

Security requirement

Hadoop

Deutsche Telekom Group

Version	2.0
Date	Dec 1, 2025
Status	Released

Publication Details

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

Reuterstrasse 65, 53315 Bonn
Germany

File name	Document number 3.98	Document type Security requirement
Version 2.0	State Dec 1, 2025	Status Released
Contact Telekom Security psa.telekom.de	Validity Dec 1, 2025 - Nov 30, 2030	Released by Stefan Pütz, Leiter SEC-T-TST

Summary

Copyright © 2025 by Deutsche Telekom AG.
All rights reserved.

Table of Contents

1.	Introduction	4
2.	Hadoop	5
2.1.	Choice of Distribution	5
2.2.	Architecture	7
2.3.	Choice of Hadoop Software Components, Use of Plugins	9
2.4.	Encryption of Data in Rest and in Motion	10
2.5.	Data, Data Access, and Roles	11
2.6.	Basic System Configuration	18
2.7.	HDFS Configuration	26
2.8.	Kerberos Configuration	26
2.9.	Knox Configuration (Hortonworks)	29

1. Introduction

This document addresses Hadoop installations storing and using not only pseudo- or anonymized data but also raw data. Installations of the former type require pseudo-/anonymization at the data sources or at the points of ingress to the Data Lake and according to predefined use cases. Installations of the latter type require the implementation of above-baseline measures to ensure adequate protection of sensitive data. The challenge being is to protect both large amounts of data and most sensitive personal data.

Further, within this document an on-premises installation of Hadoop is assumed, in contrast to managed or cloud installations. In the latter cases further requirements, e.g. in regard to commissioned data processing or processes in regard to interfaces to third parties, and strict audit requirements might be essential to define and comply with.

The requirements defined in this document are addressing the aspects of data privacy and data security for big data of “Design Guideline for Big Data projects based on Hadoop” by Dr. Lars Hofmann et al.

While the author of this document on security requirements strives to define distribution-independent requirements, they might be biased towards Hortonworks, as that is the distribution in place at the authors company.

2. Hadoop

2.1. Choice of Distribution

Req 1	Hadoop distributions supporting central management and central security management must be used.
-------	--

In practice, it will not be possible to ensure rollout and enforcement of consistent security policies if non-central management solutions are in place.

At the time of writing, Apache Ranger and Apache Sentry were available for central security management and Apache Ambari for central operational management.

ID: 3.98-1/2.0

Req 2	Hadoop distributions supporting authentication and authorization of all components of a Hadoop installation must be used.
-------	---

To ensure complete and consistent implementation of security policies as well as to prevent loopholes, authentication of users and services in regard to data ingest, data access, and for execution of jobs must be possible.

ID: 3.98-2/2.0

Req 3	Hadoop distributions must support the forwarding of user and service rights.
-------	--

A user or service must access data directly, e.g. data stored on a data node should not be accessed directly by a user, but through some other software component from the Hadoop set of components. In such typical case, the access rights of the original user must be used to grant data access, not the access rights of the respective other software component.

ID: 3.98-3/2.0

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.

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

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

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

Motivation: Software supply chains contain various attack vectors. An attacker can start at various points to manipulate software or introduce his own routines and damage or control the target environment in which the software is subsequently used. The attack can occur on the transport or transmission path or on the provisioning source itself. Accordingly, an attack is facilitated if software is not obtained from official and controlled sources or if an integrity check is omitted.

There is a particular risk for software obtained from public registries, as these are open to anyone for the provision of software projects. Perfidious attack methods are known, in which the attacker first provides completely inconspicuous, functional software for a while and as soon as it has established itself and found a certain spread, deliberately hidden

malicious code is integrated in future versions. Other methods rely on similar-sounding project names for widely used existing projects or overruling version numbers to inject manipulated software into any solutions based on them.

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

For this requirement the following threats are relevant:

- Unauthorized modification of data
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-2/9.0

2.2. Architecture

Req 5	Hadoop-specific network segments and systems must be separated from other segments and systems by stateful packet filters and by using separate VLANs. Ingress nodes can share a dedicated network segment with high volume data sources without a separate stateful packet filter, if additional security measures are implemented.
-------	--

For Hadoop's security framework to be effective, it is crucial to use all and only components from the Hadoop set of components. To prevent circumvention of security measures, such components must be segregated from non-Hadoop components at least at the data link layer, for example by using separated and dedicated VLANs, and by using stateful packet filters for connection of different network segments.

"Separation" regarding this document implies allowing only necessary and explicitly approved traffic from machines in one segment to machines in other segments while blocking all other traffic, which in turn implies the necessity to create a maintain a communication matrix.

Due to large amounts of data potentially being ingested by specific data sources, such high volume data sources and ingress node can share a common network segment which is used solely for the purpose of transferring data from specific high volume data sources to ingress edge nodes. In such cases, host based packet filters must be used on the ingress edge nodes, e.g. iptables on Linux systems.

ID: 3.98-5/2.0

Req 6	At least four network segments must be used to separate traffic; regarding data ingest, data use and data access, monitoring and utility, and name and data nodes; separated by stateful packet filters.
-------	--

Edge nodes are the nodes through which data is ingested in the Data Lake and which control access to data stored in the Data Lake, and all systems of a Hadoop installation require management and monitoring functions. To prevent by-pass of access controls those four different functions (data ingest, data use and data access, monitoring and utility, and master and data nodes from the Data Lake) require segregation from each other as well as segregation from other systems.

To note: Some Hadoop-components might not be clearly assignable to one of the four network segments, e.g. Hue, which might be used as a user access component and therefore is to be placed on an edge node or as a management component, to be placed on a monitoring and utility node.

ID: 3.98-6/2.0

Req 7	Edge nodes used to ingest data into the data lake must be dual homed, one leg homed in the data ingest segment and the other leg homed in the name and data node segment if access to such dual homed hosts is secured with an extra layer of security.
-------	---

This requirement attributes to the large amounts of ingested data flowing directly from edge nodes to data nodes. For performance reasons another packet filter in this flow might not be feasible. To ensure security in such cases, access to such dual homed edge nodes require extra protection, e.g. a shell control box through which all shell access is granted only through a proxy and which whitelists only allowed commands permitted to be executed on such edge nodes.

Independent from the architectural design decision, i.e. using dual homed or not dual homed edge nodes, access from edge nodes to the data lake must be possible to allow the edge nodes' HDFS clients access to the data lake.

ID: 3.98-7/2.0

Req 8	Non-system-management access to data in a Hadoop installation must be performed through egress edge nodes only.
-------	---

For accessing data in the Data Lake, users must connect to edge nodes, they must not access data in the data lake directly. This requirement implies that not all services available in principle in a fully accessible installation can be used or that such services require special drivers being aware of edge nodes. For more detailed requirements on egress edge nodes, see Knox configuration.

ID: 3.98-8/2.0

Req 9	Systems used for Hadoop components must not be used for non-Hadoop related applications, i.e. Hadoop components must not share a system with application not related to Hadoop.
-------	---

To avoid bypass of security policies, resulting in risk of unauthorized data access, such sharing of systems must not take place, neither by having Hadoop components as well as non-Hadoop related applications installed under the same operating systems nor by using different virtual machines on the same platform.

ID: 3.98-9/2.0

Req 10	Master services must run on master nodes separated from worker services, which must run on worker nodes, separated from management services, which must run on management nodes, and user-facing services, which must run on edge nodes.
--------	--

The Hadoop architecture often distinguishes between master services/nodes and worker services/nodes, for example name nodes and data nodes. While name nodes manage access to data, data nodes actually store blocks of data. Because of the importance of master services they should not run on the same machines as worker services, as such separation allows for:

- Resource contention: E.g., user jobs exhausting resources on a worker node will not degrade essential master nodes services or even cause a denial-of-service to master node services.
- Reduction of the effects of privilege escalation: E.g., master services require different authorizations than worker nodes; in case of mixed master/worker services on one node access rights for one type of service might be used to access data of the other type of service.
- Containment of security vulnerabilities: E.g., vulnerabilities specific to worker nodes will not influence essential master node services.
- One might like to consider installing different master services on different machines for added reliability and added protection of services, resulting again in resource contention and containment of security vulnerabilities.

Non-exhaustive list of master services:

- HDFS name node, secondary name node (standby name node), failover controller, journal node
- MapReduce job tracker and failover controller
- YARN resource manager and job history server

- Hive metastore server
- ZooKeeper server
- HBase master

Accumulo master, tracer, and garbage collector
Non-exhaustive list of worker services:

- HDFS data node
- MapReduce task tracker
- YARN node manager
- HBase region server
- Accumulo tablet server
- Solr server

Non-exhaustive list of management services:

- configuration management
- monitoring
- alerting
- software repositories
- backend databases
- DHCP servers

Non-exhaustive list of edge node services:

- HDFS HttpFS and NFS gateway
- KNOX
- client CLI tools

ID: 3.98-10/2.0

2.3. Choice of Hadoop Software Components, Use of Plugins

Req 11	Components from the Hadoop set of components must be used to cover the security framework blocks of administration (central management and consistent security), authentication (authentication of users and systems), authorization (provision access to data), audit (maintain records of data access), data protection (protection of data at rest and in motion), and perimeter security & security baseline (protection of network boundaries and systems).
--------	--

All software components must supplement and extend each other to ensure security, which is only ensured for components from the Hadoop set of components. Non-Hadoop components (i.e. components not part of the Hadoop set of components) might or might not fit, but the risk of incompatibilities and lack of long-term maintenance is quite high. Exception: As currently, i.e. at the time of writing, no Hadoop component exists covering pseudo-/anonymization, such component must be developed separately and integrated into the Hadoop installation.

ID: 3.98-11/2.0

Req 12	All non-security Hadoop components (i.e. all components allowing access to data) must use plugins enforcing the security policy.
--------	--

Some components can be used without enforcement of security policies. Such approach might be sensible in cases where only public, generally accessible data is used. Therefore, all non-security components must use according plugins, allowing for policy enforcement.

2.4. Encryption of Data in Rest and in Motion

Req 13	Credentials must be encrypted during transmission.
--------	--

For example, HTTPS, SSH, or SFTP must be used in favour of HTTP, Telnet, or FTP.

ID: 3.98-13/2.0

Req 14	Transmission of data between components in regard to system management or security management must be encrypted.
--------	--

All such data, e.g. data transmitted by Apache Ambari to its clients, must be protected in regard to confidentiality and integrity. Therefore, for such communications, TLS must be configured and used.

ID: 3.98-14/2.0

Req 15	Data transmitted to or accessed from the outside of the Hadoop installation must be encrypted.
--------	--

To ensure confidentiality and integrity of sensitive data, all such data must be encrypted.

ID: 3.98-15/2.0

Req 16	Administrative access to any Hadoop component must be encrypted.
--------	--

To ensure confidentiality and integrity of sensitive data, all such data must be encrypted.

ID: 3.98-16/2.0

Req 17	Data transmitted within the Hadoop installation must be encrypted if classified “confidential” or “strictly confidential”. Such encryption can be omitted in case the underlying cables, physical layer infrastructure and data link layer infrastructure is used exclusively by Hadoop systems within one Hadoop network segment and such infrastructure does not cross site boundaries.
--------	---

To ensure confidentiality and integrity of sensitive data, all such data must be encrypted.

The preferred option to do so is to use end-to-end encryption, i.e. encryption should be performed on the endpoints of the Hadoop components. Nevertheless, such data can be transmitted using encrypted site-to-site VPNs, if the VPN endpoints terminate within the Hadoop installation or the firewall separating the Hadoop installation's network segments. The exception stated addresses cases where data is transferred from one node to another and both respective nodes are housed within the same rack, racks in the same data centre room or adjacent data centre rooms, and a private networks infrastructure with own cables and switches is installed.

ID: 3.98-17/2.0

Req 18	Data classified “confidential” or “strictly confidential” must be stored encrypted on all parts of the Hadoop installation.
--------	---

To ensure confidentiality and integrity of sensitive data, all such data must be encrypted. Compliance with this requirement will also ensure compliance with standards requiring to keep data inaccessible even if some hat physical access to a server, e.g. PCI-DSS or HIPPA.

ID: 3.98-18/2.0

2.5. Data, Data Access, and Roles

Req 19	During transformation of data from the raw data stage into cleansed data stage, all data must be labelled.
--------	--

As authorization of data access depends on role, stage, and label, all non-raw data must be labelled.

ID: 3.98-19/2.0

Req 20	Labelling of data should reflect information classification principles as defined by corporate policy.
--------	--

Each type of label must be part of one class according to corporate policy.

ID: 3.98-20/2.0

Req 21	Data not pseudo- or anonymized must be stored in the Hadoop file system separately from other data.
--------	---

Such separation can be ensured by implementing stringent staging, which stores data of different classification in different directories, combined with according policies or ACLs.

ID: 3.98-21/2.0

Req 22	The Hadoop file system must be configured to allow access only if such access is controlled via the central security management system.
--------	---

E.g., data nodes must not accept requests access to data from users directly logged in a data node. It is understood that administrative users might have more extensive access rights, possibly allowing for circumvention of the central management system.

ID: 3.98-22/2.0

Req 23	Assuming roles for production, testing, and development is mutually exclusive, e.g. one person cannot assume roles for both production and testing or for all three group of roles, within one Hadoop-installation.
--------	---

Implementation of this requirement hinders circumvention of the role concept. One person can be developer on a development installation and assuming a production role on a production system.

ID: 3.98-23/2.0

Req 24	User must not access data nodes directly, unless for system administrative purposes. Such access must take place via edge nodes only.
--------	---

For the defined Hadoop security framework being effective it is essential to strictly gate any non-administrative access through the defined gates, i.e. the edge nodes, only.

ID: 3.98-24/2.0

Req 25	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 for the transmission of data must be recognized and effective 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

ID: 3.01-15/9.0

Req 26	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 for the storage of data must be recognised and effective for this purpose and must not have any 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

ID: 3.01-14/9.0

Req 27	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 a user account. A user account consists of a unique user name and associated authentication features. 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

This requirement also applies without restriction to any machine access to the system (Here, implementation is usually carried out through the use of so-called "technical user accounts", also known as "machine-2-machine" or "M2M" accounts).

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

For this requirement the following threats are relevant:

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

ID: 3.01-19/9.0

Req 28	User accounts must be protected with at least two authentication features from different factors.
--------	---

All user accounts of the system must be protected from unauthorized use.

This is to be achieved by using authentication features that allow the legitimate user to be uniquely identified.

Authentication features are differentiated into superordinate groups, so-called "factors":

- Factor KNOWLEDGE: "something that only the legitimate user knows"
- Factor POSSESSION: "something that only the legitimate user possesses"
- Factor INHERENCE: "something that only the legitimate user is"

At least two authentication features must be combined.

These must come from different factors ("Multi-Factor Authentication"; short: "MFA").

A combination of authentication features of the same factor (example: two different PINs) is not allowed.

Examples of acceptable authentication features

- from the KNOWLEDGE factor: PINs, passphrases (passwords*)
- from the POSSESSION factor: smart cards, FIDO 2 Level 2 security keys, device-bound passkeys, cryptographic keys stored in a TPM/HSM, OTPs (One-Time Passwords) generated in an authenticator app that has been approved for use in Deutsche Telekom
- from the INHERENCE factor: biometric features such as fingerprints, iris or retina scans, face recognition

*) Important note on the use of passwords

Passwords must be strictly avoided as an authentication feature.

They are only acceptable if more secure alternatives cannot be implemented.

Authentication features that can be easily forged or manipulated are generally not permitted.

These include:

- Phone numbers and caller IDs
- IP addresses
- Geographical locations
- VPN affiliations and network information
- Status of clients, such as workstation systems or mobile devices

Note on technical user accounts

Technical user accounts used for machine access (also known as "machine-to-machine" or "M2M" accounts) are exempt from the obligation to use multi-factor authentication, as this is usually not feasible there.

Motivation: User accounts that are not protected by appropriate authentication features can be abused by an attacker to gain unauthorized access to a system and the data and applications stored on it. Due to a variety of common attack methods, the protection of user accounts by a single authentication feature is now considered weak, especially if the classic password is used here. A combination of authentication features from different factors significantly increases security for user accounts.

Implementation example: Examples of MFA

MFA Combination	Description	Application Note
Telekom "myCard" plus PIN	1st Factor: Certificate implemented in hardware ("Possession") 2nd Factor: PIN ("Knowledge")	(none)
FIDO2 Level 2 Security Key plus fingerprint	1st Factor: Crypto key stored in hardware ("Possession") 2nd Factor: Biometric feature ("Inherence")	Biometric feature must be pre-configured, otherwise the biometric sensor will only work as a "button" and does not provide the 2nd factor ("Inherence").
Device-bound Passkey in WH4B* on Nucleus PC plus PIN	1st Factor: Crypto key stored in hardware ("Possession") 2nd Factor: PIN ("Knowledge")	(none)
Device-bound Passkey in Secure Enclave on iOS Device plus FaceID	1st Factor: Crypto key stored in hardware ("Possession") 2nd Factor: Biometric feature ("Inherence")	(none)
Password plus OTP App approved for use (e.g. MS Authenticator)	1st Factor: Password ("Knowledge") 2nd Factor: specific generated One Time Password ("Possession")	Passwords are only eligible if more secure alternatives cannot be implemented. The synchronization of the authenticator app via a cloud service as well as sharing (e.g. via a password manager) the OTP is not permitted.

*) WH4B: "Windows Hello for Business"

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: 3.01-20/9.0

Req 29 Privileged user accounts must be protected with at least two authentication features that come from different factors and provide increased resilience against phishing attacks.

A privileged user account is a user account that has advanced permissions within a system. Advanced permissions allow access to configuration settings, features, or data assets that are not available to regular users of the system. In direct dependence on the specific tasks performed by means of a privileged user account within a system, the advanced permissions assigned may be specifically limited or may include fully 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; Assign permissions; password resets)
- Accounts that are authorized to legitimize, trigger, or stop business-critical operations
- Accounts with access to data classified as SCD (Sensitive Customer Data) in the interest of the 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 entire database)

To protect privileged user accounts, at least two authentication features must be combined. These must come from different factors ("KNOWLEDGE", "POSSESSION" or "INHERENCE"). A combination of authentication features of the same factor is not permitted. At least one of the authentication features must not be vulnerable to phishing attacks.

In a phishing attack, the legitimate user is trickily manipulated to reveal his login data to the attacker, for example by means of fake login pages. A phishing attack can also be carried out against a combination of multiple authentication features.

Phishing-resilient authentication features prevent attacks based on their technical implementation. These authentication features use cryptographic methods and create a strong bond between the user, the device and the service. They also require manual user interaction during authentication, which cannot be automated; for example, to submit a fingerprint scan to a hardware token.

Considered not vulnerable to phishing attacks are

- FIDO2 Level 2 Security Keys
- Certificates implemented in hardware

Important note on the use of passwords

Passwords must be strictly avoided as an authentication feature.

They are only acceptable if more secure alternatives cannot be implemented.

Note on technical user accounts

Technical user accounts used for machine access (also known as "machine-to-machine" or "M2M" accounts) are exempt from the obligation to use multi-factor authentication, as this is usually not feasible there.

Motivation: Privileged user accounts pose an increased risk to the security of a system. If an attacker successfully compromises such a user account, they will be granted broad privileges that allow them to take control of the system or system parts, disrupt system functions, view/manipulate processed data, or influence business-critical operations. The combination of multiple authentication features of different types already significantly minimizes the risk of a user account being compromised, but for privileged user accounts, special attention must be paid to resilience against phishing attacks. The phishing attacks that can be observed are becoming increasingly tricky and difficult for users to detect, so that multifactor authentication without increased resilience against phishing can no longer be considered sufficient for privileged user accounts.

Implementation example: Examples of MFA with increased resilience to phishing attacks

MFA Combination	Description	Application Note
Telekom "myCard" plus PIN	1st Factor: Certificate implemented in hardware ("Possession") 2nd Factor: PIN ("Knowledge")	(none)
FIDO2 Level 2 Security Key plus fingerprint	1st Factor: Crypto key stored in hardware ("Possession") 2nd Factor: Biometric feature ("Inherence")	Biometric feature must be pre-configured, otherwise the biometric sensor will only work as a "button" and does not provide the 2nd factor ("Inherence").
FIDO2 Level 2 Security Key plus PIN	1st Factor: Crypto key stored in hardware ("Possession") 2nd Factor: PIN ("Knowledge")	(none)

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: 3.01-21/9.0

Req 30 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: 3.01-22/9.0

Req 31	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 data and system areas in need of protection.

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: 3.01-23/9.0

2.6. Basic System Configuration

Req 32	Hadoop installation must be configured to run in "secure mode".
--------	---

By default Hadoop runs in non-secure mode in which no actual authentication is required. By configuring Hadoop to run in secure mode, each user and service needs to authenticate using Kerberos in order to use Hadoop services.

ID: 3.98-32/2.0

Req 33	Big Data installations is the ability to quickly add and remove nodes. Assignment of configuration data like IP addresses must be centrally controlled e.g. by DHCP.
--------	--

Key advantage of Big Data installations is the ability to quickly add and remove nodes to adapt to current processing and storage requirements. To do so in a controlled and secure way and to ensure consistent and central system management, manual assignment of configuration data like IP addresses is not feasible. Therefore, DHCP should be used for data nodes.

ID: 3.98-33/2.0

Req 34 For remote administration of nodes of a Hadoop installation Secure Shell (SSH) must be used via Admin-LAN, but no other protocols. Exception: In case the Hadoop distribution used provides integrated means for administration to ensure central administration, those means can be used, even if not SSH, as long as they use encrypted protocols.

To ensure effective security of Big Data installations central administration and management is essential. Therefore, other protocols than SSH are acceptable, as long as the protocols used transmit user data encrypted.

ID: 3.98-34/2.0

Req 35 Each Hadoop component, i.e. each daemon, must run as a different Unix user.

HDFS-, YARN-, MapReduce JobHistory- and other daemons must all run as different Unix users.

ID: 3.98-35/2.0

Req 36 Unix users of Hadoop components, i.e. each daemon, must not be the root user.

As root is the privileged system user it must not be used for Hadoop components as vulnerabilities in a component can result in much greater damage and in circumvention of authorization policies if the root user is used.

ID: 3.98-36/2.0

Req 37 Hadoop components, i.e. each daemon, must share a common Unix group.

Recommended is the use of "hadoop" for the name of such group.

ID: 3.98-37/2.0

Req 38 All hosts of a Hadoop installation must be time-synchronized with trusted timeservers.

As Kerberos is essential for Hadoop security and as Kerberos is time-sensitive, all hosts must be time-synchronized with trusted timeservers. Such trust will usually be accomplished only by using timeservers controlled by a Deutsche Telekom group company. If clocks differ by five or more minutes, authentication will fail.

ID: 3.98-38/2.0

Req 39 Certificates used to secure access to components from outside a production Hadoop installation must be issued by a standard certification authority.

While the use of self-signed certificates or other certificates not signed by a recognized certification authority is technically possible, for production systems only certificates issued by standard certification authorities must be used, if used for securing access from outside of the respective Hadoop installation.

As such standard certification authority a globally operating one should be used, e.g. TeleSec of Deutsche Telekom.

ID: 3.98-39/2.0

Req 40 Authentication of users of management tools must be based on existing central directory services.

In most cases, such central directory service will be an existing Active Directory server used for authentication of users accessing office systems like workstations or of users accessing production systems.

It is understood that administrators must login as local administrators on some systems for some tasks. Nevertheless,

login on a management system using existing central directory services must occur before.

ID: 3.98-40/2.0

Req 41	The central security management of Hadoop (e.g. Apache Ranger, used by Hortonworks) must synchronise its user base via TLS.
--------	---

Typically, many users will directly or indirectly have access to parts of the Data Lake. Therefore, implementing and maintaining another user directory in addition to the one used for authentication to use office infrastructure components will not be feasible; the solution being is to synchronise with an existing directory.

ID: 3.98-41/2.0

Req 42	The central security management of Hadoop should synchronise its user base continually; such synchronisation must take place at the very least once per day. Synchronisation of administrative user can be performed manually as required.
--------	--

Such synchronisation is necessary to have changes in the user base, e.g. by termination of work contracts and new hires.

ID: 3.98-42/2.0

Req 43	Auditing must be enabled.
--------	---------------------------

To allow for investigation of misuse and to detect loopholes in the use concept, enabling of auditing is required. Auditing must be enabled at least on the central security management component (which is Apache Ranger on Hortonworks installations).

ID: 3.98-43/2.0

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

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

- There is a continuous analysis and correlation of the logging data provided by connected IT/NT Systems.
- The identification of threatening activities and ongoing attacks takes place on the basis of a continuously updated detection pattern database
- Integrated alerting processes trigger the initiation of countermeasures in the event of a find.
- Security experts are actively involved in the operation of the solution as well as the evaluation of alarms

Such a solution is not the same as a log server in terms of its function. Accordingly, the provision to a detection and response solution must always be in addition to an existing forwarding to a log server. An exception exists if the log server used provides the logging data to a detection and response solution immediately after receipt. In this case, it is sufficient for the system to forward the logging data only to the log server.

The logging data must be provided to the detection and response solution within 5 minutes ("near-real-time") from the time it is generated in the system.

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

Motivation: IT/NT systems are constantly exposed to security threats and attacks. By means of a central analysis and the correlation of recorded system events of entire IT landscapes in almost real time, current threats and ongoing attacks can be detected even beyond system boundaries and appropriate countermeasures can be initiated promptly.

The prompt delivery of logging data to the detection and response solution is essential for its effectiveness.

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

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

Example 3: Network components provide events to a SIEM that make it possible to detect suspicious communication behavior such as network scans, unauthorized data flows between network segments, or active connections or connection attempts to suspicious destinations on the Internet. In particular, gateway routers from the CNDTAG provide "Netflow" logs, while forward proxies report access to domains and/or IP addresses and firewalls log blocked connection attempts.

Example 4: Systems designed on the basis of cloud-native functions generate logging data directly from these functions using the cloud's own logging and monitoring infrastructure. In such a system, the generation of security-relevant logging data is specifically activated in the corresponding configuration options of the cloud-native functions used. The resulting logging data is continuously transferred to a SIEM (Security Information and Event Management) for further analysis, intrusion detection and alarm handling. Some cloud providers already provide some attack pattern detection in the cloud's own monitoring infrastructure and generate specific cloud alerts from it. These are also transferred to the SIEM.

Example 5: A system is based on a SaaS ("Software as a Service") application from a cloud provider. Typically, in the SaaS model, the cloud provider is responsible for all operational aspects of the application, including security management. The cloud customer itself has no influence on the underlying technical infrastructure or the operation of the application. Based on contractual agreements, recognized certificates, and documented self-disclosures from the cloud provider, it is formally verified that the cloud provider has implemented a security incident detection and response solution. Some cloud providers also generate dedicated logging data about security-relevant activities within the separate customer tenants of their SaaS application (e.g. changes to configuration settings, permission assignments or user accounts within the tenants) and make them available to cloud customers. These so-called audit logs are continuously transferred to a SIEM (Security Information and Event Management) for further analysis, attack detection and alarm handling.

For this requirement the following threats are relevant:

- Unnoticeable feasible attacks

ID: 3.01-37/9.0

Req 45 (Default) databases that are not required must be deleted on the database system.

Motivation: When installing database systems, test or practice databases are often installed which are not required once the database goes productive. In the past, vulnerabilities of these test databases have become known which allow an attacker to gain privileged rights on the database system. Such knowledge enables an attacker to access the database system. Therefore, all databases that are not required shall be deleted.

For this requirement the following threats are relevant:

- Unauthorized use of services or resources

ID: 3.16-7/9.0

Req 46 A database service must not run with root or administrative rights.

Motivation: If a database server runs with highly privileged operating-system rights (root, administrator), an attacker can completely take over a database system by exploiting a vulnerability (access to the file system or execution of files).

For this requirement the following threats are relevant:

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

ID: 3.16-18/9.0

Req 47 (Default) users and (default) roles that are not required must be deleted.

Motivation: When installing a database system, a large number of users and roles are automatically installed, which are not required to operate the database. These are envisaged, for instance, for practice and test databases. Knowledge of default users and roles allows an attacker to gain (privileged) access to the database system. Therefore, users and roles that are not needed for database functions shall be deleted.

For this requirement the following threats are relevant:

- Unauthorized access to the system

ID: 3.16-8/9.0

Req 48 Outputs and messages must not disclose information on internal structures of the system.

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

In general, this concerns information on

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

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

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

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

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

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

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

Implementation example: [Example 1]

Deactivation of the display of the product name and the installed version of a Web server in its delivered error web pages.

[Example 2]

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

For this requirement the following threats are relevant:

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

ID: 3.01-9/9.0

Req 49	All database services must be set up in accordance with the least privilege principle on operating-system level.
--------	--

Motivation: By using the least privilege principle, the risk of system corruption can be reduced.

For this requirement the following threats are relevant:

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

ID: 3.16-17/9.0

Req 50	Database functions which permit access to operating-system files must be deleted or deactivated.
--------	--

Motivation: Some database systems allow access to the operating-system level via special stored procedures or SQL functions. A multitude of vulnerabilities of these procedures are known which allow an attacker to gain unauthorized access to a system or execute malware on the target system. These procedures are often provided with unnecessarily high, mainly privileged rights or allow anyone to execute programs without authentication. For example, malware exploits the procedures MS-SQL: xp_cmdshell and Oracle: utl_tcp.

For this requirement the following threats are relevant:

- Unauthorized use of services or resources

ID: 3.16-20/9.0

Req 51	Database functions which allow access to other network services (e.g. ,SMTP, HTTP, SNMP, FTP etc.), must be deleted or deactivated.
--------	---

Motivation: Some database systems provide functions that are normally offered by an application server. For instance it is possible via special stored procedures to send e-mails or launch web queries to external systems.

For this requirement the following threats are relevant:

- Unauthorized use of services or resources

ID: 3.16-21/9.0

Req 52	Database functions which allow access to the operating system level or network services, must not be accessible by the roles / groups "Public" and / or "Everyone".
--------	---

Motivation: Withdrawing rights can prevent the use of database functions to gain unauthorized access to network services and the operating-system level. Functions to execute operating-system commands and to use network services are thus only available to authorized users, and not to everyone.

For this requirement the following threats are relevant:

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

ID: 3.16-22/9.0

Req 53	The operating-system rights for the database files and directories (program, control, trace and log files) must be assigned exclusively to the database system's operating-system account.
--------	--

Some database systems store sensitive account data in plain text in trace or log files. Therefore, access rights to sensitive files and directories on the database system MUST be set so that non-administrative users do not have any read, write or execution rights.

Access to essential system files and directories on the database system must be reserved for the database system's user account.

Motivation: By implementing restrictive access rights, the risk of manipulation can be substantially reduced.

For this requirement the following threats are relevant:

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

ID: 3.16-23/9.0

Req 54	Database accesses between various database systems must be comply with the least privilege principle.
--------	---

Motivation: Limiting the access rights reduces the attack surface.

For this requirement the following threats are relevant:

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

ID: 3.16-24/9.0

Req 55	Accesses to database systems, as well as critical database procedures and database content must be logged.
--------	--

Secure, traceable database operation requires important operating information to be logged. This includes, for instance, the logging of failed login attempts to uncover possible intrusion attempts.

Logging of security-relevant user actions shall comply with national legislation currently in force. When implementing measures resulting from this Requirement, the applicable participation rights of the responsible employee representatives/trade unions as well as the works and collective agreements shall be observed.

For this requirement the following threats are relevant:

- Unnoticeable feasible attacks

ID: 3.16-26/9.0

Req 56	Database accesses between various database systems must be made via an individual user account.
--------	---

Motivation: Accesses shall be made via a special user. Only in this way can access rights be individually restricted to the absolute minimum level.

For this requirement the following threats are relevant:

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

ID: 3.16-25/9.0

Req 57	Important database services and instances must be monitored continually for misuse scenarios.
--------	---

The monitoring of user actions for misuse shall comply with national legislation currently in force (for details see the "Security Requirement on Misuse Detection").

There must be a case study regarding the type of service/user data/number of entries and possible misuse scenarios to define the necessary level of monitoring of the cases. Insider threats

- Anomaly detection
- Accidental exposure
- Misconfiguration
- Vulnerable application
- Machine Language models
- Insecure data in transit
- ...

Motivation: There are many conceivable ways to misuse database systems. Users generate an unusually high data usage rate or operate at unusual times of day. Attackers utilize unusual and critical commands for database queries, as well as tools and malware to extend their rights. To detect misuse, database systems should be continually monitored for misuse scenarios, e.g. by means of database triggers and log monitoring.

For this requirement the following threats are relevant:

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

ID: 3.16-32/9.0

Req 58	Mass data exfiltration must be detected if more than 100K customer data entry and DPC2 or DPC3 is present.
--------	--

Mass data exfiltration detection is a critical control that monitors for unusually large volumes of data being transferred internally or leaving the corporate network or database systems. When more than 100K customer data records - particularly classified as DPC2 or DPC3 - are accessed, copied, or transferred, automated alerts should be triggered. This threshold-based detection should be integrated with SIEM and User and Entity Behavior Analytics (UEBA) tools to identify deviations from normal access patterns. The system should correlate access context (user role, time, data sensitivity, and transfer channel) to distinguish legitimate bulk operations from potential exfiltration attempts. Upon detection, predefined incident response workflows should immediately isolate the activity, notify the security operations center, and preserve forensic evidence for investigation.

Motivation: Customer data are one of the most valuable assets of the company. Therefore we must take every measures to avoid mass data leakage, malicious act or misuse. For that purpose we need to know how we are handling and using our datasets.

Implementation example: Technical guidance and implementation details are also provided on that page and are continuously maintained:

<https://yam-united.telekom.com/pages/psa/apps/wiki/wiki-sicherheitsanforderungen/list>

For this requirement the following threats are relevant:

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

ID: 3.16-33/9.0

2.7. HDFS Configuration

Req 59	For HDFS, a umask of 077 must be set.
--------	---------------------------------------

While any access to data in the HDFS should be controlled by a central security solution anyway, it is good practice to implement the security principle of “defense in depth” by setting according file permissions. In case Apache Ambari is used, such setting can be performed by setting “Service” -> “HDFS” -> “Advanced hdfs-site” and setting “fs.permissions.umask-mode = 077”.

ID: 3.98-59/2.0

Req 60	The system user used to run the HDFS daemon must not be used for any other service.
--------	---

The system user used to run the HDFS daemon must not be used for any other service.

ID: 3.98-60/2.0

Req 61	Fallback for authentication to local authentication mechanisms must be disabled.
--------	--

Fallback for authentication to local authentication mechanisms must be disabled. In case Apache Ambari is used, such setting can be performed by setting “Service” -> “HDFS” -> “Advanced ranger-hdfs-security” and setting “xasecure.add-hadoop-authorization=false”.

ID: 3.98-61/2.0

2.8. Kerberos Configuration

Req 62	Kerberos must be used as a basis for identification, authentication, and authorization; i.e., Hadoop “kerberos” authentication mechanisms must be used.
--------	---

Identification, authentication, and authorization are essential parts of granting access to data. As Kerberos is the supported service to do so by Hadoop, it must be used.

In Hadoop available is also the “simple” mechanism, which is insufficient in regard to security for production environments. It might be an option on small test or development systems if only synthetic data is used; in such case, even pseudo- or anonymized data must not be used.

ID: 3.98-62/2.0

Req 63	The KDC must be installed on two or more servers, i.e. in a master/slave configuration.
--------	---

As Kerberos is essential for security functions within a Hadoop installation, it must be configured on two or more servers to allow for redundancy.

ID: 3.98-63/2.0

Req 64	/etc/krb5.conf and /var/kerberos/krb5kdc/kdc.conf must contain installation-specific values allowing for secure operation.
--------	--

Those files can be updated on the KDC host, krb5.conf can then be copied to all other servers. The Deutsche Telekom Security Requirement Cryptographic Algorithms and IP Security (IPSec), must be honored. Care must be taken to remove potentially insecure cryptographic algorithms, cp. “supported encetypes” below. For example, one might need to delete “RC4” from this list.

Implementation example: Example of a krb5.conf:

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log[libdefaults]
dns_lookup_realm = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = DATA.LAKE
default_ccache_name = KEYRING:persistent:%{uid}[realms]
DATA.LAKE = {
  kdc = kdc.data.lake
  admin_server = kdc.data.lake
}[domain_realm]
.data.lake = DATA.LAKE
data.lake = DATA.LAKE
```

Example of a kdc.conf:

```
[kdcdefaults]
kdc_ports = 88
kdc_tcp_ports = 88[realms]
DATA.LAKE = {
  master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_encetypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal camel-
lia256-cts:normal camellia128-cts:normal
  default_principal_flags = +preauth
  max_renewable_life = 7d 0h 0m 0s
}
```

ID: 3.98-64/2.0

Req 65 Every user of a Hadoop-service and every service on each host used by a Hadoop user must have a unique Kerberos user principal name (UPN) and service principal name (SPN), resp.

The same services running on different hosts must use different SPNs, because if credentials on one node are compromised, other nodes are not affected, and because Kerberos falsely detect reply attacks if by change different services with the same SPN request and use tickets at the same time.

ID: 3.98-65/2.0

Req 66 The FQDN must be appended to a service principal's name.

Adding the FQDN to a service principal's name ensures uniqueness.

ID: 3.98-66/2.0

Req 67 For each Hadoop installation, a different Kerberos realm must be used.

While it is technically possible to use the same Kerberos realm for production, test, and development installations, administrators and user will sooner or later assume they are working in e.g. a test installation, while actually they are working in the production setup. Such mistakes can mitigate by using different realm names for different installations.

ID: 3.98-67/2.0

Req 68 All UPNs and SPNs must be registered in a network-based directory service such as Active Directory or OpenLDAP.

All users of a cluster must be provisioned on servers of a cluster. Two options are available to do so: Provisioning all users locally, e.g. in the `/etc/passwd`, or using a network-based directory service. In practice, consistency of accounts can only be ensured if using network-based directory services.

ID: 3.98-68/2.0

Req 69 Non-administrative users must not be enabled for shell access.

Regular users must not utilize a system's shell access. Best practice on Unix-like systems is to assign each such the default shell `/sbin/nologin` and to use `AllowUsers`, `DenyUsers`, `AllowGroups`, and `DenyGroups` settings in `/etc/ssh/sshd_config`.

ID: 3.98-69/2.0

Req 70 Keytab files must be accessible only by the system user running the respective service. Such access must only be read access.

File permissions of keytab files must be set to 400, the owner of the keytab file must be the system owner running the respective service.
As the keytab files contain credentials unencrypted, those files must be secured.

ID: 3.98-70/2.0

Req 71 „Java Cryptographic Extension“ must be installed on all nodes.

This requirement is a hint for administrators setting up a Hadoop installation. This package is required for cluster oper-

ation.

ID: 3.98-71/2.0

2.9. Knox Configuration (Hortonworks)

Req 72	For all but administrative access, Knox must be used for edge nodes allowing access to data in the Data Lake.
--------	---

The Apache Knox Gateway („Knox“) is a system to extend Hadoop services to users outside of a Hadoop installation while ensuring enforcement of security policies. Further, Knox simplifies security for users who access data from the Data Lake and who execute jobs.

Knox simplifies access by extending Hadoop's REST/HTTP services by encapsulating Kerberos.

Knox enhances security by exposing Hadoop's REST/HTTP services without revealing Hadoop installation network details and by providing TLS out of the box.

Knox centralizes controls by central enforcement of REST API security.

Knox integrates with Hadoop's Kerberos installation, which in turn integrates with a company's identity management solutions like Active Directory.

As of the time of writing, Knox supports the following Hadoop services, subject to specific versions:

- YARN
- WebHDFS
- WebHCat/Templeton
- Oozie
- HBase/Stargate
- Hive (via WebHCat)
- Hive (via JDBC)

ID: 3.98-72/2.0

Req 73	The security directory, \$gateway/data/security, and its contents must be readable and writeable only by the operating system user running the Knox component.
--------	--

In this directory the master secret and other confidential information is stored. Therefore, care must be taken to ensure security of this data.

ID: 3.98-73/2.0

Req 74	Web application security filters available for Knox must be installed and used.
--------	---

As of the time of writing, one filter addressing cross site request forgery (CSRF) exists.

ID: 3.98-74/2.0

Req 75	All requests to Knox must be documented to allow for both developers knowing how to access Knox and administrators of possibly used web application firewalls to configure their systems.
--------	---

As of the time of writing, one filter addressing cross site request forgery (CSRF) exists.

ID: 3.98-75/2.0

Req 76	Multiple database instances may only be operated together on one operating system instance (with physical or virtualized hardware) if they • have the same low protection requirements, • are under a single customer administrative authority and • are operated, in terms of administration, by the same group of people.
--------	--

Motivation: Operating multiple database instances together increases the risk of users gaining unauthorized access to instances for which they are neither technically nor administratively responsible.

Implementation example: To separate the individual database instances, administrators can use physical server hardware as well as virtualization solutions.

For this requirement the following threats are relevant:

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

ID: 3.16-9/9.0

Req 77	(Default) passwords on database systems must be changed.
--------	--

Motivation: Many database manufacturers provide various users with (default) passwords (sa/blank; DBSN-MP/DBSNMP, etc.) during installation. An attacker can exploit the existence of these (default) passwords to obtain what tends to be privileged access to the database system.

For this requirement the following threats are relevant:

- Unauthorized access to the system

ID: 3.16-16/9.0