Req 61	Access actions to the virtual machines or data on the virtual machine must be logged.

Another aspect that bolsters security with the virtual machines is monitoring the actions that users carry out. In this respect, Hyper-V supports the option of logging successful and unsuccessful attempts at accessing objects.

Motivation: Access logging enables you to trace from which account the – possibly unauthorized – access to a virtual hard disk or virtual machine has been gained. Suitable monitoring software can be used to automatically trigger appropriate measures to protect the virtual hard disks or virtual machines.

Implementation example: Usage of the Auditpol.exe tool to configure the monitoring policies of the file system in respect of success and failure in the case of opening, copying, modification or deletion. A monitoring product such as System Center Operations Manager (http://www.microsoft.com/en-in/server-cloud/systemcenter/ operations-manager.aspx) should also be used to generate warnings following defined actions.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-61/6.0

Req 62 Virtual machines must be synchronized with a reliable time server.

Precise time synchronization in the overall system is necessary to operate a Hyper-V system and the virtual machines. Synchronization of the virtual machine with the Hyper-V system can lead to an “out of sync” in the case of a major load on the Hyper-V system, which, in turn, causes a significant difference between the system time of the virtual machine and that of the Hyper-V system.

Motivation: Tampering on the basis of different timestamps can compromise services such as Kerberos authentication, the timestamps with logfiles, timestamp with certificate validation as well as with digital signatures and virtual machines themselves.

Implementation example: Time synchronization via the Active Directory, higher-ranking domain or the Forrest Master via the NTP protocol.

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

For this requirement the following warranty objectives are relevant:

ID: 3.49-62/6.0

Req 63 The system clock must be synchronized to an accurate reference time (Time Standard).

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

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

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

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

Implementation example: some valid time reference sources:

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-32/7.0

Req 64	Applicable retention and deletion periods must be observed for security-relevant logging data that is recorded locally.

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

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

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

- Security-related logging data must be retained for a period of 90 days.
  (*This requirement only applies if no additional forwarding to a separate log server is implemented on the system and the logging data is therefore only recorded locally.*)
- After 90 days, stored logging data must be deleted immediately.

Deviances

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

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

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

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

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-34/7.0

Req 65	Security-relevant logging data must be forwarded to a separate log server immediately after it has been generated.

Logging data must be forwarded to a separate log server immediately after it has been generated. Standardized protocols such as Syslog, SNMPv3 should be preferred.

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

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

For this requirement the following warranty objectives are relevant:

ID: 3.01-35/7.0

Req 66	For security-relevant logging data that is forwarded to the separate log server, compliance with the applicable retention and deletion periods must be ensured.

The following basic rules must be taken into account:

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

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

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

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

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

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

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

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-36/7.0

Req 67	The system must provide logging data that is required to detect the system-specific relevant forms of attack in a SIEM.

The forms of attack that are typically to be expected for the present system must be systematically analyzed and identified.
The MITRE Attack Matrix (https://attack.mitre.org) can be used as a structured guide during such an identification.

It must be ensured that the system generates appropriate logging data on events that are or may be related to these identified forms of attack and that can be used to detect an attack that is taking place.

The logging data must be sent to a SIEM immediately after the system event occurs.
SIEM (Security Information & Event Management) solutions collect event log data from various source systems, correlate it and evaluate it automatically in real time in order to detect anomalous activities such as ongoing attacks on IT/NT systems and to be able to initiate alarms or countermeasures.
The immediate receipt of system events is therefore absolutely crucial for the SIEM to fulfill its protective functions.

Note:
The immediate need to connect a system to a SIEM is specifically regulated by the separate "Operation" security requirements catalogs.
If the present system does not fall under this need, the requirement may be answered as "not applicable".

Motivation: A SIEM as an automated detection system for attacks can only be effective if it continuously receives sufficient and, above all, system-specific relevant event messages from the infrastructures and systems to be monitored. General standard event messages may not be sufficient to achieve an adequate level of detection and only allow rudimentary attack detections.

Implementation example: An example system allows end users to log in using a username and password. One of the typical forms of attack for this system would be to try to discover and take over user accounts with weak or frequently used passwords by means of automated password testing (dictionary or brute force attack). The example system is configured to record every failed login event in system protocols ("logs"). By routing this logging data in parallel to a SIEM, the SIEM can detect in real time that an attack is obviously taking place, alert it and thus enable immediate countermeasures.

ID: 3.01-37/7.0