Security requirement

Machine Learning

Deutsche Telekom Group

Version	2.0
Date	Jul 1, 2024
Status	Released

Publication Details

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

Reuterstrasse 65, 53315 Bonn
Germany

File name	Document number	Document type
	3.81	Security requirement
Version	**State**	**Status**
2.0	Jul 1, 2024	Released
Contact	**Validity**	**Released by**
Telekom Security psa.telekom.de	Jul 1, 2024 - Jun 30, 2029	Stefan Pütz, Leiter SEC-T-TST

Summary
This document describes the security requirements for machine learning algorithms.

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

Table of Contents

1.	Introduction	4
2.	AI-Compontens	5
2.1.	Machine Learning Algorithm	5
2.2.	Training of the machine learning algorithm	7
2.3.	Operation of the machine learning algorithm	9
2.4.	Access to the machine learning algorithm	14
2.5.	Access of the machine learning algorithm to other systems	23
3.	Systems using AI-Components	25

1. Introduction

This document was prepared on the basis of the requirements set out in the security guidelines applicable within the Group. It should be noted that in addition to the security guidelines, there may also be AI ethics guidelines or other requirements in the Group that must also be observed.

The security requirement serves, among other things, as the basis for release in the PSA process. It also serves as an implementation recommendation for units that do not participate in the PSA process. These requirements must already be taken into account during the planning and decision-making processes. When implementing the safety requirement, the respective overriding national, international and supranational law must be observed.

In order to create a uniform basis for understanding, some central terms are clarified below:

Term	Description
Artificial Intelligence (AI)	Generic term for technologies that enable computers to simulate human-like intelligence and perform tasks that normally require human thought.
AI-Component	A system that implements AI itself. For example, an image recognition algorithm that can be accessed via an API.
Machine Learning	Machine learning is a generic term for the "artificial" generation of knowledge from experience. An artificial system learns from examples and can generalize them after the learning phase is complete.
Machine Learning Algorithm	One of many specific algorithms used in machine learning, for example, to detect patterns in data and make predictions.
Hyperparameter	Parameters that influence the learning process of a machine learning algorithm and are set before training.
Dimensionality reduction	A process of reducing the number of features or variables in a data set to reduce complexity and improve the efficiency of machine learning algorithms.

It is also important to note that not every product must meet every single requirement. This document is divided into requirements for AI-Components and requirements for systems that use AI-Components.

If compliance with the described requirements is not feasible or only feasible to a limited extent in individual cases, a risk assessment must be carried out together with a security and/or data protection expert (according to the requirement concerned) and possible alternative protective measures must be agreed.

2. AI-Compontens

In this document an AI-Component is a system that implements AI itself. For example, an image recognition algorithm that can be accessed via an API.

2.1. Machine Learning Algorithm

Req 1	The task of the algorithm, the context in which it is to be used, and the conditions under which the algorithm is allowed to make decisions must be clearly defined.

For an algorithm to be used reliably, it is crucial that the task it is to perform is clearly defined, that the context in which it is to be used is fully understood, and that the framework conditions under which the algorithm is allowed to make decisions are precisely regulated. This is the only way to ensure that the algorithm delivers the desired results and does not lead to unexpected or undesired outcomes.

Required information for the definition of the task:

- Clear and unambiguous description of the specific task
- Example input and output data

Information needed to define the context:

- Type of data to be processed
- Source of training and test data
- List of involved stakeholders

Motivation: Clearly defining the task, the context, and the constraints under which the algorithm is allowed to make decisions not only minimizes the risk of unexpected results, but also ensures better verifiability of the algorithm.

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

For this requirement the following warranty objectives are relevant:
- Transparency

ID: 3.81-1/2.0

Req 2	The process of data collection and/or data generation must be clearly documented. This includes training, validation and test data.

It is of great importance that the process of data collection and/or data generation is clearly documented to ensure transparency, traceability as well as protection of personal data. This includes not only the documentation of the training, validation and test data, but also the description of the methods and tools used as well as the selection criteria for the data. Clear documentation is essential to ensure the quality of the data and to identify possible sources of error.

Relevant information:

- Timestamp of collection/generation
- Description of data sources/types
- Methods and tools used for data collection and/or data generation
- Selection criteria for the data
- Processes for data preparation and cleaning
- Processes for anonymization or pseudonymization of personal data
- Processes for validation and verification of data quality

- Processes for splitting data into training, validation and test data
- Processes for updating and maintaining data

Motivation: Clear documentation of the data collection and/or data generation process is critical to the quality and trustworthiness of data used for training and testing the algorithm. Insufficient documentation can lead to errors and inaccuracies in the data, which in turn can lead to incorrect decisions and unreliable results.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency

ID: 3.81-2/2.0

Req 3	The AI-Component and connected systems must be protected against misuse of the AI-Component.

Hardening must take place so that the AI component, through quasi-legitimate use, cannot be abused.

Examples of such attacks are:
- Task Escape Attacks - Using the Large Language Model (LLM) for (potentially malicious) out-of-scope tasks
- Data Exfiltration Attacks - output of internal information provided to the LLM by, for example, a database

Motivation: Machine learning algorithms often have more capabilities than are needed by the developers. These can be potential vulnerabilities as they are used to abuse the AI component.

For example, if a publicly available Large Language Model is used as the AI-Component, it is important to ensure that the LLM's responses match the context of the project.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Availability
- Confidentiality

ID: 3.81-3/2.0

Req 4	Deployed base algorithms used for transfer learning must be obtained from trusted sources and checked against existing AI-Component requirements (PSA requirements in this document).

Transfer learning refers to all methods that enable the knowledge gained from solving a particular problem to be transferred to the processing of another problem.

To ensure that transfer learning is performed effectively and safely, it is critical that the baseline algorithms used for this process come from trusted sources and comply with existing AI-Component requirements (PSA requirements in

this document). It is important that these baseline algorithms are regularly reviewed and updated to ensure that they meet the latest standards and do not pose security risks. In addition, factors such as performance, scalability, and compatibility with other systems should also be considered when selecting baseline algorithms for transfer learning.

Motivation: If the base algorithms are obtained from unreliable or insecure sources, this can lead to errors, data leaks, and even security vulnerabilities and new attack scenarios. In addition, unreliable algorithms can affect the performance of the AI-Component and make it more difficult to achieve the desired results. Therefore, it is important that the base algorithms come from trusted sources to ensure effective and secure transfer learning.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency

ID: 3.81-4/2.0

Req 5	The hyperparameters of the algorithm must be carefully selected and documented to ensure reproducibility.

Hyperparameters are external configuration variables used for training machine learning algorithms and are set manually before training an algorithm. Examples of hyperparameters are the number of nodes and layers in a neural network and the number of branches in a decision tree. Hyperparameters determine important features such as model architecture, learning rate, and model complexity.

To ensure that the results of an algorithm are reproducible, it is important to carefully select and document the hyperparameters. Accurate documentation of hyperparameters also enables further improvement of the algorithm through hyperparameter optimization. If automated hyperparameter optimization is used, this process must also be traceable.

Motivation: If the hyperparameters of a machine learning algorithm are not carefully documented, this can lead to problems with the reproducibility and optimization of the algorithm. Without accurate documentation, it is difficult to obtain the same results when the algorithm is run again, which can affect the reliability of the algorithm. In addition, the lack of documentation also makes it difficult to optimize the hyperparameters for best possible results.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Availability
- Transparency

ID: 3.81-5/2.0

2.2. Training of the machine learning algorithm

Req 6	Collected/generated data must be pre-processed and cleaned prior to use for training, validation and testing purposes.

To ensure the accuracy and reliability of training, validation and testing results, it is necessary to carefully review, clean and normalize the collected/generated data before use. In this process, erroneous, incomplete, or inconsistent data should be removed or corrected to avoid bias in the results. In addition, appropriate pre-processing of the data, such as feature extraction or dimensionality reduction, can improve the effectiveness of machine learning algorithms.

Possible cleaning procedures and preprocessing:

- Removing duplicates

- Removing outliers
- Filling missing values
- Normalization can be used to unify the scaling of the data and thus make it comparable
- Feature engineering, the preparation of data for processing in machine learning algorithms, can be used to extract relevant features from raw data and reduce the dimensionality of the data
- Removal of irrelevant features
- Balancing classes
- Each class of data (e.g. dog and cat) must be represented the same number of times in the data set

Motivation: It is important to carefully clean and prepare data before using it for training, validation, and testing purposes to achieve accurate and reliable results. Uncleaned or incomplete data can introduce bias and errors in the results and reduce the effectiveness of the machine learning algorithm. Performing cleanup and pre-processing can ensure that the data is accurate and consistent, leading to higher accuracy and reliability of results.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency

ID: 3.81-6/2.0

Req 7	The data used for training, validation and testing purposes must be representative with regard to the assigned task. In addition, the data must be free of possible bias and inconsistencies.

To ensure that the machine learning algorithm produces accurate and reliable results, the data used for training, validation, and testing purposes must be representative of the assigned task. This means that the data should have sufficient diversity and coverage to cover the different aspects of the task. In addition, it is mandatory that the data be free of possible biases and inconsistencies (bias) that may lead to inaccurate or unfair results. Examples of possible biases include unequal distribution of data points in certain categories or unequal representation of certain groups in the data.

Motivation: It is especially necessary that the data used for a machine learning algorithm be free of bias and inconsistency in order to produce accurate and fair results. If the data is biased, the results may also be skewed and lead to inaccurate or unfair decisions.

For example, an uneven distribution of data points in certain categories may cause the algorithm to inappropriately consider those categories or to draw incorrect conclusions. In addition, unequal representation of certain groups in the data may lead to unfair decisions based on prejudice or discrimination.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency

ID: 3.81-7/2.0

Req 8	Security-relevant events, such as the training of the machine learning algorithm, must be logged with an exact time stamp and a unique system designation.

It is necessary to accurately document activities, such as training the machine learning algorithm, to ensure effective

troubleshooting and traceability in the event of errors or security breaches. The following information must be recorded:

- Name of the person conducting the training
- Training timestamp
- Data set used
- Changes to training, validation and test data
- Name of the person making the changes
- Timestamp of the changes

Motivation: Accurately logging security-related events, such as training the machine learning algorithm, is essential to ensure effective troubleshooting and traceability in the event of errors or security breaches. Accurate logging allows potential security risks to be identified and fixed early, before they lead to major problems. In addition, clear identification of events enables fast and effective response to security incidents.

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

For this requirement the following warranty objectives are relevant:
- Transparency

ID: 3.81-8/2.0

Req 9	Data requiring protection, such as training, validation and test data, as well as hyperparameters and implementation of the algorithm, must be protected from unauthorized viewing, modification and deletion during storage.

To ensure the integrity and confidentiality of data used for algorithm training, validation, and testing, as well as the hyperparameters and implementation of the algorithm itself, during storage, it is essential to implement appropriate security measures. These include the use of encryption technologies, access controls, and periodic reviews of security measures. It is also necessary to ensure that all individuals who have access to this data are aware of the importance of data security and adhere to the appropriate security protocols.

Motivation: Unauthorized access to training, validation and test data can lead to unnoticed manipulation of these. If the algorithm is then trained with the manipulated data set, unforeseen behavior can occur and the integrity of the algorithm is no longer guaranteed.

If unauthorized access to the hyperparameters or to the implementation of the algorithm occurs, changes can be made that can have serious consequences. In addition, this information provides the opportunity for cloning the algorithm.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency
- Confidentiality

ID: 3.81-9/2.0

2.3. Operation of the machine learning algorithm

Req 10	The results of the algorithm must be continuously checked and validated to ensure that they are correct and meet expectations. If the results of the algorithm deviate from the expected behavior, this must be corrected immediately.

It is essential that the results of the algorithm are regularly checked and validated to ensure that they are not only correct, but also meet expectations. Inaccurate or erroneous output can have serious consequences. Therefore, it is necessary that the algorithm be continuously monitored to detect and correct potential problems early. This can be done through regular functional/ and penetration testing.

It is also necessary that the algorithm always shows the expected behavior. However, should deviations occur, these must be identified and remedied immediately to ensure reliable functioning of the algorithm. This requires that the affected machine learning algorithm is taken out of operation as long as the misbehavior has not been corrected.

Motivation: By continuously monitoring the behavior of the algorithm, potential problems can be detected and responded to at an early stage. In the absence of such monitoring, behavioral changes are not detected and cannot be remedied.

Implementation example: Depending on the area of application, for example in image recognition, random checks may be sufficient. However, if an AI-Component is used to configure productive systems, such as network elements, servers, applications, etc., the decision made by the AI-Component should be checked beforehand on a test system.

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

For this requirement the following warranty objectives are relevant:
- Integrity

ID: 3.81-10/2.0

Req 11	Security-related events must be logged with an exact timestamp and a unique system reference.

Security-relevant are any events that may have an impact on the information security (confidentiality, integrity, availability) of a system or indicate its misuse. Systems must log the occurrence of security-related events.
In order to enable the assignment and evaluation of these events, they must be logged together with a unique system name (e.g. host name, IP or MAC address) and the exact time of occurrence ("time stamp"). At a minimum, the timestamp of a logged event must include the following information:

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

The security relevance of specific events depends on the type and function of the system as well as the data processed in it. Therefore, it is not possible to specify a uniform list of events that are relevant to security. Relevant events must be identified according to the system. The following list contains typical security-related events and is accordingly to be understood as a basic guide for the identification of security-relevant events for a particular system.

Security-related Event	Information to log to the event
Failed login attempts	• affected user account • number of failed attempts • source of access (IP address, client-ID/client-name, local console, ...)

System access via user accounts (including system accounts or technical user accounts)	• affected user account • Access start/end time (login/logout) • source of access (IP address, client-ID/client-name, local console, ...)
Modifications of user accounts (including system accounts or technical user accounts)	• user account used to make the modification • user account to which the modification was made • type of modification made (create, activate, deactivate, delete, password reset, ...)
Changes to role assignements/permissions of user accounts (including system accounts or technical user accounts)	• user account used to make the modification • user account to which the modification was made • name of the affected role, type of the affected permission • type of modification made (role/permission added, role/permission removed, ...)
Use/initiation of critical system functions	• user account used to make the use • name/type of system function used
Access to sensitive data processed in the system	• user account used to access the data • Name of the affected data field (the contents of the data field should not be logged to prevent data leakage via log files) • type of access (read, modify, delete, ...)
Abnormal increase in the utilization of system resources, such as memory, CPU, network, or disk access, over an extended period of time	• threshold of the affected system resource • duration of exceeding the threshold • threshold exceedance Depending on the individual system, sensible threshold values must be defined.
Other events that may provide evidence of attempted or successful attacks against the system ***)**	• individually depending on the type of event

***)** *This requires identifying the most important forms of attack that the system can typically expect and ensuring that events relevant to their detection are actually logged by the system. As a rule, this can only be guaranteed by the developer/manufacturer of the system.*

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.

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

ID: 3.01-33/8.0

Req 12	Applicable retention and deletion periods must be observed for security-relevant logging data that is recorded locally.

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

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

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

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

Deviances

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

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

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

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

For this requirement the following threats are relevant:

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

ID: 3.01-34/8.0

Req 13	Security-relevant logging data must be forwarded to a separate log server immediately after it has been generated.

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

cols such as Syslog, HTTPS 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:
- Integrity

ID: 3.81-13/2.0

Req 14	For security-relevant logging data that is forwarded to the separate log server, compliance with the applicable retention and deletion periods must be ensured.

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

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

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

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

Implementation example: Taking into account the current legal situation and applicable data privacy regulations, the following deletion periods for forwarded security-relevant logging data from an exemplary telecommunications system are implemented on the separate log server:
- Standard System Logs: Deletion after 90 days at the latest
- Logging of public IP addresses: Deletion (or anonymization) after 7 days at the latest
- Logging of the assignment of dynamic public IP addresses by the telecommunication solution: Deletion after 7 days at the latest
- Logging of non-billing-relevant call detail records: Deletion after 7 days at the latest
- Logging of the content of e-mail and SMS: Deletion after 24 hours at the latest
- Logging of the domain queries handled by the DNS server of the telecommunications solution: Deletion after 24 hours at the latest

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

- Unnoticeable feasible attacks

ID: 3.01-36/8.0

2.4. Access to the machine learning algorithm

Req 15	Access for the use of the AI-Component must not be possible without successful authentication and authorization. This applies both to API interfaces and to all other ways of interacting with the AI-Component.

Successful authentication and authorization is required before the machine learning algorithm can be accessed or used. This applies to all types of interactions with the algorithm, including API interfaces and other access options.

This requirement applies to all systems for which Deutsche Telekom is responsible. I.e. if it is an external service (Bing Search, Chat-GPT, ...), this requirement does not apply.

Motivation: In the absence of authentication and authorization, there is a risk of unauthorized use of the algorithm and must therefore be prevented.

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

For this requirement the following warranty objectives are relevant:
- Confidentiality

ID: 3.81-15/2.0

Req 16	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 sys-

tem information must always be deactivated.

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

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

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

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

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

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

ID: 3.01-9/8.0

Req 17	User accounts must be protected with at least one authentication attribute.

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

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

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

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

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

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

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

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

ID: 3.01-20/8.0

Req 18	Privileged user accounts must be protected with at least two authentication attributes from different factors.

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

Examples of privileged user accounts:

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

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

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

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

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

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

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

For this requirement the following threats are relevant:

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

ID: 3.01-21/8.0

Req 19	It must be ensured that machine users do not use hard-coded secrets such as passwords or tokens to authenticate or authorize themselves to the AI algorithm.

To ensure the security and integrity of the system, it is important to ensure that machine users do not use hard-coded secrets such as passwords or tokens to authenticate or authorize themselves to the AI algorithm. Instead, alternative methods of authentication and authorization should be implemented to avoid potential disclosures.

Motivation: Secrets like passwords, certificates, secret keys are a high risk if they are disclosed to unauthorized persons. Especially used in code or images stored in repositories can lead to an unintended disclosure.

Implementation example: For implementation, the following Secret Management Systems can be used:

- HashiCorp. Vault
- CyberArk Conjur
- AWS KMS
- Azure Key Vault
- Google Cloud KMS
- GitLab Protected Variables

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency

ID: 3.81-19/2.0

Req 20	User accounts must ensure the unique identification of the user.

Users must be identified unambiguously by the system.

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

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

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

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

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

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

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

ID: 3.01-22/8.0

Req 21	If passwords are used as an authentication attribute, those must be stored using a suitable and approved "Password Hashing" method to protect against offline-attacks like brute force or dictionary attacks.

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

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

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

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

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

Explicitly NOT PERMISSIBLE is:

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

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

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

ID: 3.01-24/8.0

Req 22	If a password is used as an authentication attribute, a protection against online attacks like brute force and dictionary attacks that hinder password guessing must be implemented.

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

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

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

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

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

ID: 3.01-25/8.0

Req 23	If a password is used as an authentication attribute, it must have at least 12 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

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

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

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

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

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

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

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

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

ID: 3.01-26/8.0

Req 24	If a password is used as an authentication attribute for technical accounts, it must have at least 30 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

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

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

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

- lower-case letters
- upper-case letters
- digits
- special characters

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

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

ID: 3.01-27/8.0

Req 25	If a password is used as an authentication attribute, the reuse of previous passwords must be prevented.

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

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

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

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

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

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

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

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

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

ID: 3.01-28/8.0

Req 26	If a password is used as an authentication attribute, users must be able to independently change the password anytime.

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

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

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

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

ID: 3.01-29/8.0

Req 27	If a password is used as an authentication attribute, it must be changed after 12 months at the latest.

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

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

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

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

attacker can increase considerably over time.

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

ID: 3.01-30/8.0

Req 28	If passwords are used as an authentication attribute, they must not be displayed in plain text during input.

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

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

Exceptions:

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

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

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

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

ID: 3.01-31/8.0

2.5. Access of the machine learning algorithm to other systems

Req 29	The results and decisions of the machine learning algorithm must not be transferred to productive systems without prior testing.

It is crucial that the results and decisions generated by a machine learning algorithm are carefully checked and validated before they are transferred to productive systems. Although machine learning algorithms are capable of identifying complex patterns and relationships in large amounts of data, they are still not perfect and can be flawed. If these errors are not detected and corrected, they can have a serious impact on the affected system. Therefore, it is important that algorithm results and decisions are reviewed and validated to ensure that they are correct and reliable. This verification does not necessarily have to be performed by a human.

Motivation: Without prior verification of the results of the machine learning algorithm, decisions made can lead to system failures. This must be avoided as a matter of principle.

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

For this requirement the following warranty objectives are relevant:
- Availability
- Transparency

ID: 3.81-29/2.0

3. Systems using AI-Components

Req 30	Data transmitted to an AI-Component must be preprocessed before transmission. This includes denoising algorithms for images, videos and speech as well as similar preprocessing for inputs to other AI-Components, such as text-based algorithms.

This preprocessing includes various steps that must be adapted to the type of input data (images, video, speech, text, etc.). This preprocessing ensures that the data is available in a form that can be analyzed optimally by the AI component, thus making manipulation-based attacks more difficult.

Examples for pre-processing:

1. Images and Video
    - Denoising algorithms
    - Scaling to uniform size
    - Contrast adjustments
    - Color correction
    - Cropping of relevant areas
2. Speech
    - Removal of background noise
    - Normalization of speech volume
    - Restriction of frequency ranges
3. Text
    - Removal of potentially malicious words, phrases, or sentences

Motivation: By pre-processing the data, potential malicious user input can be filtered out before it reaches the AI component.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Availability
- Confidentiality

ID: 3.81-30/2.0

Req 31	It must be ensured that no confidential data is transferred by the system or the user to the AI components unless this is explicitly provided for. This includes both personal data and other information requiring protection.

It must be ensured that no personal or other data requiring protection is transferred to AI-Components by the system or the user unless this is explicitly provided for. This requires careful review and control of data flows within the system to ensure that the confidentiality and integrity of the data is maintained at all times. In any case, confidential data should be anonymized before it is transferred to an AI-Component.

Regulation for the classification of information: https://informationsdrehscheibe.telekom.de

Motivation: All information transferred to AI-Components could be viewed by the operator or third parties and also used outside the agreed processing contract.

The best protection against unwanted use of confidential data is to hand over such data only to systems that are expli-

citly designed to process it.

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

For this requirement the following warranty objectives are relevant:
- Confidentiality

ID: 3.81-31/2.0

Req 32	The system must ensure that only data that meets the requirements of the AI-Components is transferred. Thus, for example, no .exe programs may be accepted as user input for an image recognition algorithm.

In order for the AI-Component to operate effectively and securely, only data that meets the requirements for secure operation of this component may be transferred to it. It is important to avoid certain file formats or types of data to prevent incorrect or inaccurate processing. For example, .exe programs should not be accepted as input to an image recognition algorithm, as this type of data is incompatible with the algorithm's operation and can lead to undesirable results.

Motivation: Inadequate input data can have far-reaching consequences when processed. Input validation is intended to prevent undefined behavior from occurring in the AI-Component.

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

For this requirement the following warranty objectives are relevant:
- Integrity

ID: 3.81-32/2.0

Req 33	AI-Components used must be approved for use by the Group.

AI components used must have received approval from the Group before being used internally or externally in projects. If this is not the case, the AI component may not be used.

Motivation: When AI-Components are obtained from unreliable or insecure sources, this can lead to errors, data leaks, and even security breaches. In addition, unreliable algorithms can affect the performance of the application and make it more difficult to achieve the desired results.

Implementation example: The group has released OpenAI's large language models in certain cases. Thus, GPT-3.5-turbo (ChatGPT) or GPT-4, among others, may be used for internal or external projects.

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

For this requirement the following warranty objectives are relevant:
- Integrity
- Transparency

ID: 3.81-33/2.0

Req 34	A responsible person must be named who is responsible for the decisions of the machine learning algorithm so that a clear assignment is possible in the event of errors or damage.

AI-Components can cause great damage to a company in the event of misuse or in the event of an error. Therefore, it is important to ensure a clear assignment of responsibility.

Instructions and processes for all conceivable error scenarios must be defined in advance. Ethical and legal issues must also be clarified by a responsible person or department before commissioning and then updated on an ongoing basis.

The designation of a responsible person, or an area, with clear indication of the contact details, is therefore necessary.

Motivation: Clear accountability for the use of AI-Components is important to ensure that in the event of errors or damage, a clear assignment is possible and thus quick action can reduce the damage that occurs.

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

For this requirement the following warranty objectives are relevant:
- Transparency

ID: 3.81-34/2.0

Req 35	Software and hardware of the system must be covered by security vulnerability support from the supplier.

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

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

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

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

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

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

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

- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-1/8.0

Req 36	Predefined authentication attributes must be changed.

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

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

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

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

ID: 3.01-8/8.0