

Sicherheitsanforderung

SAP Netweaver Umgebungen

Deutsche Telekom Gruppe

Version	13.0
Datum	01.07.2025
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 3.22	Dokumententyp Sicherheitsanforderung
Version 13.0	Stand 01.07.2025	Status Freigegeben
Fachlicher Ansprechpartner Telekom Security psa.telekom.de	Gültigkeitsdauer 01.07.2025 - 30.06.2030	Freigegeben von Stefan Pütz, Leiter SEC-T-TST

Zusammenfassung

Das Requirement ist gültig für:

- Netweaver SAP-Systeme
- SAP HANA
- SAProuter
- SAP Web Dispatcher
- Fiori-Applikation-Server,
- Fiori-Apps

Copyright © 2025 by Deutsche Telekom AG.

Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Inhaltsverzeichnis

1.	Einleitung	5
2.	Anforderungen	6
2.1.	Allgemeine Aspekte	6
2.2.	Netz- und Kommunikationssicherheit	15
2.3.	Kryptographie ----- Wenn die Anforderungen 19 bis 44 erfüllt sind, entspricht das der Umsetzung der Sicherheitsanforderung "Kryptographische Algorithmen und Sicherheitsprotokolle" (3.50) für SAP Netweaver Umgebungen.	17
2.4.	Transport Layer Security (Umsetzungsbeispiel für die Requirements 36 - 40 in Req. 17 eingefügt)	36
2.5.	Logging	41
2.6.	Autorisierungsmanagement	47
2.7.	Allgemeine Parameter für Netweaver (ABAP und Java)	59
2.8.	Absicherung des ABAP-Stacks	60
2.8.1.	ABAP-Stack: Betriebssystemverzeichnisse	60
2.8.2.	ABAP-Stack: Standardbenutzer SAP	65
2.8.3.	ABAP-Stack: Benutzer	68
2.8.4.	ABAP-Stack: Berechtigungen	69
2.8.5.	ABAP-Stack: Logging	75
2.8.6.	ABAP-Stack: Kommunikationssicherheit	78
2.8.7.	ABAP-Stack: Parameter	81
2.8.8.	ABAP-Stack: ICM (Internet Communication Manager)	83
2.9.	Absicherung des Java-Stacks	93
2.9.1.	Java-Stack: Betriebssystemverzeichnisse	93
2.9.2.	Java-Stack: Kommunikation	98
2.9.3.	Java-Stack: Benutzer und Authentifikation	98
2.9.4.	Java-Stack: Autorisierungsmanagement	102
2.9.5.	Java-Stack: Transport Management System (TMS)	104
2.9.6.	Java-Stack: ICM (Internet Communication Manager)	104
2.10.	SAProuter	114
2.11.	Anforderungen SAP Web Dispatcher	118
2.12.	Allgemeingültige Vorgaben für Datenbanken	127
2.13.	Oracle	131
2.14.	HANA	134
2.14.1.	HANA: Datenbank User Rollen und Berechtigungen	137
2.14.2.	HANA: Netzwerkschnittstelle	139
2.14.3.	HANA: Applikation und File System	139
2.14.4.	HANA: Multitenant	142
2.14.5.	HANA: Extended Services (XS) Engine	144
2.14.5.1.	HANA: Benutzer	144
2.14.5.2.	HANA: Extended Services (XS) Engine Benutzer auf Betriebssystemebene	146
2.15.	Anbindung von externen Webservices	146
2.16.	Der Netweaver als Webapplikation	154
2.16.1.	Content-Management-Systeme (CMS)	172
2.17.	Proxy-Server	174

2.18.	SAP Fiori Applikation Server	177
2.19.	SAP Fiori-Apps	177
2.20.	Cloudnutzung	180
2.21.	SAP Cloud Plattform als PaaS	184
2.21.1.	Anforderungen Cloud Connector (nur wenn ein SaaS in der SAP-Cloud benutzt wird)	185
2.22.	Microsoft Azure Cloud via DTIT Landing Zone und Produkt "Enterprise"	186
2.22.1.	Zentrale Management Subscription "SAPaaS central Management Service"	190
2.23.	Google Cloud via DTIT Landing Zone mit Produkt "SAP on GCP"	191
2.23.1.	Zentrales Management Projekt "SAPaaS central Management Service"	194
2.24.	FCI Cloud via Landing Zone mit Produkt "SAP on FCI"	195
2.24.1.	Zentrale Management Services (administrativer Zugriff)	198

1. Einleitung

Dieses Dokument wurde auf Basis der Vorgaben aus den im Konzern gültigen Sicherheitsrichtlinien erstellt. Die Sicherheitsanforderung dient u.a. als Grundlage zur Freigabe im PSA-Prozess. Sie dient darüber hinaus als Umsetzungsempfehlung für Einheiten, die nicht am PSA-Verfahren teilnehmen. Diese Anforderungen müssen bereits während der Planungs- und Entscheidungsprozesse berücksichtigt werden. Bei der Umsetzung der Sicherheitsanforderung ist das jeweils vorrangige nationale, internationale und supranationale Recht zu beachten. Sollte die Einhaltung der beschriebenen Anforderungen in individuellen Fällen nicht oder nur eingeschränkt umsetzbar sein, ist gemeinsam mit einem Sicherheits- und/oder Datenschutzexperten (entsprechend der betroffenen Anforderung) eine Risikobewertung vorzunehmen und mögliche alternative Schutzmaßnahmen abzustimmen.

SAP spezifisches:

SAP Hinweise und Links in diesem Dokument benötigen teilweise einen Account im SAPNET.

Hinweise zu den "Systemeigenschaften"

Bereich "Komponenten"

SAP ist eine "Serveranwendung".

Der SAProuter und der SAP Web Dispatcher sind keine "Netzelement / Infrastruktur" und / oder "Netzelement/ Infrastruktur mit HTTP GUI".

Fiori ist keine "Mobile Anwendung".

Bereich "Anwendungen"

Wenn es sich bei dem System ausschließlich um ein SAP Netweaver System handelt, müssen bei den Systemeigenschaften im PSA-Portal nach der Auswahl von "SAP" die Anwendungstypen "Webanwendung" und "Web Service" nicht mehr angekreuzt werden, da die Anforderungen zu diesen in diesem Dokument integriert sind.

Bei der Auswahl von dem Anwendungstyp "SAP" muss im Bereich "Datensicherheit" der Systemeigenschaften bei der Nachfrage nach der Datenbank und des Protokolls jeweils "keine" ausgewählt werden, da die Anforderungen zu diesen in diesem Dokument integriert sind.

Bereich "Betreiber"

Der SAP Support Zugang fällt nicht unter den Zugang von Fremdfirmen zum System. Anzukreuzen ist in dem Fall vom SAP Support Zugang "Nein, kein Zugang zum System durch Fremdfirma".

2. Anforderungen

2.1. Allgemeine Aspekte

Req 1 Nicht benötigte Dienste müssen deaktiviert werden.

Nach der Installation von Systemen und Softwareprodukten sind häufig herstellerseitig voreingestellte, lokale oder aus dem Netz erreichbare Dienste aktiv, die für den Betrieb und die Funktionalität des konkreten Systems im vorgesehenen Einsatzumfeld nicht erforderlich sind.

Auf einem System dürfen jedoch grundsätzlich nur die tatsächlich benötigten Dienste aktiv sein.

Entsprechend müssen unmittelbar nach der Installation sämtliche auf einem System nicht erforderliche Dienste vollständig deaktiviert werden. Es muss sichergestellt werden, dass diese Dienste auch nach einem Neustart des Systems weiterhin deaktiviert bleiben.

Motivation: Aktive Dienste, die nicht benötigt werden, vergrößern unnötig die Angriffsfläche eines Systems und in unmittelbarer Folge das Risiko einer erfolgreichen Kompromittierung. Dieses Risiko kann zusätzlich nochmals erhöht sein, falls - wie bei nicht benötigten Diensten häufig beobachtet - eine gezielte Betrachtung und Optimierung der Konfiguration hinsichtlich Sicherheit nicht ausreichend stattfindet.

Umsetzungsbeispiel: A: Check der laufenden Prozesse auf OS-Ebene mit administrativen User mit ...

- Unix: ps -ef
- Linux: ps -aux
- Windows: Taskmanager

B: Check der erreichbaren Dienste mit ...

- Unix: netstat -an und/oder systemctl --type=service --state=active
- Linux: netstat -taupen und/oder systemctl --type=service --state=active
- Windows: netstat -an

C: Check der Konfig:

- Um Anhaltspunkte dafür zu finden, ob der Service benötigt wird oder nicht!
- Wurde der Dienst vom Administrator angepasst?
- Ist es die Standardkonfiguration und der Dienst wurde einfach gestartet?

D: Deaktivieren Sie nicht benötigte Dienste und deren Ports.

ID: 3.22-1/13.0

Req 2 Die Erreichbarkeit von aktivierten Diensten muss eingeschränkt werden.

Ein bereitgestellter Dienst muss grundsätzlich auf allen Schnittstellen des Systems vollständig deaktiviert werden, über die eine Erreichbarkeit des Dienstes für den ordnungsgemäßen Betrieb des Systems nicht erforderlich ist. Die Deaktivierung ist primär durch eine entsprechende Konfiguration des Dienstes bzw. Betriebssystems umzusetzen. In Fällen, in denen die zu Verfügung stehenden Konfigurationsoptionen eine Deaktivierung auf einzelnen Schnittstellen nicht ermöglichen, darf stattdessen auch ein lokaler Filter ("Host Firewall") auf dem System eingesetzt werden, um Zugriffe auf den Dienst über nicht erforderliche Schnittstellen zu blockieren.

Die Erreichbarkeit eines Dienstes über die erforderlichen Schnittstellen muss zudem auf legitime Kommunikationspartner eingeschränkt werden. Die Einschränkung ist durch eine entsprechende Konfiguration des Dienstes bzw. Betriebssystems oder mittels eines lokalen Filters ("Host Firewall") umzusetzen. Alternativ darf diese Aufgabe an ein netzseitiges Filterelement ausgelagert werden, sofern sich das System in einem separaten Netzwerk-Segment befindet

und die Kommunikation mit diesem Segment ausschließlich über das netzseitige Filterelement möglich ist.

Motivation: Mittels Deaktivierung von Diensten auf Schnittstellen, über die eine Erreichbarkeit nicht notwendig ist, sowie durch die Einschränkung möglicher Kommunikationspartner, kann die Angriffsfläche, die ein System bietet, stark reduziert werden.

Umsetzungsbeispiel: Ein SNMP-Dienst, über den ein System gemonitort wird, ist ausschließlich auf der dedizierten Management-Netzwerk-Schnittstelle des Systems aktiviert. Über eine Firewall ist zusätzlich reglementiert, dass einzig das legitime Monitoring System der Infrastruktur-Umgebung diesen Dienst erreichen kann.

ID: 3.22-2/13.0

Req 3	Auf dem System darf ausschließlich nur benötigte Software eingesetzt werden.
-------	--

In den Hersteller-seitig bereitgestellten Installationsroutinen für Software sind oftmals Komponenten oder einzelne Bestandteile der Software als Standard-Installation vorausgewählt, die für den Betrieb und die Funktion eines spezifischen Systems nicht notwendig sind. Hierzu zählen auch Teile einer Software, die als Anwendungsbeispiele (z. B. Default-Web-Seiten, Beispieldatenbanken, Testdaten) mit installiert, aber typischerweise im Anschluss nicht verwendet werden.

Solche Komponenten müssen bei der Installation des Systems gezielt abgewählt (nicht mit installiert) werden oder - falls eine Abwahl während der Installation nicht möglich ist - unmittelbar im Anschluss entfernt werden.

Des Weiteren darf grundsätzlich keine Software auf einem System eingesetzt sein, die für den Betrieb, die Wartung oder die Funktion des Systems nicht erforderlich ist.

Motivation: Schwachstellen in der Software eines Systems sind Einfallstore für Angreifer. Durch die Deinstallation nicht benötigter Komponenten können die potentiellen Angriffsflächen deutlich reduziert werden.

ID: 3.22-3/13.0

Req 4	Nicht benötigte Funktionen in der eingesetzten Software und Hardware müssen deaktiviert werden.
-------	---

Während der initialen Installation von Software können zunächst Funktionen standardmäßig aktiviert worden sein, die für den Betrieb und die Funktionalität des konkreten Systems nicht notwendig sind. Funktionen sind meistens ein fester Bestandteil der Software, der nicht einzeln gelöscht oder deinstalliert werden kann.

Solche Funktionen müssen unmittelbar nach der initialen Installation über die Konfigurationseinstellungen der Software deaktiviert werden, so dass sie auch nach einem Neustart des Systems dauerhaft deaktiviert bleiben.

Bereits vor der Auslieferung oder während der initialen Inbetriebnahme sind möglicherweise auch in der Hardware zunächst Funktionen standardmäßig aktiviert worden, die für den Einsatz des konkreten Systems nicht benötigt werden. Solche Funktionen, wie beispielsweise nicht benötigte Schnittstellen, müssen ebenfalls unmittelbar nach der initialen Inbetriebnahme dauerhaft deaktiviert werden.

Motivation: Die Hardware oder Software eines Systems enthält oft aktivierte Funktionen, die nicht genutzt werden. Solche Funktionen können eine unnötige Angriffsfläche für Manipulationen darstellen. Des Weiteren entsteht darüber ein Potential, dass unautorisiert auf Bereiche oder Daten des Systems zugegriffen werden kann.

Umsetzungsbeispiel: [Beispiel 1]

Deaktivierung von Debugging-Funktionen in einer Software, die im Falle der Entstörung genutzt werden, aber während des normalen Betriebs nicht aktiv sein müssen.

[Beispiel 2]

Deaktivierung von ungenutzten Netzwerk-Schnittstellen eines Servers.

ID: 3.22-4/13.0

Req 5	Eingesetzte Software muss aus vertrauenswürdigen Quellen bezogen und auf Integrität geprüft werden.
-------	---

Die auf dem System eingesetzte Software muss aus vertrauenswürdigen Quellen bezogen und vor der Installation auf ihre Integrität geprüft werden.

Diese Anforderung bezieht sich auf sämtliche Arten von Software:

- Firmware sowie Microcode für Hardware-Komponenten
- Betriebssysteme
- Software-Bibliotheken
- Applikationen und Anwendungssoftware
- vorintegrierte Anwendungslösungen, wie z. B. Software Appliances oder Container

sowie weitere Software, die ggf. zum Einsatz kommt.

Vertrauenswürdige Quellen

Als vertrauenswürdige Bezugsquellen gelten allgemein:

- die offiziellen Vertriebs- und Bereitstellungs Kanäle des Herstellers
- dritte Distributoren, sofern sie durch den Hersteller autorisiert und legitimer Bestandteil von dessen Bereitstellungs Kanälen sind
- Internet-Downloads, sofern diese von offiziellen Bereitstellungs-Servern des Herstellers oder autorisierter Distributoren erfolgen

(1) Bietet der Bereitstellungs-Server verschiedene Formen von Downloads an, müssen durch Verschlüsselung bzw. kryptografische Signaturen geschützte Downloads jenen ohne einen solchen Schutz vorgezogen werden.

(2) Sichert der Bereitstellungs-Server die Transport-Schicht mittels kryptografischer Protokolle (z. B. HTTPS, sftp), müssen die zugehörig präsentierten Server-Zertifikate oder Server-Keys/Fingerprints bei jedem Download validiert werden, um die Identität des Bereitstellungs-Servers zu bestätigen; schlägt die Validierung fehl, ist der Download abzubrechen und der Bereitstellungs-Server als nicht vertrauenswürdige Quelle zu bewerten.

Integritätsprüfung

Die Integritätsprüfung soll sicherstellen, dass die erhaltene Software frei von Manipulationen und Schadsoftware-Befall ist. Sofern vorhanden, müssen zur Prüfung die vom Hersteller implementierten Mechanismen verwendet werden.

Valide Mechanismen sind:

- physikalische Siegel bzw. fest aufgebrachte Echtheitszertifikate (bei Bereitstellung der Software auf physikalischen Datenträgern)
- Abgleich von kryptografischen Hash Werten (z. B. SHA256, SHA512) der erhaltenen Software gegen Soll-Werte, die der Hersteller separat benennt
- Prüfung von kryptografischen Signaturen (z. B. GPG, Zertifikate), mit denen der Hersteller seine Software versieht

Zusätzlich ist eine Überprüfung der Software mittels eines Anti-Virus- bzw. Anti-Malware Scanners empfohlen (falls der Hersteller keinen der zuvor erwähnten Integritätsschutz-Mechanismen für seine Software implementiert hat, ist diese Überprüfung verpflichtend).

Erweiterte Integritätsprüfung bei Bezug von Software aus öffentlichen Registries

Öffentliche Registries ermöglichen es Entwicklern, beliebige eigene Software-Projekte zur Nutzung bereitzustellen. Die Bandbreite umfasst sowohl Projekte von namhaften Anbietern mit kontrollierten Entwicklungsprozessen als auch von kleineren Anbietern oder Amateur-Entwicklern.

Beispiele für solche Registries sind:

- Code Registries (z. B. GitHub, Bitbucket, SourceForge, Python Package Index)
- Container Registries (z. B. Docker Hub)

Software aus öffentlichen Registries muss vor dem Einsatz einer erweiterten Integritätsprüfung unterzogen werden. Zusätzlich zu den im vorherigen Abschnitt beschriebenen Bestandteilen der Integritätsprüfung soll die erweiterte Prüfung explizit sicherstellen, dass die Software ihre Funktion tatsächlich wie beschrieben erfüllt, keine inhärenten Sicherheits-Risiken wie etwa gezielt implementierte Malware-Funktionen enthält und nicht von bekannten Sicherheits-Schwachstellen betroffen ist. Sofern die Software direkte Abhängigkeiten zu dritten Software-Projekten aufweist (sog. "Dependencies", sehr typisch in Open Source Software), die zum Einsatz der Software ebenfalls bezogen und installiert werden müssen, sind diese in die erweiterte Integritätsprüfung mit einzubeziehen.

Methoden für eine erweiterte Integritätsprüfung können beispielsweise sein:

- strikte Validierung der Projekt-/Paket-Namen (Vermeidung der Verwechslung mit maliziösen, gezielt nachgeahmten Software-Projekten)
- dynamische Code Analyse / strukturierte Funktionsprüfung in einer Test-Umgebung
- statische Code Analyse mittels eines Linters (z. B. Splint, JSLint, Pylint)
- Untersuchung mittels eines Sicherheitsschwachstellen-Scanners (z. B. Qualys, Nessus)
- Untersuchung mittels eines Container Security Scanners (z. B. JFrog Xray, Harbor, Clair, Docker Scan)
- Untersuchung mittels eines SCA (Software Composition Analysis) Tools bzw. Dependency-Scanners (z. B. OWASP Dependency Check, Snyk)

Die Prüfungsmethoden sind passend zur exakten Form der Software-Bereitstellung (Quellcode, Binaries/Artefakte, Container) auszuwählen und angemessen zu kombinieren.

Motivation: Softwarelieferketten enthalten diverse Angriffsvektoren. Ein Angreifer kann an verschiedenen Stellen ansetzen, um Software zu manipulieren oder gezielt eigene Routinen einzubringen und die Zielumgebung, in der die Software anschließend zum Einsatz gebracht wird, zu schädigen oder unter seine Kontrolle zu bringen. Der Angriff kann auf dem Transport- bzw. Übertragungsweg geschehen oder auch auf die Bereitstellungsquelle selbst ausgeübt werden. Entsprechend erleichtert wird ein Angriff, falls Software nicht über offizielle und kontrollierte Quellen bezogen wird oder eine Integritätsprüfung ausgelassen wird.

Besonderes Risiko zeigt sich für Software, die aus öffentlichen Registries bezogen wird, da diese jedermann für die Bereitstellung von Software-Projekten offenstehen. Perfide Angriffsmethoden sind inzwischen bekannt, in denen der Angreifer zunächst eine Zeitlang sicherheitstechnisch völlig unauffällige, funktionale Software bereitstellt und sobald diese sich etabliert und eine gewisse Verbreitung gefunden hat, in zukünftigen Versionen gezielt versteckten Schadcode integriert. Andere Methoden setzen auf ähnlich klingende Projektnamen zu weit verbreiteten Bestandsprojekten oder überstimmende Versionsnummern, um manipulierte Software in darauf aufbauende Lösungen zu injizieren.

Umsetzungsbeispiel: Bezug der Software über die offiziellen Bereitstellungskanäle des Herstellers. Nach Erhalt der Software umgehend Prüfung auf Integrität mittels kryptografischer Checksummen, wie vom Hersteller bereitgestellt, sowie Freiheit von bekannter Schadsoftware mittels Anti-Malware- / Anti-Virus-Scanner. Ablage der geprüften Software auf einer internen, geschützten Dateiablage und weitere Nutzung (z. B. Rollout auf die Zielsysteme) nur noch von dort.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-2/9.0

Req 6	Software und Hardware des Systems müssen von einem Sicherheitsschwachstellen-Support seitens des Herstellers abgedeckt werden.
-------	--

In einem System dürfen nur Software- und Hardware-Produkte eingesetzt werden, für die ein Sicherheitsschwachstellen-Support durch den Hersteller besteht.

Ein solcher Support muss umfassen, dass der Hersteller

- das Produkt kontinuierlich auf eine Betroffenheit von Sicherheitsschwachstellen beobachtet und analysiert,
- über Art, Schweregrad und Ausnutzbarkeit von im Produkt entdeckten Schwachstellen umgehend informiert
- zeitnah Produktaktualisierungen oder wirksame Workarounds zur Behebung der Schwachstellen bereitstellt.

Der Sicherheitsschwachstellen-Support muss für die gesamte Dauer bestehen, in der das betroffene Produkt im Ein-

satz verbleibt.

Support-Phasen mit eingeschränktem Leistungsumfang

Viele Hersteller bieten für ihre Produkte als Option einen zeitlich ausgedehnten Support an, welcher über die für den allgemeinen Markt vorgesehene Support-Phase hinausgeht, aber oftmals mit Einschränkungen verbunden ist. Einige Hersteller definieren ihren Support grundlegend in Abstufungen, die Einschränkungen schon während der letzten Phase vor dem absoluten Endtermin des regulären Supports enthalten können.

Wird ein Produkt innerhalb von Support-Phasen eingesetzt, die Einschränkungen unterliegen, muss explizit sichergestellt werden, dass diese Einschränkungen die Verfügbarkeit des Sicherheitsschwachstellen-Support nicht beeinträchtigen.

Open Source Software und Hardware

Open Source Produkte werden häufig von freien Organisationen oder Communities entwickelt; entsprechend ist ein vertraglich vereinbarter Sicherheitsschwachstellen-Support unter Umständen nicht verfügbar. Grundsätzlich muss auch hier sichergestellt werden, dass von der Organisation/Community (oder einem offiziell durch diese eingesetzten Dritten) für das betroffene Produkt ein umfassendes Sicherheitsschwachstellen-Management betrieben wird, welches die oben genannten Kriterien erfüllt und als zuverlässig etabliert gilt.

Motivation: Hardware- und Software-Produkte, für die kein vollumfassender Sicherheitsschwachstellen-Support durch den Hersteller besteht, stellen ein Risiko dar. Dies bedeutet, dass ein Produkt nicht ausreichend auf eine Betroffenheit durch weiterentwickelte Angriffsformen oder neu entdeckte Schwächen technischer Umsetzungen geprüft wird. Analog werden bei vorhandenen Sicherheitsschwachstellen in einem Produkt keine Nachbesserungen (z. B. Updates, Patches) bereitgestellt. Hieraus resultiert ein System, dessen Schwachstellen nicht behoben werden können, so dass sie für einen Angreifer ausnutzbar bleiben, um das System zu kompromittieren oder es negativ zu beeinträchtigen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-1/9.0

Req 7	Bekannte Schwachstellen in der Software oder Hardware des Systems müssen behoben oder gegen Missbrauch abgesichert werden.
-------	--

Bekannte Schwachstellen in Software- und Hardware Komponenten müssen mittels Installation von verfügbaren Systemaktualisierungen des Herstellers (z. B. Patches, Updates/Upgrades) behoben werden. Alternativ ist der Einsatz von Workarounds (Akut-Lösungen, welche die Schwachstelle nicht beheben, aber die Ausnutzung wirksam verhindern) zulässig. Workarounds sollten grundlegend nur temporär Verwendung finden und sobald möglich durch eine reguläre Systemaktualisierung ersetzt werden, um die Schwachstellen vollständig zu schließen.

Komponenten, die bekannte, nicht behebbare Schwachstellen enthalten, dürfen in einem System nicht eingesetzt werden.

Die Behandlung von neu bekannt werdenden Schwachstellen muss zudem für die gesamte Einsatzphase des Systems fortlaufend sichergestellt und in den kontinuierlichen Betriebsprozessen eines Security Patch Managements umgesetzt werden.

Motivation: Der Einsatz von Komponenten ohne eine Behebung enthaltener Schwachstellen erhöht deutlich das Risiko einer erfolgreichen Kompromittierung. Der Angreifer wird zusätzlich dadurch begünstigt, dass i. d. R. nicht nur detaillierte Informationen zu bereits bekannt gewordenen Schwachstellen offen verfügbar sind, sondern oft auch bereits angepasste Angriffswerkzeuge, die eine aktive Ausnutzung erleichtern.

Umsetzungsbeispiel: Im Anschluss an die initiale Installation eines Betriebssystems von einem offiziellen Installationsmedium werden alle aktuell verfügbaren Patches und Sicherheits-Updates eingespielt.

Zusätzliche Informationen:

Primäre Quellen für bereits bekannt gewordene Schwachstellen in einer Software/Hardware sind Auflistungen in den Versionshinweisen sowie die Sicherheitshinweise aus den offiziellen Meldekanälen des Herstellers oder unabhängiger CERTs. Insbesondere die Meldekanäle werden sinnvollerweise fest in kontinuierliche Prozesse des Security Patch Managements für ein System eingebunden, damit neu bekanntwerdende Schwachstellen zeitnah registriert und in operative Behebungsmaßnahmen geführt werden können.

Als ergänzende Maßnahme zur Aufdeckung von potenziell noch enthaltenen, grundsätzlich bereits bekannt gewordenen Arten von Schwachstellen können gezielte Schwachstellenuntersuchungen des Systems durchgeführt werden. Zu diesem Zweck eignen sich besonders spezialisierte Werkzeuge wie automatisierte Schwachstellen-Scanner. Beispiele hierfür sind: Tenable Nessus, Qualys Scanner Appliance.

ID: 3.22-7/13.0

Req 8	Gespeicherte schutzbedürftige Daten müssen vor nicht autorisierter Einsichtnahme, Veränderung und Löschung geschützt werden.
-------	--

Der Schutzbedarf für gespeicherte Daten ist abhängig von deren Einstufung (beispielsweise nach geltenden gesetzlichen Datenschutzvorgaben, regulatorischen Auflagen, vertraglichen Verpflichtungen), dem Schadenspotential bei deren Missbrauch, sowie weiteren relevanten Faktoren (beispielsweise dem Ort der Speicherung). Art und Umfang von Schutzmaßnahmen müssen entsprechend angemessen gewählt werden.

Gespeicherte Authentisierungsmerkmale wie beispielsweise Passwörter, Private Keys, Tokens oder Zertifikate gelten grundsätzlich als schutzbedürftig. Ebenfalls als grundsätzlich schutzbedürftig gelten Daten, die Funktionalität und sicherheitsrelevantes Verhalten eines Systems bestimmen (z. B. System-Konfigurationsdateien, Betriebssysteme und Kernel, Treiber).

Die Einhaltung der Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit muss für gespeicherte schutzbedürftige Daten durchgängig gewährleistet werden. Dies gilt auch während einer lediglich kurzfristigen Speicherung (z. B. bei der Zwischenspeicherung in einem Web Cache oder in einem temporären Ordner innerhalb einer Datenverarbeitungskette).

Grundlegend muss der Zugriff auf schutzbedürftige Daten in einem System auf Basis von technisch implementierten Berechtigungsvergaben und -kontrollen vollumfassend reglementiert werden.

Reicht eine solche technische Zugriffskontrolle allein nicht mehr aus, um den erforderlichen Schutzbedarf gespeicherter Daten zu gewährleisten, oder kann deren Wirksamkeit nicht durchgängig sichergestellt werden, müssen zusätzlich kryptografische Verfahren (z. B. Verschlüsselung, Signierung, Hashing) implementiert werden. Bei der Speicherung von Daten eingesetzte kryptografische Verfahren müssen für diesen Zweck anerkannt wirksam sein und dürfen keine bekannten Schwachstellen aufweisen.

Motivation: Die Speicherung von Daten auf einem System ohne einen adäquaten Schutz ermöglicht es einem Angreifer, diese unberechtigt einzusehen, zu nutzen, zu verbreiten, zu verändern, oder zu vernichten. Dies eröffnet potenziell zusätzliche Angriffsvektoren auf die unmittelbaren sowie angeschlossene weitere Systeme und kann zu erheblichen Ausfällen, Kontrollverlusten und Schäden sowie resultierenden Strafzahlungsforderungen und Reputationsverlusten gegenüber Kunden und Geschäftspartnern führen.

Umsetzungsbeispiel: [Beispiel 1]

Ein System exportiert Daten für einen Transport auf einen mobilen Datenträger. Da die technische Zugriffskontrolle des Systems auf Dateiberechtigungsebene nicht mehr greift, sobald der mobile Datenträger aus dem System entfernt wird, müssen zusätzliche Maßnahmen zum Schutz der Daten getroffen sein. Bevor das System die Daten auf den mobilen Datenträger schreibt, werden sie entsprechend mit einem passenden Verfahren verschlüsselt. Der zugehörige Schlüssel wird auf einem separaten Kanal ausgetauscht, so dass die Daten im legitimen Zielsystem wieder entschlüsselt und verarbeitet werden können. Ein Angreifer, der den mobilen Datenträger in seinen Besitz bringt, hat hingegen keinen Zugriff auf die Daten.

[Beispiel 2]

In der lokalen Benutzer-Datenbank eines Systems werden nur kryptografische Hashes der Passwörter gespeichert, die mit einem als sicher geltenden Password-Hashing Verfahren erzeugt wurden. Für das System sind diese Hashes ausreichend, um die Benutzer bei einer Anmeldung am System zu authentifizieren. Sollte ein Angreifer die Benutzer-Datenbank kopieren können, gelangt er somit jedoch nicht unmittelbar in den Besitz von Klartext-Passwörtern, mit denen er sich am System im Namen der Benutzer anmelden könnte.

[Beispiel 3]

Auf einem System sind die Konfigurationsdateien des Web-Servers ausschließlich für den legitimen Admin schreibbar, in dem im Dateisystem entsprechende Berechtigungen gesetzt wurden. Die Zugriffskontrolle des Betriebssystem-Kernels verweigert somit allen anderen Benutzern des Systems, Veränderungen an den Konfigurationsdateien des Web-Servers vorzunehmen; einschließlich dem Web-Server Dienste-Konto selbst, was auch die Angriffsfläche von außen bei Schwachstellen im Web-Server verkleinert.

ID: 3.22-8/13.0

Req 9	Schutzbedürftige Daten müssen während der Übertragung vor nicht autorisierter Einsichtnahme und Veränderung geschützt werden.
-------	---

Der Schutzbedarf von zu übertragenden Daten ist abhängig von deren Einstufung (beispielsweise nach geltenden gesetzlichen Datenschutzvorgaben, regulatorischen Auflagen, vertraglichen Verpflichtungen), dem Schadenspotential bei deren Missbrauch, sowie weiteren relevanten Faktoren (beispielsweise der Übertragung über öffentliche Netze). Art und Umfang der Schutzmaßnahmen müssen entsprechend angemessen gewählt werden.

Authentisierungsmerkmale wie beispielsweise Passwörter oder Tokens gelten grundsätzlich als schutzbedürftige Daten. Ebenfalls grundsätzlich als schutzbedürftig gelten Daten, die Funktionalität und sicherheitsrelevantes Verhalten eines Systems bestimmen (z. B. Updates & Patches, Konfigurationsparameter, Fernwartungen, Steuerung über APIs).

Die Einhaltung der Schutzziele Vertraulichkeit und Integrität muss während der Übertragung von schutzbedürftigen Daten durchgängig gewährleistet werden.

I.d.R. erfordert dies eine Implementation von kryptografischen Verfahren (z. B. Verschlüsselung, Signaturen, Hashes). Kryptografische Verfahren können

- vor der Übertragung direkt auf die Daten angewendet werden, was eine anschließende Übertragung der Daten auch über unsichere Kanäle vertretbar machen kann
- auf dem Übertragungs-Kanal eingesetzt sein, um einen sicheren Kanal zu schaffen und beliebige Daten, die durch diesen geleitet werden, zu schützen
- oder auch als Kombination aus beidem umgesetzt sein.

Bei der Übertragung von Daten eingesetzte kryptografische Verfahren müssen für diesen Zweck anerkannt wirksam sein und dürfen keine bekannten Schwachstellen aufweisen.

Motivation: Die Übertragung von Daten ohne einen adäquaten Schutz ermöglicht es einem Angreifer, diese unberechtigt abzufangen, zu nutzen, zu verbreiten, zu verändern oder aus der Übertragung zu entfernen. Dies eröffnet potenziell weitere Angriffsvektoren auf die unmittelbaren Zielsysteme sowie angeschlossene weitere Systeme und kann zu erheblichen Ausfällen, Kontrollverlusten und Schäden sowie resultierenden Strafzahlungsforderungen und Reputationsverlusten gegenüber Kunden und Geschäftspartnern führen.

Umsetzungsbeispiel: [Beispiel 1]

Vertrauliche Dokumente werden vor dem Versand per E-Mail an den Kunden verschlüsselt.

[Beispiel 2]

Ein Administrator konfiguriert eine neue Cloud-Anwendung über das Internet. Der Zugriff erfolgt über eine TLS-verschlüsselte Verbindung ("https").

[Beispiel 3]

Ein System bezieht automatische Software-Updates von einem Update-Server. Der Update-Server liefert die Software Updates kryptografisch signiert aus. Das System kann die empfangenen Software-Updates somit validieren und sicher ausschließen, dass diese während der Übertragung manipuliert wurden.

ID: 3.22-9/13.0

Req 10	In Ausgaben und Meldungen dürfen keine Informationen über interne Strukturen des Systems preisgegeben werden.
--------	---

Informationen über die internen Strukturen eines Systems einschließlich der dort verwendeten Komponenten und ent-

sprechende Implementationsdetails gelten grundsätzlich als schutzbedürftig.

Im Allgemeinen betrifft dies Informationen über

- Produktnamen und Produkt-Identifizierer implementierter System-Komponenten
- eingesetzte Betriebssysteme, Middleware, Backend-Software, Software-Bibliotheken und interne Applikationen sowie deren Software-Versionen
- installierte Service-Packs, Patches, Hotfixes
- Seriennummern von Komponenten sowie hinterlegte Produkt-Lizenzen
- Datenbankstrukturen

Typische Beispiele für Ausgaben und Meldungen, in denen eine Preisgabe solcher System-Informationen potenziell auftreten kann:

- Anmelde-Fenster und -dialoge
- Fehlermeldungen
- Statusausgaben
- Banner von aktiven Netzwerk-Diensten
- System-Protokolle und Log Dateien
- Debug-Protokolle, Stack Traces

Soweit es technisch umsetzbar ist, ohne die Funktion und den Betrieb des Systems zu beeinträchtigen, muss die Ausgabe betroffener System-Informationen grundsätzlich deaktiviert sein.

Ein Zugriff auf betroffene System-Informationen darf nur dazu autorisierten Benutzern des Systems möglich sein. In der Regel ist der Kreis autorisierter Benutzer auf Administratoren und Betreiber des Systems zu beschränken. Ebenfalls zulässig ist ein Zugriff für autorisierte Monitoring- und Inventarisierungs-Systeme innerhalb der Einsatzumgebung.

Eine zulässige Ausnahme dieser Einschränkungen besteht für spezifische einzelne System-Informationen, deren Preisgabe für die vorgesehene Funktion des Systems in Verbindung mit dritten Systemen technisch zwingend erforderlich ist; beispielsweise die Präsentation unterstützter Protokolle und deren Versionen während der initialen Parameter-Aushandlung in Sitzungs-Aufbauten zwischen einem Client und einem Server.

Motivation: Informationen über die internen Strukturen eines Systems können von einem Angreifer genutzt werden, um Angriffe auf das System äußerst effektiv vorzubereiten. Beispielsweise kann ein Angreifer aus der Software-Version eventuell bekannte Schwachstellen eines Produktes ableiten, um diese gezielt während des Angriffs auf das System auszunutzen.

Umsetzungsbeispiel: [Beispiel 1]

Deaktivierung der Anzeige des Produktnamens und der installierten Version in den ausgelieferten Fehler-Webseiten eines eingesetzten Web-Servers.

[Beispiel 2]

Entfernung des Produktnamens und des zugehörigen Versions-Strings aus dem Login-Banner eines eingesetzten SSH Servers.

ID: 3.22-10/13.0

Req 11	Das System muss vor Überlastsituationen geschützt werden.
--------	---

Ein System muss über Schutzmechanismen verfügen, die Überlastsituationen so weit wie möglich verhindern. Insbesondere ist eine partielle oder komplette Beeinträchtigung der Verfügbarkeit des Systems zu vermeiden.

Beispiele für mögliche Schutzmaßnahmen sind:

- Begrenzung des pro Anwendung verfügbaren Arbeitsspeichers (RAM)

- Begrenzung der maximalen Sessions einer Web-Anwendung
- Begrenzung der maximalen Größe eines Datensatzes
- Begrenzung von CPU-Ressourcen pro Prozess
- Priorisieren von Prozessen
- Begrenzung der Anzahl oder der Größe von Transaktionen eines Benutzers oder von einer IP-Adresse in einem bestimmten Zeitraum

Anmerkung:

vor Netzwerk-basierenden Angriffen mit extrem hohen Daten- oder Paketraten, den sogenannten "Distributed Denial of Service" (DDoS) Angriffen, kann ein System sich i. d. R. selbst nicht mehr schützen. Zur Abwehr von DDoS Angriffen wird eine vorgelagerte Lösung in der Netzwerk-Schicht benötigt.

Motivation: Angreifer können mit gezielten ressourcenlastigen oder großvolumigen Anfragen versuchen, die Ressourcen eines Systems aufzubrechen, so dass das System seine regulären Aufgaben bzw. vorgesehenen Aufgabenvolumen nicht mehr erfüllen kann und die Verfügbarkeit der angebotenen Dienste effektiv gestört ist. Die Begrenzung der Ressourcen, die je gestellter Anfrage an das System maximal in Anspruch genommen werden können, ist eine grundlegende Maßnahme, um die Auswirkungen solcher Denial-of-Service (DoS) Angriffe zu reduzieren.

ID: 3.22-11/13.0

Req 12 In Überlastsituationen muss sich das System berechenbar verhalten.

Selbst umfassende systemeigene Schutzmaßnahmen können in extremen Situationen möglicherweise nicht mehr verhindern, dass ein System in eine Überlastsituation gerät.

Es muss daher sichergestellt sein, dass das System in Überlastsituationen nicht in einen Zustand überwechselt, der sicherheitsrelevante Funktionen oder Eigenschaften des Systems außer Kraft setzt. Leistungseinbußen (bspw. die Verringerung des Durchsatzes von legitimen Netzwerk-Paketen oder der Anzahl beantworteter Server-Anfragen pro Zeitraum) sind in Überlastsituationen i. d. R. nicht vermeidbar, das reguläre funktionale Verhalten des Systems muss jedoch grundlegend erhalten bleiben.

Dies kann im Extremfall bedeuten, dass ein kontrolliertes Herunterfahren des Systems eher hinnehmbar ist, als ein Weiterbetrieb bei unkontrolliertem Versagen der Sicherheitsfunktionen und somit dem Verlust des Systemschutzes.

Motivation: Mittels eines Denial-of-Service-Angriffs kann ein Angreifer versuchen, ein System gezielt zu überlasten. Reagiert ein solches System dann unberechenbar oder versagt sein reguläres Verhalten, insbesondere hinsichtlich dessen Sicherheitsfunktionen, kann dies dem Angreifer eine erweiterte Angriffsfläche auf Funktionen und Daten des Systems eröffnen sowie potentiell weitere verknüpfte Systeme gefährden.

Umsetzungsbeispiel: Eine Firewall, die in Überlastsituationen ihr Filter-Regelwerk verwirft und sämtliche Pakete ohne Prüfung weiterleitet, würde die Anforderung nicht erfüllen. In diesem Fall wäre das Blockieren sämtlicher Pakete durch die Abschaltung der Firewall eher hinnehmbar als das Versagen ihrer regulär vorgesehenen Schutz Aufgabe für nachgelagerte Systeme.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.01-13/9.0

Req 13 Das System muss robust gegenüber unerwarteten Eingaben implementiert werden.

An das System übergebene Daten müssen vor der Weiterverarbeitung zunächst validiert werden, um sicherzustellen, dass die Daten dem erwarteten Daten-Typ und -Format entsprechen. Dies soll das Risiko einer Manipulation von Systemabläufen und -Zuständen durch entsprechend konstruierte Dateninhalte eliminieren. Eine Validierung ist für jegli-

che Daten durchzuführen, die an das System übergeben werden. Beispiele sind Benutzereingaben, Werte in Datenfeldern und Protokollinhalte.

Die folgenden typischen Implementierungsfehler müssen vermieden werden:

- Fehlende Validierung der Länge von übergebenen Daten
- Unzutreffende Annahmen über das Format von Daten
- Fehlende Validierung empfangener Daten auf Konformität mit der Spezifikation
- Unzureichende Behandlung von Protokollabweichungen in empfangenen Daten
- Unzureichende Begrenzung der Rekursion beim Parsing komplexer Datenformate
- Unzureichende Umsetzung von Whitelisting oder Escaping zum Schutz vor Eingaben außerhalb des validen Wertebereiches

Motivation: Ein Angreifer kann mittels gezielt konstruierter Dateninhalte versuchen, ein System, das empfangene Daten vor der internen Weiterverarbeitung nicht ausreichend validiert, in einen instabilen Zustand zu versetzen oder nicht autorisierte Aktionen innerhalb des Systems auszulösen. Das Schadenspotential entsprechender Angriffe ist individuell systemabhängig, besitzt jedoch eine theoretische Bandbreite von unkontrollierten Systemabstürzen bis hin zu einer kontrollierten Ausführung von eigens eingeschleustem Code und der daraus resultierenden vollständigen Kompromittierung eines Systems.

ID: 3.22-13/13.0

2.2. Netz- und Kommunikationssicherheit

Req 14	Der direkte Zugriff auf ein SAP-System muss immer durch ein netzwerkseitig separiertes vorgelagertes System, entsprechend den verwendeten Protokollen, geschützt sein.
--------	--

Der direkte Zugriff auf das SAP-Systeme über die Netzwerkebene muss eingeschränkt werden. Je nach Protokoll kommen unterschiedliche, dem SAP-System vorgeschaltete Systeme zum Einsatz (z. B. SAProuter, SAP Web Dispatcher, ...).

Motivation: Unberechtigter Zugriff auf Serverkomponenten kann zur Kompromittierung des Systems führen. Viele Komponenten, die auf Netzwerkebene von einem SAP-System zur Verfügung gestellt werden, sind nicht für den direkten Zugriff durch Clients vorgesehen.

Umsetzungsbeispiel: Werden DIAG- und RFC-Protokolle verwendet, muss das SAP-System über einen netzwerkseitig separierten vorgelagerten SAProuter geschützt werden.

Werden HTTP-, HTTPS- und SMTP-Protokolle verwendet, muss das SAP-System über einen netzwerkseitig separierten vorgelagerten SAP Web Dispatcher geschützt werden.

Werden DIAG-, RFC-, HTTP-, HTTPS- und SMTP-Protokolle verwendet, muss das SAP System über jeweils einen netzwerkseitig separierten vorgelagerten SAProuter und einen netzwerkseitig separierten vorgelagerten SAP Web Dispatcher geschützt werden.

Requirement-ID: 011b57bf

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.22-14/13.0

Req 15 Der direkte externe Zugriff auf ein SAP-System muss immer durch ein netzwerkseitig separiertes vorgelagertes System in der DMZ, entsprechend den verwendeten Protokollen, geschützt sein.

Es muss sichergestellt sein, dass alle Protokolle (DIAG, RFC, HTTP, HTTPS, SMTP) in der DMZ durch entsprechende Systeme (z.B. SAProuter, SAP Web Dispatcher) geschützt werden. Protokolle, die nicht auf den SAProuter und/oder den SAP Web Dispatcher verarbeitet werden, dürfen nicht bis zu diesen kommen.

Für internen und externen Zugriff sind unterschiedliche vorgelagerte Systeme zu verwenden.

Motivation: Das SAP-System muss vor unberechtigten Zugriffen und Anfragen geschützt werden.

Umsetzungsbeispiel: Werden DIAG- und RFC-Protokolle verwendet, muss das SAP System über einen SAProuter geschützt werden.

Werden HTTP-, HTTPS- und SMTP-Protokolle verwendet, muss das SAP System über einen SAP Web Dispatcher geschützt werden.

Werden DIAG-, RFC-, HTTP-, HTTPS- und SMTP-Protokolle verwendet, muss das SAP System über einen SAProuter und einen SAP Web Dispatcher geschützt werden.

Requirement-ID: 74f42f33

ID: 3.22-15/13.0

Req 16 „Vertraulich“ klassifizierte Informationen müssen verschlüsselt übertragen werden.

Grundsätzlich sind vertrauliche Daten verschlüsselt zu übermitteln, z. B. per TLS (SSL) oder SNC für CPI-C/RFC/DIAG. Die SNC Verbindung muss mit dem SNC Modus 3 aufgebaut werden (Verschlüsselung der Verbindung).

Die Verschlüsselung muss zwischen SAP-System und dem Client, zwischen SAP-Systemen und zwischen SAP-Systemen und Non-SAP-Systemen umgesetzt werden. Falls die Kommunikation zwischen zwei Systemen nicht direkt von den SAP-Systemen verschlüsselt werden kann, so muss die Kommunikation mindestens zwischen den SAProutern, die zwischen den Systemen stehen, verschlüsselt sein.

Eine Ausnahme von der Forderung nach Verschlüsselung liegt nur dann vor, wenn die Übertragung ausschließlich lokal innerhalb eines DTAG-eigenen Rechenzentrums über DTAG-interne Systeme, Netze und Leitungen geführt wird. Ein Beispiel hierfür ist ein Layer-2 Switch, an den alle beteiligten Systeme lokal angeschlossen sind. Es ist sicherzustellen, dass nur Berechtigte Zugang zu diesem Netz haben. Erfolgt eine Kommunikation über ein Zusammenspiel eines solchen vertrauenswürdigen Netzes mit anderen Netzen, so können – anstatt jede Verbindung einzeln zu schützen – auch Tunnelmechanismen mit Verschlüsselung (z. B. IPSec) zur Überbrückung nicht vertrauenswürdiger Netze eingesetzt werden.

Motivation: Wenn es einem Angreifer gelingt, die Kommunikation zu mitzuschneiden, kann er schutzbedürftige Daten (z.B. Passwörter, Anmeldetickets) lesen und auch verändern. Dies eröffnet Wege für einen unberechtigten Systemzugriff.

Umsetzungsbeispiel: S. Kapitel 2.3 Kryptographie

Requirement-ID: ae5950c0

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.22-16/13.0

Req 17 SAP-Systeme und Non-SAP-Systeme dürfen nicht in derselben Netzwerkkumgebung betrieben werden.

SAP-Systeme kommunizieren untereinander häufig mit unverschlüsselten Protokollen.

Motivation: Ein Großteil der SAP-Applikationen kommuniziert untereinander mit unverschlüsselten Protokollen. Um die Daten und Systeme vor Fremdzugriffen zu schützen, müssen NON-SAP-Systeme netzwerktechnisch von SAP-Systemen getrennt werden.

Umsetzungsbeispiel: Requirement-ID: ec19682c

ID: 3.22-17/13.0

Req 18	Es muss sichergestellt sein, dass zwischen SAP Systemen und SAProutern eine sichere Verbindung aufgebaut wird.
--------	--

Es muss sichergestellt sein, dass zwischen SAP Systemen und SAProutern sowie zwischen SAP Systemen und SAProutern untereinander eine sichere Verbindung aufgebaut wird.

Motivation: Schutz der Integrität, Authentizität und Vertraulichkeit der Daten durch Verhinderung eines unberechtigten Zugriffs auf die Daten.

Umsetzungsbeispiel: Zu beachten:

- Nutzung anerkannten und öffentlichen Zertifizierungsstelle (Certificate Authority = CA) zur Schlüsselverwaltung.
- Nutzung der SAP Cryptographic Library (*SAPCRYPTOLIB*).
- s. Kapitel 2.3 Kryptographie für weitere Details (z. B. Schlüssellänge, Verschlüsselungsalgorithmus)

Requirement-ID: c6cbb721

ID: 3.22-18/13.0

2.3. Kryptographie ----- Wenn die Anforderungen 19 bis 44 erfüllt sind, entspricht das der Umsetzung der Sicherheitsanforderung "Kryptographische Algorithmen und Sicherheitsprotokolle" (3.50) für SAP Netweaver Umgebungen.

Req 19	Es dürfen nur standardisierte kryptographische Verfahren und Primitive eingesetzt werden, die von anerkannten Organisationen veröffentlicht werden.
--------	---

Jede Abweichung muss mit der Sicherheitsorganisation der Deutschen Telekom Gruppe abgestimmt und vereinbart werden.

Die in diesem Dokument definierten Anforderungen entsprechen den zur Zeit der Veröffentlichung dieses Dokuments gültigen Sicherheitsstandards folgender anerkannter Organisationen:

- Bundesamt für Sicherheit in der Informationstechnik (BSI) [BSI]
- European Telecommunication Standards Institute (ETSI) [ETSI]
- Senior Officials Group Information Systems Security (SOGIS) [SOGIS]
- National Institute of Standards and Technology (NIST) [NIST]
- International Organization for Standardization (ISO) [ISO]
- Internet Engineering Task Force (IETF) [IETF]

Quellen:

[BSI] https://www.bsi.bund.de/DE/Home/home_node.html (letzter Aufruf: 08.10.2019)

[ETSI] <https://www.etsi.org> (letzter Aufruf: 08.10.2019)

[SOGIS] <https://www.sogis.eu/> (letzter Aufruf: 08.10.2019)

[NIST] <https://www.nist.gov/> (letzter Aufruf: 08.10.2019)

[ISO] <https://www.iso.org/home.html> (letzter Aufruf: 08.10.2019)
[IETF] <https://www.ietf.org/> (letzter Aufruf: 08.10.2019)

Motivation: Nur ausreichend untersuchte Algorithmen bieten einen verlässlichen systematischen Schutz gegen kryptographische Angriffe. Damit wird sichergestellt, dass die Empfehlungen auf einem breiten fachlichen Konsens basieren

Umsetzungsbeispiel: Die SAPcryptoLib / CommonCryptoLib ist einzurichten. Laden Sie die neuste SAP Cryptographic Library und die neuste ausführbare Datei sapgenpse vom SAP Software Download Center.

Bitte folgen Sie den SAP-Anleitungen (SAP Help, SAP Security Guides, SAP Notes) für die Einrichtung der SAP SAPcryptoLib / CommonCryptoLib und die Konfiguration von SSL/HTTPS und SNC auf dem jeweiligen SAP Produkt. Die Schritte für die Konfiguration sind unterschiedlich für die SSL/HTTPS- und SNC-Kommunikation und sogar für die verschiedenen möglichen SNC-Kommunikationsarten für AS ABAP (RFC, DIAG/SAPGUI, CPIC, Drucker, externe Programme, SAProuter, AS Java) und zudem noch von der jeweiligen Version und dem Release abhängig.

Requirement-ID: 092a7f10

ID: 3.22-19/13.0

Req 20	Für die Implementierung von kryptografischen Verfahren müssen etablierte und aktuelle Krypto-Bibliotheken genutzt werden
--------	--

Nutzerrollen: Entwicklung, Integration

Eigene Implementierungen von kryptographischen Algorithmen sollen vermieden werden. Stattdessen müssen etablierte und aktuelle kryptographische Bibliotheken genutzt werden, welche die in diesem Dokument freigegebenen Algorithmen implementiert haben. Wenn eigene Implementierungen erforderlich sind, müssen Best Practices beachtet werden und die Sicherheitsorganisation der Deutschen Telekom Gruppe zur Beratung hinzugezogen werden.

Motivation: Nur weit verbreitete kryptographischen Bibliotheken wurden in ausreichendem Maße durch Sicherheitsforscher analysiert und bieten einen hinreichenden Schutz. Diese Bibliotheken werden regelmäßig aktualisiert (u.a. bei Sicherheitsschwachstellen)

Umsetzungsbeispiel: Mit Nutzung der SAP Cryptographic Library oder T-Secure ist die Anforderung erfüllt.

Requirement-ID: b9ba557a

ID: 3.22-20/13.0

Req 21	Kryptographische Verfahren müssen in austauschbaren Modulen implementiert sein.
--------	---

Nutzerrollen: Entwicklung, Integration

Statisch implementierte kryptographische Verfahren machen einen Austausch oder die Korrektur von fehlerhaften Implementierungen aufwändiger. Insbesondere unvorhergesehene Ereignisse, wie neue Angriffe oder zukünftige Rechnerarchitekturen (z.B. Quantencomputer), können einen schnellen Austausch eines Verfahrens notwendig machen.

Motivation: Kryptografische Implementierungen sollen unabhängig von der Umgebung austauschbar sein.

Umsetzungsbeispiel: Mit Nutzung der SAP Cryptographic Library oder T-Secure ist die Anforderung erfüllt.

Requirement-ID: 176b1a12

ID: 3.22-21/13.0

Req 22	Die Anwendung muss die Möglichkeit bieten, das kryptographische Verfahren konfigurierbar zu halten und entsprechende Migrationsfunktionen bereitzustellen.
--------	--

Nutzerrolle: Entwicklung

Bei der Entwicklung kryptographischer Systemen muss eine Funktion implementiert werden, damit einzelne kryptographische Verfahren (beispielsweise Cipher-Suites) deaktivieren oder gewechselt werden können. Migrationsfunktionen sind nur für Verschlüsselungsfunktionen von persistenten Datenspeichern anwendbar und nicht jedoch für die Transportverschlüsselung.

Auch hier gilt: Unvorhergesehene Ereignisse, wie neue Angriffe oder zukünftige Rechnerarchitekturen (z.B. Quantencomputer), können einen schnellen Austausch eines Verfahrens notwendig machen.

Motivation: Krypto-Agilität wird erst durch ein umfassendes Kryptomanagement ermöglicht und erlaubt eine Risiko- und Aufwandsminimierung gegenüber notwendigen Anpassungen in einem laufenden System.

Umsetzungsbeispiel: Mit Nutzung der SAP Cryptographic Library oder T-Secure als GSS-API V2-Lib ist die Anforderung erfüllt.

Requirement-ID: e11ce0f5

ID: 3.22-22/13.0

Req 23	Als symmetrische Blockverschlüsselung müssen AES-256, AES-192 oder AES-128 verwendet werden.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Der „Advanced Encryption Standard“ (AES) ist eine im Jahre 2001 standardisierte und seitdem intensiv untersuchte Variante des „Rijndael“-Algorithmus. Der Algorithmus ist in ISO/IEC 18033-3 oder im Federal Information Processing Standard (FIPS) 197 standardisiert. AES ist eine Blockverschlüsselung die 128 Bit lange Blöcke verarbeitet und eine Schlüssellänge von 128, 192 bzw. 256 Bit unterstützt. AES-128, AES-192 und AES-256 bezeichnen die entsprechenden Schlüssellängen. Zum Schutz vor zurzeit lediglich theoretischen Angriffen mit Quantencomputern (Grover-Algorithmus) wird empfohlen bereits heute eine Schlüssellänge von 256 Bit zu wählen.

Hinweis zu Datenschutz:

Die Anforderungen in dem vorliegenden Dokument richten sich an die Daten- und Informationssicherheit im Sinne Vertraulichkeit und Integrität. Sollen kryptografische Methoden eingesetzt werden, um Anonymisierungs- bzw. Pseudonymisierungs-Anforderungen des Datenschutzes zu erfüllen, findet man die Bedingungen für den Einsatz der Methoden im Dokument „Datenschutzanforderung Anonymisierung und Pseudonymisierung (Identifikationsschutz und Datenvermeidung)“.

Motivation: AES ist intensiv untersucht, hat langanhaltender Kryptoanalyse standgehalten und bietet mit 256 Bit Schlüssellänge ausreichend Sicherheit gegenüber zukünftigen Angriffen mit Quantencomputern.

Umsetzungsbeispiel: **Achtung: Mit dem hier angegebenen Umsetzungsbeispielen sind folgende Anforderungen für alle SAP Software Produkte erfüllt: 23, 24, 25, 26, 27, 30, 38, 40, 41, 42 und 44!**

Bitte folgen Sie den SAP-Anleitungen (SAP Help, SAP Security Guides, SAP Notes) für die Einrichtung der SAP SAP-cryptoLib / CommonCryptoLib und die Konfiguration von SSL/HTTPS und SNC auf dem jeweiligen SAP Produkt. Die Schritte für die Konfiguration sind unterschiedlich für die SSL/HTTPS- und SNC-Kommunikation und sogar für die verschiedenen möglichen SNC-Kommunikationsarten für AS ABAP (RFC, DIAG/SAPGUI, CPIC, Drucker, externe Programme, SAProuter, AS Java) und zudem noch von der jeweiligen Version und dem Release abhängig.

Die Vorgaben zur Verschlüsselung der Kommunikation zum jeweiligen SAP Produkt werden in den nachfolgenden nach den SAP Produkten geordneten Abschnitten beschrieben.

AS ABAP und ICM (ab Release 6.20)

Bitte beachten Sie folgende SAP Hinweise: 1848999, 510007, 2570499, 2065806, 2450794, 2384243, 2110020,

2384290, 1433874

Setzen Sie die TLS und Cipher Einstellungen mit TA RZ10 mit den Profilparametern auf folgende Werte:

- ssl/ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- ssl/client_ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- icm/server_port_<xx> = ..., PROT=HTTPS, ..., SSLCONFIG=ssl_config_<yy>
- icm/ssl_config_<yy> = CRED=<name>.pse, CI-PHERS=544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

Hinweise:

- Wenn Sie die SSL-Konfiguration mit dem Parameter icm/ssl_config_<yy> gepflegt haben, müssen Sie die Option SSLCONFIG vom Parameter icm/server_port_<xx> auf den Wert "ssl_config_<yy>" (<yy> entsprechend dem Parameter icm/ssl_config_<xx> setzen).
- Parameter icm/ssl_config_<yy>: Die CRED Option muss angegeben werden z. B. CRED=SAPSSLsapext.pse.
- Parameter icm/ssl_config_<yy>: Die Inhalte der Option CIPHERS unterliegen denselben Regeln und derselben Syntax, die für den Profilparameter ssl/ciphersuites gelten.

Folgendes muss für SNC konfiguriert sein:

Das Zertifikat / Schlüssel für SNC müssen bei der Generierung mit sapgenpse oder der TA STRUST mindestens folgende Eigenschaften haben:

- mindestens sha256 (Option -a)
- Bitlänge mindestens 2048 (Option -s)

Setzen von folgenden SNC Profilparametern (TA RZ10):

- snc/enable = 1
- snc/gssapi_lib = <Pfad und Dateiname der SAP Cryptographic Library> z. B. /usr/sap/<SID>/SYS/exe/run/libsapcrypto.so
- snc/identity/as = p:<Distinguished_Name>
- snc/data_protection/min = 3
- snc/data_protection/max = 3
- snc/data_protection/use = 9
- snc/accept_insecure_gui = 0
- snc/accept_insecure_cplic = 0
- snc/accept_insecure_rfc = 0
- snc/accept_insecure_r3int_rfc = 0

Wenn Sie SNC zwischen dem AS ABAP und anderen Serverkomponenten konfigurieren, müssen Sie auf dem Anwendungsserver Zugriffskontrollisteneinträge (Access Control List Entries) vornehmen. Hierfür müssen Sie den vom Kommunikationspartner verwendeten SNC-Namen kennen.

Die SNC-Systemzugriffskontrollistenpflege erfolgt mit TA SM30 in der Tabelle SNCSYSACL (Sicht VSNCSYSACL und type=E).

AS Java und ICM

Bitte beachten Sie folgende SAP Hinweise: 1848999, 510007, 2065806, 2384243, 2110020, 2284059, 2540433, 2384290, 2616423, 1240081, 2708581

Setzen Sie die TLS und Cipher Einstellungen auf folgende Werte.

Eingehende Verbindungen mit der SAP CommonCryptoLib (Setzung der Parameter z. B. in der DEFAULT.PFL):

- ssl/ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

- ssl/client_ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- icm/server_port_<xx> = ..., PROT=HTTPS, ..., SSLCONFIG=ssl_config_<yy>
- icm/ssl_config_<yy> = CRED=<name>.pse, CI-PHERS=544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

Hinweise:

- Wenn Sie die SSL-Konfiguration mit dem Parameter icm/ssl_config_<yy> gepflegt haben, müssen Sie die Option SSLCONFIG vom Parameter icm/server_port_<xx> auf den Wert "ssl_config_<yy>" (<yy> entsprechend dem Parameter icm/ssl_config_<xx> setzen).
- Parameter icm/ssl_config_<yy>: Die CRED Option muss angegeben werden z. B. CRED=SAPSSLSsapext.pse.
- Parameter icm/ssl_config_<yy>: Die Inhalte der Option CIPHERS unterliegen denselben Regeln und derselben Syntax, die für den Profilparameter ssl/ciphersuites gelten.

Ausgehende Verbindungen mit IAIK JCE

Bitte lesen sie SAP Note 2284059 - Aktualisierung der SSL-Bibliothek im NW Java Server. Die folgende Liste führt die sicheren Cipher auf:

- cipherSuite=TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- cipherSuite=TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- cipherSuite=TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- cipherSuite=TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

Alle anderen im Standard vorhandenen Cipher sollten gelöscht werden (also alle mit TLS_RSA) z. B.:

- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256

Unterstützte Änderung der SSL-/TLS-Version (=> TLS 1.2 gefordert). Folgende Einstellung ist vorzunehmen:

- client.minProtocolVersion=TLS12

Das Zertifikat / Schlüssel für SNC müssen bei der Generierung mit sapgenpse mindestens folgende Eigenschaften haben:

- mindestens sha256
- Bitlänge mindestens 2048

SAP HANA, SAP S/4HANA, ABAP, HANA interner SAP Web Dispatcher

Bitte beachten Sie den HANA Security Guide und folgende SAP Hinweise: 1848999, 2829919, 2159014, 3201581, 2110020, 2256091, 1761693, 2487639, 2036111 inkl. Anhang "Frequently_Used_Config_Parameters_in_SAP_HANA_1_3.pdf", 2186744.

ABAP

Setzen Sie die TLS und Cipher Einstellungen mit TA RZ10 mit den Profilparametern auf folgende Werte:

- ssl/ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- ssl/client_ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- icm/server_port_<xx> = ..., PROT=HTTPS, ..., SSLCONFIG=ssl_config_<yy>
- icm/ssl_config_<yy> = CRED=<name>.pse, CI-PHERS=544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

Hinweise:

- Wenn Sie die SSL-Konfiguration mit dem Parameter icm/ssl_config_<yy> gepflegt haben, müssen Sie die Op-

tion SSLCONFIG vom Parameter icm/server_port_<xx> auf den Wert "ssl_config_<yy>" (<yy> entsprechend dem Parameter icm/ssl_config_<xx> setzen).

- Parameter icm/ssl_config_<yy>: Die CRED Option muss angegeben werden z. B. CRED=SAPSSLsapext.pse.
- Parameter icm/ssl_config_<yy>: Die Inhalte der Option CIPHERS unterliegen denselben Regeln und derselben Syntax, die für den Profilparameter ssl/ciphersuites gelten.

Folgendes muss für SNC konfiguriert sein:

Das Zertifikat / Schlüssel für SNC müssen bei der Generierung mit sapgenpse oder der TA STRUST mindestens folgende Eigenschaften haben:

- mindestens sha256 (Option -a)
- Bitlänge mindestens 2048 (Option -s)

Setzen von folgenden SNC Profilparametern (TA RZ10):

- snc/enable = 1
- snc/gssapi_lib = <Pfad und Dateiname der SAP Cryptographic Library> z. B. /usr/sap/<SID>/SYS/exe/run/libsapcrypto.so
- snc/identity/as = p:<Distinguished_Name>
- snc/data_protection/min = 3
- snc/data_protection/max = 3
- snc/data_protection/use = 9
- snc/accept_insecure_gui = 0
- snc/accept_insecure_cplic = 0
- snc/accept_insecure_rfc = 0
- snc/accept_insecure_r3int_rfc = 0

Wenn Sie SNC zwischen dem ABAP und anderen Serverkomponenten konfigurieren, müssen Sie auf dem Anwendungsserver Zugriffskontrolllisteneinträge (Access Control List Entries) vornehmen. Hierfür müssen Sie den vom Kommunikationspartner verwendeten SNC-Namen kennen.

Die SNC-Systemzugriffskontrollistenpflege erfolgt mit TA SM30 in der Tabelle SNCSYSACL (Sicht VSNCSYSACL und type=E).

SAP Web Dispatcher (intern HANA), s. Note 3201581 und für HANA XS Classic. Pflegen der Konfigurationsdatei (webdispatcher.ini):

- ssl/ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

Bitte beachten Sie, dass ssl/client_ciphersuites für den eingebetteten Web Dispatcher nicht relevant ist, da der eingebettete Web Dispatcher nie als HTTPS-Client agiert. Wenn ssl/ciphersuites gesetzt ist, werden sslminprotocolversion und sslciphersuites in der global.ini ignoriert (s. nachfolgend):

Sektion [communication]

- sslminprotocolversion = TLS12
- sslCipherSuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

SAP HANA, SAP S/4HANA inkl. interner SAP Web Dispatcher

SAP empfiehlt die Verschlüsselung der Kommunikation auf allen SAP HANA Channels (externe und interne), wenn es möglich ist (s. SAP Sec. Guide HANA)!

Die Konfiguration des SAP-HANA-Systems kann entweder im DBA Cockpit für SAP HANA (Transaktion ST04 oder DBACOCKPIT) in der Sicht "Configuration -> Ini files" (DE: Konfiguration -> INI-Dateien) geändert werden oder in SAP HANA Studio unter "Administration" in der Konfigurationssicht. Setzen Sie die Parameter/Properties in den entsprechenden Sektionen der global.ini (und oder multidb.ini) Datei:

Sektion [communication]

- sslcryptoprovider = commoncrypto (für SAP CommonCryptoLib)
- ssl = on or systempki (interne Kommunikation zwischen Hosts)
- ssl_local = on (interne Kommunikation innerhalb eines Hosts)
- sslCipherSuites = PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT (Setzung der validen Cipher für die Kommunikation)
- sslEnforce = true (externe Kommunikation zwischen Hosts)
- sslminprotocolversion = TLS12
- sslmaxprotocolversion = MAX

Sektion [cryptography] (wenn FIPS 140-2 Compliance nötig)

- ccl_fips_enabled = true

Sektion [ldap] (für Authentifikation über ein IAM)

- sslminprotocolversion = TLS12
- sslmaxprotocolversion = MAX
- sslciphersuites= PFS:+eAES128:-eAES_CBC::EC_HIGH:+EC_OPT (Setzung der validen Cipher für die Kommunikation)

Sektion [multidb]

- enforce_ssl_database_replication = true

Sektion [system_replication_communication]

- ssl = on (ab Support Package 10 nicht mehr nötig (Note 2256091))
- enable_ssl = on (SSL-Kommunikation für Quell- und/oder Ziel-Replikationskanäle in einem System-Replikationsszenario)

Achtung s. Note 2159014:

Die interne Kommunikation zwischen ABAP und HANA muss auch auf ABAP Seite konfiguriert werden. Betreffend für: Für global.ini -> [communication] -> ssl = on (SAP HANA interne Kommunikation nutzt SSL)
Auf SAP ABAP Seite wird die Verschlüsselung über folgende Parameter aktiviert:

- dbs/hdb/connect_property = ENCRYPT=TRUE

Achtung: Die Clients müssen auch verschlüsselt zugreifen, sonst wird die Kommunikation von HANA abgelehnt. Dies muss separat konfiguriert werden. Bitte schauen Sie sich hierzu das Kapitel 4.3.1.3 TLS/SSL Configuration on the Client des HANA Security Guides und auch die "SAP HANA Client Interface Programming Reference" in der SAP Help an.

Das oben beschriebene aktiviert TLS 1.2 für alle HANA-Ports außer den folgenden zwei:

- SAP HANA database lifecycle manager via SAP Host Agent (Port 1128/1129)
- SAP start service (sapstartsrv) (Port 5xx13/5xx14)

Um die Verschlüsselung der verbleibenden 2 Ports zu aktivieren, werden die Sicherheitseinstellungen dieser Dienste durch die Parameter bestimmt, die in den jeweiligen Konfigurationsdateien auf Betriebssystemebene angegeben sind:

- /usr/sap/hostctrl/exe/host_profile
- /usr/sap/<SID>/SYS/profile/<SID>_HDB##_<hostname>

Die Einstellungen hier sind die gleichen, wie sie auch für ABAP benötigt werden. Für weitere Informationen beachten sie bitte den Hinweis 2384290.

Wie im Hinweis erwähnt, müssen Sie die Parameter ssl/ciphersuites und ssl/client_ciphersuites wie folgt setzen und sowohl den Host-Agent als auch sapstartsrv neu starten.

- ssl/ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- ssl/client_ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- icm/server_port_<xx> = ..., PROT=HTTPS, ..., SSLCONFIG=ssl_config_<yy>

- icm/ssl_config_<yy> = CRED=<name>.pse, CIPHERS=544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT

Hinweise:

- Wenn Sie die SSL-Konfiguration mit dem Parameter icm/ssl_config_<yy> gepflegt haben, müssen Sie die Option SSLCONFIG vom Parameter icm/server_port_<xx> auf den Wert "ssl_config_<yy>" (<yy> entsprechend dem Parameter icm/ssl_config_<xx> setzen).
- Parameter icm/ssl_config_<yy>: Die CRED Option muss angegeben werden z. B. CRED=SAPSSL\$apext.pse.
- Parameter icm/ssl_config_<yy>: Die Inhalte der Option CIPHERS unterliegen denselben Regeln und derselben Syntax, die für den Profilparameter ssl/ciphersuites gelten.

SAP Web Dispatcher

Wenn der SAP Web Dispatcher die SSL-Verbindung beenden soll, muss er über Sicherheitsinformationen verfügen, die er für die SSL-Kommunikation(en) verwenden kann. Diese Informationen werden wie folgt in den Persönlichen Sicherheitsumgebungen (PSEs) des SAP Web Dispatchers abgelegt:

- Für die Verbindung, bei der er die Serverkomponente ist, werden diese Informationen in seiner SSL-Server-PSE abgelegt. Die Standard-SSL-Server-PSE verwendet den Dateinamen SAPSSLS.pse und befindet sich in /usr/sap/<SID>/<webdisp>/sec (Unix).
- Wenn der SAP Web Dispatcher auch eine SSL-Verbindung zum AS Java herstellt, werden seine für diese Verbindung zu verwendenden Sicherheitsinformationen in der SSL-Client-PSE abgelegt. Die Standard-SSL-Client-PSE verwendet den Dateinamen SAPSSLC.pse und befindet sich ebenfalls in /usr/sap/<SID>/<webdisp>/sec (Unix). Install the SAP Cryptographic Library and sapgenpse executable on the SAP Web Dispatcher's host (Unix: /usr/sap/<SID>/<webdisp>).

Bitte beachten Sie folgende SAP Hinweise: 1848999, 2110020, 2384290

Setzen Sie die TLS und Cipher Einstellungen auf folgende Werte in der Profildatei / Konfigurationsdatei.

- ssl/ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- ssl/client_ciphersuites = 544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- icm/server_port_<xx> = ..., PROT=HTTPS, ..., SSLCONFIG=ssl_config_<yy>
- icm/ssl_config_<yy> = CRED=<name>.pse, CIPHERS=544:PFS:eAES_GCM:+eAES128:!kRSA:!eAES_CBC::EC_HIGH:+EC_OPT
- wdisp/ssl_encrypt = 2 (auf jeden Fall verschlüsselt weitergeben)

Hinweise:

- Wenn Sie die SSL-Konfiguration mit dem Parameter icm/ssl_config_<yy> gepflegt haben, müssen Sie die Option SSLCONFIG vom Parameter icm/server_port_<xx> auf den Wert "ssl_config_<yy>" (<yy> entsprechend dem Parameter icm/ssl_config_<xx> setzen).
- Parameter icm/ssl_config_<yy>: Die CRED Option muss angegeben werden z. B. CRED=SAPSSL\$apext.pse.
- Parameter icm/ssl_config_<yy>: Die Inhalte der Option CIPHERS unterliegen denselben Regeln und derselben Syntax, die für den Profilparameter ssl/ciphersuites gelten.

Wenn Sie ein Zertifikat / Schlüssel mit sapgenpse generieren müssen mindestens folgende Eigenschaften vorhanden sein:

- mindestens sha256 (Option -a)
- Bitlänge mindestens 2048 (Option -s)
- CA muss Zertifikat / Schlüssel signieren

SAProuter

Die Verbindung zwischen den benachbarten SAProutern wird mit SNC geschützt. Der SAProuter kann eine SNC verschlüsselte Kommunikation auch zu / zwischen / von anderen Komponenten aufbauen (AS ABAP, DIAG/SAPGUI, Drucker, externe Programme, AS Java).

Das Zertifikat / Schlüssel für SNC müssen bei der Generierung mit sapgenpse mindestens folgende Eigenschaften haben:

- mindestens sha256 (Option -a)
- Bitlänge mindestens 2048 (Option -s)

Überprüfen Sie die Trace Files, um festzustellen, welche SAPcryptoLib / CommonCryptoLib und Cipher aktiviert sind.

- AS ABAP: dev_w0
- AS Java: dev_server<n>
- ICM: dev_icm
- HANA: dev_w0 und dev_webdisp
- SAP Web Dispatcher: dev_webdisp
- SAProuter: dev_rout

Die von den AS Java-Prozessen generierten Protokoll- und Tracedateien und die auf AS Java ausgeführten Anwendungen werden im Verzeichnis `usr\sap\<SID>\<Instanzname>\j2ee\cluster\server<n>\log` gespeichert.

Die HANA Log- und Trace-Dateien befinden sich unter `/usr/sap/<SID>/HDB<Instanznummer>/<Hostname>/trace`.

Requirement-ID: 39a81cec

ID: 3.22-23/13.0

Req 24	Bei symmetrischer Blockverschlüsselung müssen folgende Authenticated Encryption with Associated Data (AEAD) Betriebsarten verwendet werden: Galois-Counter-Mode (GCM) oder Counter with CBC-MAC (CCM).
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Wenn die Länge der zu verschlüsselnden Daten größer als die AES-Blocklänge (128 Bit) ist, muss eine Betriebsart verwendet werden um alle Daten zu verschlüsseln.

GCM und CCM sind Authenticated Encryption with Associated Data (AEAD) Betriebsarten und bieten Vertraulichkeit und Integrität für den Klartext, sowie optional Integrität für zusätzliche Daten (associated data), wie beispielsweise einem Protokoll-Header. Beide Betriebsarten sind z.B. in ISO/IEC 19772, NIST SP 800-38C und NIST SP 800-38D standardisiert. Generell ist Authenticated Encryption (GCM, CCM) den klassischen Betriebsarten (CBC, CTR, OFB) vorzuziehen solange die Anwendung AEAD unterstützt.

Die folgende Tabelle stellt die Sicherheitsdienste der verschiedenen Betriebsarten dar:

Betriebsart	Vertraulichkeit	Integrität
GCM	Ja	Ja
CCM	Ja	Ja
CTR	Ja	Nein
CBC	Ja	Nein
OFB	Ja	Nein

Anforderungen an AEAD Betriebsarten:

- Anforderungen an den Galois-Counter-Mode (GCM):
 - Der Authentication Tag muss mindestens 96 Bit lang sein.
 - Der Initialisierungsvektor (Nonce) darf sich bei gleichem Schlüssel nicht wiederholen und soll 96 Bit lang sein. Kürzere Initialisierungsvektoren sind nicht erlaubt.
- Anforderungen an den Counter Mode with CBC-MAC (CCM):
 - Der Authentication Tag muss mindestens 64 Bit lang sein.
 - Der Initialisierungsvektor (Nonce) darf sich bei gleichem Schlüssel nicht wiederholen.

Hinweis zu Datenschutz:

Die Anforderungen in dem vorliegenden Dokument richten sich an die Daten- und Informationssicherheit im Sinne Vertraulichkeit und Integrität. Sollen kryptografische Methoden eingesetzt werden, um Anonymisierungs- bzw. Pseudonymisierungs-Anforderungen des Datenschutzes zu erfüllen, findet man die Bedingungen für den Einsatz der Methoden im Dokument „Datenschutzanforderung Anonymisierung und Pseudonymisierung (Identifikationsschutz und Datenvermeidung)“.

Motivation: Werden Betriebsarten falsch implementiert, kann es zu einer Schwächung des kryptographischen Systems kommen.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-24/13.0

Req 25	Falls bei symmetrischer Verschlüsselung keine AEAD Betriebsart zur Verfügung steht, muss eine der folgenden Vertraulichkeits-Betriebsarten verwendet werden: Counter Mode (CTR), Cipher-Block Chaining (CBC) und Output Feedback (OFB).
--------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

Wenn die Länge der zu verschlüsselnden Daten größer als die AES-Blocklänge (128 Bit) ist, muss eine Betriebsart verwendet werden, um alle Daten zu verschlüsseln.

Generell ist Authenticated Encryption (GCM, CCM) den Vertraulichkeits-Betriebsarten (CBC, CTR, OFB) vorzuziehen solange es die Anwendung erlaubt.

CBC, CTR und OFB bieten lediglich Vertraulichkeit für den Klartext weshalb sie generell durch separate Mechanismen zum Integritätsschutz (z.B. Message Authentication Codes) ergänzt werden müssen. Die Vertraulichkeits-Betriebsarten sind z.B. in ISO/IEC 10116 oder NIST SP 800-38A standardisiert.

Anforderungen an Betriebsarten die lediglich Vertraulichkeit bieten (vergleiche Annex B aus ISO/IEC 10116):

- Anforderungen an den Counter Mode (CTR):
 - Der Zählerstand (=Counter) und damit auch der initiale Zählerstand (Initialisierungsvektor) sind 128 Bit lang
 - Der Zähler darf sich bei gleichem Schlüssel nicht wiederholen
 - Der Initialisierungsvektor ist so zu wählen, dass sich der Zählerstand bei gleichem Schlüssel nicht wiederholt
- Anforderungen an den Cipher Block Chaining Mode (CBC):
 - Der Initialisierungsvektor (IV) ist 128 Bit lang und muss zufällig gewählt werden
 - Der Initialisierungsvektor darf nicht beeinflussbar oder ableitbar sein
 - Es muss ein geeignetes Padding-Verfahren verwendet werden.
- Anforderungen an den Output Feedback Modus (OFB):

- Der Initialisierungsvektor (IV) ist 128 Bit lang und muss zufällig gewählt werden

Hinweis zu Datenschutz:

Die Anforderungen in dem vorliegenden Dokument richten sich an die Daten- und Informationssicherheit im Sinne Vertraulichkeit und Integrität. Sollen kryptografische Methoden eingesetzt werden, um Anonymisierungs- bzw. Pseudonymisierungs-Anforderungen des Datenschutzes zu erfüllen, findet man die Bedingungen für den Einsatz der Methoden im Dokument „Datenschutzanforderung Anonymisierung und Pseudonymisierung (Identifikationsschutz und Datenvermeidung)“.

Motivation: CBC, CTR und OFB bieten lediglich Vertraulichkeit für den Klartext, weshalb sie generell durch separate Mechanismen zum Integritätsschutz (z.B. Message Authentication Codes) ergänzt werden müssen.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-25/13.0

Req 26	Es müssen kryptographische Hash-Funktionen der SHA-3 und SHA-2 Familie mit einer Ausgabelänge von mindestens 256 Bit verwendet werden.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Die Sicherheit einer Hash-Funktion wird maßgeblich durch ihre Ausgabelänge bestimmt, welche als Suffix angegeben wird (z.B. hat SHA-3-512 eine Ausgabelänge von 512 Bit).

- Standardisierte Varianten der SHA-2 Familie sind [ISO/IEC 10118-3, FIPS 180-4]: SHA-512, SHA-384, SHA-256.
- Standardisierte Varianten der SHA-3 Familie sind [ISO/IEC 10118-3, FIPS 202]: SHA-3-512, SHA-3-384, SHA-3-256.

Die zulässigen Anwendungsgebiete für diese Hash-Funktionen sind:

- Digitale Signaturen
- Message Authentication Codes (MAC)
- Mask generation functions (MGF)
- Schlüsselableitungsfunktionen (Key Derivation Function = KDF) und passwortbasierte Schlüsselableitung (PBKDF)
- Pseudozufallszahlengeneratoren (PRNG)
- Integritätsschutz

Hinweis zu Authentizität und Vertraulichkeit:

Hash-Funktionen bieten keine Vertraulichkeit (im Sinne von Verschlüsselung) und keinen Schutz vor aktiven Angreifern (z.B. gegenüber Manipulationen). Hash-Funktionen sind öffentlich bekannte und schlüssellose Funktionen. Deshalb kann ein Angreifer, der eine Nachricht mit Hashwert manipuliert ebenfalls einfach den Hashwert der neuen Nachricht berechnen. Wenn die Authentizität einer Nachricht mit symmetrischen Verfahren garantiert werden soll müssen Message Authentication Codes (wie z.B. HMAC) verwendet werden.

Anmerkung zu SHA-224/SHA-3-224:

SHA-224/SHA-3-224 darf nur noch für Message Authentication Codes (MAC) und zur Schlüsselableitung (KDF) in Bestandssystemen genutzt werden und sollte gegen Hash Funktionen längerer Outputlänge (≥ 256 Bit) ausgetauscht werden.

Quelle:

[TR2102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Seite 39-40, Februar 2019

Motivation: Kryptographische Hash-Funktionen werden in vielen Anwendungsgebieten eingesetzt und bilden die Grundlage vieler kryptographischer Verfahren, wie digitalen Signaturen.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-26/13.0

Req 27 Folgende Message Authentication Codes (MACs) müssen verwendet werden: HMAC (basierend auf einer Hash-Funktion der SHA-3/2 Familie) oder CMAC (basierend auf AES).

Nutzerrollen: Betrieb, Entwicklung, Integration

Message Authentication Codes (MAC) dienen dem Integritätsschutz von Nachrichten in Sicherheitsprotokollen wie beispielsweise in IPSec oder TLS/SSL. Zudem werden Message Authentication Codes in anderen kryptographischen Verfahren, wie Key Derivation Functions eingesetzt.

Folgende Tabelle beschreibt die zulässigen Algorithmen und Parameter:

MAC-Verfahren	Zulässige Algorithmen	Besondere Betriebsbedingungen / Bemerkungen
HMAC [ISO/IEC 9797-2, FIPS 198]	SHA-512	Die Schlüssellänge muss mindestens 128 Bit betragen. Die Länge des MAC muss mindestens 64 Bit betragen. Es wird empfohlen mindestens 96 Bit lange MACs zu verwenden.
	SHA-3-512	
	SHA-384	
	SHA-3-384	
	SHA-256	
CMAC [ISO/IEC 9797-2, NIST SP 800-38B]	SHA-3-256	
	AES-256	Die Schlüssellänge wird durch die AES-Variante vorgegeben (128/192/256 Bit). Die Länge des MAC muss mindestens 64 Bit betragen. Es wird empfohlen mindestens 96 Bit lange MACs zu verwenden.
	AES-192	
	AES-128	

Hinweis zu CMAC:

Einfache Implementierungen des CBC-MAC sind nicht sicher gegenüber Erweiterungsangriffen. Eine in [ISO/IEC 9797] oder [NIST SP 800-38B] standardisierte und sichere Variante des CBC-MAC ist CMAC (MAC Algorithm 5 in [ISO/IEC 9797]).

Hinweis zu GMAC:

Die Authenticated-Encryption Betriebsart GCM verwendet die GMAC Konstruktion. Diese ist als eigenständiger MAC (also nicht als Bestandteil des GCM) nur unter bestimmten Bedingungen so sicher wie HMAC und CMAC siehe [BSI TR-02102-1], weshalb GMAC in dieser Anforderung nicht explizit berücksichtigt wird.

Einschränkungen zu SHA-224/SHA-3-224/SHA-1:

Die Nutzung von SHA-224/SHA-1 für Message Authentication Codes ist nur noch für bestehende Systeme zulässig wenn kein stärkeres Verfahren verfügbar ist und sollte zeitnah gegen stärkere Varianten ausgetauscht werden (mindestens SHA-256/SHA-3-256).

Hinweis zu Datenschutz:

Die Anforderungen in dem vorliegenden Dokument richten sich an die Daten- und Informationssicherheit im Sinne Vertraulichkeit und Integrität. Sollen kryptografische Methoden eingesetzt werden, um Anonymisierungs- bzw. Pseudonymisierungs-Anforderungen des Datenschutzes zu erfüllen, findet man die Bedingungen für den Einsatz der Methoden im Dokument „Datenschutzanforderung Anonymisierung und Pseudonymisierung (Identifikationsschutz und Da-

tenvermeidung)“.

Motivation: Message Authentication Codes (MAC) dienen dem Integritätsschutz von Nachrichten in Sicherheitsprotokollen wie beispielsweise in IPSec oder TLS/SSL. Zudem werden Message Authentication Codes in anderen kryptographischen Verfahren, wie Key Derivation Functions eingesetzt.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-27/13.0

Req 28	Bei der Erzeugung von Schlüsselmateriel für kryptographische Algorithmen müssen entweder physische Zufallszahlengeneratoren oder folgende kryptographisch starke Pseudozufallszahlengeneratoren (deterministische Quellen) mit ausreichender Entropie verwendet werden: HMAC-DRBG, Hash-DRBG oder CTR-DRBG.
--------	---

Grundlage für HMAC-DRBG, Hash-DRBG oder CTR-DRBG sind die Spezifikationen NIST SP800-90A, ISO18031. Wird für die Erzeugung kryptographischer Schlüssel ein Pseudozufallszahlengenerator (deterministic random bit generator = DRBG) verwendet, muss in die Berechnung des Startwertes (Seed) ein durch einen Mensch unbeeinflussbarer (nichtdeterministischer) Wert einfließen. Dieser Wert muss einen hinreichend großen Informationsgehalt (Entropie) aufweisen.

Motivation: Werden Ausgangswerte mit unterschiedlicher Wahrscheinlichkeit erzeugt kann dies ausgenutzt werden um einen Schlüssel durch statistische Analysen abzuleiten bzw. herzuleiten

Umsetzungsbeispiel: Die SAPcryptoLib / CommonCryptoLib ist einzurichten. Laden Sie die neuste SAP Cryptographic Library und die neuste ausführbare Datei sapgenpse vom SAP Software Download Center.

ID: 3.22-28/13.0

Req 29	Symmetrische Schlüssel müssen gegen unberechtigten Zugriff geschützt werden und dürfen außerhalb der Systemgrenzen nicht in ungeschützter Form übertragen, sowie gespeichert werden.
--------	--

Nutzerrolle: Entwicklung, Betrieb

Allgemeine Anforderungen an den Daten-Zugriff:

Der Datenzugriff auf Schlüsselmateriel muss auf die notwendigen Dienste und Nutzergruppen innerhalb und außerhalb der Systemgrenzen eingeschränkt werden.

Anforderung für Transport und Speicherung symmetrischer Schlüssel:

Für den Transport und Speicherung von symmetrischem Schlüsselmateriel (z.B. Pre-Shared Keys) außerhalb der Systemgrenzen müssen AES-Synthetic Initialization Vector (AES-SIV) oder AES-Keywrap verwendet werden.

Diese Anforderung ist relevant, wenn kryptographische Schlüssel außerhalb eines Key Encapsulation Verfahrens (z.B. RSA-KEM) über einen unsicheren Kanal übertragen werden müssen oder auf einem unsicheren Datenträger gespeichert werden. In diesen Fällen ist der Schlüssel ungeschützt und kann von einem Angreifer abgefangen werden. Deshalb müssen geeignete Verfahren verwendet werden, um die Vertraulichkeit des Schlüssels zu gewährleisten. Diese werden allgemein als Key-Wrap Verfahren bezeichnet und sind in [RFC5297] bzw. [SP800-38F, Algorithmen KW und KWP] spezifiziert.

Die Anforderungen an Sicherheitsparameter dieser Verfahren entsprechen den Anforderungen an AES aus Req. 5.

Hinweis zu sicherer Hardware:

Schlüssel können auch auf sicherer Hardware wie Hardware-Sicherheits-Modulen (HSMs) oder Secure Elements (wie z.B. Smart Cards) gespeichert und transportiert werden. Schlüssel werden dann von der sicheren Hardware und ggf. weiteren kryptographischen Mechanismen (z.B. Key Encryption Keys und Keywrap) geschützt.

Motivation: Symmetrische Schlüssel müssen gegen unberechtigten Zugriff geschützt werden und dürfen außerhalb der Systemgrenzen nicht in ungeschützter Form übertragen, sowie gespeichert werden.

Umsetzungsbeispiel: Limitierung des Zugriffs auf die Schlüsseldatei und des Verzeichnisses in dem der/die Schlüssel liegen:

- Verzeichnis (Unix): 700
- Schlüsseldatei (Unix): 600

Der Austausch vom Schlüssel und vom Schlüsselpasswort muss separiert und gesichert vorgenommen werden z. B. PGP/SMIME, E2E verschlüsselter Messenger, ...

ID: 3.22-29/13.0

Req 30	Verfahren zur Schlüsselvereinbarung müssen Perfect Forward Secrecy (PFS) bieten und durch geeignete Authentisierung gegen Man-in-the-Middle Angriffe geschützt werden (z.B. Authenticated Ephemeral Elliptic Curve Diffie Hellman).
--------	---

Asymmetrische Schlüsselvereinbarung wird in der Regel auf Basis des (Elliptic Curve) Diffie-Hellman Schlüsselaustausches durchgeführt (z.B. Ephemeral EC-DH oder auch MQV). Der Vorteil dieses Schlüsselaustausches ist, dass der komplette Schlüssel nie über den Kommunikationskanal transportiert wird. Zudem bietet der vergängliche (ephemeral) DH-Schlüsselaustausch, wenn die verwendeten Secrets nach Gebrauch gelöscht werden, Perfect Forward Secrecy, was bedeutet, dass selbst ein Angreifer der Zugriff auf ein Langzeitgeheimnis erhält (z.B. privaten RSA Schlüssel) die Vertraulichkeit der vorherigen Kommunikation nicht brechen kann.

Verschiedene asymmetrische Schlüsselaustauschprotokolle sind in ISO/IEC 11770-3 standardisiert. Die Eigenschaften müssen bei Implementierungen und Auswahl von Verfahren berücksichtigt werden (z.B. Vergänglichkeit und Authentifizierung). Um Implementierungsfehler zu vermeiden müssen bei Eigenimplementierungen zudem die Empfehlungen der [BSI TR 02102-1] eingehalten werden.

Hinweis zu standardisierten DH-Gruppen:

Da die Erzeugung der Systemparameter – sogenannte Diffie-Hellman Gruppen – komplex, fehleranfällig und aufwendig ist, werden häufig standardisierte Systemparameter verwendet (siehe z.B. [RFC5114]). Bei der Verwendung dieser Gruppen sind die Mindestlängen der Sicherheitsparameter aus den vorherigen Anforderungen zu beachten, da diese nicht aus der IANA Nummer der Gruppen ersichtlich sind.

Motivation: Die Schlüsselvereinbarung löst das Schlüsselaustauschproblem der symmetrischen Verschlüsselung. Mit Perfect Forward Secrecy kann ein Angreifer durch die Kompromittierung von Langzeitschlüsseln (z.B. einem RSA Schlüssel aus einem TLS-Serverzertifikat) nicht die Vertraulichkeit bereits übertragener Ciphertexte brechen. Authentisierung ist notwendig, um Man-in-the-Middle Angriffe abzuwehren.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-30/13.0

Req 31	Private Schlüssel von asymmetrischen Verfahren müssen gegen unberechtigten Zugriff geschützt werden und dürfen außerhalb der Systemgrenzen nicht in ungeschützter Form übertragen, sowie gespeichert werden.
--------	--

Nutzerrollen: Betrieb, Integration

Allgemeine Anforderungen an den Daten-Zugriff:

Der Datenzugriff auf Schlüsselmaterial muss auf die notwendigen Dienste und Nutzergruppen innerhalb und außerhalb der Systemgrenzen eingeschränkt werden.

Private Schlüssel dienen der Signatur von Nachrichten bzw. zur Entschlüsselung von verschlüsselten Nachrichten. Eine digitale Signatur (persönliche Unterschrift) ist nur mittels eines privaten Schlüssels möglich und gewährleistet die Authentizität. Private Schlüssel dürfen nicht auf zentralen Speichersystemen gespeichert werden oder verteilt werden. Zusätzlich müssen private Schlüssel von natürlichen Personen durch ein persönliches Geheimnis (PIN, Passwort) gegen unbefugte Nutzung geschützt werden.

Anmerkungen zu Fernsignaturen:

Fernsignaturen dürfen nur durch zertifizierte Vertrauensdienstleister (Trust Center wie www.telesec.de) erstellt werden. Das Zertifikat muss die Erfüllung der eIDAS Standards (electronic IDentification, Authentication and trust Services) ausweisen und muss regelmäßig überprüft, verlängert bzw. erneuert werden.

Motivation: Wenn ein Angreifer in den Besitz eines privaten Schlüssels gelangt, kann er die digitale Identität des ursprünglichen Besitzers/Systems einnehmen und sich in seinem "Namen" authentifizieren bzw. Nachrichten entschlüsseln.

Umsetzungsbeispiel: Limitierung des Zugriffs auf die Schlüsseldatei und des Verzeichnisses, in dem der/die Schlüssel liegen:

- Verzeichnis (Unix): 700
- Schlüsseldatei (Unix): 600

Der Austausch vom Schlüssel und vom Schlüsselpasswort muss separiert und gesichert vorgenommen werden z. B. PGP/SMIME, E2E verschlüsselter Messenger, ...

ID: 3.22-31/13.0

Req 32 Kryptographische Langzeitschlüssel müssen eine der Anwendung angemessene begrenzte Gültigkeit besitzen.

Nutzerrolle: Betrieb, Entwicklung, Integration

Generell wird zwischen Langzeitschlüsseln und kurzlebigen Schlüsselmaterial, wie Session Keys unterschieden. Langzeitschlüssel werden über einen längeren Zeitraum für kryptographische Operationen verwendet (z.B. TLS Serverzertifikate oder SSH Schlüssel), wohingegen Session Keys nur für eine Protokollsitzung verwendet und danach gelöscht werden. Der Begriff Langzeitschlüssel wird nicht über eine konkrete Zeitspanne definiert, sondern über die mehrfache Verwendung desselben Schlüsselmaterials.

Die Begrenzung der Gültigkeit kryptographischer Langzeitschlüssel beschränkt die Möglichkeiten der Kryptoanalyse, die Auswirkungen im Fall einer Kompromittierung des privaten Schlüssels sowie das Risiko, dass der zugrundeliegende Algorithmus oder die Schlüssellänge während des Gültigkeitszeitraums des Schlüssels als unsicher klassifiziert werden.

Die Gültigkeit asymmetrischer Langzeitschlüssel kann z.B. durch den Gültigkeitszeitraum des zugehörigen digitalen Zertifikats definiert werden. Dabei ist zu beachten, dass bei einer Zertifikatserneuerung nicht zwingend der Schlüssel erneuert werden muss (hierbei ist die Policy der ausstellenden Certificate Authority zu beachten). Grundsätzlich sollte die gesamte Gültigkeit von Schlüsseln (ggf. verteilt auf mehrere Zertifikate), die für Endanwendungen und damit häufig verwendet werden, zwischen einem und drei Jahren liegen.

In einzelnen, kontrollierbaren Fällen mit geringer Verwendung wie beispielsweise dem Schlüssel einer Certification Authority dürfen längere Gültigkeiten gewählt werden. Im Notfall (z.B. einer Kompromittierung) ist der entsprechende Schlüssel vorzeitig zu sperren.

Die folgenden Gültigkeitszeiträume werden empfohlen:

- Endentitäten (z.B. Anwender, TLS-Server): 1-3 Jahre
- Zwischenzertifizierungsstellen (Subordinate/ Intermediate CA): 8-15 Jahre
- Wurzelzertifizierungsstellen (Root CA): 20-30 Jahre

Es kann erforderlich sein, dass Schlüssel auch über ihre Gültigkeit hinaus verwendet werden müssen. Dies kann für öffentliche Schlüssel zur Prüfung von Signaturen in Langzeitanwendungen oder für private Schlüssel zur Entschlüsselung von Langzeitdaten wie beispielsweise E-Mails der Fall sein.

Motivation: Durch beschränkte Gültigkeitszeiträume von asymmetrischen Langzeitschlüsseln kann die Sicherheit der verwendeten Verfahren/Schlüssellängen in regelmäßigen Abständen angepasst werden.

Umsetzungsbeispiel: TLS-Webserverzertifikat mit einer Laufzeit von einem bis drei Jahren.

Client -> SAP Web Dispatcher => gültige CA nötig

SAP Web Dispatcher -> Applikation (OnPremise) => Selbstsigniert möglich

SapRouter -> SapRouter => Selbstsigniert möglich, da Passwortersatz - keine Abfrage bei CA.

Laufzeiten:

2048 = 1 Jahr

3072 = 3 Jahre

4096 = 5 Jahre (nur selbstsignierte)

ID: 3.22-32/13.0

Req 33	Die Sicherheit von kryptographischen Langzeitschlüsseln muss in regelmäßigen Abständen überprüft und ggf. angepasst werden.
--------	---

Nutzerrolle: Betrieb, Entwicklung, Integration

Auf Grund von steigender Rechnerleistung und neuen Angriffen müssen die Schlüssellängen der eingesetzten kryptographischen Verfahren von Zeit zu Zeit angepasst werden. Im Falle besonderer Entwicklungen der Kryptoanalyse (z.B. durch universelle Quantencomputer) kann sogar eine Anpassung des kryptographischen Verfahrens notwendig sein. Dies betrifft besonders kryptographische Langzeitschlüssel, wie z.B. Schlüsselpaare, für die ein digitales Zertifikat ausgestellt wurde oder SSH Schlüssel. Spätestens bei der Erneuerung der Zertifikate muss überprüft werden, ob das eingesetzte Verfahren und die Schlüssellänge noch als sicher gelten. Beispielsweise wird in BSI TR-02102-1 (Stand 2020) für RSA ab dem Jahr 2023 die Verwendung von 3072 Bit anstelle von 2048 Bit Schlüsseln empfohlen.

Motivation: Fortschritte im Computing machen es notwendig Schlüssellängen anzupassen.

Umsetzungsbeispiel: Überprüfung, ob die Schlüssellänge von Langzeitschlüsseln weiterhin ausreichend ist, wenn ein Zertifikat erneuert wird.

ID: 3.22-33/13.0

Req 34	Kryptographisches Schlüsselmaterial (symmetrisch und asymmetrisch) muss erneuerbar sein.
--------	--

Nutzerrolle: Betrieb, Entwicklung

Aktive Angriffe und Fortschritte in der Computertechnologie machen es notwendig kryptographisches Schlüsselmaterial auszutauschen, z.B. wenn zum Schutz vor zurzeit theoretischen Angriffen mit Quantencomputern von AES-128 auf AES-256 gewechselt wird.

Zudem hängt die Sicherheit eines kryptographischen Verfahrens von der Geheimhaltung des Schlüssels ab, weshalb feste Schlüssel bei Kompromittierung ein großes Risiko darstellen.

Die Schlüsselerneuerung (Rekeying) ist eine notwendige Maßnahme die in modernen Kryptoverfahren umgesetzt wird. Weitere Informationen können z.B. in [ISO/IEC JTC1 SC27 SD 12] gefunden werden. Erneuerte Schlüssel dürfen nicht von aktiv verwendetem Schlüsselmaterial (z.B. Verschlüsselungsschlüssel) abgeleitet werden. Der Betreiber eines Systems muss eine geeignete Lebensdauer von Schlüsseln festlegen.

Motivation: Die Sicherheit von kryptographischen Verfahren hängt von der Geheimhaltung und Güte des Schlüsselmaterials ab.

Umsetzungsbeispiel: Mit Nutzung der SAP Cryptographic Library oder T-Secure ist die Anforderung erfüllt.

ID: 3.22-34/13.0

Req 35	Kompromittierte Schlüssel oder Zertifikate müssen deaktivierbar bzw. sperrbar sein.
--------	---

Nutzerrollen: Betrieb, Entwickler, Integration

Unmittelbar nach Erkennen einer möglichen Kompromittierung von privaten Schlüsseln ist das betroffene Zertifikat zu sperren. Wenn wie z.B. bei SSH keine Zertifikate verwendet werden muss dem betroffenen Schlüsselmaterial das Vertrauen entzogen werden. Es darf beispielsweise nicht mehr möglich sein, sich mit dem privaten Schlüssel zu authentisieren. Sperrprozesse für digitale Zertifikate müssen in andere Regelprozesse integriert werden (z.B. bei Verlust eines Endgerätes).

Motivation: Sollten Angreifer in Besitz eines privaten Schlüssels gelangen (z. B. "privater RSA-Schlüssel"), können diese die Identität des Opfers übernehmen und damit beispielsweise Anwendungen oder Funktionen unberechtigt nutzen.

Umsetzungsbeispiel: Nutzung einer CA und die damit verbunden Möglichkeiten Zertifikate des Zertifikatsmanagement (auszustellen, auszutauschen und zurückziehen).

- Automatisierter Prozess zum Sperren/widerrufen ungenutzter Zertifikate mittels Certificate Revocation List (CRL) als Bestandteil des Certificate Management Protokolls (CMP)

ID: 3.22-35/13.0

Req 36	Nicht mehr benötigtes Schlüsselmaterial muss mit sicheren Verfahren unmittelbar und unwiderrbringlich vernichtet oder unbrauchbar gemacht werden.
--------	---

Nutzerrolle: Betrieb

Folgende elektronische Lösungsverfahren sind empfohlen:

- DoD-5220.22-M (ECE)
- Bruce-Schneier-Algorithmus
- Peter-Gutmann-Algorithmus
- ATA-"Secure-Erase" (bei SSD oder SSHD Festplatten) - alle Bits müssen im Zustand 1 sein

Motivation: Sollten Angreifer in Besitz eines nicht mehr verwendeten Geheimnisses gelangen (z. B. "alter AES-Key"), können aufgezeichnete Nachrichten mit dem (damals gültigen) "alten Schlüssel" entschlüsselt werden.

Umsetzungsbeispiel: - Überschreiben des Key Storage mittels DoD-5220.22-M (ECE) Algorithmus

- Zurückziehen eines Webserver Zertifikats

- Automatisierter Prozess zum Sperren/widerrufen ungenutzter Zertifikate mittels Certificate Revocation List (CRL) als Bestandteil des Certificate Management Protokolls (CMP)

ID: 3.22-36/13.0

Req 37	Folgende kryptographischen Verfahren zur Speicherung und Verifikation von Passwort-Hashes dürfen nur verwendet werden: scrypt [RFC 7914], bcrypt, Argon2, SHA-512-crypt / Crypt(3), SHA-256-crypt / Crypt(3), oder PBKDF2 [RFC8018].
--------	--

Passwörter zur Authentifizierung dürfen nur als Passwort-Hash gespeichert werden.

Sichere Verfahren zur kryptographischen Passwort-Hash-Speicherung und Verifikation:

Verfahren	Maximale Passwort-länge	Salt-Länge	Sicherheitsparameter / Work-Faktor	Zusätzliche Parameter
scrypt	80 Zeichen	128 Bit	$N=2^{18}$; $r=4$; $p=1$	keine
bcrypt	50 Zeichen	128 Bit	2^n mit $n=13$	keine
Argon2	80 Zeichen	128 Bit	$n=256$; 2^m mit $m=10$, $p=1$	keine

SHA-512-crypt / Crypt(3)	80 Zeichen	96 Bit(16 Zeichen)	n=640.000	id=6
SHA-256-crypt / Crypt(3)	80 Zeichen	96 Bit(16 Zeichen)	n=640.000	id=5
PBKDF2	80 Zeichen	128 Bit	n=640.000	keine

Anmerkung zur Passwort-Bildung:

Die aktuellen Anforderungen an die Passwort-Bildung findet man im Dokument „3.01 Technischer Basisschutz IT/NT“.

Anmerkungen zu den Parametern:

Die Parameter-Empfehlungen sind so gewählt, sodass ein Offline-Angriff einer Passwort-Hash Datei unattraktiv ist, indem die Passwort-Hash-Erzeugung verlangsamt wird.

Hinweis zu SHA-512/256-crypt:

Wichtig zu erwähnen ist, dass es nicht zulässig ist native Hashfunktionen für die Erzeugung von Passworthashes zu verwenden, da diese mit speziellen Angriffsmethoden (GPU, ASIC/FPGA) effizient gebrochen werden können. Daher ist SHA-512-crypt eine spezialisierte Implementierung basierend auf dem SHA-512 Algorithmus um Passwort-Hashwerte zu erzeugen und zu verifizieren.

Hinweis für Schlüsselableitungen:

Empfehlungen für die Passwort-basierte Ableitung von Schlüsselmaterial findet man im Kapitel „Schlüsselverwaltung“.

Quellen:

[1] <https://pthree.org/2016/06/28/lets-talk-password-hashing/> (Letzter Aufruf: 04.10.2018)

[2] https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet (Letzter Aufruf: 04.10.2018)

[3] <https://veggiespam.com/painless-password-hash-upgrades/> (Letzter Aufruf: 04.10.2018)

[4] <https://arstechnica.com/information-technology/2013/09/long-passwords-are-good-but-too-much-length-can-be-bad-for-security/> (Letzter Aufruf: 04.10.2018)

[RFC7914] C. Percival, Tarsnap, S. Josefsson, SJD AB: RFC 7914, The scrypt Password-Based Key Derivation Function, 2016

[RFC8018] K. Moriarty, Ed., Dell EMC, B. Kaliski, Verisign, A. Rusch, RSA: RFC 8018, PKCS #5: Password-Based Cryptography Specification Version 2.1, 2017

Motivation: Nur spezielle Passwort-Hash-Algorithmen sind resistent gegenüber Online und Offline Angriffen auf Passwort-Hash-Tabellen.

Umsetzungsbeispiel: Beispiel für den ABAP-Stack:

Für die lokale Anmeldung wird das Passwort als HASH gespeichert. Der SAP-Default ist nicht ausreichend sicher. Die Anforderung ist damit nicht erfüllt. Die Setzung der Einstellung erfolgt über den Profilparameter **login/password_hash_algorithm** (über TA RZ10).

SAP-Default: encoding=RFC2307, algorithm=iSSHA-1, iterations=1024, saltsize=96

Um die Anforderung zu erfüllen, muss der Parameter wie folgt gesetzt werden: "algorithm=iSSHA-256, iterations=15000 (=max) saltsize=128"

Hinweis: Der Wert der "iterations" kann je nach Leistung des Systems kleiner als 1500 gewählt werden.

ID: 3.22-37/13.0

Req 38 Folgende Signatur-Algorithmen müssen eingesetzt werden: EC-DSA, RSA-PSS, DSA/DSS.

Nutzerrollen: Entwicklung, Integration

Signaturalgorithmen werden verwendet, damit der Empfänger (einer signierten Nachricht) den Ersteller und Nachrichtintegrität eindeutig verifizieren kann.

Die Verifikation des signierten Pakets erfolgt mithilfe des öffentlichen Schlüssels. Signaturalgorithmen mit Appendix (d.h. bei denen die Signatur der Nachricht angehängt wird) benötigen immer eine kryptographische Hash-Funktion.

Folgende Tabelle stellt die zulässigen Parameter für digitale Signaturen mit Appendix dar:

Signatur-Algorithmus	Mindestlänge der Sicherheitsparameter	Zulässige Hash-Funktionen
EC-DSA [ISO/IEC 14888-3], [FIPS 186-4]	250 Bit (die Ordnung des Basispunktes der elliptischen Kurve muss mindestens 250 betragen)	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit
RSA-PSS [ISO/IEC 14888-2], [FIPS 186-4]	3000 Bit RSA Modulus (n) mit einem öffentlichen Schlüssel $e > 2^{16}$	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit
DSA / DSS [ISO/IEC 14888-3], [FIPS 186-4]	3000 Bit Primzahl p (definiert den endlichen Körper) 250 Bit Primzahl q	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit

Für weitere Informationen zu den Sicherheitsparametern asymmetrischer Verschlüsselungsverfahren wird auf [BSI TR-02102-1] verwiesen.

Einschränkungen zu SHA-224/SHA-3-224:

SHA-224/SHA-3-224 darf nur noch in Bestandssystemen verwendet werden und muss bei nächster Gelegenheit gegen einen Hashalgorithmus mit einer Output-Länge von mindestens 256 Bit ausgetauscht werden.

Einschränkungen zu RSA PKCS#1 v1.5:

RSA PKCS#1 v1.5 ist nur noch bei Bestandssystemen zulässig und sollte bei Systemänderungen gegen stärkere digitale Signaturverfahren ersetzt werden.

Freigegebene elliptische Kurven:

- brainpoolP512r1
- NIST P-521
- brainpoolP384r1
- NIST P-384
- brainpoolP320r1
- brainpoolP256r1
- NIST P-256
- Curve25519/Ed25519

Abkürzungen:

DSA = Digital Signature Algorithm

DSS = Digital Signature Standard

EC = Elliptic Curve

PKCS = Public Key Cryptography Standards

PSS = Probabilistic Signature Scheme

Motivation: Eine sichere digitale Signatur stellt die Grundlage für eine eindeutige (digitale) Identität dar.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-38/13.0

Req 39	Zertifikate müssen von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt werden.
--------	---

Im Rahmen der PSM Betrachtung kann es möglich sein, selbstsignierte Zertifikate/Schlüssel zu verwenden.

Motivation: Nur wenn die Zertifizierungsstelle (Certificate Authority, CA) vertraut werden kann, ist die Authentizität eines Servers bzw. eines Web Services gewährleistet.

Umsetzungsbeispiel: Eine vertrauenswürdige CA wäre z. B. die TeleSec.

ID: 3.22-39/13.0

2.4. Transport Layer Security (Umsetzungsbeispiel für die Requirements 36 - 40 in Req. 17 eingefügt)

Req 40	Es muss TLS in der Version 1.2 oder 1.3 verwendet werden.
--------	---

Benutzerrollen: Betrieb, Entwicklung, Integration

TLS (Transport Layer Security) ist ein Protokoll zur sicheren Übertragung von Informationen über TCP/IP-basierte Verbindungen und der Nachfolger von SSL (Secure Sockets Layer). Es schützt die Vertraulichkeit, Integrität und Authentizität der Informationen bzw. Kommunikationspartner.

TLS in der Version 1.2 [RFC 5246] und Version 1.3 [RFC 8446] bietet Cipher Suites mit Authenticated Encryption Associated Data (AEAD). AEAD bietet neben dem Schutz der Vertraulichkeit auch Integrität sowie Authentizität der übertragenen Inhalte.

Quellen:

[RFC 5246] T. Dierks, E. Rescorla: RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2, 2008

[RFC 8446] E. Rescorla: RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3, 2018

Motivation: Es muss TLS in der Version 1.2 oder 1.3 verwendet werden.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-40/13.0

Req 41	Es müssen Cipher-Suiten mit Perfect Forward Secrecy (PFS) gemäß folgender Tabellen bei TLS verwendet werden.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Cipher Suites spezifizieren die kryptographischen Verfahren einer Verbindung. Perfect Forward Secrecy (kurz PFS, auch Forward Secrecy) bedeutet, dass Verbindungsinhalte auch bei Kenntnis der Langzeit-Schlüssel der Kommunikationspartner nicht nachträglich entschlüsselt werden können. In TLS v1.2 werden Cipher Suites wie folgt definiert: TLS_AKE_WITH_Enc_Hash. Nachfolgend wird die Bedeutung der einzelnen Bestandteile erläutert:

- AKE (Authenticated Key Exchange): Authentisiertes Schlüsseleinschließungsverfahren für das Handshake-Protokoll
- Enc (Encryption): Verschlüsselungsverfahren mit Betriebsart für das Record-Protokoll
- Hash: Hashfunktion für einen HMAC zur Schlüsselableitung. Sofern Enc kein AEAD-Verschlüsselungsverfahren ist, wird der HMAC außerdem zur Integritätssicherung verwendet.

Die nachfolgende Tabelle listet die zulässigen Cipher Suites mit PFS in TLS v1.2 sowie die Referenzspezifikationen auf. Dabei wurde der Designphilosophie von TLS v1.3 gefolgt, weshalb in dieser Tabelle lediglich AEAD Konstruktionen aufgelistet sind. Zulässige Cipher Suites mit PFS in TLS v1.2

Priorität	Cipher Suite	Referenzspezifikation
-----------	--------------	-----------------------

HOCH	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	RFC 5289
HOCH	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	RFC 5289
NIEDRIG	TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	RFC 5288
NIEDRIG	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	RFC 5288
HOCH	TLS_ECDHE_ECDSA_WITH_CHACHA20POLY1305_SHA256	RFC 7905
HOCH	TLS_ECDHE_RSA_WITH_CHACHA20POLY1305_SHA256	RFC 7905
NIEDRIG	TLS_DHE_RSA_WITH_CHACHA20POLY1305_SHA256	RFC 7905
HOCH	TLS_ECDHE_ECDSA_WITH_AES_256_CCM	RFC 7251
NIEDRIG	TLS_DHE_RSA_WITH_AES_256_CCM	RFC 6655
HOCH	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	RFC 5289
HOCH	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	RFC 5289
NIEDRIG	TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	RFC 5288
NIEDRIG	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	RFC 5288
HOCH	TLS_ECDHE_ECDSA_WITH_AES_128_CCM	RFC 7251
NIEDRIG	TLS_DHE_RSA_WITH_AES_128_CCM	RFC 6655

In Bestandsystemen sind außerdem die Cipher Suites aus der folgenden Tabelle erlaubt.
Zusätzliche Cipher Suites mit PFS in TLS v1.2 mit AES-CBC:

Priorität	Cipher Suite	Referenzspezifikation
HOCH	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	RFC 5289
HOCH	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	RFC 5289
NIEDRIG	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256	RFC 5246
NIEDRIG	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	RFC 5246

HOCH	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	RFC 5289
HOCH	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	RFC 5289
NIEDRIG	TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	RFC 5246
NIEDRIG	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	RFC 5246

Anmerkung zu Cipher Suites für TLS v1.2:

Die Tabelleneinträge sind nach dem symmetrischen Verschlüsselungsverfahren (Enc) sortiert. Für das authentifizierte Schlüsseleinigungsverfahren (AKE) werden Verfahren basierend auf elliptischen Kurven (ECDHE_ECDSA) bevorzugt. DHE (diskreter Logarithmus) Schlüsselvereinbarungen sind anfälliger gegen DoS (DHeater) Angriffe als ECDHE (elliptische Kurven). Daher sollte ECDHE bevorzugt werden.

In TLS v1.3 werden Cipher Suites wie folgt definiert: TLS_AEAD_Hash.

Nachfolgend wird die Bedeutung der einzelnen Bestandteile erläutert:

- AEAD: Authentisiertes Verschlüsselungsverfahren für das Record-Protokoll
- Hash: Hashfunktion für HMAC und HKDF im Handshake-Protokoll

Die nachfolgende Tabelle listet die empfohlenen Cipher Suites mit PFS in TLS v1.3 sowie die Referenzspezifikationen auf.

Zulässige Cipher Suites mit PFS in TLS v1.3

Cipher Suite	Referenz-spezifikation
TLS_AES_256_GCM_SHA384	RFC 8446
TLS_CHACHA20_POLY1305_SHA256	RFC 8446
TLS_AES_128_GCM_SHA256	RFC 8446
TLS_AES_128_CCM_SHA256	RFC 8446

Quellen:

[1] Bundesamt für Sicherheit in der Informationstechnik, TR-02102-2, Version 2022-01, 24.01.2022

[2] <https://ciphersuite.info/cs/?security=secure&sort=asc>

[3] <https://ciphersuite.info/cs/?singlepage=true&security=recommended#>

Motivation: Der Einsatz von modernen Cipher Suites mit Perfect Forward Secrecy schützt die Transport-Sicherheit in TLS.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben

ID: 3.22-41/13.0

Req 42 Für TLS müssen die Diffie-Hellman Gruppen aus der folgenden Tabelle verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Die Diffie-Hellman Gruppe wird zum Schlüsselaustausch mit Perfect Forward Secrecy (PFS) verwendet. Generell kann zwischen Elliptic Curve Gruppen und Finite Field Gruppen (mod p) unterschieden werden. Die folgende Tabelle beinhaltet die zulässigen Diffie-Hellman Gruppen

Zulässige Diffie-Hellman Gruppen für den Einsatz in TLS:

Diffie-Hellman Gruppe	IANA-Nr.	Referenzspezifikation
brainpoolP512r1	33	RFC 7027
secp521r1	25	RFC 8422
brainpoolP384r1	27	RFC 7027
secp384r1	24	RFC 8422
brainpoolP256r1	26	RFC 7027
secp256r1	23	RFC 8422
ffdhe4096	258	RFC 7919
ffdhe3072	257	RFC 7919

Hinweis zu DH Group 256:

Die Diffie-Hellman Gruppe 256 (IANA-Nr. 256) besitzt eine Schlüssellänge von 2048 Bit [1] [2] und darf nur bei Bestandssystemen bis Ende des Jahres 2025 eingesetzt werden[2]. Die Gruppe muss bei nächster Gelegenheit gegen ein stärkeres Verfahren (gemäß o.g. Aufzählung) ersetzt werden.

Hinweis zu DH Gruppen 255, 257 und 258:

Diese Diffie-Hellman Gruppen sind anfälliger gegenüber dem DHater (DoS) Angriff als elliptische Kurven Diffie-Hellman. Daher wird empfohlen Brainpool oder NIST (secp) Kurven für die Diffie-Hellman Schlüsselvereinbarung zu nutzen.

Quellen:

[1] Bundesamt für Sicherheit in der Informationstechnik, TR-02102-2, Version 2022-01, 24.01.2022

[2] SOG-IS Crypto Evaluation Scheme: Agreed Cryptographic Mechanisms, v1.2, Januar 2020

Motivation: Standardisierte Diffie-Hellman Gruppen verwenden sichere Parameter und beschleunigen den Schlüsselaustausch.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben

ID: 3.22-42/13.0

Req 43 Für TLS müssen digitale Zertifikate von einer geeigneten Zertifizierungsstelle mit ausreichender Schlüssellänge und begrenzter Gültigkeit verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Digitale Zertifikate werden in TLS zur Authentisierung während des TLS Handshakes verwendet. TLS Server müssen ein geeignetes TLS Zertifikat verwenden, Clients benötigen ein Zertifikat falls gegenseitige Authentisierung gefordert ist (d.h. mutual authentication).

TLS Zertifikate für Webserver, die aus dem Internet erreichbar sind müssen von einer öffentlichen Zertifizierungsstelle ausgestellt werden, die von Browsern und Betriebssystemen als Vertrauenswürdig eingestuft ist. Diese können z.B. über den Dienst „TeleSec ServerPass“ beauftragt werden (siehe <https://www.telesec.de/de/serverpass>). Bei Schlüssellängen, Gültigkeit und weiteren Konfigurationsmöglichkeiten muss die Certificate Policy der Zertifizierungsstelle beachtet werden.

Bei Webservern, die ausschließlich für interne Anwendungen verwendet werden und die nicht über das Internet erreichbar sind können digitale Zertifikate von einer privaten (internen) Public Key Infrastruktur verwendet werden.

Die Mindestanforderungen aus der folgenden Tabelle müssen für jede Form von TLS-Zertifikaten, d.h. auch für Client-zertifikate, beachtet werden:

Algorithmenfamilie	Schlüssellänge	Hashfunktion
--------------------	----------------	--------------

Elliptic Curve	250 Bit	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit
Digital Signature Algorithm (DSA)	3000 Bit	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit
RSA	3000 Bit	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit

Hinweise zu DSA und RSA Zertifikaten:

Für DSA und RSA dürfen Schlüssellängen kleiner als 3000 Bit nur noch in Bestandssystemen bis zum Jahre 2023 [BSI TR 02102-1] eingesetzt werden und sollten zum nächstmöglichen Zeitpunkt ersetzt werden. Aufgrund der besseren Performanz, sind elliptische Kurven Zertifikate (EC-DSA) zu bevorzugen (falls technisch unterstützt und realisierbar). RSA-PKCS#1 v1.5 darf nur noch in Bestandssystemen eingesetzt werden und sollte, wenn technisch möglich, zum nächstmöglichen Zeitpunkt ersetzt werden [BSI TR 02102-1].

Einschränkungen zu SHA-224/SHA-3-224:

SHA-224/SHA-3-224 darf nur noch in Bestandssystemen verwendet werden und muss bei nächster Gelegenheit gegen einen Hashalgorithmus mit einer Output-Länge von mindestens 256 Bit ausgetauscht werden.

Die Gültigkeit von TLS Zertifikaten sollte 3 Jahre nicht überschreiten.

Quellen:

[BSI TR 02102-1] Bundesamt für Sicherheit in der Informationstechnik: Kryptographische Verfahren: Empfehlungen und Schlüssellängen, TR-02102-1, Version 2022-01, 24.01.2020

[1] Bundesamt für Sicherheit in der Informationstechnik, TR-02102-2, Version 2022-01, 24.01.2022

[2] SOG-IS Crypto Evaluation Scheme: Agreed Cryptographic Mechanisms, v1.2, Januar 2020

Motivation: Digitale Zertifikate bilden die Basis der Authentisierung und schaffen Vertrauen. Ohne ausreichend starke Authentisierung sind Man-in-the-Middle Angriffe möglich.

Umsetzungsbeispiel: Nutzung von Zertifikaten einer anerkannten CA

ID: 3.22-43/13.0

Req 44	Die Sicherheit der TLS-Implementierung und -Konfiguration von Webservern muss regelmäßig überprüft werden.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Sichere Cipher Suites enthalten die wesentlichen Bestandteile eines Sicherheitsprotokolls:

- Authentication-Algorithmus
- Key Exchange-Algorithmus
- Encryption-Algorithmus
- MAC-Algorithmus

Die Sicherheit ist jedoch bestimmt von der jeweilig verwendeten Implementierung / Software. Desweiteren ist sie abhängig von der eingesetzten Protokollversion. Daher müssen die eingesetzten Ciphersuites auf Basis von Security Best Practices ausgewählt werden.

Empfohlene Informationsquellen und Testwerkzeuge:

- Cipher Suite Info (<https://ciphersuite.info/>)
- SSL Labs (<https://www.ssllabs.com/ssltest/>)
- Testssl (<https://testssl.sh/>)
- DTSP Internal Scan Platform (<https://dtsp.telekom-dienste.de/>)

Motivation: Nur sichere Cipher Suites erlauben eine minimale Angriffsfläche und schützen vor Downgrade-Angriffen.

Umsetzungsbeispiel: Die Umsetzung der Anforderung ist in Anforderung 23 unter dem Punkt "Umsetzungsbeispiel" beschrieben.

ID: 3.22-44/13.0

2.5. Logging

Req 45	Die Systemzeit muss zu einer exakten Referenzzeit (Zeitnormal) synchronisiert werden.
--------	---

Als Zeitnormal ist eine Zeitgeber-Quelle heranzuziehen, die ein Zeitsignal auf Basis der koordinierten Weltzeit ("UTC" = "Universal Time Coordinated") liefert.

Anmerkung: die anhand der Zeitgeber-Quelle zu UTC synchronisierte Systemzeit darf auf dem lokalen System mittels entsprechender Zeitzone-Konfiguration bei der Ausgabe von Zeitinformationen in die Ortszeit transformiert werden, sofern die so stattfindende Zeitzoneanpassung eindeutig erkennbar und nachvollziehbar ist.

Systeme, die zu einer gemeinsamen Sicherheits-Domäne gehören, müssen sich einheitlich an derselben Zeitgeber-Quelle synchronisieren.

Motivation: Zeitsynchronisation kann auf einem System eine technische Voraussetzung sein, um zeitkritische Abhängigkeiten in beispielsweise der Nutzung von Zertifikaten oder Authentisierungsmechanismen korrekt bedienen zu können. Dringend erforderlich sind außerdem exakte Zeitstempel in Protokollierungseinträgen, damit diese bei Sicherheitsvorfällen oder Problemanalysen in eindeutiger zeitlicher Relation auswertbar sind.

Umsetzungsbeispiel: Einige zulässige Zeitgeber-Quellen:

- vertrauenswürdig eingestuftes NTP ("NetworkTimeProtocol") Server im IP-Netzwerk
- DCF77-Funksignal über physisch angebundenes Empfängermodul
- GPS-Funksignal über physisch angebundenes Empfängermodul

ID: 3.22-45/13.0

Req 46	Sicherheitsrelevante Ereignisse müssen mit einem exakten Zeitstempel und einer eindeutigen Systembezeichnung protokolliert werden.
--------	--

Sicherheitsrelevant sind jegliche Ereignisse, die Auswirkungen auf die Informationssicherheit (Vertraulichkeit, Integrität, Verfügbarkeit) eines Systems haben können oder dessen mißbräuchliche Verwendung anzeigen. Systeme müssen das Auftreten von sicherheitsrelevanten Ereignissen protokollieren.

Um eine Zuordnung und Auswertung dieser Ereignisse zu ermöglichen, müssen diese zusammen mit einer eindeutigen Systembezeichnung (z. B. Hostname, IP- oder MAC-Adresse) und dem genauen Zeitpunkt des Auftretens ("Zeitstempel") protokolliert werden. Der Zeitstempel eines protokollierten Ereignisses muss mindestens folgende Informationen beinhalten:

- Datum des Ereignisses (Jahr, Monat, Tag)
- Uhrzeit des Ereignisses (Stunden, Minuten, Sekunden)
- Zeitzone, auf die sich diese Zeitangaben beziehen

Die Sicherheitsrelevanz konkreter Ereignisse ergibt sich individuell aus Art und Funktion des Systems sowie den darin verarbeiteten Daten. Daher kann keine einheitliche Liste aus Ereignissen, die Sicherheitsrelevanz besitzen, vorgegeben werden. Relevante Ereignisse müssen passend zum System identifiziert werden. Die folgende Auflistung enthält typische sicherheitsrelevante Ereignisse und ist entsprechend als eine grundlegende Orientierungshilfe bei der Identifikation der sicherheitsrelevanten Ereignisse für ein bestimmtes System zu verstehen.

Sicherheitsrelevantes Ereignis	Informationen, die zum Ereignis zu protokollieren sind
--------------------------------	--

Fehlerhafte Anmeldeversuche	• betroffenes Benutzerkonto, • Anzahl der Fehlversuche • Quelle des Zugriffs (IP-Adresse, Client-ID/Client-Name, lokale Konsole, ...)
Systemzugriffe über Benutzerkonten (einschliesslich System-Konten bzw. technischer Benutzerkonten)	• betroffenes Benutzerkonto • Startzeit/Endzeit des Zugriffs (Anmeldung/Abmeldung) • Quelle des Zugriffs (IP-Adresse, Client-ID/Client-Name, lokale Konsole, ...)
Änderungen an Benutzerkonten (einschliesslich System-Konten bzw. technischer Benutzerkonten)	• Benutzerkonto, mit dem die Änderung vorgenommen wurde • Benutzerkonto, an dem die Änderung vorgenommen wurde • Art der vorgenommenen Änderung (Neueinrichtung, Aktivierung, Deaktivierung, Löschung, Passwort-Rücksetzung, ...)
Änderungen der Rollenzuweisungen/Berechtigungen von Benutzerkonten (einschliesslich System-Konten bzw. technischer Benutzerkonten)	• Benutzerkonto, mit dem die Änderung vorgenommen wurde • Benutzerkonto, an dem die Änderung vorgenommen wurde • Bezeichnung der betroffenen Rolle / Typ der betroffenen Berechtigung • Art der vorgenommenen Änderung (Rolle/Berechtigung hinzugefügt, Rolle/Berechtigung entfernt, ...)
Nutzung/Initiierung von kritischen Systemfunktionen	• Benutzerkonto, mit dem die Nutzung vorgenommen wurde • Bezeichnung/Typ der genutzten Systemfunktion
Zugriff auf sensible Daten ("Sensitive Data"), die im System verarbeitet werden	• Benutzerkonto, mit dem der Zugriff vorgenommen wurde • Bezeichnung des betroffenen Datenfelds (der Inhalt des Datenfelds sollte dabei nicht mit protokolliert werden, um einen Datenabfluss über Log-Dateien zu verhindern) • Art des Zugriffs (Lesen, Ändern, Löschen, ...)
Abnormaler Anstieg in der Auslastung von Systemressourcen, wie etwa Arbeitsspeicher, CPU, Netzwerk oder Festplattenzugriffen über einen längeren Zeitraum	• Schwellwert der betroffenen Systemressource • Dauer der Überschreitung des Schwellwerts • Höhe der Überschreitung des Schwellwerts Es müssen abhängig vom individuellen System sinnvolle Schwellwerte definiert werden.
Sonstige Ereignisse, die für versuchte oder erfolgreiche Angriffe gegen das System Hinweise liefern können *)	• individuell abhängig von der Art des Ereignisses

*) Dies erfordert eine Identifizierung der wichtigsten, für das System typischerweise zu erwartenden Angriffsformen und die Sicherstellung, dass zu deren Erkennung relevante Ereignisse durch das System tatsächlich protokolliert werden. In der Regel kann das nur durch den Entwickler/Hersteller des Systems gewährleistet werden.

Bei der Protokollierung müssen die jeweils geltenden gesetzlichen und betrieblichen Bestimmungen beachtet werden. Letztere umfassen insbesondere auch Vereinbarungen, die mit den betrieblichen Sozialpartnern getroffen wurden. Diese Bestimmungen besagen unter anderem, dass die Protokollierung von Ereignissen nur zweckgebunden erfolgen darf. Die Protokollierung von Ereignissen zum Zweck der Arbeitskontrolle von Mitarbeitern ist nicht zulässig.

Zudem muss - wie für jegliche Daten, die von einem System verarbeitet werden - auch für Protokollierungsdaten ein angemessener Schutzbedarf berücksichtigt und umgesetzt werden; dies gilt für die Speicherung, die Übertragung sowie den Zugriff. Insbesondere falls Wirkdaten in den Protokollierungsdaten enthalten sind, muss derselbe Schutzbedarf berücksichtigt werden, der auch für die reguläre Verarbeitung dieser Wirkdaten innerhalb des Quell-Systems Anwendung findet.

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

ID: 3.22-46/13.0

Req 47	Für sicherheitsrelevante Protokollierungsdaten, die lokal erfasst werden, müssen geltende Aufbewahrungszeiträume und Löschfristen eingehalten werden.
--------	---

Aus Sicht der IT-Sicherheit wird auf einem System die lokale Erfassung von sicherheitsrelevanten Protokollierungsdaten nicht zwingend gefordert. Da lokal erfasste Protokollierungsdaten bei Systemstörungen beschädigt oder von einem erfolgreichen Angreifer gezielt manipuliert werden können, sind sie für sicherheitstechnische bzw. forensische Analysen nur bedingt belastbar. Entsprechend ist für die IT-Sicherheit relevant, dass Protokollierungsdaten an einen separaten Log Server weitergeleitet werden.

Eine lokale Erfassung kann dennoch erfolgen; beispielsweise falls bei der Erzeugung der Protokollierungsdaten eine lokale Speicherung aufgrund technischer Abläufe zunächst unabdingbar ist oder es begründete betriebliche Interessen gibt, Protokollierungsdaten auch lokal vorzuhalten.

Folgende Grundregelung ist bei einer lokalen Erfassung von Protokollierungsdaten zu berücksichtigen:

- Sicherheitsrelevante Protokollierungsdaten müssen für einen Zeitraum von 90 Tagen aufbewahrt werden.
(Diese Anforderung gilt nur, falls auf dem System keine zusätzliche Weiterleitung an einen separaten Log Server umgesetzt ist und Protokollierungsdaten daher ausschließlich lokal erfasst werden.)
- Nach Ablauf von 90 Tagen muss eine umgehende Löschung von gespeicherten Protokollierungsdaten erfolgen.

Abweichungen

Abweichende Aufbewahrungszeiträume und Löschfristen können durch gesetzliche oder regulatorische Anforderungen (insbesondere im Zusammenhang mit personenbezogenen Daten) bestehen oder durch vertragliche Vereinbarungen definiert sein. In diesen Fällen sind die geltenden Zeiträume individuell mit einem Project Security Manager (PSM) bzw. Datenschutzberater (DSB) abzustimmen oder werden durch diesen vorgegeben.

Motivation: Protokollierungsdaten sind ein immens wichtiges Werkzeug der IT-Sicherheit zur Abwendung, Detektierung und Aufklärung von Systemstörungen, Sicherheits- und Datenschutzvorfällen. Auf der anderen Seite unterliegt die Erfassung von Protokollierungsdaten wie jede andere Datenverarbeitung auch gesetzlichen und regulatorischen Auflagen. Entsprechend müssen Richtlinien eingehalten werden, die beides in Einklang bringen.

Umsetzungsbeispiel: Unter Berücksichtigung der gültigen Gesetzeslage und zutreffenden Datenschutz-Regularien werden folgende Löschfristen für lokal erfasste sicherheitsrelevante Protokollierungsdaten auf einem beispielhaften TK-System umgesetzt:

- Standard System Logs: Löschung nach spätestens 90 Tagen
- Protokollierungen von öffentlichen IP-Adressen: Löschung (oder Anonymisierung) nach spätestens 7 Tagen
- Protokollierungen der Vergabe von dynamischen öffentlichen IP-Adressen durch die TK-Lösung: Löschung nach spätestens 7 Tagen

- Protokollierungen nicht abrechnungsrelevanter Call Detail Records: Löschung nach spätestens 7 Tagen
- Protokollierungen der Kommunikations-Inhalte von E-Mail und SMS: Löschung nach spätestens 24 Stunden
- Protokollierungen der behandelten Domain-Abfragen des DNS-Servers der TK-Lösung: Löschung nach spätestens 24 Stunden

ID: 3.22-47/13.0

Req 48 Sicherheitsrelevante Protokollierungsdaten müssen umgehend nach deren Erzeugung an einen Log-Server weitergeleitet werden.

Die sicherheitsrelevanten Protokollierungsdaten des Systems müssen umgehend nach deren Erzeugung an einen Log-Server weitergeleitet werden.

Im Sinne der Anforderung muss der dafür genutzte Log-Server die folgenden Kriterien erfüllen:

- Der Log-Server muss dediziert für die Vorhaltung von Protokollierungsdaten eingesetzt sein. Er darf keine weiteren Funktionen oder Dienste abbilden, die nicht in Relation zu dieser Aufgabe stehen. Eine zulässige Ausnahme ist die Kombination eines Log Servers mit einer SIEM (Security Information and Event Management) Lösung.
- Der Log-Server muss vom System separiert sein. Er darf nicht als zusätzliches Modul unmittelbar mit auf dem System betrieben werden.
- Der Log-Server muss zentral für die Einsatzumgebung bereitstehen, zu der auch das System gehört.

Zur Übertragung der Protokollierungsdaten an den Log-Server sollten bevorzugt auf diesen Zweck spezialisierte Standard-Protokolle (z. B. Syslog oder SNMPv3) zum Einsatz kommen.

Motivation: Werden Protokollierungsdaten nur lokal gespeichert, können diese im Fall einer erfolgreichen Kompromittierung des Systems durch einen Angreifer manipuliert werden, um dessen Angriff und auf dem System durchgeführte Aktionen zu verschleiern. Aus diesem Grund muss die Weiterleitung von Protokollierungsdaten an einen Log-Server sehr zeitnah nach dem Auftreten eines sicherheitsrelevanten Ereignisses im Quell-System erfolgen. So stehen die angefallenen Protokollierungsdaten separiert vom Quell-System und unabhängig von dessen Zustand jederzeit für Analysen bereit. Die Verwendung eines zentralen Log-Servers für eine Einsatzumgebung stellt sicher, dass auf die Logs sämtlicher IT/NT-Systeme dieser Einsatzumgebung schnell und einheitlich zugegriffen und der Zugriff selbst optimal in die Incident-Prozesse der Umgebung integriert werden kann.

Umsetzungsbeispiel: Für SAP müssen die entsprechenden Protokolldaten umgehend nach deren Erzeugung (Real Time) an ein externes System weitergeleitet werden.

Folgende sicherheitskritische Logs (am Beispiel von ABAP) sind auf einem geschützten und separaten System zu sichern:

- SAP Security Audit Log
- Systemprotokollierung
- Gateway Protokollierung
- HTTP Server und Client Protokollierung
- Business Transaktions Protokoll

Das SAP Enterprise Thread Detection (ETD) könnte zur Speicherung und Verwaltung der Logdateien genutzt werden.

ID: 3.22-48/13.0

Req 49 Für sicherheitsrelevante Protokollierungsdaten, die an den Log Server weitergeleitet werden, muss die Einhaltung geltender Aufbewahrungszeiträume und Löschfristen sichergestellt werden.

Folgende Grundregelung ist zu berücksichtigen:

- sicherheitsrelevante Protokollierungsdaten müssen auf dem Log Server für einen Zeitraum von 90 Tagen aufbewahrt werden.
- nach Ablauf von 90 Tagen muss eine umgehende Löschung der auf dem Log Server gespeicherten Protokollierungsdaten erfolgen.

Abweichungen

Abweichende Aufbewahrungszeiträume und Löschfristen können durch gesetzliche oder regulatorische Anforderungen bestehen (insbesondere im Zusammenhang mit personenbezogenen Daten) oder durch vertragliche Vereinbarungen definiert sein. In diesen Fällen sind die geltenden Zeiträume individuell mit einem Project Security Manager (PSM) bzw. Datenschutzberater (DSB) abzustimmen oder werden durch diesen vorgegeben.

Log Server in Verantwortung eines Dritten

Sollte der ausgewählte Log Server nicht innerhalb derselben operativen Verantwortung liegen wie das Quellsystem der Protokollierungsdaten, muss sichergestellt werden, dass dem verantwortlichen Betreiber des Log Servers die gültigen Parameter für die zu empfangenden Protokollierungsdaten bekannt sind und durch diesen im Sinne der hier genannten Regelungen eingehalten werden.

Motivation: Protokollierungsdaten sind ein immens wichtiges Werkzeug der IT-Sicherheit zur Abwendung, Detektierung und Aufklärung von Systemstörungen, Sicherheits- und Datenschutzvorfällen. Auf der anderen Seite unterliegt die Erfassung von Protokollierungsdaten wie jede andere Datenverarbeitung auch gesetzlichen und regulatorischen Auflagen. Entsprechend müssen Richtlinien eingehalten werden, die beides in Einklang bringen.

Umsetzungsbeispiel: Unter Berücksichtigung der gültigen Gesetzeslage und zutreffenden Datenschutz-Regularien werden folgende Löschfristen für ausgeleitete sicherheitsrelevante Protokollierungsdaten eines beispielhaften TK-Systems auf dem separaten Log Server umgesetzt:

- Standard System Logs: Löschung nach spätestens 90 Tagen
- Protokollierungen von öffentlichen IP-Adressen: Löschung (oder Anonymisierung) nach spätestens 7 Tagen
- Protokollierungen der Vergabe von dynamischen öffentlichen IP-Adressen durch die TK-Lösung: Löschung nach spätestens 7 Tagen
- Protokollierungen nicht abrechnungsrelevanter Call Detail Records: Löschung nach spätestens 7 Tagen
- Protokollierungen der Kommunikations-Inhalte von E-Mail und SMS: Löschung nach spätestens 24 Stunden
- Protokollierungen der behandelten Domain-Abfragen des DNS-Servers der TK-Lösung: Löschung spätestens nach 24 Stunden

ID: 3.22-49/13.0

Req 50	SETAM oder SIEM: Sicherheitsrelevante Protokollierungsdaten müssen echtzeitnah an eine Erkennungs- und Reaktionslösung für Sicherheitsvorfälle bereitgestellt werden.
--------	---

Eine zulässige Erkennungs- und Reaktionslösung im Sinne dieser Anforderung muss die folgenden Kriterien erfüllen:

- Es erfolgt eine kontinuierliche Analyse bzw. Korrelation der aus angebundenen IT/NT-Systemen bereitgestellten Protokollierungsdaten.
- Die Identifizierung bedrohlicher Aktivitäten und laufender Angriffe findet auf Basis einer fortlaufend aktuell gehaltenen Erkennungsmuster-Datenbank statt.
- Integrierte Alarmierungsprozesse lösen im Fall eines Fundes die Einleitung von Gegenmaßnahmen aus.
- Sicherheitsexperten sind aktiv involviert in den Betrieb der Lösung sowie die Auswertung von Alarmen.

Eine solche Lösung ist in ihrer Funktion nicht mit einem Log Server gleichzusetzen. Entsprechend muss die Bereitstellung an eine Erkennungs- und Reaktionslösung stets zusätzlich zu einer vorhandenen Weiterleitung an einen Log Server erfolgen. Eine Ausnahme besteht, falls der genutzte Log Server seinerseits die Protokollierungsdaten umgehend nach Erhalt zusätzlich an eine Erkennungs- und Reaktionslösung bereitstellt. In diesem Fall ist es ausreichend, dass

das System die Protokollierungsdaten nur an den Log Server weiterleitet.

Die Protokollierungsdaten müssen ab dem Zeitpunkt ihrer Erzeugung im System innerhalb von 5 Minuten an die Erkennungs- und Reaktionslösung bereitgestellt werden ("echtzeitnah").

Die konkret zu unterstützenden technischen Schnittstellen und Protokolle für die Bereitstellung werden durch die im Einsatzumfeld des Systems verfügbare Erkennungs- und Reaktionslösung sowie die technische Basis des Systems bestimmt.

Motivation: IT/NT-Systeme sind permanent sicherheitstechnischen Bedrohungen und Angriffen ausgesetzt. Mittels einer zentralen Analyse und der Korrelation erfasster System-Ereignisse ganzer IT-Landschaften in nahezu Echtzeit können aktuelle Bedrohungen und laufende Angriffe auch über Systemgrenzen hinaus erkannt und zeitnah passende Gegenmaßnahmen eingeleitet werden. Die unverzügerte Bereitstellung von Protokollierungsdaten an die Erkennungs- und Reaktionslösung ist essenziell für deren Effektivität.

Umsetzungsbeispiel: **Beispiel 1:** Ein System auf Basis eines klassischen Servers erfasst sicherheitsrelevante Protokollierungsdaten im lokalen Syslog oder Windows Event Log des Betriebssystems. Die Protokollierungsdaten werden kontinuierlich zur weiteren Analyse, Angriffserkennung und Alarmbehandlung an ein SIEM (Security Information and Event Management) weitergeleitet.

SAP-Applikationslogs betreffend:

- Nutzung eines SIEM für nicht DTAG SAP-Systeme.
- Nutzung SETAM (SAP Real Time Threat Detection Monitoring) für DTAG SAP-Systeme. SETAM gilt als SIEM für SAP-Applikationen.

Beispiel 2: Auf einem System wird ein EDR ("Endpoint Detection and Response") Agent ausgeführt. Der EDR-Agent leitet die von ihm auf dem System erfassten sicherheitsrelevanten Protokollierungsdaten kontinuierlich an das zentrale Backend der EDR-Lösung aus, auf dem die weitere Analyse, Angriffserkennung und Alarmbehandlung erfolgt.

Beispiel 3: Netzwerk-Komponenten stellen Ereignisse an ein SIEM bereit, die es ermöglichen, verdächtiges Kommunikationsverhalten wie Netzwerk-Scans, unzulässige Datenflüsse zwischen Netz-Segmenten oder aktive Verbindungen bzw. Verbindungsversuche zu verdächtigen Zielen im Internet zu entdecken. Im speziellen liefern Gateway Router aus dem CNDTAG "Netflow" Logs, während Forward Proxies Zugriffe auf Domänen und/oder IP-Adressen melden und Firewalls blockierte Verbindungsversuche loggen.

Beispiel 4: Auf Basis von Cloud-nativen Funktionen entwickelte Systeme erzeugen Protokollierungsdaten mittels der Cloud-eigenen Protokollierungs- und Überwachungsinfrastruktur direkt aus diesen Funktionen heraus. In einem solchen System wird die Erzeugung sicherheitsrelevanter Protokollierungsdaten in den entsprechenden Konfigurationsoptionen der eingesetzten Cloud-nativen Funktionen gezielt aktiviert. Die entstehenden Protokollierungsdaten werden kontinuierlich zur weiteren Analyse, Angriffserkennung und Alarmbehandlung an ein SIEM (Security Information and Event Management) übertragen. Einige Cloud-Anbieter leisten eine gewisse Angriffsmustererkennung bereits in der Cloud-eigenen Überwachungsinfrastruktur und erzeugen daraus spezifische Cloud-Alarme. Diese werden zusätzlich an das SIEM übertragen.

Beispiel 5: Ein System basiert auf einer SaaS ("Software as a Service") Applikation eines Cloud-Anbieters. Typischerweise verantwortet im SaaS Modell der Cloud-Anbieter alle operativen Aspekte der Applikation, inkl. des Sicherheitsmanagements. Der Cloud-Kunde selbst hat keinen Einfluss auf die zugrunde liegende technische Infrastruktur oder den Betrieb der Applikation. Anhand von Vertragsvereinbarungen, anerkannten Zertifikaten und dokumentierten Selbstauskünften des Cloud-Anbieters wird formell überprüft, dass der Cloud-Anbieter eine Erkennungs- und Reaktionslösung für Sicherheitsvorfälle implementiert hat. Einige Cloud-Anbieter erzeugen zusätzlich dedizierte Protokollierungsdaten über sicherheitsrelevante Aktivitäten innerhalb der separaten Kunden-Mandanten ihrer SaaS Applikation (beispielsweise zu Änderungen von Konfigurationseinstellungen, Berechtigungsvergaben oder Benutzer-Konten innerhalb der Mandanten) und stellen diese den Cloud-Kunden zu Verfügung. Diese sogenannten Audit-Logs werden kontinuierlich zur weiteren Analyse, Angriffserkennung und Alarmbehandlung an ein SIEM (Security Information and Event Management) übertragen.

SAP-Applikationslogs betreffend:

- Nutzung eines SIEM für nicht DTAG SAP-Systeme.
- Nutzung SETAM (SAP Real Time Threat Detection Monitoring) für DTAG SAP-Systeme. SETAM gilt als SIEM für SAP-Applikationen.

2.6. Autorisierungsmanagement

Req 51	Die Nutzung von schutzbedürftigen Funktionen des Systems sowie der Zugriff auf interne oder vertrauliche Daten darf nicht ohne vorhergehende Authentifizierung und Autorisierung ermöglicht werden.
--------	---

Die Nutzung von schutzbedürftigen Funktionen des Systems sowie der Zugriff auf Daten, die als intern oder vertraulich eingestuft sind, darf erst nach einer eindeutigen Identifizierung und erfolgreichen Authentifizierung des Benutzers anhand eines Benutzerkontos ermöglicht werden. Ein Benutzerkonto besteht aus einem eindeutigen Benutzernamen und zugehörigen Authentisierungsmerkmalen. Zudem muss verifiziert werden, dass der Benutzer innerhalb der ihm im System zugeordneten Benutzerrolle für den Zugriff auf die betroffenen Funktionen und Daten legitimiert ist.

Eine Ausnahme hiervon sind Funktionen und Daten, die uneingeschränkt öffentlich genutzt werden dürfen; beispielsweise der Bereich einer Web-Seite im Internet, in dem nur öffentliche Informationen bereitgestellt werden.

Beispiele für Funktionen, die eine vorhergehende Authentifizierung erfordern:

- Fernzugriff auf Netzdienste (wie SSH, SFTP, Web-Dienste)
- lokaler Zugriff auf die Management-Konsole
- lokale Nutzung von Betriebssystem und Anwendungen

Diese Anforderung gilt ohne Einschränkung auch für jegliche maschinellen Zugriffe auf das System (Hier erfolgt die Umsetzung in der Regel durch den Einsatz von sogenannten "technischen Benutzerkonten", auch "Machine-2-Machine" oder "M2M"-Konten genannt).

Motivation: Die eindeutige Authentifizierung und Autorisierung von Zugriffen auf ein System sind elementar, um Funktionen und Daten vor Missbrauch zu schützen.

ID: 3.22-51/13.0

Req 52	Die Rollen für alle SAP Benutzer müssen über GRC umgesetzt sein.
--------	--

Die Rollen für alle SAP Benutzer müssen über GRC umgesetzt sein.

Motivation: Gesetzliche Vorgaben und Richtlinien können zentrale mit geringen Aufwand umgesetzt werden.

ID: 3.22-52/13.0

Req 53	Benutzerkonten müssen die eindeutige Identifizierung des Benutzers sicherstellen.
--------	---

Benutzer müssen eindeutig von einem System identifiziert werden können.

Dies wird typischerweise dadurch erreicht, dass für jeden Benutzer ein individuelles Benutzerkonto genutzt wird.

Sogenannte Gruppenkonten, die sich dadurch auszeichnen, dass sie von mehreren Personen gemeinsam genutzt werden, dürfen nicht verwendet werden. Dies gilt uneingeschränkt auch für privilegierte Benutzerkonten. Die meisten Systeme verfügen nach der Grundinstallation zunächst nur über ein einzelnes Benutzerkonto mit administrativen Berechtigungen. Soll das System von mehreren Personen administriert werden, muss jede dieser Personen ein persönliches, individuelles Benutzerkonto verwenden, dem entsprechende administrative Berechtigungen bzw. Rollen zugewiesen sind.

Eine Besonderheit stellen sogenannte technische Benutzerkonten dar. Diese werden für Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt und können damit nicht einer

konkreten Person zugewiesen werden. Solche Benutzerkonten müssen individuell pro System oder pro Anwendung vergeben werden. Hierbei ist allerdings sicherzustellen, dass eine missbräuchliche Nutzung solcher Benutzerkonten nicht möglich ist.

Möglichkeiten, um eine missbräuchliche Nutzung solcher Benutzerkonten durch Personen zu verhindern, sind:

- Konfiguration eines den Sicherheitsanforderungen entsprechenden Passwortes, dass nur einem kleinstmöglichen Kreis von Administratoren bekannt ist.
- Konfiguration des entsprechenden Benutzerkontos, so dass dieses nur lokal verwendet werden kann und eine interaktive Anmeldung nicht möglich ist.
- Verwendung eines Verfahrens mit öffentlichem und privatem Schlüssel oder Zertifikaten für die Authentifizierung des entsprechenden Benutzerkontos.
- Einschränkung des Zugriffs über ein Netz auf legitimierte Systeme.

Weitere Lösungen sind im Einzelfall auf ihre Verwendbarkeit zu prüfen.

Motivation: Die eindeutige Identifizierung eines Benutzers ist Voraussetzung, um einem Benutzer Berechtigungen zuweisen zu können, die für die Durchführung seiner Aufgaben auf dem System notwendig sind. Nur so kann eine adäquate Zugriffskontrolle auf Daten und Dienste eines Systems erreicht und der missbräuchliche Zugriff verhindert werden. Des Weiteren wird hierdurch ermöglicht, dass Tätigkeiten und Aktionen auf einem System protokolliert und einem Benutzer zugeordnet werden können.

ID: 3.22-53/13.0

Req 54	Benutzerkonten müssen mit mindestens zwei Authentisierungsmerkmalen aus unterschiedlichen Faktoren geschützt werden.
--------	--

Sämtliche Benutzerkonten des Systems müssen vor einer unberechtigten Nutzung geschützt werden. Dies ist durch Verwendung von Authentisierungsmerkmalen umzusetzen, die eine eindeutige Identifizierung des legitimen Benutzers ermöglichen.

Authentisierungsmerkmale werden in übergeordneten Gruppen, sogenannten "Faktoren", differenziert:

- Faktor WISSEN: "etwas, das nur der legitime Benutzer weiss"
- Faktor BESITZ: "etwas, das nur der legitime Benutzer besitzt"
- Faktor INHÄRENZ: "etwas, das nur der legitime Benutzer ist"

Es müssen mindestens zwei Authentisierungsmerkmale miteinander kombiniert werden.

Diese müssen aus unterschiedlichen Faktoren stammen ("Multi-Faktor-Authentisierung"; kurz: "MFA").

Eine Kombination von Authentisierungsmerkmalen desselben Faktors (Beispiel: zwei unterschiedliche PINs) ist nicht zulässig.

Beispiele für zulässige Authentisierungsmerkmale

- aus dem Faktor WISSEN: PINs, Passphrases (Passwörter*)
- aus dem Faktor BESITZ: Smartcards, FIDO2 Level 2 Security Keys, Device-bound Passkeys, in einem TPM/HSM gespeicherte kryptografische Schlüssel, OTPs (One-Time Passwords) generiert in einer zum Einsatz in der Deutschen Telekom freigegebenen Authenticator App
- aus dem Faktor INHÄRENZ: biometrische Merkmale wie Fingerabdruck, Iris- bzw. Retina-Scans, Gesichtserkennung

*) Wichtiger Hinweis zum Einsatz von Passwörtern

Passwörter müssen als Authentisierungsmerkmal strikt vermieden werden.

Sie sind nur dann zulässig, falls sicherere Alternativen nicht umgesetzt werden können.

Grundsätzlich nicht zulässig sind Authentisierungsmerkmale, die leicht gefälscht oder manipuliert werden können.
Dazu gehören:

- Rufnummern und Anrufer-IDs
- IP-Adressen
- geografische Standorte
- VPN-Zugehörigkeiten und Netzwerkinformationen
- Status von Clients, etwa von Arbeitsplatzsystemen oder mobilen Endgeräten

Hinweis zu technischen Benutzerkonten

Für den maschinellen Zugriff eingesetzte technische Benutzerkonten (auch unter dem Begriff "Machine-to-Machine" bzw. "M2M"-Konten bekannt) sind von der Verpflichtung zur Multi-Faktor-Authentisierung ausgenommen, da eine solche dort i.d.R. nicht umsetzbar ist.

Motivation: Benutzerkonten, die nicht über angemessene Authentisierungsmerkmale geschützt sind, können von einem Angreifer missbraucht werden, um unberechtigt auf ein System und die darauf gespeicherten Daten und Anwendungen zuzugreifen. Aufgrund vielfältiger gängiger Angriffsmethoden ist der Schutz von Benutzerkonten durch ein einzelnes Authentisierungsmerkmal mittlerweile als schwach zu bewerten, insbesondere, falls hier das klassische Passwort zum Einsatz kommt. Eine Kombination von Authentisierungsmerkmalen aus unterschiedlichen Faktoren erhöht die Sicherheit für Benutzerkonten signifikant.

Umsetzungsbeispiel: Beispiele für MFA

MFA Kombination	Beschreibung	Anmerkung zur Anwendung
Telekom "myCard" plus PIN-Eingabe	1. Faktor: in Hardware implementiertes Zertifikat ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)
FIDO2 Level 2 Security Key plus Fingerabdruck	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: Biometrisches Merkmal ("Inhärenz")	Die Biometrie-Funktion muss vorab konfiguriert werden, da der Biometrie-Sensor sonst nur als "Taste" funktioniert und somit der 2. Faktor ("Inhärenz") nicht gegeben wäre.
Device-bound Passkey in WH4B* auf Nucleus PC plus PIN	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)
Device-bound Passkey in Secure Enclave auf iOS Device plus FaceID	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: Biometrisches Merkmal ("Inhärenz")	(keine)
Passwort plus freigegebene OTP App (z. B. MS Authenticator)	1. Faktor: Passwort ("Wissen") 2. Faktor: spezifisch generiertes One Time Password ("Besitz")	Passwörter sind nur dann zulässig, falls sicherere Alternativen nicht umgesetzt werden können. Die Synchronisation der Authenticator App über einen Clouddienst sowie das Teilen (z. B. über einen Passwortmanager) des OTP ist nicht zulässig.

*) WH4B: "Windows Hello for Business"

Req 55	Privilegierte Benutzerkonten müssen mit mindestens zwei Authentisierungsmerkmalen geschützt werden, die aus unterschiedlichen Faktoren stammen und erhöhte Resilienz gegen Phishing-Angriffe bieten.
--------	--

Ein privilegiertes Benutzerkonto ist ein Benutzerkonto, welches über erweiterte Berechtigungen innerhalb eines Systems verfügt. Erweiterte Berechtigungen ermöglichen einen Zugriff auf Konfigurationseinstellungen, Funktionen oder Datenbestände, der regulären Benutzern des Systems nicht zu Verfügung steht. In direkter Abhängigkeit zu den besonderen Aufgaben, die mittels eines privilegierten Benutzerkontos innerhalb eines Systems wahrgenommen werden, können die zugewiesenen erweiterten Berechtigungen spezifisch limitiert sein oder einen vollständig uneingeschränkten Systemzugriff umfassen.

Beispiele für privilegierte Benutzerkonten:

- Konten für Administrations-, Wartungs- oder Entstörungsaufgaben
- Konten für Benutzer-Verwaltungsaufgaben (z. B. Anlegen/Löschen von Benutzern; Zuweisen von Berechtigungen; Passwort-Rücksetzungen)
- Konten, die zur Freigabe, Auslösung oder Unterbindung von geschäftskritischen Vorgängen berechtigt sind
- Konten mit Zugriff auf Daten, die im Interesse des DTAG-Konzerns, dessen Kunden oder der Öffentlichkeit als SCD (Sensitive Customer Data) eingestuft sind
- Konten, die in hohem Umfang auf lt. EU-DSGVO als „personenbezogen“ definierte Daten zugreifen können (z. B. Massenabruf von größeren Teilen oder des vollständigen Datenbestands)

Zum Schutz privilegierter Konten müssen mindestens zwei Authentisierungsmerkmale miteinander kombiniert werden.

Diese müssen aus unterschiedlichen Faktoren ("WISSEN", "BESITZ" oder "INHÄRENZ") stammen.

Eine Kombination von Authentisierungsmerkmalen desselben Faktors ist nicht zulässig.

Mindestens eines der Authentisierungsmerkmale darf nicht anfällig gegenüber Phishing-Angriffen sein.

Bei einem Phishing-Angriff wird der legitime Nutzer trickreich manipuliert, um seine Anmeldedaten dem Angreifer offenzulegen, beispielsweise mittels gefälschter Anmelde-Seiten. Ein Phishing-Angriff kann auch gegen eine Kombination aus mehreren Authentisierungsmerkmalen erfolgen.

Phishing-resiliente Authentisierungsmerkmale verhindern Angriffe anhand ihrer technischen Umsetzung. Diese Authentisierungsmerkmale nutzen kryptographische Verfahren und stellen eine starke Bindung zwischen dem Nutzer, dem Gerät und dem Dienst her. Sie erfordern während der Authentisierung zusätzlich eine manuelle Nutzerinteraktion, die nicht automatisiert werden kann; beispielsweise, einen Fingerabdruck-Scan an einem Hardware Token abzugeben.

Als nicht anfällig gegenüber Phishing-Angriffen gelten

- FIDO2 Level 2 Security Keys
- in Hardware implementierte Zertifikate

Wichtiger Hinweis zum Einsatz von Passwörtern

Passwörter müssen als Authentisierungsmerkmal strikt vermieden werden.

Sie sind nur dann zulässig, falls sicherere Alternativen nicht umgesetzt werden können.

Hinweis zu technischen Benutzerkonten

Für den maschinellen Zugriff eingesetzte technische Benutzerkonten (auch unter dem Begriff "Machine-to-Machine" bzw. "M2M"-Konten bekannt) sind von der Verpflichtung zur Multi-Faktor-Authentisierung ausgenommen, da eine solche dort i.d.R. nicht umsetzbar ist.

Motivation: Privilegierte Benutzerkonten stellen ein erhöhtes Risiko für die Sicherheit eines Systems dar. Kompromittiert ein Angreifer erfolgreich ein solches Benutzerkonto, erhält er weitreichende Berechtigungen, mit denen er das Sy-

stem oder Systemanteile unter seine Kontrolle bringen, die Systemfunktionen stören, verarbeitete Daten einsehen/manipulieren oder geschäftskritische Vorgänge beeinflussen kann. Die Kombination von mehreren Authentisierungsmerkmalen unterschiedlicher Art minimiert das Risiko der Kompromittierung eines Benutzerkontos bereits signifikant, jedoch muss für privilegierte Benutzerkonten in besonderem Maße auf Resilienz gegen Phishing-Angriffe geachtet werden. Die zu beobachtenden Phishing-Angriffe werden zunehmend trickreicher und für Nutzer schwerer zu erkennen, so dass eine Multifaktor-Authentifizierung ohne erhöhte Resilienz gegen Phishing nicht mehr als ausreichend für privilegierte Benutzerkonten zu bewerten ist.

Umsetzungsbeispiel: Beispiele für MFA mit erhöhter Resilienz gegen Phishing-Angriffe

MFA Kombination	Beschreibung	Anmerkung zur Anwendung
Telekom "myCard" plus PIN-Eingabe	1. Faktor: in Hardware implementiertes Zertifikat ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)
FIDO2 Level 2 Security Key plus Fingerabdruck	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: Biometrisches Merkmal ("Inhärenz")	Die Biometrie-Funktion muss vorab konfiguriert werden, da der Biometrie-Sensor sonst nur als "Taste" funktioniert und somit der 2. Faktor ("Inhärenz") nicht gegeben wäre.
FIDO2 Level 2 Security Key plus PIN-Eingabe	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)

ID: 3.22-55/13.0

Req 56 Vordefinierte, nicht benötigte Benutzerkonten müssen gelöscht oder zumindest deaktiviert werden.

Auf vielen Systemen existieren nach der initialen Installation vordefinierte, aber nicht genutzte Benutzerkonten (z. B. "Gast").

Diese vordefinierten Benutzerkonten müssen unmittelbar nach der initialen Installation gelöscht oder zumindest deaktiviert werden; sollten diese Maßnahmen nicht umsetzbar sein, muss das entsprechende Benutzerkonto für einen Fernzugriff gesperrt werden. In jedem Fall müssen auch deaktivierte oder gesperrte Benutzerkonten mit einem Authentisierungsmerkmal (z. B. einem Passwort oder einem SSH Key) versehen werden, so dass im Fall einer Fehlkonfiguration die unberechtigte Nutzung eines solchen Benutzerkontos verhindert wird.

Ausgenommen von der Anforderung, vordefinierte Benutzerkonten zu löschen oder zu deaktivieren, sind Benutzerkonten, die ausschließlich der internen Nutzung auf dem entsprechenden System dienen und die für die Funktionalität einer oder mehrerer Anwendungen des Systems notwendig sind. Auch für ein solches Benutzerkonto muss sichergestellt werden, dass ein Fernzugriff oder eine lokale Anmeldung nicht möglich ist und dass ein Benutzer des Systems ein solches Benutzerkonto nicht missbräuchlich nutzen kann.

Motivation: In einem Produkt standardmäßig vordefinierte Benutzerkonten sind typischerweise allgemein bekannt und können von einem Angreifer sehr gezielt für Brute-Force- und Wörterbuchangriffe ausgewählt werden. Werden diese Benutzerkonten in einem konkreten System gar nicht benötigt, stellt deren Existenz dann eine unnötige Angriffsfläche dar. Ein besonderes Risiko stellen vordefinierte Benutzerkonten dar, die ohne Passwort oder mit einem weithin bekannten Standardpasswort vorkonfiguriert sind. Solche Benutzerkonten können unmittelbar von einem Angreifer missbraucht werden, falls ihre Absicherung aufgrund der nicht weiter vorgesehenen Nutzung im konkreten System versäumt wurde.

ID: 3.22-56/13.0

Req 57 Vordefinierte Authentisierungsmerkmale müssen geändert werden.

Nach der Übernahme oder Erstinstallation eines Systems existieren üblicherweise vordefinierte Authentisierungsmerkmale (z. B. Passwörter, SSH Keys, SSL/TLS Zertifikate) im System, wie sie durch Hersteller, Entwickler, Lieferanten oder automatisierte Installationsroutinen vergeben wurden.

Solche vordefinierten Authentisierungsmerkmale müssen unmittelbar nach der Übernahme bzw. Installation des Systems auf neue, individuelle Werte geändert werden.

Motivation: Von Dritten vordefinierte Werte in Authentisierungsmerkmalen können nicht als vertrauenswürdig angesehen werden, da sie kein kontrolliertes Geheimnis darstellen. Betroffene Authentisierungsmerkmale können von Unbefugten dafür missbraucht werden, um auf Systeme zuzugreifen und diese zu kompromittieren. Dieses Risiko ist signifikant erhöht, falls für Authentisierungsmerkmale allgemein bekannte Standard-Werte verwendet werden (z. B. ein Standardpasswort für das Administrator-Benutzerkonto in einem bestimmten Software-Produkt).

ID: 3.22-57/13.0

Req 58 Die Berechtigungen von Benutzern und Anwendungen müssen auf den zur Erfüllung ihrer Aufgaben benötigten Umfang beschränkt werden.

Die Berechtigungen auf einem System müssen so weit eingeschränkt werden, dass ein Benutzer nur auf Daten zugreifen und Funktionen nutzen kann, die er im Rahmen seiner Arbeit benötigt. Entsprechende Berechtigungen sind auch für den Zugriff auf Dateien zu vergeben, die Bestandteil des Betriebssystems oder von Anwendungen sind oder von diesen erzeugt werden (z. B. Konfigurations- und Protokollierungsdateien).

Neben dem Zugriff auf Daten muss auch die Ausführung von Anwendungen und deren Bestandteilen mit möglichst niedrigen Berechtigungen erfolgen. Anwendungen sollten nicht mit Administrator- oder Systemberechtigungen ausgeführt werden.

Motivation: Werden einem Benutzer zu weitreichende Berechtigungen auf einem System gewährt, kann er in einem Umfang auf Daten und Anwendungen zugreifen, der zur Erfüllung der zugewiesenen Aufgaben nicht erforderlich ist. Dies erzeugt ein unnötig erhöhtes Risiko im Falle eines Missbrauchs, insbesondere auch, falls der Benutzer oder dessen Benutzerkonto durch einen Angreifer kompromittiert werden.

Anwendungen, die über zu weitreichende Berechtigungen verfügen, können von einem Angreifer missbraucht werden, um unbefugte Zugänge zu erhalten oder zu erweitern und damit einen Zugriff auf schutzbedürftige Daten und Systembereiche zu erlangen.

ID: 3.22-58/13.0

Req 59 Sitzungen müssen gegen eine unautorisierte Übernahme ("Session Hijacking") geschützt werden.

In Schnittstellen, die Funktionen für Sitzungen zum System bereitstellen, müssen technische Maßnahmen implementiert werden, die verhindern, dass die Sitzung eines legitimen Benutzers von einem nicht autorisierten Dritten übernommen und weitergeführt werden kann.

Ein derartiger Schutz kann beispielsweise durch die Umsetzung einer für das spezifische System sinnvollen Kombination aus den folgenden Optionen erreicht werden:

- Auf der Transportschicht: Verwendung des TCP-Protokolls (mit dessen Sequenz-Nummern) und entsprechenden Filterlisten
- Auf der Sitzungsschicht: Verwendung des TLS-Protokolls
- Auf der Applikationsschicht: Aushandlung eines zufälligen geheimen Sitzungsschlüssels zwischen Sender und Empfänger zwecks Autorisierung allen Sitzungsverkehrs (z. B. Session-ID, Session Cookie, Session Token)
- Nutzung kryptographischer Verfahren zum Schutz von Sitzungsschlüsseln vor Ausspähung oder Veränderung

Motivation: Ungeschützte Sitzungen können potenziell von einem Angreifer gekapert und weitergeführt werden, um darüber im Kontext des betroffenen Benutzers unberechtigte Zugriffe auf das System auszuüben.

ID: 3.22-59/13.0

Req 60	Das System muss es Benutzern ermöglichen, sich von ihrer laufenden Sitzung abzumelden.
--------	--

Das System muss über eine Funktion verfügen, die es dem angemeldeten Benutzer ermöglicht, sich jederzeit abzumelden. Die Wiederaufnahme einer abgemeldeten Sitzung darf nicht ohne erneute und erfolgreiche Authentifizierung des Benutzers möglich sein.

Motivation: Ein Benutzer muss die vollständige Kontrolle über von ihm aufgebaute Sitzungen behalten, um jederzeit situationsgerecht seinen Zugriff auf ein System zu beenden und damit über diesen Zugriff exponierte Daten und Funktionen schützen zu können. Zudem muss der Benutzer davon ausgehen können, dass durch ihn gezielt beendete Sitzungen nicht im Anschluss durch unberechtigte Dritte wieder aufgenommen und weitergeführt werden können.

ID: 3.22-60/13.0

Req 61	Sitzungen müssen nach einer dem Verwendungszweck angepassten Zeit der Inaktivität automatisch beendet werden.
--------	---

Es ist notwendig, dass Sitzungen auf einem System nach einer vorgegebenen Zeit der Inaktivität automatisch beendet werden.

Aus diesem Grund muss ein Time-Out für Sitzungen gesetzt werden. Der hier zu wählende Zeitraum richtet sich nach der Verwendung des Systems und ggf. nach der physischen Umgebung. So ist beispielsweise der Time-Out für eine Anwendung, die in einer ungesicherten Umgebung genutzt wird, kürzer (wenige Minuten) zu wählen als der Time-Out für eine Anwendung, die von Betriebsmitarbeitern für System-Monitoring-Aufgaben in einem zugangsgeschützten Bereich genutzt wird (60 Minuten und länger).

Motivation: Für eine offene, aber ungenutzte Sitzung besteht ein Risiko, dass ein nicht legitimer Benutzer diese unbemerkt übernimmt und weiterführt, um im Namen des betroffenen Benutzers einen nicht autorisierten Zugriff auf das System und enthaltene Daten auszuüben.

ID: 3.22-61/13.0

Req 62	Falls Passwörter als Authentisierungsmerkmal genutzt werden, müssen diese mindestens 12 Zeichen lang sein und drei der folgenden Zeichenkategorien beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.
--------	--

Ein System darf nur Passwörter akzeptieren, welche die folgende Komplexität erfüllen:

- Mindestlänge 12 Zeichen
- Bestehend aus mindestens drei der folgenden vier Zeichenkategorien:
 - Kleinbuchstaben
 - Großbuchstaben
 - Ziffern
 - Sonderzeichen

Die maximal mögliche Länge für Passwörter sollte 25 Zeichen oder mehr betragen. Dies unterstützt Endbenutzer bei Bedarf mit mehr Freiraum, um individuelle merkbare Passwörter bilden zu können, und hilft so, unerwünschte Verhaltensweisen im Umgang mit Passwörtern zu vermeiden, welche durch stark eingeschränkte maximale Passwortlängen provoziert werden können.

Das System muss bei der Vergabe eines Passwortes sicherstellen, dass das Passwort diesen Bildungsregeln entspricht; vorzugsweise mittels technischer Maßnahmen, falls solche aber nicht umsetzbar sein sollten, ersatzweise mittels organisatorischer Maßnahmen.

Bei Verwendung eines zentralen Systems für die Authentisierung der Benutzer [vgl. Wurzel-Anforderungsdokument "[i]3.69 IAM (Identity Access Management) - Framework"] darf diese Aufgabe an das zentrale System weitergereicht bzw. dorthin ausgelagert werden.

Zulässige Abweichung in der Passwort-Mindestlänge

Unter sicherheitsbezogenen Kriterien lassen sich für ein System potenziell Bedingungen identifizieren, die eine Reduzierung der Passwort-Mindestlänge ermöglichen:

- Allgemein zulässig ist eine Reduzierung der Passwort-Mindestlänge für Systeme, die innerhalb des Authentisierungsprozess zusätzlich zum Passwort weitere unabhängige Authentisierungsmerkmale verwenden (Umsetzung von 2-Faktor oder Multi-Faktor Authentisierung).
- Eine Reduzierung der Passwort-Mindestlänge muss grundsätzlich durch einen fachlichen Sicherheitsberater (z. B. ein PSM der Telekom Security) individuell bewertet und als zulässig bestätigt werden. In der Bewertung müssen umgebende technische, organisatorische und gesetzliche Rahmenparameter berücksichtigt sowie der systemspezifische Schutzbedarf und die potenzielle Schadenshöhe bei Sicherheitsvorfällen einbezogen werden.
- Der absolute Mindestwert von 8 Zeichen Länge darf für Passwörter grundsätzlich nicht unterschritten werden.

Motivation: Passwörter mit der genannten Komplexität bieten eine zeitgemäße Robustheit gegen Angriffe bei gleichzeitig akzeptabler Benutzerfreundlichkeit. In der Praxis haben sich Passwörter mit dieser Komplexität bewährt. Triviale und zu kurze Passwörter sind anfällig für Brute-Force- und Wörterbuch-Angriffe und von einem Angreifer leicht zu ermitteln. Ein ermitteltes Passwort kann dann von einem Angreifer für den unautorisierten Zugriff auf das System und dessen Daten genutzt werden

Umsetzungsbeispiel: Beispiel für ABAP:

```
login/min_password_lng = 12  
login/min_password_lowercase = 1  
login/min_password_uppercase = 1  
login/min_password_digits = 1  
login/min_password_specials = 1  
login/password_compliance_to_current_policy = 1
```

ID: 3.22-62/13.0

Req 63	Falls Passwörter als Authentisierungsmerkmal für technische Benutzerkonten genutzt werden, müssen diese mindestens 30 Zeichen lang sein und drei der folgenden Zeichenkategorien beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.
--------	--

Technische Benutzerkonten zeichnen sich dadurch aus, dass sie nicht von Personen verwendet werden. Stattdessen werden sie für die Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt.

Ein System darf für technische Benutzerkonten nur Passwörter verwenden, welche die folgende Komplexität erfüllen:

- Mindestlänge 30 Zeichen
- Bestehend aus mindestens drei der folgenden vier Zeichenkategorien:
 - Kleinbuchstaben
 - Großbuchstaben
 - Ziffern
 - Sonderzeichen

Motivation: Technische Benutzerkonten sind aufgrund ihrer Verwendung in Machine-to-Machine (M2M) Kommunikationsszenarien oft mit Berechtigungen ausgestattet, die für einen Angreifer von hohem Interesse sein können, um Infrastrukturen zu kompromittieren. Ohne Mechanismen einer umfassenden Kompromittierungserkennung kann dann das Risiko, dass ein Passwort von einem Angreifer ermittelt oder gebrochen wird, über die Zeit deutlich ansteigen. Ei-

ne signifikante Erhöhung der Passwort-Länge wirkt diesen Risiken entgegen und lässt sich zudem in M2M Szenarien besonders leicht umsetzen, da die Handhabung eines sehr langen Passworts für eine Maschine (im Gegensatz zu einer Person) keine besondere Herausforderung darstellt.

ID: 3.22-63/13.0

Req 64	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss eine Änderung des eigenen Passwortes jederzeit durch den Benutzer möglich sein.
--------	---

Das System muss eine Funktion bieten, die es einem Benutzer ermöglicht, sein Passwort jederzeit zu ändern.

Bei der Verwendung eines zentralen Systems darf die Funktion auf dieses System weitergereicht oder ausgelagert werden.

Motivation: Nur durch die Möglichkeit, dass ein Benutzer sein Authentisierungsmerkmal jederzeit selbsttätig ändern kann, ist ein zeitnahe Änderung möglich, wenn der Benutzer den Verdacht hat, dass dieses in den Besitz Dritter gelangt sein könnte.

ID: 3.22-64/13.0

Req 65	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss nach spätestens 12 Monaten ein Passwort-Wechsel erfolgen.
--------	---

Die maximal zulässige Nutzungsdauer für Passwörter beträgt 12 Monate.
Erreicht ein Passwort die maximal zulässige Nutzungsdauer, muss es geändert werden.

Das System muss hierzu den Benutzer automatisch bei dessen nächster Systemanmeldung auf die abgelaufene Nutzungsdauer hinweisen und ihn unmittelbar durch einen Passwort-Änderungsdialog führen. Der Zugriff auf das System darf ohne eine erfolgreich abgeschlossene Passwort-Änderung nicht weiter zugelassen werden.

Für technische Benutzerkonten (M2M bzw. Machine-2-Machine), die für die Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt werden, müssen ebenfalls automatisierte Lösungen zur Einhaltung der zulässigen Nutzungsdauer für Passwörter implementiert werden.

Falls eine automatisierte Abbildung des Prozesses zur Passwort-Änderung nicht umsetzbar ist, muss ersatzweise eine wirksame organisatorische Maßnahme angewendet werden, die eine verbindliche manuelle Passwort-Änderung bei Ablauf der zulässigen Nutzungsdauer sicherstellt.

Motivation: Passwörter sind im Gegensatz zu moderneren Authentisierungsmerkmalen leichter angreifbar. Ohne spezifische Maßnahmen einer verlässlichen technisch automatisierten Kompromittierungserkennung kann das Risiko, dass ein Passwort von einem Angreifer ermittelt oder gebrochen wird, über die Zeit deutlich steigen.

Umsetzungsbeispiel: Beispiel ABAP setzen der Parameter:

- login/password_expiration_time = 365
- rfc/reject_expired_passwd = 1

ID: 3.22-65/13.0

Req 66	Falls Passwörter als Authentisierungsmerkmal verwendet werden, muss die Wiederverwendung bereits eingesetzter Passwörter unterbunden werden.
--------	--

Für jedes Benutzerkonto muss eine Historie der bisher genutzten Passwörter aufgezeichnet werden. Wird eine Passwort-Änderung für ein Benutzerkonto initiiert, ist das neue Passwort gegen diese Passwort-Historie abzugleichen. Bei erkannter Wiederverwendung eines Passworts muss die Passwort-Änderung abgelehnt werden. Die Umsetzung dieses Validierungsvorgangs muss im System auf Basis von technischen Maßnahmen erfolgen. Bei Verwendung eines zentralen IAM Systems für die Authentisierung der Benutzer darf die Umsetzung an das zentrale IAM System weitergereicht bzw. dorthin ausgelagert werden [vgl. Wurzel-Anforderungsdokument "3.69 IAM (Identity Access Manage-

ment) - Framework"].

Generell soll mittels der Passwort-Historie sichergestellt werden, dass ein schon einmal verwendetes Passwort niemals erneut einsetzbar ist.

Aufgrund technischer Limitierungen kann eine Passwort-Historie jedoch in vielen IT/NT-Produkten nicht uneingeschränkt aufgezeichnet werden. In diesem Fall ist die folgende Grundregelung zu berücksichtigen:

- ein bereits genutztes Passwort darf für den Zeitraum von mindestens 60 Tagen nicht erneut einsetzbar sein (gemessen ab dem Zeitpunkt, an dem das betroffene Passwort durch ein anderes ersetzt wurde)
- in Systemen, in denen der Zeitraum von mindestens 60 Tagen nicht abgebildet werden kann, muss der längstmögliche Zeitraum konfiguriert werden. Zusätzlich muss durch einen Project Security Manager (PSM) bestätigt werden, dass der konfigurierte Zeitraum im Gesamtkontext des Systems noch ausreichend im Sinne der Sicherheitsanforderung ist.

Anmerkung:

Einige IT/NT-Produkte bieten keinen technischen Konfigurationsparameter an, mit dem die Passwort-Historie direkt an einen Zeitraum gebunden werden kann, sondern ermöglichen nur, die Anzahl der aufzuzeichnenden Passwörter zu definieren. In solchen Fällen darf der Zeitraum alternativ durch eine Verknüpfung der folgend beschriebenen, i. d. R. breit verfügbaren Konfigurationsparameter sichergestellt werden. Innerhalb der resultierenden Policy kann ein Benutzer nur einmal pro Tag sein Passwort ändern und aufgrund der Anzahl der aufgezeichneten Passwörter effektiv nach frühestens 60 Tagen ein altes Passwort erneut einsetzen.

- Minimum Passwort-Alter: 1 Tag
- Passwort-Historie: Aufzeichnung der letzten 60 benutzten Passwörter

In dieser Umsetzungsvariante ist zu beachten, dass das Minimum Passwort-Alter nicht mehr als einen Tag betragen sollte, um den Benutzer nicht unangemessen hinsichtlich der grundsätzlichen Notwendigkeit einzuschränken, das Passwort jederzeit selbstständig ändern zu können.

Motivation: Benutzer favorisieren in der Regel Passwörter, die sie sich gut merken können und verwenden diese in Folge oft wiederkehrend über lange Zeiträume, falls das System dies zulässt. Aus Sicht des Benutzers ist das Verhalten nachvollziehbar, führt aber effektiv zu einer erheblichen Reduzierung der Schutzwirkung dieses Authentisierungsparameters. Mit etwa entsprechenden Kenntnissen über den Benutzer oder erbeuteten Informationen aus zurückliegenden System-Kompromittierungen kann ein Angreifer so Zugriff auf vermeintlich geschützte Benutzerkonten erlangen. Insbesondere auch in Situationen, in denen im Rahmen einer akuten Risikobehandlung zentral neue Initialpasswörter vergeben werden, die Benutzer diese jedoch unmittelbar aus Bequemlichkeit auf ein vorheriges Passwort zurückändern, bestünde hohe Gefahr, dass ein Angreifer seinen illegalen Zugriff erneut aufnehmen kann. Es ist daher wichtig zu vermeiden, dass Benutzer alte Passwörter erneut einsetzen.

Umsetzungsbeispiel:

Beispielsweise gesetzte Werte für Windows password-history = 24 , Linux Passwort-History = 60, SAP-Netweaver ABAP = login/password_history_size = 5

ID: 3.22-66/13.0

Req 67	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss deren Speicherung unter Einsatz eines für diesen Zweck als ausreichend sicher anerkannten "Password Hashing"-Verfahrens erfolgen, um Offline-Angriffe wie Wörterbuch- und Brute-Force-Attacken stark zu erschweren.
--------	---

Diese Anforderung bezieht sich auf die Speicherung von Passwörtern in sämtlichen Arten von Benutzerdatenbanken, die im vorliegenden System eingesetzt werden, um darüber eingehende Zugriffe (lokal oder aus der Ferne) von Benutzern oder anderen Systemen zu authentisieren.

Gelingt es einem Angreifer, die Kopie einer Benutzerdatenbank des Systems in seinen Besitz zu bringen, kann er diese in einer unabhängigen Umgebung einem automatisierten Wörterbuch- oder Brute-Force-Angriff aussetzen, um enthaltene Passwörter zu ermitteln. Durch Einsatz spezialisierter Werkzeuge in Kombination mit hoher Rechenleistung

können bei unzureichenden Schutzmaßnahmen hierbei hohe Erkennungsraten in verhältnismäßig kurzer Zeit erreicht werden. Aufgrund der Unabhängigkeit vom Quellsystem bleibt ein solcher Offline-Angriff auf eine Benutzerdatenbank zudem unbemerkt.

Folgende Maßnahme ist umzusetzen, da diese den größtmöglichen Schutz bei Offline-Angriffen gewährleistet:

- Passwörter müssen unter Verwendung einer für diesen Zweck anerkannt wirksamen, zum aktuellen Stand der Technik als sicher eingestuft, kryptografischen Einwegfunktion gespeichert werden (bekannt als "Password-Hashing" Verfahren)

Hinweis: Zulässige Passwort-Hashing-Verfahren sind im Sicherheitsanforderungsdokument "3.50 / Kryptographische Algorithmen und Sicherheitsprotokolle" beschrieben.

In expliziter Deutlichkeit ist als NICHT ZULÄSSIG hervorzuheben:

- jegliche Speicherung von Passwörtern im Klartext
- jegliche Speicherung von Passwörtern in direkt rückrechenbaren Formaten
- jegliche Speicherung von Passwörtern mittels umkehrbarer Verschlüsselung

Hinweis:

"direkt rückrechenbare Formate" bezieht sich in diesem Kontext auf Formate, die ein Passwort lediglich kodieren, also ohne Einbeziehung eines geheimen Schlüssels transformieren. Da ein Passwort bei Anwendung eines solchen Formats nicht mehr in seinem originalen Klartext lesbar bleibt, werden diese Formate leicht als Schutzmaßnahme missverstanden. Effektiv bieten sie jedoch keinen Schutz, da die dahinterliegende Kodierung fest definiert ist und von einem Angreifer sehr leicht für die Rückrechnung der Inhalte in den ursprünglichen Klartext genutzt werden kann.

Beispiele für direkt rückrechenbare Formate sind: "base64", "rot13"

"Umkehrbar" sind alle Verfahren der Verschlüsselung, die unter Verwendung des passenden Schlüssels ermöglichen, einen verschlüsselten Inhalt zurück in den originalen Inhalt zu transformieren. Entsprechend stellt sich bei umkehrbarer Verschlüsselung stets die Herausforderung, den Schlüssel sicher aufzubewahren und vor unberechtigtem Zugriff zu schützen. Die Umkehrbarkeit ist auf vielen Einsatzgebieten der Verschlüsselung, z. B. für den Transport von vertraulichen Nachrichten, eine benötigte Grundeigenschaft, für das Speichern von Passwörtern jedoch kontraproduktiv: ein gespeichertes Passwort muss zwar mittels technischer Methoden vergleichbar bleiben, aber darf nicht mehr in seinen Klartext rückwandelbar sein, um es vor unberechtigter Einsicht bestmöglich zu schützen.

Beispiele für umkehrbare Verschlüsselung sind: "AES", "CHACHA20", "3DES", "RSA"

Motivation: Ohne entsprechenden Schutz kann ein Angreifer im Besitz der Benutzerdatenbank durch bloßes Durchprobieren von Wörterbuchlisten oder automatisch erzeugten Zeichenkombinationen versuchen, massenhaft Passwörter zu ermitteln, um so entsprechende Benutzerkonten missbräuchlich zu nutzen. Passwörter, die im Klartext oder direkt rückrechenbaren Formaten gespeichert werden, sind bei einem Offline-Angriff vollkommen schutzlos.

Umsetzungsbeispiel: Beispiel für ABAP (s. SAP Hinweis 2140269). Setzen der Parameter:

- login/password_hash_algorithm = encoding=RFC2307, algorithm=iSSHA-256, iterations=15000, saltsize=128
- login/password_charset = 2
 - Achtung: Bei login/password_charset = 2 speichert das System Passwörter in einem Format, das Systeme mit älteren Kernels nicht interpretieren können. Stellen Sie daher sicher, dass alle beteiligten Systeme die neue Passwortcodierung unterstützen, bevor Sie den Profilparameter auf den Wert 2 setzen.

ID: 3.22-67/13.0

Req 68	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss ein Schutz gegen Online-Angriffe wie Wörterbuch- und Brute-Force-Attacken vorhanden sein, der das Erraten von Passwörtern stark erschwert.
--------	--

Online Wörterbuch- und Brute-Force Angriffe haben zum Ziel, durch automatisiertes Ausprobieren an einer regulären Zugangsschnittstelle des Systems die Passwörter von Benutzerkonten zu erraten.

Um dies zu verhindern, muss eine der folgenden Maßnahmen oder eine Kombination aus diesen umgesetzt werden:

- mittels technischer Maßnahmen abgebildete Erzwingung einer Wartezeit nach einem fehlgeschlagenen Anmeldeversuch, bevor eine erneute Anmeldung ermöglicht wird. Die Wartezeit sollte sich zudem mit jedem weiteren fehlgeschlagenen Anmeldeversuch automatisch deutlich erhöhen (z. B. durch Verdopplung).
- automatisiertes Sperren des Benutzerkontos nach einer vorgegebenen Anzahl von aufeinanderfolgenden fehlgeschlagenen Anmeldeversuchen (typischerweise 5). Allerdings ist bei dieser Lösung zu bedenken, dass diese einen Entsperrungs-Prozess erforderlich macht und ein Angreifer dies gezielt nutzen kann, um Konten zu sperren und somit vorübergehend unbrauchbar zu machen.
- Verwendung von CAPTCHA ("Completely Automated Public Turing test to tell Computers and Humans Apart"), um automatisierte Anmeldeversuche von Maschinen ("Robots" oder "Bots") bestmöglich zu unterbinden. Ein CAPTCHA stellt dazu eine kleine Aufgabe, i. d. R. basierend auf grafischen oder akustischen Elementen, die von einer Maschine nur schwer zu lösen ist. Es ist entsprechend zu berücksichtigen, dass CAPTCHA problematisch bzgl. der Barrierefreiheit sein können.

Um eine höhere Sicherheit zu erreichen, ist es oftmals sinnvoll, zwei oder mehrere der genannten Maßnahmen miteinander zu kombinieren. Dies ist abhängig vom Einzelfall zu prüfen und entsprechend umzusetzen.

Motivation: Ohne entsprechenden Schutz kann ein Angreifer durch bloßes Durchprobieren von Wörterbuchlisten oder automatisch erzeugten Zeichenkombinationen versuchen ein Passwort zu ermitteln, um so das entsprechende Benutzerkonto missbräuchlich zu nutzen.

Umsetzungsbeispiel: Beispiel ABAP:

login/fails_to_user_lock = 5

login/failed_user_auto_unlock = 0

ID: 3.22-68/13.0

Req 69	Falls Passwörter als Authentisierungsmerkmal genutzt werden, darf deren Darstellung während der Eingabe nicht im Klartext erfolgen.
--------	---

Passwörter dürfen während deren Eingabe nicht in lesbarem Klartext auf Bildschirmen oder sonstigen Ausgabegeräten dargestellt werden. Eine Darstellung während der Eingabe darf keine Rückschlüsse auf die tatsächlich im Passwort verwendeten Zeichen ermöglichen.

Diese Anforderung gilt für jegliche Arten von Eingabemasken und -feldern für Passwörter.

Beispiele hierfür sind Dialoge zur Passwort-Vergabe, zur Passwort-basierenden Anmeldung an Systemen oder zur Änderung von bestehenden Passwörtern.

Ausnahmeregelungen:

- Innerhalb eines Eingabefelds ist eine optionale Klartext-Darstellung eines Passworts zulässig, sofern diese Klartext-Darstellung einem validen Zweck dient, nur temporär anhält, dazu fallbezogen durch den legitimen Benutzer explizit zu aktivieren ist und von diesem umgehend auch wieder deaktiviert werden kann. Ein valider Zweck wäre beispielsweise, dem legitimen Benutzer bei Bedarf eine unkomplizierte visuelle Überprüfung zu ermöglichen, sein Passwort in einem Anmelde-Dialog korrekt eingegeben zu haben, bevor er die Anmeldung letztendlich final vollzieht. Eine solche optionale Klartext-Darstellung eines Passworts muss vollständig in der Kontrolle des legitimen Benutzers verbleiben, damit er über deren Aktivierung/Deaktivierung situationsgerecht entscheiden kann. In der Standard-Einstellung des Systems muss die Klartext-Darstellung deaktiviert sein.
- Das auf vielen mobilen Endgeräten (Smartphones) typische Verhalten, jedes einzelne Zeichen bei der Eingabe eines Passworts sehr kurz im Klartext anzuzeigen, um dem Benutzer die Eingabekontrolle zu erleichtern, ist dort grundlegend zulässig. Allerdings darf dabei niemals das vollständige Passwort im Klartext auf dem Bild-

schirm dargestellt werden.

Motivation: Im Fall einer Klartext-Darstellung besteht das Risiko, dass dritte Personen ein Passwort während der Eingabe zufällig oder gezielt über die Bildschirm-Ausgabe ausspähen können.

Umsetzungsbeispiel: In der Darstellung auf dem Bildschirm wird während der Eingabe eines Passworts jedes einzelne Zeichen einheitlich durch ein "*" ersetzt.

ID: 3.22-69/13.0

Req 70	Falls Passwörter als Authentisierungsmerkmal verwendet werden, muss sich das neue Passwort in mindestens um 1 Zeichen vom alten Passwort unterscheiden.
--------	---

Falls Passwörter als Authentisierungsmerkmal verwendet werden, muss sich das neue Passwort in mindestens um 1 Zeichen vom alten Passwort unterscheiden.

Motivation: Sicherstellen, dass die Passwörter nicht zu identisch sind.

Umsetzungsbeispiel: Falls Passwörter als Authentisierungsmerkmal verwendet werden, muss sich das neue Passwort in mindestens um 1 Zeichen vom alten Passwort unterscheiden.

ID: 3.22-70/13.0

2.7. Allgemeine Parameter für Netweaver (ABAP und Java)

Req 71	Auf einem Netweaver-System mit ABAP-Stack muss der JAVA-Stack deaktiviert sein.
--------	---

Auf einem SAP-System muss die Entscheidung getroffen sein, ob es sich um ein Java oder ein Abap-System handelt (Der dual-Stack war viele Jahre ein von SAP unterstütztes Applikationsmodell).

Motivation: Dual-Stack-Systeme werden nicht mehr von SAP supportet.

Umsetzungsbeispiel: Check Parameter "J2EE Engine = 0

ID: 3.22-71/13.0

Req 72	Der SAP-Software-Manager muss aktiviert sein.
--------	---

Es ist sicherzustellen, dass auf jedem SAP-System der SAP-Software-Update Manager läuft. Damit wird gewährleistet, dass das System über den Solution-Manager gemanagt werden kann.

Motivation: Mit dem SAP-Software-Manager auf dem lokalen SAP-System und einem zentralen Solution-Manager-System wird sichergestellt, dass ein zentrales und globales Management möglich ist.

ID: 3.22-72/13.0

Req 73	Nach Abschluss der Standardinstallation müssen geeignete Verschlüsselungsbibliotheken installiert werden.
--------	---

Nach Abschluss der Standardinstallation müssen geeignete Verschlüsselungsbibliotheken installiert werden. Es können die SNC-Implementationen der Firma SAP oder T-Systems zum Einsatz kommen.

Motivation: In älteren Releases stehen im SAP-Java-Stack (JVM) standardmäßig keine starken Verschlüsselungsmechanismen zur Verfügung. Ohne dass die Verschlüsselungsbibliothek heruntergeladen und installiert wird, können keine starken Verschlüsselungsalgorithmen eingesetzt werden. Dies gibt daher Angreifern die Möglichkeit zum Angriff auf das System, da es zu schwache Mechanismen nutzt. Darüberhinaus braucht das JRE "unlimited encryption

capabilities“, denn im Default sind diese „limited“. Diese werden durch local_policy.jar gesteuert und MÜSSEN mit den „policy files“ der SAP manuell aktualisiert werden.

Umsetzungsbeispiel:

1. JVM – Java Virtual Machine Abfrage welche Bibliothek geladen ist:
Visual administrator (VA): Dispatcher | Libraries core_lib. Die Datei iaik_jce.jar muss in der Liste der geladenen jar-files enthalten sein.
Die Exportversion iaik_jce_export.jar darf nicht gelistet sein.
Ab SAP NetWeaver 7.1 wird hierfür der NetWeaver Administrator verwendet.
2. JFRE – Java Runtime Environment:
Ab Release 7.0 wird die starke Verschlüsselung des JRE mit ausgeliefert. Es ist notwendig die „limited policy files“ der 1.4 - J2SE-Plattform auszutauschen. Siehe Hinweis 739043: <https://service.sap.com/sap/support/notes/739043> Für den Release 6.40 sind die Nacharbeiten für „Enable Strong Encryption“ durchzuführen, bzw. zu überprüfen.
Siehe Hinweis 85833: <https://service.sap.com/sap/support/notes/85833>.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.22-73/13.0

2.8. Absicherung des ABAP-Stacks

2.8.1. ABAP-Stack: Betriebssystemverzeichnis

Req 74	Der Zugriff auf die SAP-Installationslogs muss stark eingeschränkt sein.
--------	--

Für den ABAP- und Java-Stack werden die Installationsdateien (Logs) auf dem Standard SAP Betriebsverzeichnis /usr/sap/<SAPSID>/SYS gespeichert. Dies ist der Defaultwert von der Variablen DIR_INSTALL, welche den Ort für das SAP Installationsverzeichnis angibt.

Motivation: In Installationslogs befinden sich Daten mit einem hohen Schutzbedarf. Nach Installationsabschluss darf der Zugriff nur durch berechtigte User erfolgen.

Umsetzungsbeispiel: Ändern der Zugriffsrechte für die Installationslogs mit folgenden Befehlen:

```
chown root <Dateiname>
chmod 700 <Dateiname>
```

Wenn möglich, können die Installationslogs auch gelöscht werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit

ID: 3.22-74/13.0

Req 75	Das HOME-Verzeichnis des "<sapsid>adm" Benutzers und dessen Inhalt müssen vor nicht autorisierten Zugriff geschützt werden.
--------	---

Das HOME-Verzeichnis des "<sapsid>adm" Benutzers muss vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Motivation: Im HOME-Verzeichnis befinden sich besonders schutzbedürftige Informationen.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigung	OS-User/OS-Gruppe
<home Verzeichnis von <sapsid>adm>	700	<sapsid>adm/sapsys
<home Verzeichnis von <sapsid>adm>/*	700	<sapsid>adm/sapsys

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>  
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys <home Verzeichnis von <sapsid>adm>  
chmod 700 <home Verzeichnis von <sapsid>adm>
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen

ID: 3.22-75/13.0

Req 76	Das Verzeichnis "/sapmnt/<SAPSID>" mit seinen Unterverzeichnissen und Inhalten müssen vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.
--------	---

Das Verzeichnis "/sapmnt/<SAPSID>" mit seinen Unterverzeichnissen muss vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Motivation: Unter dem Verzeichnis "/sapmnt/<SAPSID>" befinden sich einige Verzeichnisse und Dateien die für die Applikation essentiell sind. In der Regel liegen die Dateien auf einem globalen Host und werden mittels NFS, SMB verteilt. Ein Zugriff auf Dateien und Verzeichnisse innerhalb des Verzeichnisses können dazu führen, dass Information und Daten verändert werden können oder die Applikation sich fehlerhaft verhält.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigungen	OS-User/OS-Gruppe	Anmerkungen
--------------------	---------------------------	-------------------	-------------

(1) /sapmnt/<SAPSID>/exe (ohne "Inplace" Datenbank)	(2)700	<sapsid>adm/sapsys	Setzung des s-Bits für die Gruppe. Wert (2)700 für ABAP und JavaSysteme ohne "Inplace" Datenbank (SAP HANA, SAP Web Dispatcher). Nur der User <sid>adm muss auf die Dateien zugreifen können.
(2) /sapmnt/<SAPSID>/exe (mit "Inplace" Datenbank z. B. Oracle)	(2)750	<sapsid>adm/dba	Setzung des s-Bits für die Gruppe. Wert (2)750 für ABAP und Java Systeme mit "Inplace" Datenbank wie z. B. Oracle. Nur der User <sid>adm muss auf die Dateien zugreifen können. Wenn eine Oracle Datenbank genutzt wird, liegen hier die ebenfalls die brtools auf die der User ora<sid> zugreifen können muss.
/sapmnt/<SAPSID>/global	700	<sapsid>adm/sapsys	
/sapmnt/<SAPSID>/profile	755	<sapsid>adm/sapsys	

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys /sapmnt/<SAPSID>
chmod 751 /sapmnt/<SAPSID>
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.22-76/13.0

Req 77	Das Verzeichnis "/usr/sap/<SAPSID>" und dessen Inhalt müssen vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.
--------	--

Unter dem Verzeichnis "/usr/sap/<SAPSID>" befinden sich Verzeichnisse und Dateien, die für die Applikation essentiell sind. Diese Dateien liegen auf dem lokalen Host.

Motivation: Ein Zugriff auf Dateien und Verzeichnisse innerhalb des Verzeichnisses können dazu führen, dass Information und Daten verändert werden können oder die Applikation sich fehlerhaft verhält.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigungen	OS-User/OS-Gruppe	Anmerkungen
/usr/sap/<SAPSID>	751	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/<Instance ID>	755	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/<Instance ID>/sec	700	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/SYS	755	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/SYS/*	755	<sapsid>adm/sapsys	
(1) /usr/sap/<SAPSID>/SYS/exe (ohne "Inplace" Datenbank)	(2)700	<sapsid>adm/sapsys	Setzung des s-Bits für die Gruppe. Wert (2)700 für ABAP und Java Systeme ohne "Inplace" Datenbank (SAP HANA, SAP Web Dispatcher). Nur der User <sid>adm muss auf die Dateien zugreifen können.
(2) /usr/sap/<SAPSID>/SYS/exe (mit "Inplace" Datenbank z. B. Oracle)	(2)750	<sapsid>adm/dba	Setzung des s-Bits für die Gruppe. Wert (2)750 für ABAP und Java Systeme mit "Inplace" Datenbank wie z. B. Oracle. Nur der User <sid>adm muss auf die Dateien zugreifen können. Wenn eine Oracle Datenbank genutzt wird, liegen hier die ebenfalls die brtools auf die der User ora<sid> zugreifen können muss.

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys /usr/sap/<SAPSID>
chmod 751 /usr/sap/<SAPSID>
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.22-77/13.0

Req 78	Das SAP-Transportverzeichnis und dessen Inhalt müssen vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.
--------	---

Das SAP-Transportverzeichnis muss vor nicht autorisierten Zugriff geschützt sein. SMB oder NFS freigaben können vorhanden sein.

Motivation: Das Transportverzeichnis ist für Datenaustausch zwischen den Systemen eines Verbundes (Abnahme-, Test-, Produktion-, ...) zuständig und kann schutzbedürftige Daten (Programme, neuen Benutzer, ...) enthalten. Unbefugte Änderungen an Dateien im SAP-Transportverzeichnis können schwerwiegende Folgen für das System haben.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigungen	OS-User/OS-Gruppe	Anmerkungen
/usr/sap/trans	(2)770	<sapsid>adm/sapsys	Setzung des s-Bits für die Gruppe. (2)770 wenn möglich, wenn nicht (2)775.
/usr/sap/trans/*	770	<sapsid>adm/sapsys	
/usr/sap/trans/.sapconf	775	<sapsid>adm/sapsys	

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys /usr/sap/trans
chmod 775 /usr/sap/trans
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.22-78/13.0

Req 79	Auf Betriebssystemebene müssen SAP Tools (Programme) vor unberechtigten Zugriff geschützt werden.
--------	---

Es muss sichergestellt sein, dass für die SAP-Tools die auf Betriebssystemebene benötigt werden, eigene Benutzer, Benutzergruppen und Verzeichnisse eingerichtet werden, um den unberechtigten Zugriff zu verhindern.

Motivation: Es muss verhindert werden, dass einen unberechtigten Zugriff, Dateien und Dateiinhalte verändert. Die Änderungen können zu einem fehlerhaften Verhalten und/oder zu einem Ausfall führen.

Umsetzungsbeispiel: Es müssen mindestens die folgenden Tools (Programme) vor unberechtigten Zugriff geschützt werden:

Toolname	Verzeichnis	Berechtigung	Besitzer/Gruppe	Funktion
tp		770	<sapsid>adm/sapsys	Durchführen von Transporten
sapevt	/usr/sap/<SAPSID>/SYS/exe/run	750	<sapsid>adm/sapsys	Netweaver Events auf Betriebssystemebene starten
saposcol	/usr/sap/<SAPSID>/SYS/exe/run	4710	root/sapsys	Sammelt Informationen vom Betriebssystem

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.22-79/13.0

2.8.2. ABAP-Stack: Standardbenutzer SAP

Req 80	Der Benutzer SAP* muss geschützt werden.
--------	--

Der Benutzer SAP* ist der zentrale Administrator innerhalb des SAP-Systems und wird während der Installation vom SAP System angelegt. Es muss sichergestellt sein, dass der Benutzer nur in absoluten Ausnahmefällen benutzt wird.

Es muss zwischen dem Benutzer SAP* in der SAP-Anwendung und dem initialen Benutzer SAP* unterschieden werden. Der Benutzer SAP* in der SAP-Anwendung kann über die Transaktion SU01 bearbeitet werden. Während der initiale SAP* Benutzer mit dem Passwort "PASS" im SAP Kernel definiert ist und nicht verändert werden kann. So kann dieser nicht gesperrt werden und das Passwort und die Berechtigungen können nicht geändert werden. Ferner hat der initiale SAP* Benutzer weitreichende Berechtigungen. Der Zugriff auf diesen mächtigen Benutzer wird über einen Parameter und die Existenz des Benutzers SAP* in der SAP-Anwendung kontrollierbar. Der Benutzerstammsatz SAP* in der SAP-Anwendung muss ebenfalls stark eingeschränkt werden.

Motivation: Unberechtigter Zugriff mit dem Benutzer kann zur Dateneinsicht und -änderung mit einem Systemausfall führen.

Umsetzungsbeispiel: Machen Sie sich mit der Transaktion SCC4 einen Überblick, was für Mandanten auf dem System existieren und prüfen Sie diese auf Gültigkeit.

Gehen Sie mit der Transaktion SU01 in die Benutzerpflege auf dem jeweiligen Mandanten und prüfen bzw. setzen Sie folgendes für den Benutzerstammsatz des SAP*:

- Existiert in dem Mandanten.
- Muss gesperrt sein.
- Das Passwort muss deaktiviert (damit entfällt die Änderung des Standardpasswortes).
- Muss der Gruppe SUPER zugeordnet sein.
- Sämtliche Berechtigungen müssen gelöscht sein.

Zudem ist der Systemparameter „login/no_automatic_user_sapstar=1“ zu setzen (Transaktion RZ10).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Unbemerkt durchführbare Angriffe

ID: 3.22-80/13.0

Req 81 Der Benutzer DDIC muss vor nicht autorisierten Zugriff geschützt werden.

Der Benutzer DDIC ist ein zentraler Administrator innerhalb des SAP-Systems und wird während der Installation vom SAP System angelegt. Es muss sichergestellt sein, dass der Benutzer nur in Ausnahmefällen benutzt wird.

Motivation: Unberechtigter Zugriff mit dem Benutzer kann zur Dateneinsicht und -änderung mit einem Systemausfall führen.

Umsetzungsbeispiel: Machen Sie sich mit der Transaktion SCC4 einen Überblick, was für Mandanten auf dem System existieren und prüfen Sie diese auf Gültigkeit.

Gehen Sie mit der Transaktion SU01 in die Benutzerpflege auf dem jeweiligen Mandanten und setzen Sie folgendes für den Benutzerstammsatz des DDIC:

- Muss gesperrt sein.
- Das Passwort muss deaktiviert (damit entfällt die Änderung des Standardpasswortes).
- Muss der Gruppe SUPER zugeordnet sein.

Im Standard wird der Benutzer DDIC für den Transport Hintergrund Job (RDDIMPDP) benutzt. Um den DDIC Benutzer sperren zu können muss ein neuer Benutzer für die Ausführung des Job wie folgt angelegt werden:

- Der neue Benutzer benötigt SAP_ALL und S_A.SYSTEM Berechtigungen, weil die Jobaufrufe von unterschiedlichen Anwendungen erfolgen.
- Der neue Benutzer muss dem Usertyp "system" zugeordnet werden, damit dieser nicht für eine unberechtigte Dialoganmeldung missbraucht wird.
- Der neue Benutzer muss in allen Mandanten angelegt werden, die für Importe benutzt werden.
- Richte den Job RDDIMPDP so ein, dass der neue Benutzer der Besitzer ist (mit Transaktion SM37).

ID: 3.22-81/13.0

Req 82 Der Benutzer EARLYWATCH darf nur im Mandanten 066 existieren und darf nur dann aktiviert werden, wenn er benötigt wird.

Der Benutzer EARLYWATCH wird für den Early-Watch-Service in Mandant 066 benötigt. Er darf nur im Mandanten 066 existieren und darf nur dann aktiviert werden, wenn er benötigt wird.

Motivation: Über den Mandanten 066 und dem Benutzer EARLYWATCH werden Systeminformationen gesammelt, die z. B. für einen Going-Life-Check oder einen EARLYWATCH-Bericht gesammelt werden.

Umsetzungsbeispiel: Gehen Sie mit der Transaktion SU01 in die Benutzerpflege in den Mandanten 066 und setzen Sie folgendes für den Benutzerstammsatz des EARLYWATCH:

- Muss gesperrt sein.
- Das Passwort muss deaktiviert (damit entfällt die Änderung des Standardpasswortes).
- Muss der Gruppe SUPER zugeordnet sein..

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-82/13.0

Req 83	Der Benutzer SAPCPIC muss vor nicht autorisierten Zugriffen geschützt sein und darf nicht für RFC-Verbindungen eingesetzt werden.
--------	---

SAPCPIC darf nicht als Service-Benutzer für RFC-Destinationen eingesetzt werden. Stattdessen sollten dazu anwendungsspezifische Benutzer eingerichtet werden.

Motivation: Der SAPCPIC ist ein unspezifizierter Systembenutzer, der für den Aufruf von Programmen und Funktionsbausteinen im SAP-System verwendet werden kann. Es ist nicht ersichtlich, wer den Benutzer nutzt. Deswegen sollte dieser Benutzers nicht verwendet werden. Stattdessen sollten dazu eigene anwendungsspezifische Benutzer eingerichtet werden. Durch eine Namenskonvention kann bei größeren Systemen oder Landschaften eine klarere Struktur und eine der Verwendung angepassten Berechtigungsvergabe erfolgen.

Umsetzungsbeispiel: Gehen Sie mit der Transaktion SU01 in die Benutzerpflege auf dem jeweiligen Mandanten:

1. Bevorzugte Option: Löschen Sie den Benutzer SAPCPIC, wenn dieser nicht benötigt wird!
2. Option: Wenn der Benutzer nicht gelöscht werden kann, ...
 - ... ändern Sie das Standardpasswort
 - ... ordnen Sie den Benutzer der Gruppe SUPER zu
 - ... der Benutzer hat den Benutzertyp "system"

ID: 3.22-83/13.0

Req 84	Der Benutzer TMSADM ist ein Systembenutzer, der nur für die Kommunikation vom Transport Management System (TMS) genutzt werden darf.
--------	--

Das Benutzerkonto TMSADM wird für die Kommunikationsinfrastruktur vom TMS benötigt. Beim Initialisieren und Konfigurieren des TMS-Services im Mandant 000 wird ein Standardpasswort angelegt, das geändert werden muss. Durch die Verwendung des TMS werden RFC-Verbindungen zu Systemen im Transportverbund (nach den konfigurierten Transportwegen) erstellt. Über diese Verbindungen kommuniziert der Benutzer TMSADM mit anderen Systemen im Verbund.

Motivation: Es ist zu verhindern, dass Transporte nur von autorisierten Personen durchgeführt werden können und Transportinhalte auch nur für diese sichtbar sind. Somit ist die Sicherung des TMSADM Benutzers sehr wichtig. Beispielsweise können Informationen über die Transportverzeichnisse und deren Inhalte vorhanden sein.

Umsetzungsbeispiel: Hierzu sind die folgenden Aktionen durchzuführen:

1. Der Benutzerstammsatz existiert nur im Mandanten 000.

2. Der Benutzerstammsatz TMSADM muss der Gruppe SUPER zugeordnet sein.
3. Das Standardpasswort muss geändert werden.

Der TMSADM Benutzer kann nicht gesperrt werden!

Weitere Informationen zum Setzen des TMSADM Passwortes erhalten Sie in folgenden SAP Hinweisen: 761637, 1488406, 1610103, 1414256, 1552894, 1726102

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-84/13.0

2.8.3. ABAP-Stack: Benutzer

Req 85 Alle Benutzer müssen über einen geeigneten Benutzertyp verfügen, der sich nach der Art der Benutzung richtet.

In einem SAP-System gibt es verschiedenen Benutzertypen: Dialog-, System-, Kommunikation- und Servicebenutzer. Je nach Art der Verwendung muss ein passender Benutzertyp ausgewählt werden.

Motivation: Die verschiedenen Benutzertypen haben unterschiedliche Eigenschaften. Die Verwendung eines falschen Benutzertyps kann zusätzliche Angriffsflächen eröffnen (z. B. Möglichkeit zur interaktiven Anmeldung für technische Benutzer.

Umsetzungsbeispiel: Der Benutzertyp muss zur Funktion des Benutzers passen. Ist der Benutzer ein Anwender benötigt dieser den Benutzertyp "Dialog". Geht es um eine Kommunikation innerhalb des Systems oder zwischen Systemen wird der Usertyp "Kommunikation" für den Benutzer ausgewählt. Sind Jobs auf dem System durchzuführen, bekommt der Benutzer den Benutzertyp "System". Der Benutzertyp "Service" ist nur in begründeten Ausnahmefällen zu verwenden.

Eigenschaft	Dialog (A)	Kommunikation (C)	System (B)	Service (S)
GUI Anmeldung	Ja	Nein	Nein	Ja
RFC Anmeldung	Ja	Ja	Ja	Ja
Erzwingung der Kennwortänderung	Ja	Ja	Nein	Nein
Kennwortablauf	Ja	Ja	Nein	Nein
Anmeldeticket kann erzeugt werden	Ja	Ja	Nein	Nein

Bitte führen Sie zur Prüfung den Report RSUSR200 aus.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-85/13.0

Req 86 Für die Systemadministration muss ein Notfallbenutzer (mit vollständigen Rechten) angelegt werden.

Es muss ein Notfallbenutzer mit vollständigen Rechten (SAP_ALL) vorhanden sein. Die Benutzung muss über das „Central Administrator Access“ (CAA) authentifiziert und angemeldet werden. Für diesen Notfallbenutzer muss das SAP-Auditing eingeschaltet sein.

Motivation: Bei einem Notfall muss die SAP-System-Administration (SAP-BASIS) noch Zugriff auf das System haben. Hierzu wird ein Notfallbenutzer mit vollständigen Rechten eingerichtet.

Umsetzungsbeispiel: Verwendung des CAA (Central Administrator Access) Tools der T-Systems.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.22-86/13.0

2.8.4. ABAP-Stack: Berechtigungen

Req 87 Eine Trennung in der Benutzerverwaltung zwischen Benutzer-, Profil- und Rollenadministration muss umgesetzt werden.

Motivation: Es muss verhindert werden, dass die Security Policies (need-to-know, need-to-do, segregation of duties) umgangen werden.

Umsetzungsbeispiel: Hierfür liefert SAP Template Rollen aus, die noch auf die Begebenheiten des Benutzer- und Berechtigungskonzeptes angepasst werden müssen:

Template:	Administrator
SAP_ADM_PR	Profilmanagement
SAP_ADM_AU	Rollenmanagement
SAP_ADM_US	Benutzeradministration. Die Benutzeradministration wird über die Transaktion SU01 und die Rollen- und Profilpflege über die Transaktion PFCG ausgeführt.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-87/13.0

Req 88 Auf SAP-Ebene definierte Betriebssystemkommandos müssen vor nicht autorisiertem Zugriff geschützt sein (Transaktionen SM49 und SM69)

Der Zugriff aus der Applikation (SAP-Ebene) auf das Betriebssystem (z.B. Konsole) muss ebenso abgesichert werden, wie der direkte Zugriff aus der Applikation auf Programme des Betriebssystems (z.B. Ping, ls, nc, ...)

Motivation: Die mit den Transaktionen SM69 und SM49 definierten ausführbaren Betriebssystemkommandos werden unter dem SAP-Administrationsuser auf Betriebssysteme (<sid>adm) ausgeführt. Über Programme können Sicherheitsrelevante Informationen gelesen, verändert und/oder gelöscht werden. Beispielsweise kann mit SAP Tool "tp" ein SAP-Transport auf Betriebssystemebene ausgeführt werden oder es kann der gesamte SAP-Verzeichnisbaum gelöscht und somit ein Systemausfall provoziert werden.

Umsetzungsbeispiel: Die Validierung der Einstellungen erfolgt über mehrere Schritte:

1. Über die Benutzerinformationssystem TA SUIM muss geprüft werden, in welcher Rolle/Profil die TAs SM49 und SM69 (Berechtigungsobjekt S_TCODE) enthalten sind und welchen Benutzer diese zugewiesen wurden.
2. Zusätzlich muss das Berechtigungsobjekt S_LOG_COM geprüft werden. In dem Feld "COMMAND" wird das logische Kommando eingetragen. Es können generische Eingaben für das logische Kommando erfolgen wie z. B. für eine Gruppe aus dem Finanzbereich tätig ist könnte der generische Wert "ZFI*" sein (darunter können dann mehrere logische Kommandos definiert werden, die alle mit einem anderen Betriebssystemkommando verknüpft sind).
3. Über den Abgleich mit dem Berechtigungskonzept und Tätigkeitsprofil muss abgeglichen werden, ob das Need-to-do-Prinzip wirklich eingehalten wurde.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.22-88/13.0

Req 89	Das Ausführen von Betriebssystemkommandos über den Report RSBDCOS0 muss vor nicht autorisiertem Zugriff geschützt sein.
--------	---

RSBDCOS0 ist ein ABAP Report/Programm, das eine Konsole/Shell öffnet und erlaubt, Betriebssystemkommandos auf Betriebssystemebene aus dem SAPGUI heraus auszuführen.

Um den Report RSBDCOS0 auszuführen, werden verschiedene Berechtigungsobjekte benötigt. Siehe hierzu den SAP Hinweis 2297349.

Motivation: Die mit den Report RSBDCOS0 ausgeführten Betriebssystemkommandos, die unter dem SAP-Administrationsuser auf Betriebssysteme (<sid>adm) ausgeführt werden, können weitreichende Änderungen am dem System vornehmen. Über diese Kommandos können Sicherheitsrelevante Informationen gelesen, verändert und/oder gelöscht werden. Beispielsweise kann mit SAP Tool "tp" ein SAP-Transport auf Betriebssystemebene ausgeführt werden oder es kann der gesamte SAP-Verzeichnisbaum gelöscht und somit ein Systemausfall provoziert werden. Ein unberechtigter Zugriff muss verhindert werden.

Umsetzungsbeispiel: Die Validierung der Einstellungen erfolgt über mehrere Schritte (siehe auch SAP Hinweis 2297349):

Über das Benutzerinformationssystem TA SUIM muss geprüft werden, in welcher Rolle/Profil folgende Objekte mit den aufgeführten Ausprägungen enthalten sind und welchen Usern diese zugeordnet wurden:

Objekt	Felder	Kommentar
S_TCODE	TCD=SE38 und/oder SA38 und SM69 und SM49	
S_RZL_ADM	ACTVT=01	

S_LOG_COM	COMMAND=RSBDCOS0 OPSYSTEM=<Name des OS> HOST=<Hostname>	OPSYSTEM und HOST Werte können Wildcards * sein, z. B.: OPSYSTEM=* HOST=*
S_C_FUNCT	PROGRAM=RSBDCOS0 ACTVT=16 (Execute) CFUNCNAME=SYSTEM	Ohne diese Berechtigung wird bei Ausführung der Short Dump CALL_C_FUNCTION_NO_AUTHORITY erzeugt

Die Möglichkeit zur Ausführung des Reportes RSBDCOS0 benötigt ebenfalls, dass die TAs SM69 und SM49 nicht gesperrt sind (überprüfbar mit der TA SM01 (neuer SM01_DEV und SM01_CUS, s. SAP Hinweis 2234192).

Über den Abgleich mit dem Berechtigungskonzept und Tätigkeitsprofil muss geprüft werden, ob das Need-to-do-Prinzip wirklich eingehalten wurde.

ID: 3.22-89/13.0

Req 90 Die Download-Berechtigung für Benutzer muss begrenzt sein.

Die Download-Berechtigung ist über ein Berechtigungskonzept so (aufgabenbezogen) einzuschränken, dass Benutzerkreise, für deren effiziente Arbeitsabläufe die Downloadberechtigung wichtig ist, weiterhin über diese verfügen können, anderen jedoch die Berechtigung entzogen wird.

Motivation: Durch einen uneingeschränkten Download oder Export, entsteht der Verlust der Vertraulichkeit von schutzbedürftigen Daten, bis hin zum Verstoß gegen das Datenschutzgesetz durch Zweckentfremdung.

Umsetzungsbeispiel: Standardmäßig ist der Download oder der Export von SAP-Reports für alle Benutzer freigeschaltet. Entzug der Berechtigung S_GUI Aktivität 61

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Störung der Verfügbarkeit

ID: 3.22-90/13.0

Req 91 Die ICF-Kommunikation muss auf Benutzerebene eingeschränkt werden.

Das Internet-Communication-Framework (ICF) ist der Teil eines Netweaver-Systems, dass für die Bereitstellung von Services (Webseiten und Webapplikationen) zuständig ist. Damit der Http-Request bei Eingabe einer bestimmten URL im System zugeordnet werden kann, muss dieser an einen Handler im ICF-Service gebunden (konfiguriert) sein. Die Zuordnung von Benutzern für interne, freigeschaltete Services, ist über das Berechtigungskonzept zu realisieren.

Motivation: Über die ICF-Dienste gibt es viele Möglichkeiten außerhalb der SAP-GUI (Protokoll DIAG) auf das Netweaver-System und die Daten zuzugreifen. Sobald das Protokoll HTTP verwendet wird, kommen ICF-Dienste zum Einsatz. Die ICF-Dienste sind nicht standardmäßig geschützt. Das heißt, jeder Benutzer kann auf den ICF- Dienst zugreifen, wenn dieser aktiv ist.

Umsetzungsbeispiel: Die Zuordnung der Dienste und Handler erfolgt bei jedem freigeschalteten Service und den dazugehörigen Benutzer in der Transaktion SICF. Die folgenden Berechtigungen können durch den Systemadministrator an einen Benutzer zugewiesen werden:

Berechtigungsobjekt: S_ICFREC

Feld: ACTVT (Aktivität):

- 02 Ändern

- 03 Anzeigen
- 06 Löschen
- 16 Ausführen
- 63 Aktivieren
- 70 Verwalten, Administrieren D1 Kopieren
- DL Download
- UL Upload

Feld: USER (Benutzername)

Feld: CLIENT (Mandant)

Feld: SYSTEM (System-ID)

Die Berechtigungen sind restriktiv zu vergeben (Need-to-do-principle, Least Privilege Prinzip).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-91/13.0

Req 92	Administrative Transaktionen, die für den Betrieb und die Wartung des Systems eingesetzt werden, müssen für normale Benutzer unzugänglich sein.
--------	---

Administrative Transaktionen, die für den Betrieb und die Wartung des Systems eingesetzt werden, müssen für normale Benutzer unzugänglich sein. Als administrative Transaktionen gelten z.B. PFCG, RZ10, RZ11, SCC4, SE16, SE38, SE80, SM21, SM31, STMS, SUIM, SU01, SU02 ...

Motivation: Dies kann erreicht werden, indem normalen Benutzern restriktive Profile/Rollen zugewiesen werden, die den Autorisierungscode S_TCODE nur mit ausdrücklich gestatteten Transaktionen enthalten.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-92/13.0

Req 93	Transaktionen und Module die nicht benötigt werden, müssen deaktiviert sein.
--------	--

Um die Sicherheit in einem SAP-System zu gewährleisten, sind Transaktionen und Module, die nicht benutzt werden zu deaktivieren oder zu löschen. Ist dies nicht Möglich, muss die nicht Benutzung über Rollen und Berechtigungen sicherzustellen werden .

Motivation: Nach der Installation stehen in einem SAP-System alle verfügbaren Module und Transaktionen (allen Mandanten) zur Verfügung. Durch den Entzug von Berechtigungen und das Deaktivieren von Modulen wird das System für die Mandanten angepasst und das "Need to know" umgesetzt. Über das aktivieren oder deaktivieren von Modulen und Transaktionen wird auch der Funktionsumfang an die Lizenz angepasst.

Umsetzungsbeispiel: Die folgende Transaktionen sollten für Benutzer in den Mandanten via Transaktion SM01 gesperrt sein:

K A 1 0 , K A 1 2 , K A 1 6 , K A 1 8 , S A R A ; O B R 1 ; O I C P , O O S B ;
 OMDL,OMEH,OMWF,OOUS,OPF0,OTZ1,OY27,OY28,OY29,OY30; OMEI,OMWG,OOPR,OP15,OPE9,OTZ2,OY21;
 OMG7,OMWK,OPF1,OTZ3,OY20; OOSP; OVZ6;SCC1;SCC5;SE01;SE09,SE10;SE16,SM30,SM31;SM69;SU12.

Req 94 Das Ausführen der Transaktion RSUDO muss durch passende Berechtigungen geregelt sein.

Die Transaktion RSUDO kann genutzt werden um Transaktionen oder Prozeduren mit höheren Berechtigungen zu starten (ähnlich wie sudo oder "ausführen als") und muss über Rollen und Berechtigungen geschützt sein.

Motivation: Insbesondere in Z-Programmen muss geprüft werden ob das RSUDO genutzt wird.

Req 95 Die Nutzung und Konfiguration von Trusted-/Trusting-Beziehungen zwischen SAP Systemen muss restriktiv sein.

Bei Trusted-Systemen handelt es sich um eine definierte Vertrauensbeziehung zwischen SAP-Systemen (Trusting System). Die Vertrauensbeziehung bewirkt, dass Zugriffe zwischen Trusted-Systemen ohne explizite Authentifizierung möglich ist .

Motivation: Bei Einrichtung einer Vertrauensbeziehung und fehlerhafter Konfiguration kann ein beliebiger User vom dem Trusted System auf einem beliebigen User mit dem Berechtigungsobjekt S_RFCACL des Trusting Systems zugreifen und im Rahmen dessen Berechtigungen alle Aktionen ausführen.

Umsetzungsbeispiel: Das Trusting System sollte das System mit der niedrigeren Schutzstufe sein, so dass das System mit der hohen Schutzstufe sich die Informationen vom System mit der niedrigen Schutzstufe holt; beispielsweise darf ein sich ein DE4 System Daten von einem DE3 System abholen. Trusting Systeme anzeigen mit Transaktionscode SMT2 oder über die Tabellen RFCTRUST für die Trusted Systeme und RFCSYSCAL für die Trusting Systeme. In der RFC-Destination sollte für die betreffenden RFC-Verbindungen auf dem Trusted System die Checkbox "aktueller Benutzer" aktiviert sein.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Req 96 Bei der Nutzung des SAP eigenen Single-Sign-On (SSO) müssen die Benutzerberechtigungen entsprechend der Systemumgebung angepasst sein.

In SAP-Netweaver-Systemen können Anmelde-Daten auf verschiedenen SAP-Systemen mit Hilfe des SAP eigenen SSO genutzt werden. Dabei sind der Benutzername und Passwort auf allen Systemen bekannt (gleich). Die Berechtigungen müssen entsprechend der Aufgaben auf den unterschiedlichen Systemen angepasst werden.

Motivation: Vereinfachung für den Benutzer und zügigeres Arbeiten über verschiedene SAP-Systeme hinweg. Beispielsweise Anmeldung am SAP-Netweaver-Portal und Weiterleitung an das System, das für die zuständige Backendsystem.

Umsetzungsbeispiel: SAP-Systeme können Login-Ticket-Schlüssel erstellen oder verarbeiten.

- Über den Profilparameter login/accept_sso2_ticket wird gesteuert, ob das System Login-Ticket-Schlüssel annimmt. (0= keine Login-Tickets verarbeiten).
- Über den Profilparameter login/create_sso2_ticket wird gesteuert, ob das System Login-Schlüssel erzeugt. (0= keine Erzeugung).
- Eine zweite Art des Single-Sign-On ist, dass der Account auf allen System eingerichtet ist.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.22-96/13.0

Req 97	Es muss gewährleistet werden, dass für das Direct-Input-Verfahren ein 4-Augenprinzip eingeführt ist, der Einsatz des Verfahrens dokumentiert wird und dazu eine Prozessbeschreibung besteht.
--------	--

Es muss gewährleistet werden, dass für das Direct-Input-Verfahren ein 4-Augenprinzip eingeführt ist, der Einsatz des Verfahrens dokumentiert wird und dazu eine Prozessbeschreibung besteht.

Motivation: Beim Direct-Input-Verfahren werden die Daten mit reduzierten Plausibilitätsprüfungen direkt in die Datenbank geschrieben. Zudem ist es möglich, vorhandene Datenbestände nicht protokolliert zu verändern. Bei einer fehlerhaften oder unbedachten Anwendung ist die Datenintegrität des Systems gefährdet.

Umsetzungsbeispiel: Zur Umsetzung nötig:

1. Es muss eine Prozessanweisung für das Direct Input Verfahren existieren.
2. In dieser Prozessanweisung ist beschrieben, dass ein 4-Augenprinzip umgesetzt wird.
3. In dieser Prozessanweisung ist beschrieben, wie die Nutzung des Direct Input Verfahren dokumentiert wird.

Zur Überprüfung, welche User das Verfahren benutzen dürfen, muss folgendes ausgeführt werden:

- Die Benutzer mit den entsprechenden Berechtigungen können über den Report RSUSR002 ermittelt werden. Dafür müssen 3 Auswertungen vorgenommen werden, für die Transaktionen BMV0, SXDA und LSMW jeweils eine. Um die genauen Abfragedaten zu erhalten, rufen Sie bitte folgenden Link auf: https://www.dsag.de/sites/default/files/150505_leitfaden_best-practice-sap-erp_rz.pdf

ID: 3.22-97/13.0

Req 98	Eine Berechtigungsprüfung für den Start von Web-Dynpro-ABAP-Anwendungen muss durchgeführt werden (Bedingung: Einsatz von S/4HANA).
--------	--

Mit der Nutzung von S/4HANA wird das Berechtigungsobjekt S_START eingeführt. Die Startberechtigungsprüfung gegen das Berechtigungsobjekt S_START (Web-Dynpro-ABAP-Anwendungen) wurde bereits mit den ABAP Basis-Releases 7.03 und 7.30 ausgeliefert.

Motivation: Ein unberechtigter Zugriff kann zur Einsicht, Änderung und oder Löschung von Daten führen.

Umsetzungsbeispiel: Überprüfen Sie zuerst, ob die Startberechtigungsprüfung für WebDynpro aktiv ist. Starten Sie Transaktion SU25. Unter "Anpassung der Berechtigungsprüfungen(optional)" starten Sie "Web Dynpro Startberechtigungsprüfung aktivieren (S_START)". In der darauffolgenden ALV-Liste sehen Sie zwei Zeilen:

Web Dynpro Anwendungskonfiguration	R3TR	WDCA
Web Dynpro Applikation	R3TR	WDYA

Wenn diese beiden Zeilen in der Spalte "Inaktiv" nicht markiert sind (Häkchen gesetzt), ist die Startberechtigungsprüfung für die entsprechenden Objekte aktiv.

Sie haben folgende Möglichkeiten zur Konfiguration:

- Sie können Ihre Berechtigungsrollen mit geeigneten Berechtigungen zum Berechtigungsobjekt S_START ergänzen. Die grundlegende Vorgehensweise ist bereits im Hinweis 1413011 beschrieben. Zusätzlich haben Sie die Möglichkeit, in Transaktion SUIM unter "Rollen" den Report "Suche in Rollen nach startbaren Anwendungen" zu starten, mit dem Rollen in einer vereinfachten Weise angepaßt werden können. Wählen Sie hierzu im Eingabefeld "Anwendungstyp" den Wert "Web Dynpro Applikation" bzw. "Web Dynpro Anwendungskonfiguration" aus.
- Sie können allen Benutzern, die WebDynpro-Anwendungen ausführen sollen, die Rolle SAP_BC_WEBDYNPRO_ALL (oder eine Kopie dieser Rolle) zuordnen, die eine Gesamtberechtigung zum Starten aller WebDynpro-Anwendungen und WebDynpro-Anwendungskonfigurationen enthält.
- Nicht empfohlen/erlaubt: Sie können die Startberechtigungsprüfung für WebDynpro-Anwendungen und WebDynpro-Anwendungskonfigurationen abschalten.

ID: 3.22-98/13.0

2.8.5. ABAP-Stack: Logging

Req 99	Änderungen an kritischen Tabellen müssen über die "Tabellenprotokollierung" oder über die "Protokollierung von Änderungsbelegen" protokolliert werden.
--------	--

Die Tabellenprotokollierung ist im Standard nicht aktiv und muss eingeschaltet werden, um gesetzliche Vorgaben zu entsprechen.

Motivation: Ohne die Tabellenprotokollierung besteht ein Risiko eines unberechtigten und unkontrollierten Zugriffs auf Tabellen. So kann z. B. ein Angreifer den Bank Account ändern und somit Geld auf einen anderen Account transferieren. Eine nicht aktivierte Protokollierung kann außerdem zu einer strafrechtlichen Verfolgung führen. SAP bietet die Möglichkeit, Änderungen an Tabellen zu protokollieren. Über die Tabellenprotokollierung werden maßgeblich Customizing-Daten aufgezeichnet. Somit kann nachvollzogen werden, welche Benutzer in Tabellen Datensätze verändert, hinzugefügt oder gelöscht haben. Standardmäßig sind in SAP Systemen inkl. SAP S/4 HANA bereits ca. 40.000 Tabellen zur Protokollierung vorgesehen. Hierunter befinden sich auch rechnungsrelevante Tabellen, deren Änderungsprotokollierung gesetzlich vorgeschrieben ist. Customizing-Tabellen gelten als Verfahrungsanweisung und sind (daher gemäß §257 HGB zehn Jahre lang) aufbewahrungspflichtig. Die Tabellenprotokollierung ist im Standard NICHT aktiv und ist somit ein Muss. Insbesondere sind auch die unternehmenseigenen Tabellen hiervon betroffen. Eigene Tabellen mit rechnungslegungsrelevanten Inhalten müssen ebenso protokolliert werden wie die SAP-Standardtabellen. Änderungen können über ZWEI Wege erfolgen, die auch separat aktiviert werden müssen. Entweder werden Änderungen direkte im SAP-System durchgeführt oder durch Transporte über das Transportsystem. Die direkten Änderungen im System werden durch Setzen des Profilparameters rec/client aufgezeichnet, die Änderungen durch Transporte durch Setzen des Transportparameters RECCLIENT. Alle Änderungen werden in der Tabelle DBTABLOG gespeichert. Die Logdaten können über die SAP-Transaktion SCU3 eingesehen werden. Ein Logeintrag liefert Informationen zu dem Zeitpunkt, zu dem Benutzer und dem Applikationsserver, auf dem die Änderung vorgenommen wurde.

Änderungen an betriebswirtschaftlichen Daten (Stammdaten oder Belegdaten wie Verkaufsbelege, Kreditkartendaten oder Lieferantendaten), die häufigen Änderungen unterliegen, werden im SAP System durch Änderungsbelege protokolliert. Die Protokollierung wird über sogenannte Änderungsbelegobjekte gesteuert. In solchen Objekten werden die Tabellen angegeben, die protokolliert werden. Die Objekte werden über die TA SCDO bzw. SCDO_NEW verwaltet. Zu jedem Änderungsobjekt sind Tabellen definiert, für die Änderungsbelege geschrieben werden sollen. Änderungen an diesen Tabellen sind gemäß § 257 HGB aufzeichnungspflichtig und sind ebenfalls nach §257 HGB zehn Jahre lang aufbewahrungspflichtig. Alle Änderungsbelegobjekte werden in der Tabelle TCDOB gespeichert. Die Daten der Protokollierung werden in den Tabellen CDHDR und CDPOS abgelegt. In den Logs ist der Benutzername, das Datum der Änderung die genutzte Transaktion sowie die Art der Änderung und die Änderungsnummer des Belegs einsehbar.

Umsetzungsbeispiel: 1.1) Die Tabellenprotokollierung wird in den technischen Einstellungen zu einer Tabelle gesetzt (TA SE13 oder TA RDDPRCHK_AUDIT). Um sich alle protokollierten Tabellen anzuzeigen, muss die Tabelle DD09L mit der TA SE16N angezeigt werden. Berücksichtigen Sie alle kritischen Systemtabellen (s. SAP Hinweis 112388) und besonders auch die relevanten unternehmenseigenen Tabellen für die Protokollierung! Diese Einstellungen werden

grundsätzlich im Entwicklungssystem vorgenommen und in das Produktionssystem importiert!

1.2) Aktivieren Sie die Tabellenprotokollierung für direkte Änderungen an Tabellen durch Setzen des folgenden SAP Profilparameter:

- rec/client = ALL (oder wenigstens alle produktiven Mandanten. Wert "ALL" ist möglich, da z. B. in 000 und 066 keine Tabellenänderungen stattfinden)

1.3) Um die Änderungen an Tabellen über Transporte zu protokollieren, setzen Sie den folgenden Transportparameter auf dem Transportdomänencontroller in der TA STMS auf den gleichen Wert wie den SAP Profilparameter rec/client (s. SAP Hinweis 163694):

- RECCLIENT = <gleicher Wert wie SAP Profilparameter rec/client>

2) SAP-Systeme verwenden für Business-Objekte nicht automatisch Änderungsbelege. Sie müssen den Änderungsbeleg selbst aktivieren. Berücksichtigen Sie besonders auch die relevanten unternehmenseigenen Tabellen für die Protokollierung! So aktivieren Sie einen Änderungsbeleg für ein Objekt:

1. TA SCD0: Legen Sie den Änderungsbeleg an
2. TA SE11: Aktivieren Sie den Änderungsbeleg des Objekts in der Datenelementpflege
3. TA SCD0: Generieren Sie einen Verbucher für das Objekt.
4. Fügen Sie die entsprechenden Aufrufe in die betreffenden Programme ein.

Änderungsbelege für ein Objekt können Sie ebenfalls über die Transaktion SCD0 anzeigen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-99/13.0

Req 100	Das SAP Security Audit Log (SAL) muss aktiviert sein und sicherheitsrelevante Ereignisse müssen protokolliert werden (bis SAP NW 7.4).
---------	--

Systeme müssen sicherheitsrelevante Ereignisse protokollieren. Um Log-Informationen auf kritische Transaktionen und Programmteile zu bekommen, muss das SAL aktiviert sein.

Motivation: Die Erkennung von (gezielten) Angriffen, sowie die Erkennung erfolgreicher Kompromittierungen von Systemen innerhalb der Netze des DTAG-Konzerns erfordert eine umfangreiche Protokollierung von sicherheitsrelevanten Ereignissen auf Systemen, die von solchen Angriffen direkt betroffenen oder mittelbar daran beteiligt sein können. Systeme müssen sicherheitsrelevante Ereignisse protokollieren können. Bei der Protokollierung müssen die jeweils geltenden gesetzlichen, regulatorischen und betrieblichen Bestimmungen beachtet werden. Diese besagen unter anderem, dass die Protokollierung von Ereignissen nur zweckgebunden erfolgen darf. Die Protokollierung von Ereignissen, zum Zwecke der Arbeitskontrolle von Mitarbeitern, ist nicht zulässig. Die Protokollierung folgender Ereignisse muss aktivierbar sein:

- Authentisierungen von Benutzern und Systemen (erfolgreich und nicht erfolgreich)
- Administrative Zugriffe (inkl. ausgeführter Funktionen, erfolgreich und nicht erfolgreich)
- Unerlaubte/Fehlerhafte Zugriffsversuche (z. B. Zugriff auf für einen Account nicht berechtigte Funktionen oder Zugriff auf eine nicht existente Funktion)
- Administration von Konten
- Änderung von Gruppenzugehörigkeiten von Konten.
- Weitere sicherheitsrelevante Systemmeldungen, die jeweils systemspezifisch festgelegt werden müssen.

Umsetzungsbeispiel:

1. Ausführen der TA RZ10 und setzen folgender Parameter:
 - rsau/enable = 1 (Aktivierung des SAL)
 - rsau/selection_slots = 10 (max. Anzahl Auditfilter)
 - rsau/user_selection = 1 (aktiviert generische Usernameneingabe, welches die Zahl Filter sparen kann)
 - Systemneustart erforderlich
2. Ausführen der TA SM19 zum Konfigurieren von Filtern in der statischen Konfiguration nach sicherheitsrelevanten Ereignissen wie Sie im Langtext oben beschrieben sind. SAP anwendungsbezogen können weitere Aktionen oder Benutzer geloggt werden wie ...
 1. Debugging mit Replace, Systemänderbarkeit geändert, Änderungsbeläge wurden ohne Archivierung gelöscht
 2. Alle Aktionen der Benutzer DDIC, SAP*, Notfalluser, weitere unpersonifizierte Benutzer z. B. kann ein Filter mit DDIC erstellt werden, der für all Mandanten gültig ist durch das Einfügen des "*" im Feld Mandant.
 3. Benutzer mit den Profilen SAP_ALL, SAP_NEW, S_A.SYSTEM, S_A.ADMIN, S_A.DEVELOP, S_ADMI_ALL, R3_BASIC, S_A.CUSTOMIZ, S_ABAP_ALL, S_ADMI_SAP, S_ENTW, S_RZL_ADM
3. Aktivieren Sie die Filter, in dem der Haken bei "aktiv" gesetzt wird.
4. Führen Sie einen Restart durch, um die durchgeführten Einstellungen zu aktivieren.

ID: 3.22-100/13.0

Req 101 Sicherheitsrelevante Ereignisse müssen protokolliert werden (ab SAP NW 7.5).

Systeme müssen sicherheitsrelevante Ereignisse protokollieren können. Bei der Protokollierung müssen die jeweils geltenden gesetzlichen, regulatorischen und betrieblichen Bestimmungen beachtet werden. Diese besagen unter anderem, dass die Protokollierung von Ereignissen nur zweckgebunden erfolgen darf. Die Protokollierung von Ereignissen, zum Zwecke der Arbeitskontrolle von Mitarbeitern, ist nicht zulässig. Die Protokollierung folgender Ereignisse muss aktivierbar sein:

- Authentisierungen von Benutzern und Systemen (erfolgreich und nicht erfolgreich)
- Administrative Zugriffe (inkl. ausgeführter Funktionen, erfolgreich und nicht erfolgreich)
- Unerlaubte/Fehlerhafte Zugriffsversuche (z. B. Zugriff auf für einen Account nicht berechnigte Funktionen oder Zugriff auf eine nicht existente Funktion)
- Administration von Konten
- Änderung von Gruppenzugehörigkeiten von Konten.
- Weitere sicherheitsrelevante Systemmeldungen, die jeweils systemspezifisch festgelegt werden müssen.

Motivation: Die Erkennung von (gezielten) Angriffen, sowie die Erkennung erfolgreicher Kompromittierungen von Systemen innerhalb der Netze des DTAG-Konzerns erfordert eine umfangreiche Protokollierung von sicherheitsrelevanten Ereignissen auf Systemen, die von solchen Angriffen direkt betroffenen oder mittelbar daran beteiligt sein können.

Umsetzungsbeispiel: Zur Aktivierung des SAL führen Sie folgende Schritte aus (beachten Sie dabei den SAP Hinweis 2191612):

1. Führen Sie TA RSAU_CONFIG aus: Und richten in der statischen Konfiguration in einem Profil Filter ein. Nur die statische Konfiguration ist nach einem Systemstart noch aktiv.
2. Eine Aufzeichnung der Protokolle in der Datenbank ist nicht empfehlenswert, da Sie die Protokolle auf ein anderes sicheres System ablegen müssen (Felder Aufzeichnungsziel und Aufzeichnungsart).
3. Definieren Sie genug Filter.
4. Sie sollten die generische Benutzerselektion auswählen. Das kann bei Usern, die gleich Zeichenfolgen haben zu definierende Filter sparen.

5. Richten Sie Filter nach sicherheitsrelevanten Ereignissen ein wie Sie im Langtext oben beschrieben sind. SAP anwendungsbezogen können weitere Aktionen oder Benutzer geloggt werden wie ...
 1. Debugging mit Replace, Systemänderbarkeit geändert, Änderungsbeläge wurden ohne Archivierung gelöscht
 2. Alle Aktionen der Benutzer DDIC, SAP*, Notfalluser, weitere unpersonifizierte Benutzer z. B. kann ein Filter mit DDIC erstellt werden, der für all Mandanten gültig ist durch das Einfügen des "*" im Feld Mandant.
 3. Benutzer mit den Profilen SAP_ALL, SAP_NEW, S_A.SYSTEM, S_A.ADMIN, S_A.DEVELOP, S_ADMI_ALL, R3_BASIC, S_A.CUSTOMIZ, S_ABAP_ALL, S_ADMI_SAP, S_ENTW, S_RZL_ADM
6. Aktivieren Sie den Filter.
7. Führen Sie einen Restart durch um die durchgeführten Einstellungen zu aktivieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.22-101/13.0

2.8.6. ABAP-Stack: Kommunikationssicherheit

Req 102	Externe Zugriffe auf das SAP-Gateway müssen über die Konfigurationsdatei „secinfo“ eingeschränkt werden.
---------	--

Um nur zulässige Zugriffe auf das Sap-System zu bekommen, muss das interne Sap-Gateway genutzt werden. Dieses geschieht durch die Konfiguration der Datei "secinfo". Kann das interne SAP-Gateway via IP erreicht werden, können Programme bei unkonfigurierter oder fehlender "secinfo"-Datei auf dem SAP-System gestartet werden.

Motivation: Angreifer können über das SAP-Gateway Programme im SAP-System starten, wenn die „secinfo“ nicht gepflegt ist. WICHTIG: Existiert eine „secinfo“, aber keine „reginfo“, wird via „secinfo“ auch die Registrierung am SAP-Gateway blockiert. Deshalb muss die „reginfo“ ebenfalls gepflegt werden, weil das SAP-System sonst nicht mehr wie gewohnt für Registrierungen verfügbar ist und SAP-Entwickler in der Praxis mit diesem Fehlerbild nicht vertraut sind.

Umsetzungsbeispiel: secinfo und reginfo sollten über gw/sec_info = \$(DIR_GLOBAL)/secinfo gw/reg_info = \$(DIR_GLOBAL)/reginfo ins Global-Verzeichnis verschoben werden, um den Pflegeaufwand je Instanz einzusparen. Beispiel secinfo – mehrere Zeilen nach folgendem Muster (#VERSION=2 ist kein Kommentar!)

```
#VERSION=2
P TP=* USER=* USER-HOST=local HOST=local
P TP=* USER=* USER-HOST=internal HOST=internal
```

Weiter Informationen: http://help.sap.com/saphelp_nw70ehp3/helpdata/en/e2/16d0427a2440fc8bfc25e786b8e11c/content.htm

Requirement-ID: 5fd17246

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Req 103 Registrierungen von Diensten am SAP-Gateway müssen über die Konfigurationsdatei „reginfo“ eingeschränkt werden.

Registrierungen von Diensten am SAP-Gateway müssen über die Konfigurationsdatei „reginfo“ eingeschränkt werden. Wenn die Datei "regInfo" nicht gepflegt ist, können externe Programme eigene Dienste über dieses Gateway anbieten und somit die Verarbeitung der Daten ungewollt verändern.

Motivation: Durch die Konfiguration der der "reginfo"-Datei, wird verhindert, dass sich externe unbekannte Programme am Sap-System anmelden und in die Art der Verarbeitung der Daten ungewollt eingreifen. Vorhandenen, bereits registrierten Dienste können nicht überschrieben werden

Umsetzungsbeispiel: secinfo und reginfo sollten über gw/sec_info = \$(DIR_GLOBAL)/secinfo gw/reg_info = \$(DIR_GLOBAL)/reginfo ins Global-Verzeichnis verschoben werden, um den Pflegeaufwand je Instanz einzusparen.

Beispiel reginfo: (#VERSION=2 ist kein Kommentar!)

```
#VERSION=2
P TP=* HOST=local
P TP=* HOST=internal
```

Requirement - ID: 3f7813be

Req 104 Der Zugriff von externen SAP-RFC-Diensten muss über SAP-Gateway kontrolliert sein.

Externe Zugriffe via RFC auf das SAP System müssen über das SAP-Gateway gefiltert werden. Dies erfolgt über 2 Parameter und die Pflege einer ACL (Access Control List).

Der Parameter gw/acl_file gibt den Namen einer ACL-Datei an. Mit einer ACL (Access Control List) können Sie konfigurieren, wer eine Verbindung zum Gateway aufbauen darf. Ist der Parameter nicht gesetzt, gilt keine Zugriffskontrolle!

Der Parameter gw/acl_mode definiert das Verhalten des Gateways, falls eine ACL-Datei (gw/sec_info oder gw/reg_info) nicht existiert. Mit dem Wert 1 werden externe und registrierte Server nur innerhalb des Systems (Applikationsserver des gleichen Systems) erlaubt. Alle anderen Server werden abgelehnt oder müssen in den jeweiligen Dateien (gw/sec_info oder gw/reg_info und gw/acl_file) gepflegt werden.

Mit der ACL wird gesteuert, welche Verbindungen das Gateway akzeptiert und welche nicht. Grundlage hierfür sind die IP-Adressen der Clients. Als letzte Regel wird ein generelles Verbot eingefügt (deny 0.0.0.0/0). Eine Zeile der ACL muss der folgenden Syntax entsprechen:

```
<permit | deny> <ip-address> [tracelevel] [# comment]
```

Motivation: Um die Kontrolle über die Daten nicht zu verlieren, muss ein unberechtigter Zugriff verhindert werden.

Umsetzungsbeispiel: Führen Sie alle Punkte aus, um eine sichere Konfiguration umzusetzen:

1) Ausführen der TA RZ10 und setzen folgender Parameter für die DEFAULT.PFL:

- gw/acl_file = <Verzeichnis>/<Dateiname>
- gw/monitor =1
- gw/acl_mode = 1

2) Beispiel für Einträge in die ACL Datei:

```
permit 10.1.2.0/24      # permit client network
deny   0.0.0.0/0       # deny the rest
```

https://help.sap.com/saphelp_em92/helpdata/de/48/b0e64ba49c2883e10000000a42189c/content.htm?no_cache=true

Requirement-ID: ccb4c58b

ID: 3.22-104/13.0

Req 105	Der Zugriff auf den SAP-Message Server muss begrenzt sein, um Überlastsituationen zu verhindern.
---------	--

Es muss gewährleistet sein, dass Anwender über den SAP-Message-Server zugreifen und somit eine Lastverteilung auf den SAP-System vorliegt. Die folgenden Parameter sind zu setzen:

- Maximale Anzahl von HTTP-Ports, die der Message-Server zusätzlich aufmachen kann.
- Maximale Anzahl von HTTP-Clients, die sich parallel am Message-Server anmelden können.
- Timeout für Netzwerk-Operationen: wenn in dieser Zeitspanne keine weiteren Aktionen erfolgen, wird die Verbindung zum HTTP-Client unterbrochen. Die Angabe des Timeout erfolgt in Sekunden.
- Maximale Länge eines HTTP-Headers, der eingelesen werden kann.

SAP empfiehlt generell für das HTTP-Loadbalancing von Webanwendungen den SAP Web Dispatcher zu verwenden (s. SAP Note 1040325).

Motivation: Das Setzen der Parameter minimieren das Risiko der Kompromittierung oder DoS (Denial of Service) Angriffen.

Umsetzungsbeispiel: Die folgenden Defaultwerte sind eingestellt und können je nach Projektkontext geändert werden:

- ms/http_max_ports:20
- ms/http_max_clients:1000
- ms/http_timeout:20
- ms/http_bufferIn:65636

Requirement-ID: dfca5a9f

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.22-105/13.0

Req 106	Systeme, die mit der RFC-Callback-Funktion arbeiten, müssen eine Whitelist gepflegt haben.
---------	--

In der Regel sollte auf die RFC-Callback-Funktion verzichtet werden.

Die Gefahr bei der RFC-Callback-Funktion ist, dass dabei die Rechte des aufrufenden Users aus dem Ursprungssystem verwendet werden können und somit oft weitgehende Rechte missbraucht werden können.

Mittels dem Funktionsaufruf "CALL FUNCTION <function> DESTINATION "BACK"> ist es möglich, auf einem SAP-System schädliche RFC-Funktionen auszuführen, wenn sich das aufrufende System via RFC an ein kompromittiertes System wendet. Das kompromittierte System kann mit DESTINATION "BACK" nämlich einen Rückkanal aufbauen.

Um das Risiko eines Angriffs zu minimieren, bietet SAP die Erstellung von Whitelists.

Weitere Informationen: <https://rz10.de/sap-security/sap-rfc-sicherheit/rfc-callback-positivliste-generieren/>

Motivation: Manipulierte RFC-Callback-Funktionen können einen unberechtigten Zugriff auf das angegriffene System erlauben.

Umsetzungsbeispiel: Der Profilparameter `rfc/callback_security_method` muss auf 3 gesetzt sein.

Wenn es erforderlich ist die RFC-Callback-Funktion zu nutzen, muss folgendes umgesetzt werden (Schritte zur Erstellung der Whitelist):

1. Um RFC Callback Aufrufe sichtbar zu machen, muss der Profilparameter `rfc/callback_security_method` auf „2“ (Simulationsmodus) gesetzt werden.
2. Zudem muss das Security Audit Log (SAL) TA SM20 aktiviert sein und in der Detailauswahl unter der Audit-klasse „RFC Funktionsaufruf“ die folgenden 3 Optionen anhängen:
 - RFC-Callback ausgeführt (Destination &A, Gerufen &B, Callback &C)
 - RFC-Callback abgewiesen (Destination &A, Gerufen &B, Callback &C)
 - RFC-Callback im Simulations-Modus (Destination &A, Gerufen &B, Callback &C)
3. Nach einer gewissen Laufzeit im Simulationsmodus bietet SAP die Möglichkeit, über die SM59 eine Whitelist / Positivliste zu generieren. Dazu die SM59 öffnen und den Button „RFC-Callback-Positivlisten generieren“ auswählen.
4. Unerwünschte Callback-Aufrufe können hier aussortiert werden. Die Positivliste kann anschließend über „Positivlisten-Generierung und Aktivieren“ generiert und aktiviert werden.
5. Anschließend wird der Profilparameter `rfc/callback_security_method` auf „3“ gesetzt. Damit sind nur die Callback-Aufrufe möglich, die in der Whitelist erlaubt werden. Falls die Callback-Funktion nicht genutzt werden soll, bleibt die Whitelist leer.

Es empfiehlt sich vor dem Blockieren aller Nicht-Whitelist Einträge (Parameterwert 3) die eigentlichen Aufrufe zu protokollieren. Wenn Sie den `rfc/callback_security_method` Parameter auf 2 gestellt und damit den Simulationsmodus gewählt haben, können Sie alle Callback Aufrufe in der SM20 nachvollziehen (Voraussetzung hierfür ist Punkt 3.). Und es ist möglich eine Whiteliste in der Transaktion SM59 zu generieren und zu aktivieren und unerwünschte Callbackaufrufe dabei auszusortieren.

Um sich die aktuellen Einträge der Whitelist einzusehen, rufen Sie in der SE16 die Tabelle RFCCBWHITELIST auf. Dort sind alle Einträge gelistet. Um zu sehen, welche Verbindungen auch aktiv sind, reicht ein Blick in die Tabelle RFCCBWHITELIST_A.

Requirement-ID: 7033e1c0

ID: 3.22-106/13.0

2.8.7. ABAP-Stack: Parameter

Req 107	Die Abwärtskompatibilität beim Passwort muss verhindert werden.
---------	---

Es muss verhindert werden, dass die Abwärtskompatibilität eingeschaltet ist. Die Abwärtskompatibilität wird nur dann benötigt, wenn sich Benutzer von SAP-Systeme kleiner 6.0, auf einem aktuellen Sap-System anmelden müssen.

Motivation: Das parallel erzeugte Passwort, das für die Anmeldung von Benutzern aus Alt-Sap-Systemen erforderlich ist wird als „BCODE- Hash“ mit sehr eingeschränkten Zeichensatz gespeichert Der BCODE ist mit einfacher PC-Hardware schnell zu entschlüsseln.

Umsetzungsbeispiel: Ausführen der Transaktion RZ10 und setzen des Parameters wie folgt:

- `login/password_downwards_compatibility = 0`

- Ausführen des Reports CLEANUP_PASSWORD_HASH_VALUES und Löschung alter überflüssiger Kennworthashwerte mit Hilfe des Reports.

SAP Hinweis 1458262, Verfahren für ABAP-Systeme mit SAP_BASIS 7.x:

Setzen Sie den Profilparameter login/password_downwards_compatibility auf den Wert 0, und starten Sie den (diesem SAP-Hinweis beigelegten) Report CLEANUP_PASSWORD_HASH_VALUES, um zu prüfen, ob überflüssige Kennwort-Hash-Werte in Ihrem System gespeichert sind. (Es wird angeboten, diese zu löschen.) Beachten Sie, dass Sie anschließend funktionale Fehler, die durch die Nutzung veralteter Frontend-/Middleware-Komponenten verursacht werden, möglicherweise nicht mehr ausgleichen können (Nähere Informationen finden Sie dazu in SAP-Hinweis 1023437).

ID: 3.22-107/13.0

Req 108 Das SAP-System darf keine Mehrfachanmeldungen zulassen.

Das SAP-System darf keine Mehrfachanmeldungen eines Benutzers vom Typ Dialog zulassen, aber in einer Session, dürfen mehrer Tabs geöffnet werden.

Motivation: In einer Session, dürfen mehrer Tabs geöffnet werden. Aber ein Benutzer darf sich nicht mehrmals am System anmelden. Dabei wird über den Parameter login/disable_multi_gui_login gesteuert, ob sich Benutzer nur einmal oder mehrfach an einem Mandanten anmelden dürfen. Mit dem Wert 1 wird die Mehrfachanmeldung deaktiviert.

Der Parameter login/multi_login_users steuert, welche Benutzer sich mehrfach an einem Mandanten anmelden können. Dieser muss leer sein.

Umsetzungsbeispiel: Ausführen der TA RZ10 und setzen folgender Parameter:

- login/disable_multi_gui_login = 1
- login/multi_login_users = <empty>

Requirement-ID: 352984ee

ID: 3.22-108/13.0

Req 109 Die Berechtigungsprüfung auf Berechtigungsobjekte muss aktiviert sein.

Der Profilparameter auth/object_disabling_active besitzt standardmäßig den Wert Y und ermöglicht dadurch über die Transaktionen SU25 bzw. AUTH_SWITCH_OBJECTS ein globales Deaktivieren von Berechtigungsobjekten. So könnte beispielsweise die Prüfung auf das Berechtigungsobjekt S_TCODE systemweit ausgeschaltet werden. Das muss verhindert werden!

Motivation: Wird das Ausschalten von Berechtigungsprüfungen aktiviert, nimmt das SAP-System keine Berechtigungsprüfungen auf die ausgewählten Berechtigungsobjekte vor. Somit kann ein unberechtigter Zugriff auf Daten erfolgen.

Umsetzungsbeispiel: 1) Ausführen der TA RZ10 und setzen des folgenden Parameters:

- auth/object_disabling_active = N

2) Das Recht zum Ausführen der TAs SU25 bzw. AUTH_SWITCH_OBJECTS muss sehr restriktiv vergeben werden (Berechtigungsobjekt S_TCODE).

Requirement-ID: eeee0592

ID: 3.22-109/13.0

Req 110 Beim Aufruf von Funktionsbausteinen von einem entfernten System oder Programm muss eine Berechtigungsprüfung stattfinden.

Beim Aufruf von RFC-Funktionsbausteinen von einem RFC-Client-Programm oder einem anderen System aus, wird im aufgerufenen System eine Berechtigungsprüfung auf das Berechtigungsobjekt S_RFC ausgeführt. Dabei wird auf den Namen der Funktionsgruppe, zu der der Funktionsbaustein gehört, geprüft.

Der Parameter `auth/rfc_authority_check` regelt, ob für Systemfunktionsbausteine eine Anmeldung und eine Berechtigungsprüfung durchgeführt wird (s. SAP Hinweis 176064).

Motivation: Unberechtigte Anmeldung am System und unberechtigter Zugriff über Funktionen und damit auf Daten müssen über automatische Systemprüfungen verhindert werden.

Umsetzungsbeispiel: Der Parameter `auth/rfc_authority_check` muss auf einen Wert 1 oder höher gesetzt werden. Mit Ausführen der TA RZ10 kann man den Parameter setzen.

Für eine höhere Sicherheit sollte der Parameter entweder auf 6 oder besser auf 9 konfiguriert werden:

- 6=Anmeldung und Berechtigungsprüfung erforderlich, jedoch nicht beim Funktionsbaustein RFC_PING. Der Funktionsbaustein RFC_PING wird ohne Anmeldung und ohne Berechtigungsprüfung ausgeführt
- 9 =Anmeldung und Berechtigungsprüfung erforderlich für alle Funktionsbausteine

Mit dem Wert 9 wird verhindert, dass ein Angreifer anonym auf RFC-Funktionen zugreifen und auf diesem Weg an zusätzliche Systeminformationen zur Angriffsvorbereitung gelangen kann. Potenzielle Funktionen hierzu wären:

RFC_PING: Einfache Prüfung der Erreichbarkeit

RFC_SYSTEM_INFO: Informationen zu dem System (OS, Datenbank, Version, Identifier)

RFC_GET_LOCAL_DESTINATIONS: Liefert alle momentan aktiven RFC-Destinationen an derselben Datenbank

RFC_GET_LOCAL_SERVERS: Informationen zu den lokalen Servern

SYSTEM_INVISIBLE_GUI: Setzt die Sichtbarkeit der aktuellen SAP GUI auf unsichtbar.

Requirement-ID: 033e7aef

ID: 3.22-110/13.0

Req 111 SAP Logintickets müssen sicher übertragen werden.

Wenn Logintickets genutzt werden, müssen diese sicher übertragen werden. Dies kann über einen Parameter gesteuert werden.

Motivation: Unberechtigter Systemzugriff über ein abgegriffenes SAP Loginticket muss verhindert werden.

Umsetzungsbeispiel: Setzen Sie folgenden Parameter mit Transaktion RZ10:

- `login/ticket_only_by_https = 1`
damit schickt der Browser das Loginticket nur bei einer bestehenden HTTPS-Verbindungen mit.

ID: 3.22-111/13.0

2.8.8. ABAP-Stack: ICM (Internet Communication Manager)

Req 112 Der Zugriff auf das SAP-System über den ICM muss durch Verwendung eines ACL-Filters, eines Authentication-Handlers und/oder dem HTTP Rewrite-Handler beschränkt werden.

Der Zugriff auf ein SAP-System muss über verschiedene Verfahren abgesichert werden. Je nach Art des Zugriffes können ACL-Filter (ACL = Access Control List), Authentication-Handler oder HTTP Rewrite-Handler zum Schutz verwenden.

det werden. Das ICM bietet verschiedene Filtermechanismen an. Es ist empfehlenswert den einfachsten Mechanismus zu verwenden, mit welchem sich die Sicherheitsanforderungen lösen lassen. Das heißt z.B. wenn ACLs ausreichen, verwenden Sie diese; die nächste Stufe wäre der Authentication Handler und die letzte Stufe schließlich der Rewrite-Handler. Dadurch können Sie eine unnötig komplexe Konfiguration vermeiden, was wiederum auch zur Sicherheit des Systems beiträgt.

URLs sollten gefiltert werden da ansonsten Informationen über die Infrastruktur und die Konfiguration ablesbar werden (s. auch SAP Note 870127).

Filtermechanismus	ACL-Dateien	Authentication Handler	HTTP Rewrite-Handler
Verwendung	Verwenden Sie ACL-Dateien, um Zugriffe auf bestimmte Client-IP-Adressen oder Client-IP-Adressbereiche einzuschränken, wenn die Beschränkung nicht vom Inhalt des HTTP-Requests (also auch nicht von der URL) abhängt, und keine HTTP-Fehlerseite gewünscht ist.	Verwenden Sie den Authentication Handler, um URL-Filter einzurichten (als „White List“ oder „Black List“). Regeln im Authentication Handler können sich ebenfalls auf bestimmte Client-IP-Adressen, oder auf IP-Adressen des Servers beziehen.	Verwenden Sie den HTTP Rewrite-Handler für Filter, die sich nicht durch ACL-Dateien oder den Authentication Handler abbilden lassen. Der Rewrite Handler ist ein mächtiges Werkzeug für unterschiedliche Filtermechanismen. Er ermöglicht es, anhand eines Regelwerks zahlreiche Daten eines HTTP-Request abzufragen und miteinander zu verknüpfen.
Dynamisches Nachladen der Konfigurationsdatei	Ist möglich	Ist möglich	Ist möglich
Positiv- bzw. Negativlisten	• Ja, beide • Auch gemischt möglich	• Ja, beide • Auch gemischt möglich	• Ja, beide • Auch gemischt möglich
Filterung auf URLs, Behandlung von Groß- und Kleinschreibung	Nein	• Ja • Standardeinstellung ist case-insensitiv • Ist konfigurierbar	• Ja • Groß- und Kleinschreibung ist pro Filterregel konfigurierbar
Security Logging	Ja	Ja	Nein
Filterung auf Client IP Adressen, inklusive Netzmasken	Ja	Ja	Ja
Parameter	icm/server_port_<xx> mit Option ACLFILE	icm/HTTP/auth_<xx> mit Option PERMFILE	icm/HTTP/mod_<xx> mit Option FILE

Motivation: Verhindern Sie unberechtigten Webzugriff auf Ihr Backendsystem, um Daten- und Vertraulichkeitsverlust vorzubeugen.

Umsetzungsbeispiel: Für welche Filtertechnik (ACL Datei, Authentication Handler und / oder HTTP Rewrite-Handler) Sie sich auch entscheiden, beachten Sie folgendes:

- Werten Sie das Connection-Log inhaltlich aus (s. das geschriebene LOGFILE über den Parameter `icm/HTTP/logging_xx`), um die relevanten URLs zu ermitteln und diese mit genauen Regeln in Ihrer URL-

Permission-Tabelle zu beschreiben (White List).

- Filtern / sperren Sie auf alle Fälle folgende Pfade in Ihrer URL-Permission-Tabelle:
 - /sap/public/icman/*
 - /sap/wdisp/info
- Wenn Sie Negativlisten (Deny-Einträge) in URL-Filtern angeben, verwenden Sie case-insensitive Filter, da der Application Server ABAP URLs case-insensitiv behandelt.
- Anstatt der Filterung der URLs kann - wenn möglich - alternativ eine Deaktivierung von URLs vorgenommen werden.

1) ACL Dateien / Filter für das System einstellen.

Die Option ACLFILE gibt die Datei an, die als Zugriffskontrollliste verwendet wird (ACL = Access Control List). Damit kann eine Zugriffskontrollliste eingerichtet werden, welche Verbindungen das ICM akzeptiert und welche nicht. Für jeden Port des ICMs kann eine eigene ACL-Datei angelegt werden. Die Zeichenkette gehorcht folgender Syntax:

```
icm/server_port_<xx> = PROT=<Protokollname>, PORT=<Port oder Servicen-ame>[, TIMEOUT=<timeout>, PROCTIMEOUT=<proctimeout>, EXTBIND=1, HOST=<Rechnername>, VCLIENT=<SSL Client Verifikation>, SSLCONF=ssl_config_<xx>, ACLFILE=<Pfad zur ACL-Datei>]
```

Beschreibung der Zugangspunkte

```
icm/server_port_0 = PROT=HTTP , PORT=0 , TIMEOUT=30 , PROCTIMEOUT=60, ACLFILE=<Pfad zur ACL-Datei>
icm/server_port_1 = PROT=HTTPS , PORT=1443, TIMEOUT=30 , PROCTIMEOUT=60, ACLFILE=<Pfad zur ACL-Datei>
```

Eine Zeile der ACL Datei muss der folgenden Syntax entsprechen:

```
<permit | deny> <ip-address> [tracelevel] [# comment]
```

Die Regeln werden sequentiell von oben beginnend geprüft („top down“). Die erste zutreffende Regel bestimmt das Ergebnis („first match“). Wenn keine Regel zutrifft, wird die Verbindung abgelehnt. Zum eindeutigen Verständnis sollte dennoch ein explizites deny (deny 0.0.0.0/0) als letzte Regel eingetragen werden.

Beispiel Inhalt ACL-Datei:

```
permit 10.1.2.0/24 # permit client network
```

```
...
```

```
deny 0.0.0.0/0 # deny the rest
```

2) Authentication Handler für das System einstellen.

Zum Schutz des Backend-Systems (AS ABAP oder AS Java) gibt es einen HTTP-Subhandler (Filter), der Requests anhand von verschiedenen Kriterien abblocken kann. Wenn der Filter aktiviert ist, wird er bei jedem HTTP(S)-Request zum ICM durchlaufen, bevor der Request zu einem anderen HTTP-Handler (Dateizugriff, Cache, Administration, Redirect) oder zum Backend-System (AS ABAP oder AS Java) geschickt wird. Sie können Requests nach den folgenden Kriterien filtern: URL, Client-IP-Adresse, Server-IP-Adresse, Benutzername/Benutzergruppe/Benutzerkennwort, Mustersuche in der URL. Um die Zugriffsbeschränkungen über Authentication Handler einzurichten, verwenden Sie den Parameter mit folgender Syntax:

```
icm/HTTP/auth_<xx> = PREFIX=<URL-Präfix>
[, PERMFILE=<permission file>, AUTHFILE=<authentication file>,
FILTER=<name>]
```

Beispiel für die Setzung des Parameters:

```
icm / HTTP / auth _ 0      =    P R E F I X = / ,      P E R M F I L E = D : \ u s r \ s a p \ W D P \ S Y S \ p r o f i l e \ p e r m _ f i l t e r . t x t
```

Die Suboption PERMFILE wird die Art des Zugriffsschutzes mit der Permission-Datei bestimmt. Definieren Sie eine Permission-Datei und legen Sie die Regeln an. Die Regeln werden in der Reihenfolge angewendet, in der sie in der Datei erscheinen, von oben nach unten, und die erste Regel, die auf die Anfrage zutrifft, wird angewendet. Es gibt eine implizite "deny all"-Regel, die am Ende der Datei zur Laufzeit hinzugefügt wird. Es wird daher empfohlen, "positive Regeln" zu verwenden (erstellen Sie nur S- oder P-Regeln, da alles andere standardmäßig verweigert wird).

Die Permission-Datei hat den folgenden Aufbau:

- Kommentarzeilen beginnen mit einem # und werden ignoriert
- Andere Zeilen haben die Form: P/D/S <URI-pattern> <USER> <GROUP> <CLIENT-IP> <SERVER-IP>
- Der Buchstabe am Zeilenanfang hat folgende Bedeutung:
 - P (Permit) lässt den Request durch
 - D (Deny) lehnt den Request ab und schickt dem Client eine entsprechende Nachricht
 - S (Secure) erlaubt für den URL-Präfix nur sichere Verbindungen (HTTPS)
 - <URI-pattern> ist der Abschnitt der URL, der im Abschnitt Cache-Key als translated path bezeichnet ist
- Für das URI-Pattern können Sie das Wildcard-Zeichen * verwenden, allerdings nur am Anfang oder am Ende des <URI-pattern> -Strings.
- Für die Client- und Server-IP-Adresse können Sie einen genauen Vergleich, das Wildcard-Zeichen * oder die Netzmasken Syntax verwenden.

Beispiel einer White List mit Erzwingung von HTTPS durch Setzung des "S" am Zeilenanfang:

```
# This is the "permission file" used by the ICM/SAP Web Dispatcher Au-
thentication handler
# webadmin and ping only from Admin-workstations <IP-Admin workstation> allowed
S /sap/bc/ping * * <IP-Admin workstation/net class> *
S /sap/wdisp/admin/* * * <IP-Admin workstation/net class> *
# Access from Webapps and WebGUI (https only) from internal IPs
S /sap/bc/webdynpro/* * * <internal IP/net class> *
S /sap/bc/gui/sap/its/webgui * * <internal IP/net class> *
# Access from Webapps and WebGUI (https only) from Internet
S /sap/bc/srt/rfc/meinwebservice * * * * *
# Everything not listed above will be denied because of the final implicit rule "D * * * * *"
```

3) HTTP Rewrite-Handler für das System einstellen

Der HTTP-Rewrite-Handler ist ein sehr leistungsfähiges Werkzeug. Sie können ihn für verschiedene Aktionen verwenden:

- Löschen, Hinzufügen und Erweitern von HTTP-Header-Feldern:
Sie können HTTP-Header-Felder löschen oder hinzufügen oder sie mit zusätzlichen Werten erweitern.
- URLs umschreiben (rewrite):
Sie können sowohl den URL-Pfad als auch den Query-String einer eingehenden HTTP-Anfrage in einen anderen, zuvor definierten URL-Pfad oder Query-String umschreiben. Sie können die Ausführung von URL-Änderungen mit einer oder mehreren Bedingungen verknüpfen.
- URLs umleiten (redirect) oder filtern:
Sie können die URL einer eingehenden HTTP-Anfrage an einen anderen Anwendungsserver oder eine andere URL umleiten. Sie können eingehende HTTP-Anfragen auch nach bestimmten Mustern filtern.
- Keine Änderungsaktion / nop-Aktion (no operation):
Modifikatoren (Optionen) wie break, restart und noescape können nach einer Änderungsaktion (Regel) auftreten. Damit Sie auch Modifikatoren ohne Definitionen verwenden können, gibt es die nop-Aktion.

Um die Zugriffsbeschränkungen über HTTP Rewrite-Handlers einzurichten, verwenden Sie den Parameter mit folgender Syntax:

```
icm/HTTP/mod_<xx> = PREFIX=<URL-Präfix>[, FILE=<Aktionsdatei>]
```

Empfehlung von SAP: Um auf allen Instanzen die gleichen Filterregeln zu haben, legen Sie die Datei mit den Filterregeln in einem globalen Verzeichnis ab und setzen den Parameter im Default-Profil DEFAULT.PFL. Die Filterregeln stehen in der Datei icm_filter_rules.txt im Verzeichnis \$(DIR_GLOBAL)/security/data:

```
icm/HTTP/mod_0 = PREFIX=/ , FILE=$(DIR_GLOBAL)/security/data/
icm_filter_rules.txt
```

Die Aktivierung des HTTP Rewrite-Handlers erfolgt über eine Aktionsdatei (durch Option FILE). Die Aktionsdatei ist eine Textdatei mit jeweils einer Aktion (Regel) pro Zeile. Kommentarzeilen beginnen mit einem Gatter (#). Die Reihenfolge in der Aktionsdatei beschreibt auch die Ausführungsreihenfolge. Sie können mehrere URL-Umschreibungen

hintereinander ausführen. Die Ausführung der HTTP-Headerfeld-Modifikationen können Sie mit einer bzw. mehreren Bedingungen verknüpfen. Anders als bei der Verwendung des Authentication Handlers wird am Ende keine implizite "deny all"-Regel hinzugefügt. Beachten Sie zudem, dass eine "Negativliste" erstellt werden muss, statt einer "Positivliste" wie im Authentication Handler.

Verwenden Sie die folgende Syntax, um eine Regel in der Aktionsdatei zu definieren (s. SAP Help):
<operation> <pattern> <dest> [<option>]

Erzwingen Sie HTTPS durch Nutzung des Rewrite-Handler:

Um die Verwendung von unverschlüsseltem HTTP zu verhindern, müssen Sie das Protokoll des Rewrite-Handlers von HTTP auf HTTPS ändern. Dies verhindert Fehlermeldungen im Browser, wenn Benutzer versehentlich mit einer HTTP-URL auf das System zugreifen. Beachten Sie, dass nicht alle HTTP-Clients diese Umleitung befolgen. Während die Redirect-Konfiguration sicherstellt, dass kein HTTP-Zugriff auf das System möglich ist, ist es möglich, dass einzelne Benutzer des Systems, z. B. Web-Service-Endpunkte, von HTTP auf HTTPS umgestellt werden müssen.

Beispiel für strikte Nutzung von HTTPS:

```
# We allow access to the WEB GUI, but only through HTTPS
# The indentation is optional
if %{SERVER_PROTOCOL} !stricmp "HTTPS" [and]
if %{PATH} regimatch ^/sap/bc/gui/sap/its/webgui(.*?)
RegForbiddenURL ^/sap/bc/gui/sap/its/webgui(.*?) - [break]
```

Beispiel Verbot Ausführen eines Ping:

```
# We deny access to the "ping" service
RegForbiddenURL ^/sap/public/ping(.*?) - [break]
```

ID: 3.22-112/13.0

Req 113 Der ICM Server (Port) muss sicher konfiguriert werden.

Der ICM muss so konfiguriert sein, dass nur sichere Protokolle Zugriff haben. Dieses geschieht durch das Setzen des Parameters `icm/server_port_<xx>` zur Nutzung des HTTPS-Protokolls. Mit den Optionen PROT und PORT spezifizieren Sie das Protokoll und den dazugehörigen Port durch die Portnummer oder den Servicenamen. Auf einen Port kann genau nur ein Service gebunden werden.

Motivation: Eine unverschlüsselte Kommunikation erlaubt das Mitlesen dieser mit gefährlichem Potenzial. User-IDs mit den zugehörigen Passwörtern können abgegriffen werden und für eine unberechtigte Anmeldung missbraucht werden, welches wiederum einen unberechtigten Datenzugriff zur Folge hat.

Umsetzungsbeispiel: Durch Setzung des Parameters `icm/server_port_<xx>` wird die Nutzung von HTTPS erzwungen und Zugriffe können kontrolliert werden. Dieser Profilparameter des ICMs konfiguriert den Server Port. Die Zeichenkette gehorcht folgender Syntax:

```
icm/server_port_<xx> = PROT=<Protokollname>, PORT=<Port oder
Servicename>[, TIMEOUT=<timeout>, PROCTIMEOUT=<proctimeout>,
EXTBIND=1, HOST=<Rechnername>, VCLIENT=<SSL Client Verifikation>,
SSLCONFIG=ssl_config_<xx>, ACLFILE=<Pfad zur ACL-Datei>]
```

Beschreibung der Zugangspunkte (_0 und _1 sind im AS ABAP vordefinierte Werte für ausgehende Verbindungen, _2 ist ein Beispiel für eine eingehende HTTPS Verbindung, _3 ist ein Beispiel für eine ausgehende HTTPS Verbindung):

```
icm/server_port_0 = PROT=HTTP,PORT=0,TIMEOUT=30,PROCTIMEOUT=60
icm/server_port_1 = PROT=SMTP,PORT=0,TIMEOUT=120,PROCTIMEOUT=120
icm/server_port_2 = PROT=HTTPS , PORT=443, TIMEOUT=30 , PROCTI-
MEOUT=60, ACLFILE=<Pfad zur ACL-Datei>
icm/server_port_3 = PROT=HTTPS,PORT=0,TIMEOUT=30,PROCTIMEOUT=60,
ACLFILE=<Pfad zur ACL-Datei>
```

Bitte beachten Sie, dass der Wert 0 bei PORTS bedeutet, dass für das angegebene Protokoll kein Port für eingehende Verbindungen geöffnet wird. Ports für eingehende Verbindungen müssen aus Sicherheitsgründen explizit konfiguriert werden.

Achtung: Mit dem ICM werden von SAP vorgelegte Werte ausgeliefert. Diese müssen auf Gültigkeit überprüft werden und ggf. mit " deaktiviert werden z. B.:

```
icm/server_port_0 ="  
icm/server_port_1 ="
```

ID: 3.22-113/13.0

Req 114 Die Administration des ICMs über die webbasierte Administrationsoberfläche muss gesichert sein.

Sie müssen den Parameter `icm/HTTP/admin_<xx>` setzen, wenn Sie den ICM über die webbasierte Administrationsoberfläche (Browser) überwachen und administrieren wollen.

Motivation: Die Weboberfläche zur Administration des ICMs darf nicht für alle Benutzer erreichbar sein. Diese muss geschützt sein, um einen unberechtigten Zugriff zu verhindern.

Ebenfalls ist die Anzeige von Systeminformationen zu verhindern, da diese Informationen zu einem Angriff auf das System missbraucht werden könnten.

Umsetzungsbeispiel: Der Parameter `icm/HTTP/admin_<xx>` hat folgende Syntax:

```
icm/HTTP/admin_<xx> = PREFIX=<URL-Präfix>, DOCROOT=<Wurzelverzeichnis  
der Administrationsdateien> [, AUTHFILE=<Dateiname der Authentifizie-  
rungsdatei>, PORT=<TCP/IP-Port, auf dem Adminrequests akzeptiert wer-  
den>, HOST=<Rechnername oder IP-Adresse, auf dem Admin-Requests akzep-  
tiert werden>, CLIENTHOST=<Rechnername oder IP Adresse, von dem Admin-  
requests akzeptiert werden>, ALLOWPUB=<Wert> ]
```

Beispiel zur Setzung des Parameters im Profil:

```
i c m / H T T P / a d m i n _ 0 = P R E -  
F I X = / s a p / a d m i n , D O C R O O T = $ ( D I R _ I C M A N _ R O O T /  
a d m i n , P O R T = 1 4 4 3 , H O S T = l o c a l h o s t ; < I P H o s t  
ICM>, C L I E N T H O S T = l o c a l h o s t ; < H o s t n a m e n v o n A d m i n i s t r a t o r C l i -  
e n t s > , [ A L L O W P U B = F A L S E ]
```

Setzung des Wertes für PREFIX: URL-Präfix, für den dieser HTTP-Subhandler aufgerufen werden soll. Im Standard wird der URL-Präfix für die Verwaltung auf `/sap/admin` gesetzt.

Setzung des Wertes für DOCROOT: Wurzelverzeichnis der Administrationsdateien.

Setzung des Wertes für AUTHFILE: Dateiname der Authentifizierungsdatei (Defaultwert: `icmauth.txt`), in der die Hash-Werte der Passwörter für die Admin-Benutzer liegen. Wenn keine Benutzerauthentifizierung erfolgen muss (z.B. weil diese im Authorisation Handler bereits erfolgt), so kann dies mit der Angabe von `AUTHFILE=none` erreicht werden.

Setzung des Wertes für PORT (MUSS): Im Default werden Admin-Requests auf allen lokalen TCP/IP-Ports angenommen. Das ist nicht erlaubt. Eine Einschränkung auf einen TCP/IP-Port, der nur lokal erreichbar ist und auf denen Admin-Requests angenommen werden, muss umgesetzt werden. So kann die Verwendung von HTTPS leicht umgesetzt werden.

! Hinweis: Dieser Port muss über den Parameter `icm/server_port_<xx>` angegeben worden sein. Dafür muss ein HTTPS-Port eingerichtet werden, sonst werden die Administrator Passwörter bei der Anmeldung im Klartext übertragen. Unsichere Protokolle sind nicht erlaubt!

Beispiel:

```
icm/server_port_<xx>=.... PROT=HTTPS, PORT=1443, ....
```

Setzung der Werte für HOST (MUSS): Wird hier nichts angegeben, werden Admin-Requests von allen Rechnernamen akzeptiert. Dies ist nicht erlaubt. Die Einschränkung des Zugriffs auf den lokalen Host muss umgesetzt werden, damit können nur die lokalen Benutzer die webbasierte Oberfläche benutzen. Geben Sie dafür den Wert `localhost` oder `127.0.0.1` und die `<IP des ICM Hosts>` an.

Setzung der Werte für CLIENTHOST (optional): Wird hier nichts angegeben, werden Admin-Requests von jedem Clientrechner akzeptiert. Die Administration muss auf bestimmte Clientrechner (Rechner der ICM Administratoren) eingeschränkt werden. Geben Sie dafür den Wert `localhost` und die `<Rechnernamen oder IP-Adressen der Administrato-`

ren> ein.

Setzung des Wertes für ALLOWPUB: Ist ALLOWPUB=TRUE gesetzt, so werden unter dem Pfad "public/index.html" öffentliche und lesende Zugriffe auf bestimmte Administrationsseiten ohne Anmeldung zugelassen (z.B. "Monitor", "Aktive Services", "Core Thread Status", "Hostname Buffer", "Release Information" und "MPI Status"). Der Zugriff auf diese Seiten ohne Anmeldung muss verhindert werden. Auch wenn der Zugriff mit den Subparametern HOST und CLIENTHOST eingeschränkt ist (s. oben), dürfen Systeminformationen, die für einen Angriff auf das System genutzt werden könnten, nur über eine Anmeldung am System einsehbar sein. ALLOWPUB=FALSE muss gesetzt werden.

Für weitere Informationen s. SAP Note 870127 und die SAP Online Hilfe.

ID: 3.22-114/13.0

Req 115 Die Autorisierungsdatei "icmauth.txt" muss vor unberechtigten Zugriffen geschützt sein.

In der Autorisierungsdatei "icmauth.txt" muss vor unberechtigten Zugriffen geschützt sein, da Benutzer und ihre Gruppenzugehörigkeit im Klartext und Passwörter (verschlüsselt) enthalten sind. Der Dateipfad ist entsprechend den Konventionen des zugrunde liegenden Betriebssystems.

Motivation: Das "icmauth.txt"-File ist gegen Fremdzugriffe zu schützen, um den Einfallsvektor für einen Angriff möglichst klein zu halten. Das Verschlüsselte Passwort ist leicht zu knacken und für einen unberechtigten Angriff zu nutzen.

Umsetzungsbeispiel: Diese Datei muss aus Sicherheitsgründen nur für den Administrations-Benutzer des SAP-Systems (<SID>adm) lesbar und änderbar sein. Setzen von restriktiven Zugriffsrechten auf die Autorisierungsdatei icmauth.txt mit folgenden Befehlen (UNIX):

```
chmod 700 <autorisierungsdatei>
```

Der Ablageort der Datei sowie deren Namen ist im Parameter `icm/authfile` spezifiziert.

ID: 3.22-115/13.0

Req 116 Der Zugriff auf das ICM Administrationsverzeichnis muss restriktiv gesetzt sein.

Das Administrationsverzeichnis des ICM muss restriktiv eingestellt sein. Das Administrationsverzeichnis ist im Standard mit <DIR_ICMAN_ROOT>/admin definiert und enthält die Dateien für die Administration über die Weboberfläche.

Motivation: In dem Administrationsverzeichnis befinden sich Daten (z. B. Datei icmauth.txt) mit einem hohen Schutzbedarf.

Umsetzungsbeispiel: Anpassen der Zugriffsrechte für das Administrationsverzeichnis mit folgenden Befehlen:

```
chown <sid>adm:sapsys <Verzeichnisname>
chmod 750 <Verzeichnisname> oder wenn möglich restriktiver =>
700
```

ID: 3.22-116/13.0

Req 117 Die Web Administration erfolgt personifiziert und nicht über den User "icmadm".

Um den ICM von einem Browser aus zu überwachen und zu administrieren, benötigen Sie einen Benutzer.

Der User icmadm ist der Standard User für die Webadministrationsoberfläche und wird in der Datei icmauth.txt hinterlegt, wenn die Einrichtung des Administrationsuser durchgeführt wird.

Für die nachvollziehbare Administration werden personifizierte Administratoren angelegt.

Motivation: Unberechtigter Administrationszugriff kann zur Dateneinsicht und -änderung mit einem Systemausfall führen.

Umsetzungsbeispiel: Bei der Einrichtung der Administrationsuser müssen folgende Schritte ausgeführt werden:

1. Legen Sie als <sid>adm den icmadm Administrationsbenutzer mit icmon an. Hierdurch wird die Berechtigungsdatei icmauth.txt generiert.
2. Legen Sie personalisierte Administrationsuser mit icmon an.
3. Probieren Sie den Zugriff mit den personalisierten Usern.
4. Ändern Sie das Passwort des Standardusers icmadm gemäß den Passwortrichtlinien.

ID: 3.22-117/13.0

Req 118 Der Zugriff auf das Kommando "icmbnd" muss restriktive vergeben werden.

Der Zugriff auf das Kommando "icmbnd" muss restriktive vergeben werden. Im Standard ordnet der ICM die zu benutzenden Ports sich selbst zu (empfohlen). Um die privilegierten Ports mit Portnummer 0 bis einschließlich 1023 („well known ports“) an den ICM binden zu lassen, muss das externe Bindeprogramm icmbnd genutzt werden. Durch Setzen der Option EXTBIND=1 bei der Parameterdefinition von icm/server_port_<xx> wird das Programm icmbnd vom ICM aufgerufen, um die Ports < 1023 zu setzen.

Das Programm icmbnd kann auch direkt aufgerufen werden, um Ports zu konfigurieren (nicht empfohlen).

Das Programm ist unter Pfad, der mit folgenden Parametern im Profil definiert wird, zu finden:

- exe/icmbnd
- DIR_EXECUTABL

Motivation: Unberechtigtes Ausführen des Programmes icmbnd kann zu einer Fehlkonfiguration führen. Es können Angriffe über den ICM auf die nachgelagerten Systeme ausgeführt werden oder auch Programme, die in der Regel nicht über den Ports < 1023 erreichbar sind, erreichbar gebracht werden.

Umsetzungsbeispiel: Setzen von eingeschränkten Zugriffsrechten auf die Datei icmbnd:

```
chown root:sapsys icmbnd
```

```
chmod 4750 icmbnd
```

ID: 3.22-118/13.0

Req 119 Es dürfen keine Informationen zum System in "Fehlerseiten" ausgegeben werden.

Die Standardfehlerseiten müssen durch benutzerspezifisch definierte Fehlerseiten ersetzt werden.

Benutzerspezifisch definierte Fehlerseiten dürfen keine Versionsinformationen des Webserver und verwendeter Module/Erweiterungen enthalten. Die Fehlermeldungen dürfen keine internen Informationen, wie z. B. interne Servernamen, Fehlercodes usw. enthalten.

Aus Sicherheitsgründen sollten Information über das System und oder die Anwendung nicht an den Client weitergegeben (sichtbar) werden.

Sie können über den Parameter is/HTTP/show_detailed_errors einstellen, ob die Fehlerseite mit oder ohne Details an den Client geschickt werden soll. Setzung:

- TRUE: Die Fehlerseite enthält Detailinformationen über den Grund des Fehlers (Name des Applikationsservers, Datum, Modul, Zeile, Komponente, Beschreibung des Fehlers).
- FALSE: Die Fehlerseite enthält keine Detailinformationen über den Grunde des Fehlers. Bei manchen Fehlern wird ein Error Tag (z.B. {00060025}) ausgegeben, mit dem Sie in der jeweiligen Trace-Datei (dev_icm bzw. dev_webdisp) die Fehlerdetails finden können.

Motivation: Es dürfen keine Informationen zum System in "Fehlerseiten" ausgegeben werden, die zu einem Angriff auf das System ausgenutzt werden könnten.

Umsetzungsbeispiel: Der Parameter `is/HTTP/show_detailed_errors` sollte auf "false" gesetzt werden. Überprüfung des ICM Profils:

`is/HTTP/show_detailed_errors = FALSE`

Weitere Informationen dazu finden Sie in der SAP Online Hilfe im Kapitel "Fehlerbehandlung durch den SAP Web Dispatcher".

ID: 3.22-119/13.0

Req 120 Die Informationen zum Server in der HTTP Antwort an den Client müssen limitiert werden.

Der Parameter steuert, ob das Server-Headerfeld bei HTTP-Antworten des Servers an den Client eingefügt werden soll oder nicht. Das Server-Header-Feld umfasst die Werte der Parameter `is/server_name` und `is/server_version`. Aus Sicherheitsgründen empfiehlt SAP diese Informationen nicht an den Client weiterzugeben.

Motivation: Es dürfen keine Informationen an den Client weitergegeben werden, die zu einem Angriff ausgenutzt werden könnten.

Umsetzungsbeispiel: Setzen Sie dafür den Parameter `is/HTTP/show_server_header` wie folgt:

`is/HTTP/show_server_header = FALSE`

ID: 3.22-120/13.0

Req 121 Das Sicherheitslogging für den ICM muss aktiviert sein.

Mit dem Parameter `icm/security_log` können Sie die Ausgabe des Sicherheits-Protokolls aus dem ICM und SAP Web Dispatcher steuern.

Administratoren können das Sicherheitsprotokoll (`icm/security_log`) verwenden, um mögliche unberechtigte Zugriffe auf das System zu erkennen. Die folgenden Unregelmäßigkeiten werden protokolliert:

- Daten mit ungültiger Syntax
- Versuchter Zugriff auf Objekte, die nicht existieren (NOT found)
- Zugriff auf Objekte, die aufgrund von Filterregeln nicht erlaubt sind (permission denied)
- Anmeldefehler bei der Webadministration (im ICM und Web Dispatcher)

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Umsetzungsbeispiel: Setzen Sie den Parameter `icm/security_log` für die Aktivierung der Security Protokollierung.

Syntax des Parameters:

`icm/security_log = LOGFILE=<file name>, LEVEL=<security level>, MAXSIZEKB=<max size in KB>, SWITCHTF=<options>, FILEWRAP=on`

LOGFILE: Name der Datei. Durch Zeitangaben wird der Name eindeutig.

LEVEL: Gibt den Grad (1 - 3) der Protokollierung an. Mit Level 1 wird nur der Grund für den Eintrag protokolliert. Mit Level 2 werden zusätzliche Informationen über den Zustand der Verbindung und der Beginn der Daten, die den Eintrag verursachen, protokolliert (Standard, wenn LEVEL nicht angegeben wird). Mit Level 3 werden alle Daten, die den Eintrag verursachen, protokolliert.

MAXSIZEKB: Gibt die Größe der Protokolldatei in KB an. Wird diese Größe überschritten, wird die aktuelle Datei geschlossen und eine neue Datei (mit einem neuen Namen) geöffnet. Durch Definition von LOGFILE wird der Name eindeutig und damit wird ein Überschreiben verhindert.

SWITCHTF: Eine neue Protokolldatei kann nicht nur angelegt werden, wenn die Datei eine bestimmte Größe erreicht hat, sondern auch, wenn sich die Zeitdaten ändern (eine neue Datei stündlich, täglich oder monatlich möglich).

FILEWRAP: Wenn FILEWRAP=on aktiv ist, wird bei jedem Öffnen einer neuen Datei die bestehende Protokolldatei zurückgesetzt und überschrieben (!). Es gibt also immer nur eine Protokolldatei mit den aktuellen Protokolldaten. Wenn Sie diese Option weglassen, wird nach Überschreiten der Größe eine neue Datei geschrieben (siehe MAXSIZEKB und SWITCHTF).

Beispiel Setzung des Parameters (Option FILEWRAP wird **nicht** gesetzt!):

```
icm/security_log = LOGFILE=dev_icm_seclog-%d-%m-%y_%h:%t:%s, LEVEL=2,
MAXSIZEKB=10000, SWITCHTF=day
```

ID: 3.22-121/13.0

Req 122 Das HTTP Logging muss aktiviert sein.

Mit den Parametern icm/HTTP/logging_<xx> (eingehende Requests) und icm/HTTP/logging_client_<xx> (ausgehende Requests) können Sie die Ausgabe des Verbindungs-Protokolls aus dem ICM und SAP Web Dispatcher steuern und protokollieren.

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Umsetzungsbeispiel: Setzen Sie folgende zwei Parameter:

1) Protokollierung eingehender Verbindungen

Syntax des Parameters:

```
icm/HTTP/logging_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file name>
[, LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBytes>,
SWITCHTF=<options>, FILEWRAP=on]
```

PREFIX: URL-Präfix, der für diesen HTTP-Subhandler aufgerufen wird (z. B. /).

LOGFILE: Name der Ausgabedatei im Dateisystem. Der ICM / Web Dispatcher prüft bei jedem Neustart, ob die als Logdatei angegebene Datei existiert. Ist dies der Fall, schreibt er weiter in diese Datei, ist dies nicht der Fall, legt er eine neue Datei an. An den eigentlichen Dateinamen können Sie einen anderen Zeitstempel anhängen, welches die Datei eindeutig macht.

LOGFORMAT: Es gibt die verschiedene Formate für Protokolldateien. Sie können das "NCSA Combined Log Format", das anderen Webservern bekannt ist, durch Verwendung der folgenden Formatzeichenfolge reproduzieren: %h %l %u %t "%r" %s %b "%{referer}i" "%{user-agent}i"

FILTER: Die Filtereigenschaft legt fest, dass eine HTTP-Anfrage nur dann protokolliert wird, wenn ein bestimmtes Header-Feld (z. B. ein HTTP-Header-Feld) in der Anfrage oder Antwort enthalten ist.

MAXSIZEKB: Gibt die Größe der Protokolldatei in KB an. Wird diese Größe überschritten, wird die aktuelle Datei geschlossen und eine neue Datei (mit einem neuen Namen) geöffnet. Durch Definition von LOGFILE wird der Name eindeutig und damit wird ein Überschreiben verhindert.

SWITCHTF: Eine neue Protokolldatei kann nicht nur angelegt werden, wenn die Datei eine bestimmte Größe erreicht hat, sondern auch, wenn sich die Zeitdaten ändern (eine neue Datei stündlich, täglich oder monatlich möglich).

FILEWRAP: Wenn FILEWRAP=on aktiv ist, wird bei jedem Öffnen einer neuen Datei die bestehende Protokolldatei zurückgesetzt und überschrieben (!). Es gibt also immer nur eine Protokolldatei mit den aktuellen Protokolldaten. Wenn Sie diese Option weglassen, wird nach Überschreiten der Größe eine neue Datei geschrieben (siehe MAXSIZEKB und SWITCHTF).

Beispiel Setzung des Parameters (Option FILEWRAP wird nicht gesetzt!):

```
icm/HTTP/logging_0 = PREFIX=/, LOGFILE=
incoming_icm_log-%d-%m-%y_%h:%t:%s, LOGFORMAT=%h %l %u %t "%r" %s
%b "%{referer}i" "%{user-agent}i", MAXSIZEKB=10000, SWITCHTF=day
```

2) Protokollierung ausgehender Verbindungen

Syntax des Parameters:

```
icm/HTTP/logging_client_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file
name> [, LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBy-
tes>, SWITCHTF=<options>, FILEWRAP=on]
```

Die Optionen sind dieselben wie bei `icm/HTTP/logging_<xx>`.

Beispiel Setzung des Parameters (Option FILEWRAP wird nicht gesetzt!):

```
icm/HTTP/logging_client_<xx> = PREFIX=/, LOGFILE=outgoing_icm_log-%d-%m-%y_%h:%t:%s, LOGFORMAT=%h %l %u %t "%r" %s %b "%{referer}i" "%{user-agent}i", MAXSIZEKB=10000, SWITCHTF=day
```

Achtung bezüglich der Parameter `icm/HTTP/logging_<xx>` und `icm/HTTP/logging_client_<xx>`: Die Protokolldateien für eingehende und ausgehende Anfragen müssen unterschiedliche Namen haben, d.h. Sie können nicht beide Richtungen in eine Datei schreiben.

ID: 3.22-122/13.0

2.9. Absicherung des Java-Stacks

2.9.1. Java-Stack: Betriebssystemverzeichnis

Req 123	Verzeichnisse und Dateien, in denen Konfigurationsdateien der Java-Stack-Installation auf dem Server abgelegt werden, müssen mit restriktiven Zugriffsberechtigungen ausgestattet werden.
---------	---

Alle Verzeichnisse und Dateien unterhalb von `JC<nr>` müssen dem `SIDadm` gehören. Sie dürfen von anderen Betriebssystem-Benutzern nicht überschrieben oder geändert werden. Bei Double-Stack Systemen (ABAP+JAVA = nur noch in Ausnahmefällen zulässig) ist zu beachten, dass es sich beim JAVA-Stack um eine Add-In Installation handelt. Der Java-Stack läuft unter der gleichen SID wie der ABAP-Stack, das Installationsverzeichnis liegt im Verzeichnis des ABAP-Stack.

Motivation: Wenn der Zugriff auf die Ressourcen des Java-Stacks nicht eingeschränkt und geschützt wird, sind Angreifer möglicherweise in der Lage, ohne Befugnis auf die Funktionalität und die Daten zuzugreifen. Unberechtigter Zugriff auf Ressourcen in Verbindung mit Schwächen bei der Umsetzung der Autorisierung auf Anwendungsebene (z. B. keine ausdrücklichen Autorisierungsprüfungen etc.) kann dann zu einer Kompromittierung des gesamten Java-Stacks führen.

Umsetzungsbeispiel: in das Verzeichnis wechseln

```
cd JC
```

`find . \! -user sidadm | xargs ls -lad` alle Ergebniseinträge müssen auf `sidadm` geändert werden:

```
chown sidadm <Datei/Verzeichnisliste>
```

```
find . \( -type f -o -type d \) -perm +022 | xargs ls -lad
```

Beispieloutput mit fehlerhaften Schreibberechtigungen:

- `-rwxrwxr-x 1 sidadm sapsys 625826 2010-09-11 00:38 ./exe/libsapcsa.so.old`
- `-rwxrwxr-x 1 sidadm sapsys 7062188 2010-09-14 21:34 ./exe/sapstartsrv.old`
- `drwxrwxr-x 6 sidadm sapsys 4096 2009-02-24 17:01 ./igs`
- `drwxr-xrwx 2 sidadm sapsys 4096 2010-10-09 20:35 ./log`

Im Beispiel müssten dann folgende Kommandos zum Entfernen abgesetzt werden:

```
chmod o-w log
```

```
chmod g-w igs exe/libsapcsa.so.old exe/sapstartsrv.old
```

Falls das `find`-Kommando nur eine Ausgabe wie folgt liefert, bedeutet das lediglich, dass keine Einträge mit fehlerhaften Schreibberechtigungen gefunden wurden und kein Handlungsbedarf besteht:

```
drwxr-xr-x 7 sidadm sapsys 4096 2010-10-09 20:44
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-123/13.0

Req 124 Der Zugriff auf die SAP-Installationslogs muss stark eingeschränkt sein.

Für den ABAP- und Java-Stack werden die Installationsdateien (Logs) auf dem Standard SAP Betriebsverzeichnis /usr/sap/<SAPSID>/SYS gespeichert. Dies ist der Defaultwert von der Variablen DIR_INSTALL, welche den Ort für das SAP Installationsverzeichnis angibt.

Motivation: In Installationslogs befinden sich Daten mit einem hohen Schutzbedarf. Nach Installationsabschluss darf der Zugriff nur durch berechtigte User erfolgen.

Umsetzungsbeispiel: Ändern der Zugriffsrechte für die Installationslogs mit folgenden Befehlen:

```
chown root <Dateiname>
chmod 700 <Dateiname>
```

Wenn möglich, können die Installationslogs auch gelöscht werden.

ID: 3.22-124/13.0

Req 125 Das HOME-Verzeichnis des "<sapsid>adm" Benutzers und dessen Inhalt müssen vor nicht autorisierten Zugriff geschützt werden.

Das HOME-Verzeichnis des "<sapsid>adm" Benutzers muss vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Motivation: Im HOME-Verzeichnis befinden sich besonders schutzbedürftige Informationen.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigung	OS-User/OS-Gruppe
<home Verzeichnis von <sapsid>adm>	700	<sapsid>adm/sapsys
<home Verzeichnis von <sapsid>adm>/*	700	<sapsid>adm/sapsys

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys <home Verzeichnis von <sapsid>adm>
chmod 700 <home Verzeichnis von <sapsid>adm>
```

ID: 3.22-125/13.0

Req 126 Das Verzeichnis "/sapmnt/<SAPSID>" mit seinen Unterverzeichnissen und Inhalten müssen vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Das Verzeichnis "/sapmnt/<SAPSID>" mit seinen Unterverzeichnissen muss vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Motivation: Unter dem Verzeichnis "/sapmnt/<SAPSID>" befinden sich einige Verzeichnisse und Dateien die für die Applikation essentiell sind. In der Regel liegen die Dateien auf einem globalen Host und werden mittels NFS, SMB verteilt. Ein Zugriff auf Dateien und Verzeichnisse innerhalb des Verzeichnisses können dazu führen, dass Information und Daten verändert werden können oder die Applikation sich fehlerhaft verhält.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigungen	OS-User/OS-Gruppe	Anmerkungen
(1) /sapmnt/<SAPSID>/exe (ohne "Inplace" Datenbank)	(2)700	<sapsid>adm/sapsys	Setzung des s-Bits für die Gruppe. Wert (2)700 für ABAP und JavaSysteme ohne "Inplace" Datenbank (SAP HANA, SAP Web Dispatcher). Nur der User <sid>adm muss auf die Dateien zugreifen können.
(2) /sapmnt/<SAPSID>/exe (mit "Inplace" Datenbank z. B. Oracle)	(2)750	<sapsid>adm/dba	Setzung des s-Bits für die Gruppe. Wert (2)750 für ABAP und Java Systeme mit "Inplace" Datenbank wie z. B. Oracle. Nur der User <sid>adm muss auf die Dateien zugreifen können. Wenn eine Oracle Datenbank genutzt wird, liegen hier die ebenfalls die brtools auf die der User ora<sid> zugreifen können muss.
/sapmnt/<SAPSID>/global	700	<sapsid>adm/sapsys	
/sapmnt/<SAPSID>/profile	755	<sapsid>adm/sapsys	

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>  
chmod <OS-Berechtigung> <Verzeichnisname>
```

ID: 3.22-126/13.0

Req 127 Das Verzeichnis "/usr/sap/<SAPSID>" und dessen Inhalt müssen vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Unter dem Verzeichnis "/usr/sap/<SAPSID>" befinden sich Verzeichnisse und Dateien, die für die Applikation essentiell sind. Diese Dateien liegen auf dem lokalen Host.

Motivation: Ein Zugriff auf Dateien und Verzeichnisse innerhalb des Verzeichnisses können dazu führen, dass Information und Daten verändert werden können oder die Applikation sich fehlerhaft verhält.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigungen	OS-User/OS-Gruppe	Anmerkungen
/usr/sap/<SAPSID>	751	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/<Instance ID>	755	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/<Instance ID>/sec	700	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/SYS	755	<sapsid>adm/sapsys	
/usr/sap/<SAPSID>/SYS/*	755	<sapsid>adm/sapsys	
(1) /usr/sap/<SAPSID>/SYS/exe (ohne "Inplace" Datenbank)	(2)700	<sapsid>adm/sapsys	Setzung des s-Bits für die Gruppe. Wert (2)700 für ABAP und JavaSysteme ohne "Inplace" Datenbank (SAP HANA, SAP Web Dispatcher). Nur der User <sid>adm muss auf die Dateien zugreifen können.
(2) /usr/sap/<SAPSID>/SYS/exe (mit "Inplace" Datenbank z. B. Oracle)	(2)750	<sapsid>adm/dba	Setzung des s-Bits für die Gruppe. Wert (2)750 für ABAP und Java Systeme mit "Inplace" Datenbank wie z. B. Oracle. Nur der User <sid>adm muss auf die Dateien zugreifen können. Wenn eine Oracle Datenbank genutzt wird, liegen hier die ebenfalls die brtools auf die der User ora<sid> zugreifen können muss.

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
```

```
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys /usr/sap/<SAPSID>
```

```
chmod 751 /usr/sap/<SAPSID>
```

ID: 3.22-127/13.0

Req 128	Das SAP-Transportverzeichnis und dessen Inhalt müssen vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.
---------	---

Das SAP-Transportverzeichnis muss vor nicht autorisierten Zugriff geschützt sein. SMB oder NFS freigaben können vorhanden sein.

Motivation: Das Transportverzeichnis ist für Datenaustausch zwischen den Systemen eines Verbundes (Abnahme-, Test-, Produktion-, ...) zuständig und kann schutzbedürftige Daten (Programme, neuen Benutzer, ...) enthalten. Unbefugte Änderungen an Dateien im SAP-Transportverzeichnis können schwerwiegende Folgen für das System haben.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigungen	OS-User/OS-Gruppe	Anmerkungen
/usr/sap/trans	(2)770	<sapsid>adm/sapsys	Setzung des s-Bits für die Gruppe. (2)770 wenn möglich, wenn nicht (2)775.
/usr/sap/trans/*	770	<sapsid>adm/sapsys	
/usr/sap/trans/.sapconf	775	<sapsid>adm/sapsys	

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys /usr/sap/trans
chmod 775 /usr/sap/trans
```

ID: 3.22-128/13.0

Req 129	Auf Betriebssystemebene müssen SAP Tools (Programme) vor unberechtigten Zugriff geschützt werden.
---------	---

Es muss sichergestellt sein, dass für die SAP-Tools die auf Betriebssystemebene benötigt werden, eigene Benutzer, Benutzergruppen und Verzeichnisse eingerichtet werden, um den unberechtigten Zugriff zu verhindern.

Motivation: Es muss verhindert werden, dass einen unberechtigten Zugriff, Dateien und Dateiinhalte verändert. Die Änderungen können zu einem fehlerhaften Verhalten und/oder zu einem Ausfall führen.

Umsetzungsbeispiel: Es müssen mindestens die folgenden Tools (Programme) vor unberechtigten Zugriff geschützt werden:

Toolname	Verzeichnis	Berechtigung	Besitzer/Gruppe	Funktion
sapevt	/usr/sap/<SAPSID>/SYS/exe/run	750	<sapsid>adm/sapsys	Netweaver Events auf Betriebssystemebene starten

saposccl	/usr/sap/<SAPSID>/SYS/exe/run	4710	root/sapsys	Sammelt Informationen vom Betriebssystem
----------	-------------------------------	------	-------------	--

ID: 3.22-129/13.0

2.9.2. Java-Stack: Kommunikation

Req 130 Für die Administration des AS Java benötigte Tools dürfen nur aus dem AdminLAN genutzt werden.

Um den Angriffsvektor zu verkleinern, ist es notwendig, dass Administrations-Werkzeuge und die dazugehörigen Ports nur über das AdminLAN (eine separate Netzwerkschnittstelle, die vom Office-LAN getrennt ist) erreichbar sind. Tools die Klartextprotokolle nutzen sind auch im geschützten AdminLAN nicht mehr zulässig (z. B. der Shell-Administrator nutzt telnet). Allerdings kann der Shell Administrator local aufgerufen werden.

Motivation: Administrative Zugriffe auf Systeme erfolgen in der Regel mit höheren Privilegien. Diese Trennung ist erforderlich, um verschiedene Arten von Zugriffen und Datenströmen voneinander zu separieren.

Umsetzungsbeispiel: Administrative Webzugriffe über den Netweaver Administrator (NWA) werden über einen Sprungserver in das AdminLAN realisiert.

Für das Tool Shell Administrator ist aus Sicherheitsgründen der Zugriff auf den AS Java über Telnet auf den Host 127.0.0.1 (localhost) beschränkt.

ID: 3.22-130/13.0

2.9.3. Java-Stack: Benutzer und Authentifikation

Req 131 Der Benutzer administrator, J2EE_ADMIN_<SID> oder J2EE_ADMIN muss vor nicht autorisierten Zugriffen geschützt sein.

Im AS Java darf der administrator (Datenquelle Datenbank), J2EE_ADMIN_<SID> (Datenquelle ABAP) oder J2EE_ADMIN Benutzer nicht gelöscht werden, da dieser Benutzer für bestimmte Anwendungen auf dem AS Java für Verwaltungs- und Installationsaufgaben benötigt wird z. B. Software Deployment und Undeployment. Der User muss gesperrt werden. Für die täglichen administrativen Aufgaben müssen personifizierte Benutzer mit Administrationsberechtigungen für die AS Java Administratoren angelegt.

Für die bestimmten Anwendungen auf dem AS Java für Verwaltungs- und Installationsaufgaben z. B. Software Deployment und Undeployment muss der der Default Administrationsbenutzer von den personifizierten Administratoren entsperrt und nach der Durchführung der Tasks wieder gesperrt werden.

Motivation: Der AS Java Default Administrator Benutzer ist zwingend notwendig und wird für Verwaltungs- und Installationsaufgaben zwingend benötigt. Allerdings muss ein unberechtigter Zugriff durch diesen technischen Benutzer verhindert werden.

Umsetzungsbeispiel: Gehen Sie wie folgt vor (s. SAP Hinweis 2526747):

- Legen Sie für die AS Java Administratoren personifizierte Benutzer an
- Setzen Sie das Passwort gemäß der DTAG Sicherheitsvorgaben für den AS Java Default Administrationsbenutzer
- Sperren Sie den Standarduser

Für die SAP AS Java Benutzerverwaltung wird der Netweaver Administrator (NWA) verwendet.

Wenn die Userverwaltung für AS Java im AS ABAP durchgeführt wird, werden die Transaktionen SU01 oder PFCG für die Benutzer- und Rollenverwaltung verwendet.

ID: 3.22-131/13.0

Req 132	Der Benutzer guest, J2EE_GUEST_<SID> oder J2EE_GUEST muss vor nicht autorisierten Zugriffen geschützt sein.
---------	---

Im SAP Java-Stack darf der guest (Datenquelle Datenbank), J2EE_GUEST_<SID> (Datenquelle ABAP) oder J2EE_GUEST Benutzer nicht gelöscht werden, da der Gast-Zugang zwingend notwendig ist und für alle Bereiche ohne Berechtigungsprüfung verwendet wird z. B. für den Anmeldebildschirm und für das Starten des Systems. Der Benutzer muss ein dem DTAG Sicherheitsvorgaben entsprechendes Passwort erhalten und er muss gesperrt werden (s. SAP Hinweis 2526747).

Achtung: Der Benutzername für den J2EE_GUEST kann frei definiert werden. Wenden Sie die Vorgaben auf dem von Ihnen als J2EE_GUEST Benutzer definierten Benutzer an.

Motivation: Gast-Zugang ist zwingend notwendig und wird für alle Bereiche ohne Berechtigungsprüfung verwendet (für den Anmeldebildschirm und für das Starten des Systems). Allerdings muss ein unberechtigter Zugriff durch diesen technischen Benutzer verhindert werden.

Umsetzungsbeispiel: Gehen Sie wie folgt vor (s. SAP Hinweis 2526747):

- Setzen Sie das Passwort gemäß der DTAG Sicherheitsvorgaben für den AS Java Default Administrationsbenutzer
- Sperren Sie den Standarduser

Für die SAP AS Java Benutzerverwaltung wird der Netweaver Administrator (NWA) verwendet.

Wenn die Userverwaltung für AS Java im AS ABAP durchgeführt wird, werden die Transaktionen SU01 oder PFCG für die Benutzer- und Rollenverwaltung verwendet.

ID: 3.22-132/13.0

Req 133	Bei Verwendung von Adobe Document Services (ADS) muss der Kommunikationsbenutzer adsuser sicher aufgesetzt sein.
---------	--

Der adsuser wird für die Kommunikation zwischen dem AS Java und den Adobe Document Services (ADS) verwendet. ADS gibt es nur in Verbindung mit einem AS Java. Die Datenquelle für den Benutzer kann Java oder ABAP sein, je nachdem wie das Benutzermangement in der UME definiert wurde. Um den adsuser Benutzer sicher aufzusetzen, muss das Passwort gemäß den DTAG Sicherheitsvorgaben, der richtige Benutzertyp und die richtige Rolle gesetzt sein.

Motivation: Ein unberechtigter Zugriff durch diesen technischen Benutzer muss verhindert werden.

Umsetzungsbeispiel: Der adsuser ist sicher konfiguriert, wenn, ...

- ... Sie das Passwort gemäß der DTAG Sicherheitsvorgaben setzen. Hinweis: Das gesetzte Passwort muss dem Zieldienst bekannt gemacht werden.
- ... der Benutzer den Benutzertyp "service user" für Java oder "system" für ABAP zugeordnet ist.
- ... der Benutzer nur die Rolle ADSCaller erhält

Für die SAP Web AS Java Benutzerverwaltung wird der Netweaver Administrator (NWA) verwendet. Für die SAP Web AS ABAP Benutzer- und Rollenverwaltung werden die Transaktionen SU01 oder PFCG verwendet.

ID: 3.22-133/13.0

Req 134	Der Servicebenutzer ADS_AGENT muss sicher aufgesetzt sein.
---------	--

Der Servicebenutzer ADS_AGENT im SAP Web AS ABAP wird für die Kommunikation zwischen dem SAP Web AS ABAP und den Adobe Document Services benötigt, wenn Sie PDF-basierte Druckformulare in der ABAP-Umgebung verwenden. Um den ADS_AGENT Benutzer sicher aufzusetzen, muss das Passwort gemäß den DTAG Sicherheitsvor-

gaben, der richtige Benutzertyp und die richtige Rolle gesetzt sein.

Motivation: Ein unberechtigter Zugriff durch diesen technischen Benutzer muss verhindert werden.

Umsetzungsbeispiel: Der ADS_AGENT ist sicher konfiguriert, wenn, ...

- ... Sie das Passwort gemäß der DTAG Sicherheitsvorgaben setzen. Hinweis: Das gesetzte Passwort muss dem Zieldienst bekannt gemacht werden.
- ... der Benutzer den Benutzertyp "service".
- ... der Benutzer nur die Rolle SAP_BC_FP_ICF (Dual Stack) oder SAP_BC_FPADS_ICF (Single Stack) erhält.

Der Benutzer wird im ABAP definiert. Verwenden Sie hierfür die Transaktion SU01.

ID: 3.22-134/13.0

Req 135 Der Systembenutzer sapjsf oder sapjsf_XYZ muss sicher aufgesetzt sein.

sapjsf bzw. sapjsf_XYZ ist ein Systembenutzer und behandelt die Kommunikation zwischen der User Management Engine (UME) und der Benutzerverwaltung des SAP NetWeaver Application Server (AS) ABAP. Es ist ein User, der zum Installationszeitpunkt des AS Java oder nachträglich im ABAP angelegt wird (ABAP als Datenquelle für Java), damit sich das Java über RFC mit oder ohne SNC am ABAP anmelden kann.

Empfehlung: Weil mehrere AS Java an einen AS ABAP angebunden werden können, sollte der Nutzer in „sapjsf_XYZ“ umbenannt werden, um Verwechslungen vorzubeugen, wobei XYZ ein Namensverweis auf den entsprechenden AS Java ist.

Motivation: Ein unberechtigter Zugriff durch diesen technischen Benutzer muss verhindert werden.

Umsetzungsbeispiel: Der sapjsf bzw. sapjsf_XYZ Benutzer ist sicher konfiguriert, wenn, ...

- ... Sie das Passwort gemäß der DTAG Sicherheitsvorgaben setzen.
- ... der Benutzer den Benutzertyp "system" für ABAP zugeordnet ist.
- ... der Benutzer nur folgende Rolle (ab Release 6.20) erhält. Die von Ihnen verwendete Rolle hängt davon ab, ob Änderungen an Verwaltungsdaten oder die Erstellung neuer Benutzer aus der UME erforderlich sind oder nicht:
 - SAP_BC_JSF_COMMUNICATION_RO bietet alle Berechtigungen für den Lesezugriff auf Benutzerdaten, für die Authentifizierung von Remotebenutzern und mehrere RFC-Berechtigungen auf niedriger Ebene. Beispielsweise können Benutzer weiterhin ihr eigenes Kennwort ändern. Diese Rolle bietet eine ausreichende Berechtigung, wenn Sie keine administrativen Änderungen an der UME vornehmen möchten: Fügen Sie beispielsweise einen neuen Benutzer hinzu oder ändern Sie einen Nachnamen.
 - SAP_BC_JSF_COMMUNICATION entspricht der oben genannten Rolle, bietet jedoch zusätzlich die Berechtigung zum Ändern und Löschen aller benutzerbezogenen Daten.

Wenn Sie das Kennwort dieses Benutzers in Transaktion SU01 ändern, müssen Sie auch das für diesen Benutzer in der AS-Java-Datenbank abgelegte Kennwort entsprechend ändern. Für diese Vorgehensweise ist ein Neustart des AS Java erforderlich. Daher sollten Sie diese erforderliche Ausfallzeit des AS Java einplanen. Ferner müssen Sie noch in der RFC-Verbindung UMEBackendConnection im AS Java das neue Kennwort für den Servicebenutzer für die UME-ABAP-Kommunikation eingeben (Destination-Service im SAP NetWeaver Administrator). Testen Sie die Verbindung und geben Sie bei Fehlschlag die Verbindungsdaten erneut ein, bis die Verbindung erfolgreich war. Starten Sie den AS Java neu. Der AS Java unterstützt nun das neue Kennwort des Systembenutzers für die UME-ABAP-Kommunikation. Wenn der AS Java nicht starten kann, aktivieren Sie den Notfallbenutzer und pflegen Sie das Kennwort des Systembenutzers in der RFC-Destination. Bitte lesen Sie ggf. weitere Details in der SAP Hilfe nach.

ID: 3.22-135/13.0

Req 136	Beim Einsatz eines AS Java muss die Benutzerverwaltung an ein "Active Directory" oder LDAP-Service gekoppelt sein.
---------	--

Die Benutzerverwaltung eines AS Java muss an eine übergeordnete Benutzerverwaltung (Instanz) gebunden sein (z.B. AD oder das LDAP), da auf dem AS Java kein vergleichbares Sicherheitsniveau in der lokalen Benutzerverwaltung erreicht werden kann, wie es bei der Benutzung eines AS ABAP erreicht wird.

Motivation: Die Benutzerverwaltung von AS Java kann bietet nicht den gleichen Schutzbedarf wie das AD oder ein LDAP-Service.

Umsetzungsbeispiel: Anbindung an ein zentrales IAM.

ID: 3.22-136/13.0

Req 137	System Cookies müssen über das HttpOnly Attribut geschützt werden.
---------	--

Java EE-Anwendungen können System-Cookies verwenden, um Benutzerdaten (wie Sitzungsverfolgung, Anmeldedaten usw.) zu verfolgen. Diese Cookies enthalten sensible Informationen über den Benutzer. Um einen möglichen Missbrauch von Sitzungsdaten zu verhindern, sollten Cookies daher nicht für clientseitige Skripte zugänglich sein. Um den Sicherheitsschutz von System-Cookies zu erhöhen, können Sie die Verwendung des zusätzlichen System-Cookie-Attributs HttpOnly.

Die Deklaration eines Cookies als "HttpOnly" erhöht die Sicherheit Ihres Systems, da der Zugriff auf dieses Cookie im Webbrowser durch clientseitige Skripte, Applets, Plugins usw. unterbunden wird. Dies kann Nebenwirkungen haben, da einige Anwendungen solche Technologien verwenden und ebenfalls auf diese Informationen angewiesen sind. Diese Anwendungen funktionieren möglicherweise nicht mehr richtig, weil sie nicht auf diese Informationen zugreifen können. Bitte prüfen Sie, ob Anwendungen diese Informationen benötigen.

*Motivation: Auf Cookies, welche das Attribut **HttpOnly** besitzen, kann nicht per JavaScript unberechtigt zugegriffen werden. Dies stellt einen Schutz gegenüber Cross-Site-Scripting dar.*

Umsetzungsbeispiel: Sichern Sie die Session Cookies mit folgenden Einstellungen ab:

- Setzen Sie die HTTP-Service-Property "SystemCookiesDataProtection" auf den Wert "true". Damit wird das Attribut HttpOnly für System-Cookies aktiviert.
- Setzen sie die ICM-Property "disable_url_session_tracking" auf den Wert "false". Damit ist das URL-Session-Tracking aktiviert.

Hinweis: Wenn Sie beide Parameter auf "true" setzen, kann AS Java HTTP-Sitzungen nicht korrekt behandeln.

ID: 3.22-137/13.0

Req 138	Zur sicheren Übertragung von System Cookies müssen Sie das "secure" Attribute setzen.
---------	---

Sie verwenden die HTTP-Service-Property SystemCookiesHTTPSPProtection, um das Secure-Attribut auf das Web-Session-Tracking und die Lastausgleichs-Cookies (sapdb_) zu setzen, auf die es sich bezieht. Wenn die Eigenschaft auf true gesetzt ist, wird das Attribut Secure auf das System-Cookie gesetzt und als sicher gekennzeichnete Cookies werden nur übertragen, wenn der Kommunikationskanal ein sicherer ist. In diesem Fall ist die Verwendung von SSL zwingend erforderlich, da diese Cookies nicht über einfaches HTTP übertragen werden und die Sitzungsverfolgung über HTTP nicht funktioniert.

Sie verwenden die Property SecuritySessionIDHTTPSPProtection des HTTP-Dienstes, um das Security Session Identifier Cookie vor der Übertragung über eine ungesicherte HTTP-Verbindung zu schützen. Wenn die Eigenschaft auf true gesetzt ist, wird das Secure-Attribut für das Cookie gesetzt und der Browser sendet es nur über HTTPS.

Motivation: Verhindern Sie einen unberechtigten Zugriff auf das Session Cookie, in dem Sie sicherstellen, dass das Cookie nur über eine gesicherte Verbindung transportiert wird.

Umsetzungsbeispiel: Um das System Cookie vor einer unsicheren Übertragung zu schützen, setzen Sie das HTTP-Service-Property "SystemCookiesHTTPSPProtection" auf den Wert "true".

ID: 3.22-138/13.0

Req 139 Sichern Sie die Cookies des Logon Tickets, indem Sie das Attribut HttpOnly setzen.

Logon Tickets sind Cookies, die für die Benutzerauthentifizierung und Single Sign-On auf AS Java verwendet werden.

Motivation: Verhindern Sie unberechtigten Zugriff auf das Logon Ticket Cookie.

Umsetzungsbeispiel: Um dieses Attribut für Anmeldetickets festzulegen, setzen Sie die Property "ume.logon.httponlycookie" der User Management Engine (UME) auf den Wert "true".

ID: 3.22-139/13.0

2.9.4. Java-Stack: Autorisierungsmanagement

Req 140 Eine Trennung zwischen Benutzer- und Rollenadministration in der Benutzerverwaltung- und Systemadministration muss umgesetzt werden.

Eine Trennung zwischen Rollen- und Benutzeradministration sowie Benutzerverwaltung und Systemadministration ist notwendig.

Motivation: In der Standardkonfiguration ist eine Trennung zwischen Benutzer- und Rollenadministration bei den J2EE-Administratoren nicht gegeben. Um Missbrauch zu verhindern, sind diese Bereiche organisatorisch voneinander zu trennen.

Umsetzungsbeispiel:

Verantwortung	Java
Rollenverwaltung	Contentadministration
Profilverwaltung	Contentadministration
Benutzerverwaltung	Benutzeradministration

Empfehlung: Keine lokale Userverwaltung für Anwendungsnutzer in der UME einsetzen, sondern Kopplung an die ABAP-Userverwaltung (falls Dualstack), an Active Directory oder LDAP.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-140/13.0

Req 141 Es muss sichergestellt werden, dass Administrationsberechtigungen nur an die zuständigen Administratoren vergeben werden.

In der Standardkonfiguration des AS Java wird in der Benutzerverwaltung nicht zwischen Benutzer und Administratoren unterschieden. Es muss sichergestellt werden, dass über das Berechtigungskonzept Unterscheidungen herbeigeführt werden.

Motivation: Segregation of Duties einführen, damit nicht unberechtigte administrative Aktionen auf dem AS Java durchführen.

Umsetzungsbeispiel: Abgleich mit dem kundenspezifischen Berechtigungskonzepts welche Benutzer eine bestimmte UME-action besitzt:

Alle vorhanden Rollen über das J2EE UME IdentityManagement exportieren (<protokoll>://<host>:<port>/useradmin => Rollen (alle suche und markieren) => Exportieren).

Nach der Feststellung, welche Rollen die kritischen Berechtigungen enthalten, können diese nach der Benutzerzuweisung geprüft werden. Aus in der Gruppe der Administratoren darf keine Benutzer die Berechtigung " UME-action" besitzt:

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-141/13.0

Req 142 Die Selbstregistrierungsfunktion für Benutzer muss deaktiviert werden.

Selbstregistrierung bedeutet, dass sich neue Benutzer bei der Anmeldung selbst registrieren können. In der Standardkonfiguration ist die User Management Engine (UME) so eingerichtet, dass eine Grundform der Selbstregistrierung möglich ist. Auf dem Anmeldebild können sich Benutzer durch Ausfüllen eines Formulars selbst registrieren. Danach sind sie registrierte Benutzer und können auf alle Inhalte zugreifen, die der Standardgruppe "Alle" zugeordnet sind. Sie müssen nicht von einem Administrator genehmigt werden. Wenn Sie die Selbstregistrierung mit einem Genehmigungs-Workflow einrichten möchten, müssen Sie Benutzerverwaltung für die Verwendung von Unternehmen konfigurieren.

Die Selbstregistrierungsfunktion für Benutzer sollte deaktiviert werden, wenn sie für die Anwendung nicht erforderlich ist. (Beispiel: eBest-Selbstregistrierung)

Motivation: Die Selbstregistrierungsfunktion der UME ist standardmäßig aktiviert. In dieser Situation können Angreifer ein Konto im Java-Stack erstellen und dann eventuell bestehende Schwächen im Autorisierungskonzept ausnutzen. Dies kann dann zu einem unbefugten Zugriff auf die Daten und Funktionen des AS Java führen.

Umsetzungsbeispiel: Dies erreicht man, indem man den Parameter ume.logon.selfreg auf FALSE setzt.

Oder man im Netweaver Administrator (NWA) in der Konfiguration unter dem Reiter "Benutzerverwaltungs-UI" die folgenden Checkboxes **deaktiviert**:

- Selbstregistrierung von Gastbenutzern aktivieren
- Selbstregistrierung von Firmen aktivieren

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.22-142/13.0

Req 143 Sicherheitsrelevante Ereignisse müssen protokolliert werden.

Systeme müssen sicherheitsrelevante Ereignisse protokollieren können. Bei der Protokollierung müssen die jeweils geltenden gesetzlichen, regulatorischen und betrieblichen Bestimmungen beachtet werden. Diese besagen unter anderem, dass die Protokollierung von Ereignissen nur zweckgebunden erfolgen darf. Die Protokollierung von Ereignissen, zum Zwecke der Arbeitskontrolle von Mitarbeitern, ist nicht zulässig. Die Protokollierung folgender Ereignisse muss aktivierbar sein:

- Authentisierungen von Benutzern und Systemen (erfolgreich und nicht erfolgreich)
- Administrative Zugriffe (inkl. ausgeführter Funktionen, erfolgreich und nicht erfolgreich)
- Unerlaubte/Fehlerhafte Zugriffsversuche (z. B. Zugriff auf für einen Account nicht berechnigte Funktionen oder Zugriff auf eine nicht existente Funktion)
- Administration von Konten
- Änderung von Gruppenzugehörigkeiten von Konten.
- Weitere sicherheitsrelevante Systemmeldungen, die jeweils systemspezifisch festgelegt werden müssen.

Motivation: Motivation: Die Erkennung von (gezielten) Angriffen, sowie die Erkennung erfolgreicher Kompromittierungen von Systemen innerhalb der Netze des DTAG-Konzerns erfordert eine umfangreiche Protokollierung von sicherheitsrelevanten Ereignissen auf Systemen, die von solchen Angriffen direkt betroffenen oder mittelbar daran beteiligt sein können.

Umsetzungsbeispiel: Nutzen Sie den SAP NetWeaver Administrator, um Log und Trace Dateien zu konfigurieren. Öffnen Sie den SAP NetWeaver Administrator und wählen Sie: Troubleshooting -> Logs and Traces -> Log Configuration

Navigieren Sie zu "security audit log" und setzen Sie die "severity" auf "Info".

ID: 3.22-143/13.0

2.9.5. Java-Stack: Transport Management System (TMS)

Req 144	Beim Einsatz von AS Java muss das sichere Erweiterte Change und Transport System (CTS+) verwendet werden.
---------	---

Um Nicht-ABAP-Objekte kontrolliert transportieren zu können, muss das Erweiterte Change und Transport System (CTS+) verwendet werden.

Motivation: Durch die Nutzung des Erweiterten Change und Transport System (CTS+) werden eine systemübergreifende Synchronisation von Änderungen (z. B. Portaländerungen) möglich, Workflow Szenarien können flexibel konfiguriert werden, fachliche und technische Genehmigungsschritte geplant und unter anderem revisionssicheres Tracking und Reporting des CTS (Change and Transport System) eingesetzt werden.

Umsetzungsbeispiel: CTS+ aktivieren.

S. auch auf SAPHelp:

http://help.sap.com/saphelp_nw70ehp2/helpdata/de/bb/6fab6036a146baa58e42fac032ab7b/frameset.htm

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechnigte Einsichtnahme oder Abgriff von Daten
- Unberechnigte Modifikation von Daten
- Unberechnigte Nutzung von Diensten oder Ressourcen

ID: 3.22-144/13.0

2.9.6. Java-Stack: ICM (Internet Communication Manager)

Req 145	Der Zugriff auf das SAP-System über den ICM muss durch Verwendung eines ACL-Filters, eines Authentication-Handlers und/oder dem HTTP Rewrite-Handler beschränkt werden.
---------	---

Der Zugriff auf ein SAP-System muss über verschiedene Verfahren abgesichert werden. Je nach Art des Zugriffes können ACL-Filter (ACL = Access Control List), Authentication-Handler oder HTTP Rewrite-Handler zum Schutz verwendet werden. Das ICM bietet verschiedene Filtermechanismen an. Es ist empfehlenswert den einfachsten Mechanismus

mus zu verwenden, mit welchem sich die Sicherheitsanforderungen lösen lassen. Das heißt z.B. wenn ACLs ausreichen, verwenden Sie diese; die nächste Stufe wäre der Authentication Handler und die letzte Stufe schließlich der Rewrite-Handler. Dadurch können Sie eine unnötig komplexe Konfiguration vermeiden, was wiederum auch zur Sicherheit des Systems beiträgt.

URLs sollten gefiltert werden da ansonsten Informationen über die Infrastruktur und die Konfiguration ablesbar werden (s. auch SAP Note 870127).

Filtermechanismus	ACL-Dateien	Authentication Handler	HTTP Rewrite-Handler
Verwendung	Verwenden Sie ACL-Dateien, um Zugriffe auf bestimmte Client-IP-Adressen oder Client-IP-Adressbereiche einzuschränken, wenn die Beschränkung nicht vom Inhalt des HTTP-Requests (also auch nicht von der URL) abhängt, und keine HTTP-Fehlerseite gewünscht ist.	Verwenden Sie den Authentication Handler, um URL-Filter einzurichten (als „White List“ oder „Black List“). Regeln im Authentication Handler können sich ebenfalls auf bestimmte Client-IP-Adressen, oder auf IP-Adressen des Servers beziehen.	Verwenden Sie den HTTP Rewrite-Handler für Filter, die sich nicht durch ACL-Dateien oder den Authentication Handler abbilden lassen. Der Rewrite Handler ist ein mächtiges Werkzeug für unterschiedliche Filtermechanismen. Er ermöglicht es, anhand eines Regelwerks zahlreiche Daten eines HTTP-Request abzu prüfen und miteinander zu verknüpfen.
Dynamisches Nachladen der Konfigurationsdatei	Ist möglich	Ist möglich	Ist möglich
Positiv- bzw. Negativlisten	• Ja, beide • Auch gemischt möglich	• Ja, beide • Auch gemischt möglich	• Ja, beide • Auch gemischt möglich
Filterung auf URLs, Behandlung von Groß- und Kleinschreibung	Nein	• Ja • Standardeinstellung ist case-insensitiv • Ist konfigurierbar	• Ja • Groß- und Kleinschreibung ist pro Filterregel konfigurierbar
Security Logging	Ja	Ja	Nein
Filterung auf Client IP Adressen, inklusive Netzmasken	Ja	Ja	Ja
Parameter	icm/server_port_<xx> mit Option ACLFILE	icm/HTTP/auth_<xx> mit Option PERMFILE	icm/HTTP/mod_<xx> mit Option FILE

Motivation: Verhindern Sie unberechtigten Webzugriff auf Ihr Backendsystem, um Daten- und Vertraulichkeitsverlust vorzubeugen.

Umsetzungsbeispiel: Für welche Filtertechnik (ACL Datei, Authentication Handler und / oder HTTP Rewrite-Handler) Sie sich auch entscheiden, beachten Sie folgendes:

- Werten Sie das Connection-Log inhaltlich aus (s. das geschriebene LOGFILE über den Parameter `icm/HTTP/logging_xx`), um die relevanten URLs zu ermitteln und diese mit genauen Regeln in Ihrer URL-Permission-Tabelle zu beschreiben (White List).

- Wenn Sie Negativlisten (Deny-Einträge) in URL-Filtern angeben, verwenden Sie case-insensitive Filter, da der Application Server ABAP URLs case-insensitiv behandelt.
- Anstatt der Filterung der URLs kann - wenn möglich - alternativ eine Deaktivierung von URLs vorgenommen werden.

1) ACL Dateien / Filter für das System einstellen.

Die Option ACLFILE gibt die Datei an, die als Zugriffskontrollliste verwendet wird (ACL = Access Control List). Damit kann eine Zugriffskontrollliste eingerichtet werden, welche Verbindungen das ICM akzeptiert und welche nicht. Für jeden Port des ICMs kann eine eigene ACL-Datei angelegt werden. Die Zeichenkette gehorcht folgender Syntax:

```
icm/server_port_<xx> = PROT=<Protokollname>, PORT=<Port oder Servicen-
ame>[, TIMEOUT=<timeout>, PROCTIMEOUT=<proctimeout>, EXTBIND=1,
HOST=<Rechnername>, VCLIENT=<SSL Client Verifikation>, SSLCON-
FIG=ssl_config_<xx>, ACLFILE=<Pfad zur ACL-Datei>]
```

Beschreibung der Zugangspunkte

```
icm/server_port_0 = PROT=HTTP , PORT=0 , TIMEOUT=30 , PROCTIMEOUT=60,
ACLFILE=<Pfad zur ACL-Datei>
icm/server_port_1 = PROT=HTTPS , PORT=1443, TIMEOUT=30 , PROCTI-
MEOUT=60, ACLFILE=<Pfad zur ACL-Datei>
```

Eine Zeile der ACL Datei muss der folgenden Syntax entsprechen:

```
<permit | deny> <ip-address> [tracelevel] [# comment]
```

Die Regeln werden sequentiell von oben beginnend geprüft („top down“). Die erste zutreffende Regel bestimmt das Ergebnis („first match“). Wenn keine Regel zutrifft, wird die Verbindung abgelehnt. Zum eindeutigen Verständnis sollte dennoch ein explizites deny (deny 0.0.0.0/0) als letzte Regel eingetragen werden.

Beispiel Inhalt ACL-Datei:

```
permit 10.1.2.0/24 # permit client network
...
```

```
deny 0.0.0.0/0 # deny the rest
```

2) Authentication Handler für das System einstellen.

Zum Schutz des Backend-Systems (AS ABAP oder AS Java) gibt es einen HTTP-Subhandler (Filter), der Requests anhand von verschiedenen Kriterien abblocken kann. Wenn der Filter aktiviert ist, wird er bei jedem HTTP(S)-Request zum ICM durchlaufen, bevor der Request zu einem anderen HTTP-Handler (Dateizugriff, Cache, Administration, Redirect) oder zum Backend-System (AS ABAP oder AS Java) geschickt wird. Sie können Requests nach den folgenden Kriterien filtern: URL, Client-IP-Adresse, Server-IP-Adresse, Benutzername/Benutzergruppe/Benutzerkennwort, Musteruche in der URL. Um die Zugriffsbeschränkungen über Authentication Handler einzurichten, verwenden Sie den Parameter mit folgender Syntax:

```
icm/HTTP/auth_<xx> = PREFIX=<URL-Präfix>
[,PERMFILE=<permission file>, AUTHFILE=<authentication file>,
FILTER=<name>]
```

Beispiel für die Setzung des Parameters:

```
i c m / H T T P / a u t h _ 0      =      P R E F I X = / ,      P E R M F I -
L E = D : \ u s r \ s a p \ W D P \ S Y S \ p r o f i l e \ p e r m _ f i l t e r . t x t
```

Die Suboption PERMFILE wird die Art des Zugriffsschutzes mit der Permission-Datei bestimmt. Definieren Sie eine Permission-Datei und legen Sie die Regeln an. Die Regeln werden in der Reihenfolge angewendet, in der sie in der Datei erscheinen, von oben nach unten, und die erste Regel, die auf die Anfrage zutrifft, wird angewendet. Es gibt eine implizite "deny all"-Regel, die am Ende der Datei zur Laufzeit hinzugefügt wird. Es wird daher empfohlen, "positive Regeln" zu verwenden (erstellen Sie nur S- oder P-Regeln, da alles andere standardmäßig verweigert wird).

Die Permission-Datei hat den folgenden Aufbau:

- Kommentarzeilen beginnen mit einem # und werden ignoriert
- Andere Zeilen haben die Form: P/D/S <URI-pattern> <USER> <GROUP> <CLIENT-IP> <SERVER-IP>
- Der Buchstabe am Zeilenanfang hat folgende Bedeutung:
 - P (Permit) lässt den Request durch

- D (Deny) lehnt den Request ab und schickt dem Client eine entsprechende Nachricht
- S (Secure) erlaubt für den URL-Präfix nur sichere Verbindungen (HTTPS)
- <URI-pattern> ist der Abschnitt der URL, der im Abschnitt Cache-Key als translated path bezeichnet ist
- Für das URI-Pattern können Sie das Wildcard-Zeichen * verwenden, allerdings nur am Anfang oder am Ende des <URI-pattern> -Strings.
- Für die Client- und Server-IP-Adresse können Sie einen genauen Vergleich, das Wildcard-Zeichen * oder die Netzmasken Syntax verwenden.

Beispiel einer White List mit Erzwingung von HTTPS durch Setzung des "S" am Zeilenanfang:

```
# This is the "permission file" used by the ICM/SAP Web Dispatcher Authentication handler
# webadmin and ping only from Admin-workstations <IP-Admin workstation> allowed
S /sap/bc/ping ** <IP-Admin workstation/net class> *
S /sap/wdisp/admin/** ** <IP-Admin workstation/net class> *
# Access from Webapps and WebGUI (https only) from internal IPs
S /sap/bc/webdynpro/** ** <internal IP/net class> *
S /sap/bc/gui/sap/its/webgui ** <internal IP/net class> *
# Access from Webapps and WebGUI (https only) from Internet
S /sap/bc/srt/rfc/meinwebservice * * * * *
# Everything not listed above will be denied because of the final implicit rule "D * * * * *
```

3) HTTP Rewrite-Handler für das System einstellen

Der HTTP-Rewrite-Handler ist ein sehr leistungsfähiges Werkzeug. Sie können ihn für verschiedene Aktionen verwenden:

- Löschen, Hinzufügen und Erweitern von HTTP-Header-Feldern:
Sie können HTTP-Header-Felder löschen oder hinzufügen oder sie mit zusätzlichen Werten erweitern.
- URLs umschreiben (rewrite):
Sie können sowohl den URL-Pfad als auch den Query-String einer eingehenden HTTP-Anfrage in einen anderen, zuvor definierten URL-Pfad oder Query-String umschreiben. Sie können die Ausführung von URL-Änderungen mit einer oder mehreren Bedingungen verknüpfen.
- URLs umleiten (redirect) oder filtern:
Sie können die URL einer eingehenden HTTP-Anfrage an einen anderen Anwendungsserver oder eine andere URL umleiten. Sie können eingehende HTTP-Anfragen auch nach bestimmten Mustern filtern.
- Keine Änderungsaktion / nop-Aktion (no operation):
Modifikatoren (Optionen) wie break, restart und noescape können nach einer Änderungsaktion (Regel) auftreten. Damit Sie auch Modifikatoren ohne Definitionen verwenden können, gibt es die nop-Aktion.

Um die Zugriffsbeschränkungen über HTTP Rewrite-Handlers einzurichten, verwenden Sie den Parameter mit folgender Syntax:

```
icm/HTTP/mod_<xx> = PREFIX=<URL-Präfix>[, FILE=<Aktionsdatei>]
```

Empfehlung von SAP: Um auf allen Instanzen die gleichen Filterregeln zu haben, legen Sie die Datei mit den Filterregeln in einem globalen Verzeichnis ab und setzen den Parameter im Default-Profil DEFAULT.PFL. Die Filterregeln stehen in der Datei icm_filter_rules.txt im Verzeichnis \$(DIR_GLOBAL)/security/data:

```
i c m / H T T P / m o d _ 0 = P R E -
F I X = / , F I L E = $(DIR_GLOBAL)/security/data/icm_filter_rules.txt
```

Die Aktivierung des HTTP Rewrite-Handlers erfolgt über eine Aktionsdatei (durch Option FILE). Die Aktionsdatei ist eine Textdatei mit jeweils einer Aktion (Regel) pro Zeile. Kommentarzeilen beginnen mit einem Gatter (#). Die Reihenfolge in der Aktionsdatei beschreibt auch die Ausführungsreihenfolge. Sie können mehrere URL-Umschreibungen hintereinander ausführen. Die Ausführung der HTTP-Headerfeld-Modifikationen können Sie mit einer bzw. mehreren Bedingungen verknüpfen. Anders als bei der Verwendung des Authentication Handlers wird am Ende keine implizite "deny all"-Regel hinzugefügt. Beachten Sie zudem, dass eine "Negativliste" erstellt werden muss, statt einer "Positivliste" wie im Authentication Handler.

Verwenden Sie die folgende Syntax, um eine Regel in der Aktionsdatei zu definieren (s. SAP Help):
<operation> <pattern> <dest> [<option>]

Erzwingen Sie HTTPS durch Nutzung des Rewrite-Handler:

Um die Verwendung von unverschlüsseltem HTTP zu verhindern, müssen Sie das Protokoll des Rewrite-Handlers von HTTP auf HTTPS ändern. Dies verhindert Fehlermeldungen im Browser, wenn Benutzer versehentlich mit einer HTTP-URL auf das System zugreifen. Beachten Sie, dass nicht alle HTTP-Clients diese Umleitung befolgen. Während die Redirect-Konfiguration sicherstellt, dass kein HTTP-Zugriff auf das System möglich ist, ist es möglich, dass einzelne Benutzer des Systems, z. B. Web-Service-Endpunkte, von HTTP auf HTTPS umgestellt werden müssen.

Beispiel für strikte Nutzung von HTTPS:

```
# We allow access to the WEB GUI, but only through HTTPS
# The indentation is optional
if %{SERVER_PROTOCOL} !stricmp "HTTPS" [and]
if %{PATH} regimatch ^/sap/bc/gui/sap/its/webgui(.*?)
RegForbiddenURL ^/sap/bc/gui/sap/its/webgui(.*?) - [break]
```

Beispiel Verbot Ausführen eines Ping:

```
# We deny access to the "ping" service
RegForbiddenURL ^/sap/public/ping(.*?) - [break]
```

ID: 3.22-145/13.0

Req 146 Der ICM Server (Port) muss sicher konfiguriert werden.

Der ICM muss so konfiguriert sein, dass nur sichere Protokolle Zugriff haben. Dieses geschieht durch das Setzen des Parameters `icm/server_port_<xx>` zur Nutzung des HTTPS-Protokolls. Mit den Optionen PROT und PORT spezifizieren Sie das Protokoll und den dazugehörigen Port durch die Portnummer oder den Servicenamen. Auf einen Port kann genau nur ein Service gebunden werden.

Motivation: Eine unverschlüsselte Kommunikation erlaubt das Mitlesen dieser mit gefährlichem Potenzial. User-IDs mit den zugehörigen Passwörtern können abgegriffen werden und für eine unberechtigte Anmeldung missbraucht werden, welches wiederum einen unberechtigten Datenzugriff zur Folge hat.

Umsetzungsbeispiel: Durch Setzung des Parameters `icm/server_port_<xx>` wird die Nutzung von HTTPS erzwungen und Zugriffe können kontrolliert werden. Dieser Profilparameter des ICMs konfiguriert den Server Port. Die Zeichenkette gehorcht folgender Syntax:

```
icm/server_port_<xx> = PROT=<Protokollname>, PORT=<Port oder
Servicename>[, TIMEOUT=<timeout>, PROCTIMEOUT=<proctimeout>,
EXTBIND=1, HOST=<Rechnername>, VCLIENT=<SSL Client Verifikation>,
SSLCONFIG=ssl_config_<xx>, ACLFILE=<Pfad zur ACL-Datei>]
```

Beschreibung der Zugangspunkte (_0 bis _3 sind im AS Java vordefinierte Werte für ausgehende Verbindungen, _4 ist ein Beispiel für eine eingehende HTTPS Verbindung, _5 ist ein Beispiel für eine ausgehende HTTPS Verbindung):

```
icm/server_port_0 = PROT=HTTP , PORT=5$ (SAPSYSTEM) 00, TIMEOUT=60 , PROCTIMEOUT=600
icm/server_port_1 = PROT=P4 , PORT=5$ (SAPSYSTEM) 04
icm/server_port_2 = PROT=IIOP , PORT=5$ (SAPSYSTEM) 07
icm/server_port_3 = PROT=TELNET , PORT=5$ (SAPSYSTEM) 08,
HOST=localhost
icm/server_port_4 = PROT=HTTPS , PORT=443, TIMEOUT=30 , PROCTIMEOUT=60, ACLFILE=<Pfad zur ACL-Datei>
icm/server_port_5 = PROT=HTTPS, PORT=0, TIMEOUT=30, PROCTIMEOUT=60,
ACLFILE=<Pfad zur ACL-Datei>
```

Bitte beachten Sie, dass der Wert 0 bei PORTS bedeutet, dass für das angegebene Protokoll kein Port für eingehende Verbindungen geöffnet wird. Ports für eingehende Verbindungen müssen aus Sicherheitsgründen explizit konfiguriert werden.

Achtung: Mit dem ICM werden von SAP vorgelegte Werte ausgeliefert. Diese müssen auf Gültigkeit überprüft werden und ggf. mit " deaktiviert werden z. B.:

```
icm/server_port_0 ="
icm/server_port_1 ="
...
```

ID: 3.22-146/13.0

Req 147 Die Administration des ICMs über die webbasierte Administrationsoberfläche muss gesichert sein.

Sie müssen den Parameter `icm/HTTP/admin_<xx>` setzen, wenn Sie den ICM über die webbasierte Administrationsoberfläche (Browser) überwachen und administrieren wollen.

Motivation: Die Weboberfläche zur Administration des ICMs darf nicht für alle Benutzer erreichbar sein. Diese muss geschützt sein, um einen unberechtigten Zugriff zu verhindern.

Ebenfalls ist die Anzeige von Systeminformationen zu verhindern, da diese Informationen zu einem Angriff auf das System missbraucht werden könnten.

Umsetzungsbeispiel: Der Parameter `icm/HTTP/admin_<xx>` hat folgende Syntax:

```
icm/HTTP/admin_<xx> = PREFIX=<URL-Präfix>, DOCROOT=<Wurzelverzeichnis
der Administrationsdateien> [, AUTHFILE=<Dateiname der Authentifizie-
rungsdatei>, PORT=<TCP/IP-Port, auf dem Adminrequests akzeptiert wer-
den>, HOST=<Rechnername oder IP-Adresse, auf dem Admin-Requests akzep-
tiert werden>, CLIENTHOST=<Rechnername oder IP Adresse, von dem Admin-
requests akzeptiert werden>, ALLOWPUB=<Wert> ]
```

Beispiel zur Setzung des Parameters im Profil:

```
i c m / H T T P / a d m i n _ 0 = P R E -
F I X = / s a p / a d m i n , D O C R O O T = $ ( D I R _ I C M A N _ R O O T /
a d m i n , P O R T = 1 4 4 3 , H O S T = l o c a l h o s t ; < I P      H o s t
ICM>, C L I E N T H O S T = l o c a l h o s t ; < H o s t n a m e n   v o n   A d m i n i s t r a t o r   C l i -
e n t s > , [ A L L O W P U B = F A L S E ]
```

Setzung des Wertes für PREFIX: URL-Präfix, für den dieser HTTP-Subhandler aufgerufen werden soll. Im Standard wird der URL-Präfix für die Verwaltung auf `/sap/admin` gesetzt.

Setzung des Wertes für DOCROOT: Wurzelverzeichnis der Administrationsdateien.

Setzung des Wertes für AUTHFILE: Dateiname der Authentifizierungsdatei (Defaultwert: `icmauth.txt`), in der die Hash-Werte der Passwörter für die Admin-Benutzer liegen. Wenn keine Benutzerauthentifizierung erfolgen muss (z.B. weil diese im Authorisation Handler bereits erfolgt), so kann dies mit der Angabe von `AUTHFILE=none` erreicht werden.

Setzung des Wertes für PORT: Im Default werden Admin-Requests auf allen lokalen TCP/IP-Ports angenommen. Das ist nicht erlaubt. Eine Einschränkung auf einen TCP/IP-Port, der nur lokal erreichbar ist und auf denen Admin-Requests angenommen werden, muss umgesetzt werden. So kann die Verwendung von HTTPS leicht umgesetzt werden.

! Hinweis: Dieser Port muss über den Parameter `icm/server_port_<xx>` angegeben worden sein. Dafür muss ein HTTPS-Port eingerichtet werden, sonst werden die Administrator Passwörter bei der Anmeldung im Klartext übertragen. Unsichere Protokolle sind nicht erlaubt!

Beispiel:

```
icm/server_port_<xx>=.... PROT=HTTPS, PORT=1443, ....
```

Setzung der Werte für HOST: Wird hier nichts angegeben, werden Admin-Requests von allen Rechnernamen akzeptiert. Dies ist nicht erlaubt. Die Einschränkung des Zugriffs auf den lokalen Host muss umgesetzt werden, damit können nur die lokalen Benutzer die webbasierte Oberfläche benutzen. Geben Sie dafür den Wert `localhost` oder `127.0.0.1` und die `<IP des ICM Hosts>` an.

Setzung der Werte für CLIENTHOST: Wird hier nichts angegeben, werden Admin-Requests von jedem Clientrechner akzeptiert. Dies ist nicht erlaubt. Die Administration muss auf bestimmte Clientrechner (Rechner der ICM Administrato-

ren) eingeschränkt werden. Geben Sie dafür den Wert localhost und die <Rechnernamen oder IP-Adressen der Administratoren> ein.

Setzung des Wertes für ALLOWPUB: Ist ALLOWPUB=TRUE gesetzt, so werden unter dem Pfad "public/index.html" öffentliche und lesende Zugriffe auf bestimmte Administrationsseiten ohne Anmeldung zugelassen (z.B. "Monitor", "Aktive Services", "Core Thread Status", "Hostname Buffer", "Release Information" und "MPI Status"). Der Zugriff auf diese Seiten ohne Anmeldung muss verhindert werden. Auch wenn der Zugriff mit den Subparametern HOST und CLIENTHOST eingeschränkt ist (s. oben), dürfen Systeminformationen, die für einen Angriff auf das System genutzt werden könnten, nur über eine Anmeldung am System einsehbar sein. ALLOWPUB=FALSE muss gesetzt werden.

ID: 3.22-147/13.0

Req 148 Die Autorisierungsdatei "icmauth.txt" muss vor unberechtigten Zugriffen geschützt sein.

In der Autorisierungsdatei "icmauth.txt" muss vor unberechtigten Zugriffen geschützt sein, da Benutzer und ihre Gruppenzugehörigkeit im Klartext und Passwörter (verschlüsselt) enthalten sind. Der Dateipfad ist entsprechend den Konventionen des zugrunde liegenden Betriebssystems.

Motivation: Das "icmauth.txt"-File ist gegen Fremdzugriffe zu schützen, um den Einfallsvektor für einen Angriff möglichst klein zu halten. Das Verschlüsselte Passwort ist leicht zu knacken und für einen unberechtigten Angriff zu nutzen.

Umsetzungsbeispiel: Diese Datei muss aus Sicherheitsgründen nur für den Administrations-Benutzer des SAP-Systems (<SID>adm) lesbar und änderbar sein. Setzen von restriktiven Zugriffsrechten auf die Autorisierungsdatei icmauth.txt mit folgenden Befehlen (UNIX):

```
chmod 700 <autorisierungsdatei>
```

Der Ablageort der Datei sowie deren Namen ist im Parameter `icm/authfile` spezifiziert.

ID: 3.22-148/13.0

Req 149 Der Zugriff auf das ICM Administrationsverzeichnis muss restriktiv gesetzt sein.

Das Administrationsverzeichnis des ICM muss restriktiv eingestellt sein. Das Administrationsverzeichnis ist im Standard mit <DIR_ICMAN_ROOT>/admin definiert und enthält die Dateien für die Administration über die Weboberfläche.

Motivation: In dem Administrationsverzeichnis befinden sich Daten (z. B. Datei icmauth.txt) mit einem hohen Schutzbedarf.

Umsetzungsbeispiel: Anpassen der Zugriffsrechte für das Administrationsverzeichnis mit folgenden Befehlen:

```
chown <sid>adm:sapsys <Verzeichnisname>
```

```
chmod 750 <Verzeichnisname> oder wenn möglich restriktiver =>  
700
```

ID: 3.22-149/13.0

Req 150 Die Web Administration erfolgt personifiziert und nicht über den User "icmadm".

Um den ICM von einem Browser aus zu überwachen und zu administrieren, benötigen Sie einen Benutzer.

Der User icmadm ist der Standard User für die Webadministrationsoberfläche und wird in der Datei icmauth.txt hinterlegt, wenn die Einrichtung des Administrationsuser durchgeführt wird.

Für die nachvollziehbare Administration werden personifizierte Administratoren angelegt.

Motivation: Unberechtigter Administrationszugriff kann zur Dateneinsicht und -änderung mit einem Systemausfall führen.

Umsetzungsbeispiel: Bei der Einrichtung der Administrationsuser müssen folgende Schritte ausgeführt werden:

1. Legen Sie als <sid>adm den icmadm Administrationsbenutzer mit icmon an. Hierdurch wird die Berechtigungsdatei icmauth.txt generiert.
2. Legen Sie personalisierte Administrationsuser mit icmon an.
3. Probieren Sie den Zugriff mit den personalisierten Usern.
4. Ändern Sie das Passwort des Standardusers icmadm gemäß den Passwortrichtlinien.

ID: 3.22-150/13.0

Req 151 Der Zugriff auf das Kommando "icmbnd" muss restriktive vergeben werden.

Der Zugriff auf das Kommando "icmbnd" muss restriktive vergeben werden. Im Standard ordnet der ICM die zu benutzenden Ports sich selbst zu (empfohlen). Um die privilegierten Ports mit Portnummer 0 bis einschließlich 1023 („well known ports“) an den ICM binden zu lassen, muss das externe Bindeprogramm icmbnd genutzt werden. Durch Setzen der Option EXTBIND=1 bei der Parameterdefinition von icm/server_port_<xx> wird das Programm icmbnd vom ICM aufgerufen, um die Ports < 1023 zu setzen.

Das Programm icmbnd kann auch direkt aufgerufen werden, um Ports zu konfigurieren (nicht empfohlen).

Das Programm ist unter Pfad, der mit folgenden Parametern im Profil definiert wird, zu finden:

- exe/icmbnd
- DIR_EXECUTABL

Motivation: Unberechtigtes Ausführen des Programmes icmbnd kann zu einer Fehlkonfiguration führen. Es können Angriffe über den ICM auf die nachgelagerten Systeme ausgeführt werden oder auch Programme, die in der Regel nicht über den Ports < 1023 erreichbar sind, erreichbar gebracht werden.

Umsetzungsbeispiel: Setzen von eingeschränkten Zugriffsrechten auf die Datei icmbnd:

```
chown root:sapsys icmbnd
```

```
chmod 4750 icmbnd
```

ID: 3.22-151/13.0

Req 152 Es dürfen keine Informationen zum System in "Fehlerseiten" ausgegeben werden.

Die Standardfehlerseiten müssen durch benutzerspezifisch definierte Fehlerseiten ersetzt werden.

Benutzerspezifisch definierte Fehlerseiten dürfen keine Versionsinformationen des Webserver und verwendeter Module/Erweiterungen enthalten. Die Fehlermeldungen dürfen keine internen Informationen, wie z. B. interne Servernamen, Fehlercodes usw. enthalten.

Aus Sicherheitsgründen sollten Information über das System und oder die Anwendung nicht an den Client weitergegeben (sichtbar) werden.

Motivation: Es dürfen keine Informationen zum System in "Fehlerseiten" ausgegeben werden, die zu einem Angriff auf das System ausgenutzt werden könnten.

Umsetzungsbeispiel: Der Parameter is/HTTP/show_detailed_errors sollte auf "false" gesetzt werden. Überprüfung des ICM Profils:

```
is/HTTP/show_detailed_errors = FALSE
```

Weitere Informationen dazu finden Sie in der SAP Online Hilfe im Kapitel "Fehlerbehandlung durch den SAP Web Dispatcher".

ID: 3.22-152/13.0

Req 153 Die Informationen zum Server in der HTTP Antwort an den Client müssen limitiert werden.

Der Parameter steuert, ob das Server-Headerfeld bei HTTP-Antworten des Servers an den Client eingefügt werden soll oder nicht. Das Server-Header-Feld umfasst die Werte der Parameter `is/server_name` und `is/server_version`. Aus Sicherheitsgründen empfiehlt SAP diese Informationen nicht an den Client weiterzugeben.

Motivation: Es dürfen keine Informationen an den Client weitergegeben werden, die zu einem Angriff ausgenutzt werden könnten.

Umsetzungsbeispiel: Setzen Sie dafür den Parameter `is/HTTP/show_server_header` wie folgt:
`is/HTTP/show_server_header = FALSE`

ID: 3.22-153/13.0

Req 154 Das Sicherheitslogging für den ICM muss aktiviert sein.

Mit dem Parameter `icm/security_log` können Sie die Ausgabe des Sicherheits-Protokolls aus dem ICM und SAP Web Dispatcher steuern.

Administratoren können das Sicherheitsprotokoll (`icm/security_log`) verwenden, um mögliche unberechtigte Zugriffe auf das System zu erkennen. Die folgenden Unregelmäßigkeiten werden protokolliert:

- Daten mit ungültiger Syntax
- Versuchter Zugriff auf Objekte, die nicht existieren (NOT found)
- Zugriff auf Objekte, die aufgrund von Filterregeln nicht erlaubt sind (permission denied)
- Anmeldefehler bei der Webadministration (im ICM und Web Dispatcher)

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Umsetzungsbeispiel: Setzen Sie den Parameter `icm/security_log` für die Aktivierung der Security Protokollierung.

Syntax des Parameters:

`icm/security_log = LOGFILE=<file name>, LEVEL=<security level>, MAXSIZEKB=<max size in KB>, SWITCHTF=<options>, FILEWRAP=on`

LOGFILE: Name der Datei. Durch Zeitangaben wird der Name eindeutig.

LEVEL: Gibt den Grad (1 - 3) der Protokollierung an. Mit Level 1 wird nur der Grund für den Eintrag protokolliert. Mit Level 2 werden zusätzliche Informationen über den Zustand der Verbindung und der Beginn der Daten, die den Eintrag verursachen, protokolliert (Standard, wenn LEVEL nicht angegeben wird). Mit Level 3 werden alle Daten, die den Eintrag verursachen, protokolliert.

MAXSIZEKB: Gibt die Größe der Protokolldatei in KB an. Wird diese Größe überschritten, wird die aktuelle Datei geschlossen und eine neue Datei (mit einem neuen Namen) geöffnet. Durch Definition von LOGFILE wird der Name eindeutig und damit wird ein Überschreiben verhindert.

SWITCHTF: Eine neue Protokolldatei kann nicht nur angelegt werden, wenn die Datei eine bestimmte Größe erreicht hat, sondern auch, wenn sich die Zeitdaten ändern (eine neue Datei stündlich, täglich oder monatlich möglich).

FILEWRAP: Wenn FILEWRAP=on aktiv ist, wird bei jedem Öffnen einer neuen Datei die bestehende Protokolldatei zurückgesetzt und überschrieben (!). Es gibt also immer nur eine Protokolldatei mit den aktuellen Protokolldaten. Wenn Sie diese Option weglassen, wird nach Überschreiten der Größe eine neue Datei geschrieben (siehe MAXSIZEKB und SWITCHTF).

Beispiel Setzung des Parameters (Option FILEWRAP wird **nicht** gesetzt!):

`icm/security_log = LOGFILE=dev_icm_seclog-%d-%m-%y_%h:%t:%s, LEVEL=2, MAXSIZEKB=10000, SWITCHTF=day`

ID: 3.22-154/13.0

Mit den Parametern `icm/HTTP/logging_<xx>` (eingehende Requests) und `icm/HTTP/logging_client_<xx>` (ausgehende Requests) können Sie die Ausgabe des Verbindungs-Protokolls aus dem ICM und SAP Web Dispatcher steuern und protokollieren.

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Umsetzungsbeispiel: Setzen Sie folgende zwei Parameter:

1) Protokollierung eingehender Verbindungen

Syntax des Parameters:

```
icm/HTTP/logging_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file name>
[, LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBytes>,
SWITCHTF=<options>, FILEWRAP=on]
```

PREFIX: URL-Präfix, der für diesen HTTP-Subhandler aufgerufen wird (z. B. /).

LOGFILE: Name der Ausgabedatei im Dateisystem. Der ICM / Web Dispatcher prüft bei jedem Neustart, ob die als Logdatei angegebene Datei existiert. Ist dies der Fall, schreibt er weiter in diese Datei, ist dies nicht der Fall, legt er eine neue Datei an. An den eigentlichen Dateinamen können Sie einen anderen Zeitstempel anhängen, welches die Datei eindeutig macht.

LOGFORMAT: Es gibt die verschiedene Formate für Protokolldateien. Sie können das "NCSA Combined Log Format", das anderen Webservern bekannt ist, durch Verwendung der folgenden Formatzeichenfolge reproduzieren: `%h %l %u %t "%r" %s %b "%{referer}i" "%{user-agent}i"`

FILTER: Die Filtereigenschaft legt fest, dass eine HTTP-Anfrage nur dann protokolliert wird, wenn ein bestimmtes Header-Feld (z. B. ein HTTP-Header-Feld) in der Anfrage oder Antwort enthalten ist.

MAXSIZEKB: Gibt die Größe der Protokolldatei in KB an. Wird diese Größe überschritten, wird die aktuelle Datei geschlossen und eine neue Datei (mit einem neuen Namen) geöffnet. Durch Definition von LOGFILE wird der Name eindeutig und damit wird ein Überschreiben verhindert.

SWITCHTF: Eine neue Protokolldatei kann nicht nur angelegt werden, wenn die Datei eine bestimmte Größe erreicht hat, sondern auch, wenn sich die Zeitdaten ändern (eine neue Datei stündlich, täglich oder monatlich möglich).

FILEWRAP: Wenn FILEWRAP=on aktiv ist, wird bei jedem Öffnen einer neuen Datei die bestehende Protokolldatei zurückgesetzt und überschrieben (!). Es gibt also immer nur eine Protokolldatei mit den aktuellen Protokolldaten. Wenn Sie diese Option weglassen, wird nach Überschreiten der Größe eine neue Datei geschrieben (siehe MAXSIZEKB und SWITCHTF).

Beispiel Setzung des Parameters (Option FILEWRAP wird nicht gesetzt!):

```
icm/HTTP/logging_0 = PREFIX=/, LOGFILE=
incoming_icm_log-%d-%m-%y_%h:%t:%s, LOGFORMAT=%h %l %u %t "%r" %s
%b "%{referer}i" "%{user-agent}i", MAXSIZEKB=10000, SWITCHTF=day
```

2) Protokollierung ausgehender Verbindungen

Syntax des Parameters:

```
icm/HTTP/logging_client_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file
name> [, LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBy-
tes>, SWITCHTF=<options>, FILEWRAP=on]
```

Die Optionen sind dieselben wie bei `icm/HTTP/logging_<xx>`.

Beispiel Setzung des Parameters (Option FILEWRAP wird nicht gesetzt!):

```
icm/HTTP/logging_client_<xx> = PREFIX=/, LOGFILE=
outgoing_icm_log-%d-%m-%y_%h:%t:%s, LOGFORMAT=%h %l %u %t "%r" %s
%b "%{referer}i" "%{user-agent}i", MAXSIZEKB=10000, SWITCHTF=day
```

Achtung bezüglich der Parameter `icm/HTTP/logging_<xx>` und `icm/HTTP/logging_client_<xx>`: Die Protokolldateien für eingehende und ausgehende Anfragen müssen unterschiedliche Namen haben, d.h. Sie können nicht beide Richtungen in eine Datei schreiben.

2.10. SAProuter

Req 156 Der SAProuter muss auf einem gehärteten Image installiert sein.

Der SAProuter ist eine Applikation (keine Hardware und kein Router) die das nachgelagerte SAP-System vor ungewollten Zugriffen schützt. Ein SAProuter steht in der Regel in der DMZ und ist somit ein System das verschiedenen Angriffen standhalten muss.

Motivation: So wie Webserver durch Proxyserver geschützt werden, so SAP Systeme durch den SAProuter in Funktion eines Proxies für die SAP Protokolle (DIAG, RFC, SNC) genutzt.

ID: 3.22-156/13.0

Req 157 Der SAProuter muss der einzige Service sein, der auf dem Host/VM läuft.

Der SAProuter ist eine Applikation die ähnliche Funktionen wie eine Firewall oder Proxy-Server übernimmt. In der Regel wird das System in ungeschützten Bereichen aufgebaut. Damit bei einem Angriff oder einer Übernahme nicht andere Applikationen in Mitleidenschaft gezogen werden, ist das System, als Single-Applikation zu installieren.

Motivation: Durch die Installation in ungeschützten Bereichen, sollten keine weiteren DMZ-Applikationen auf dem System installiert sein. Durch die Vereinzelung wird im Falle eines Angriffes oder Verlust der Applikation keine weitere Applikation beeinträchtigt.

ID: 3.22-157/13.0

Req 158 Der SAProuter darf nicht im gleichen SubNetz (IP) wie das zu schützende System stehen.

Um die SAP-Applikation zu schützen und sicherzustellen, dass der SAProuter nicht umgangen werden kann, muss die Applikation in einem anderen IP-Bereich wie der SAP-Router liegen.

Motivation: Der Schutz des SAP-System durch den SAProuter kann nur gewährleistet werden, wenn dieser nicht umgangen werden kann.

Umsetzungsbeispiel: SAP Anwendung = 192.168.100.1
SAProuter = 10.10.10.10

ID: 3.22-158/13.0

Req 159 Auf dem Server, auf dem der SAProuter läuft, darf kein IP-Forwarding aktiviert sein.

Auf dem SAP-Router-Betriebssystem, darf die Funktionalität "IP-Forwarding" nicht aktiv sein, da sonst die Funktionalität des SAP-Routers umgangen wird.

Motivation: Mit dem aktivieren der IP-Forward funktion, wird der SAP-Router umgangen.

ID: 3.22-159/13.0

Req 160 Der Zugriff auf das Verzeichnis, die Unterverzeichnisse und die Dateien des SAProuters muss eingeschränkt sein.

Der SAProuter wird unter seinem eigenen User gestartet, gestoppt und administriert. Dabei ist erfolgt der Zugriff personalisiert und ist über Berechtigungen zentralisiert geregelt (z. B. LDAP, sudo).

Motivation: Schutz des Regelwerks, dass den SAProuter steuert.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigung	OS-User/OS-Gruppe
<Verzeichnis vom SAProuter>	700	<sapsid>adm/sapsys
<Verzeichnis vom SAProuter>/*	700	<sapsid>adm/sapsys

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <sapsid>adm:sapsys <Verzeichnis vom SAProuter>
chmod 700 <Verzeichnis vom SAProuter>
```

ID: 3.22-160/13.0

Req 161 Der Besitzer der ausführbaren SAProuter-Datei darf keine root-Rechte besitzen und oder die ausführbare SAProuter-Datei darf nicht der Gruppe root angehören.

Der Besitzer der ausführbaren SAProuter-Datei darf keine root-Rechte besitzen und oder die ausführbare SAProuter-Datei darf nicht der Gruppe root angehören.

Motivation: Schutz des System, falls der SAP-Router nach einer Verwundbarkeit der Applikation fehlfunktionen hat.

ID: 3.22-161/13.0

Req 162 Wenn SNC eingesetzt wird, müssen die SNC-Einträge in der saproustab-Datei in den ersten Zeilen stehen.

Die Einträge in der saproustab-Datei werden von oben nach unten gelesen und ausgewertet (First Match). Die erste Zeile, die passt, wird als Regel für die Verbindung genutzt.

Motivation: Durch das nach oben ziehen der SNC-Regeln wird sichergestellt, dass SNC-Regeln vor unverschlüsselten Verbindungen genutzt werden.

Umsetzungsbeispiel: Inhalt saproustab:

```
# Initiating SNC for all connections to host2 :
KT = "p:CN=saprou2, OU=TEST01, O=myCompany, C=US"      host2      *
# Accepting all connections
```

```
D      *      *      *
```

```
https://help.sap.com/doc/saphelp_nw73ehp1/7.31.19/en-us/65/8d09ab5c7e46028f633bb01a09b380/
content.htm?no_cache=true
```

```
https://help.sap.com/doc/saphelp_nw75/7.5.5/en-US/48/6c7a3fc1504e6ce10000000a421937/
content.htm?no_cache=true
```

ID: 3.22-162/13.0

Req 163 Die letzte Zeile in der saproustab-Datei muss alle anderen Verbindungen abweisen.

Um sicherzustellen, dass unberechtigte mögliche Verbindungen abgewiesen werden, muss eine Deny-Regel ans Ende eingefügt werden.

Motivation: Sicherstellen, dass keine Verbindung regellos ist. In Normalfall sollten Verbindungen die keine Regel fin-

den, keinen Zugriff auf das System bekommen.

Umsetzungsbeispiel: Letzte Zeilen in der saproutab:

KD * * * *

D * * * *

Dabei verhindert "KD" alle SNC Verbindungen und "D" alle unverschlüsselten Verbindungen (für SAP Protokolle RFC und DIAG).

ID: 3.22-163/13.0

Req 164 In der "saproutab"-Datei dürfen nur die benötigten Verbindungen vorhanden sein.

In der Konfigurationsdatei für den SAProuter, dürfen nur die Verbindungen gepflegt sein, die für das Nutzen der Applikation nötig sind.

Motivation: Es sollten nur Verbindungen zugelassen werden, die für die Nutzung der Applikation benötigt werden, um den Schutz für die Applikation möglichst hoch zu halten.

Umsetzungsbeispiel: Prüfen Sie, ob nur die benötigten Verbindungen in der saproutab eingepflegt sind (s. SAP Hinweis 1895350).

ID: 3.22-164/13.0

Req 165 Es dürfen in der Route Permission Tabelle (saproutab) keine Wildcards (*) verwendet werden.

In den P-/KP- oder S-/KS-Zeilen der Route Permission Tabelle (saproutab, Regelwerk für die Zugriffe eines SAP-Routers) dürfen keine Wildcards (*) für den Zielrechner (<dest-host>) und den Zielport (<dest-serv>) verwendet werden. Die Verwendung von IP-Adressbereichen für Ziel Host und oder von Portbereichen für den Service ist nicht erlaubt.

Motivation: Durch das Eingrenzen auf die tatsächlichen Zielhosts und oder Services wird ein unberechtigtes Zugreifen auf andere Host und andere Services verhindert.

Umsetzungsbeispiel: Beispiel Eintrag in der saproutab:

P/S <IP-Adresse/Hostname> <vollständige IP-Adresse/Hostname> 3299 <Passwort>

ID: 3.22-165/13.0

Req 166 Der SAProuter darf nur für die Protokolle DIAG, RFC und SNC genutzt werden.

Nur DIAG, RFC und SNC dürfen als Protokolle von dem SAProuter weitergeleitet werden. Eine Ausnahme ist der SAP Support-Fall, der auch andere Protokolle weiterleiten können muss z. B. https.

Motivation: Andere Protokolle wie https sollen z. B. über den SAP Web Dispatcher qualifiziert werden und es darf keine Umgehung über den SAProuter stattfinden. Über den SAProuter werden nur die SAP Protokolle DIAG und RFC geleitet.

Umsetzungsbeispiel: Durch Nutzung des "S" (im Falle von SNC "KS") anstatt des "P" (im Falle von SNC "KP") für alle positiven Einträge werden nur die SAP-Protokolle für die Verbindung zugelassen:

KS <host 1> <host2> <Port> <Passwort>

S <host 3> <host4> <Port> <Passwort>

ID: 3.22-166/13.0

Req 167 Jeder SAProuter muss eine (spezifische) Route-Permission-Tabelle (Dateiname: saprountab) besitzen.

Der Defaultpfad ist ./saprountab, wenn man in dem Ordner steht, in dem die Datei "saprouter" liegt.

Motivation: Ohne eine Regeldatei ist der SAProuter nicht sinnvoll zu nutzen.

Umsetzungsbeispiel: Starte den SAProuter mit der Option -R <path>.

<Pfad zur SAProuter Datei>SAProuter -r -R <Pfad zur SAProuter Datei>/saprountab

ID: 3.22-167/13.0

Req 168 Es muss verhindert werden, dass der SAProuter Verbindungen zu sich selbst (loopback) aufbaut.

In der Default Konfiguration lässt der SAProuter keine Route auf sich selber zu. Der "loopback" aus dem SAProuter auf sich selber kann explizit mit der Option -X zugelassen werden. Diese Option kann die Sicherheit ihres Systems beeinträchtigen, da potenzielle Angreifer den SAProuter von einem externen Client aus administrieren könnten (s. SAP Hinweis 1853140).

Motivation: Werden Loopback-Verbindungen zugelassen, können Änderungen der SAProuter Konfiguration durch unberechtigten Zugriff erfolgen.

Umsetzungsbeispiel: Starte den SAProuter OHNE der Option -X <path>.

<Pfad zur SAProuter Datei>/SAProuter -r -R <Pfad zur SAProuter Datei>/saprountab

ID: 3.22-168/13.0

Req 169 Die SAProuter Protokollierung muss eingeschaltet (SAP note 1895350) sein.

Der Loginhalt bei Setzen der Option -G zeigt die Verbindungen, der Option -T zeigt die Verbindungsfehler.

Beim Setzen von Option -G sollte auch die Option -J gesetzt werden. Durch Setzen von Option -J wird die Größe der Logdatei auf den angegebenen Wert begrenzt und damit die Erzeugung einer beliebig großen Logdatei verhindert. Wenn die Logdatei-Größe erreicht ist, wird eine neue Logdatei geöffnet und die Alte wird unter dem Namen <log file name>_a_<start date>_<start time>-<end date>_<end time> gesichert.

Eine weitere Option, die bei Verwendung der Optionen -G und -T verwendet werden sollte, ist die Option -E. Die Option -E verhindert das Überschreiben der Logdateien bei einem Neustart des SAProuters.

Motivation: Durch das Setzen der Log funktionalität, kann erst erkannt werden, welche Verbindungen über den SAP-Router laufen und welche abgelehnt wurden.

Umsetzungsbeispiel: Um die SAProuter Verbindungen zu protokollieren müssen die Optionen -G <Pfad> und -T <Pfad> gesetzt werden. Außerdem ist darauf zu achten, dass die Logdateien nicht zu groß werden. Dazu sind geeignete Werte für die Option -J zu setzen.

<path>SAProuter -r -G <path> -T <path> -J <Größe in bytes> -E

Req 170 Der SAProuter muss als Service gestartet werden.

Um Sicherzustellen, dass der SAProuter nach einem Neustart des Betriebssystems in der Umgebung gestartet wird, muss dieser via Skript gestartet werden.

Motivation: Um Sicherzustellen, dass der SAProuter nach einem Neustart des Betriebssystems in der Umgebung gestartet wird, sollte dieser via Skript gestartet werden.

Umsetzungsbeispiel: Der SAProuter muss mit dem OS-Start automatisch gestartet werden:

```
<path>SAProuter -r
```

Wenn er ohne die Option -S gestartet wird, wird der Standard-Port 3299 verwendet.

ID: 3.22-170/13.0

2.11. Anforderungen SAP Web Dispatcher

Req 171 Der Zugriff auf das SAP-System über den SAP Web Dispatcher muss durch Verwendung eines ACL-Filters, eines Authentication-Handlers und/oder dem HTTP Rewrite-Handler beschränkt werden.

Der Zugriff auf ein SAP-System muss über verschiedene Verfahren abgesichert werden. Je nach Art des Zugriffes können ACL-Filter (ACL = Access Control List), Authentication-Handler oder HTTP Rewrite-Handler zum Schutz verwendet werden. Das SAP Web Dispatcher bietet verschiedene Filtermechanismen an. Es ist empfehlenswert den einfachsten Mechanismus zu verwenden, mit welchem sich die Sicherheitsanforderungen lösen lassen. Das heißt z.B. wenn ACLs ausreichen, verwenden Sie diese; die nächste Stufe wäre der Authentication Handler und die letzte Stufe schließlich der Rewrite-Handler. Dadurch können Sie eine unnötig komplexe Konfiguration vermeiden, was wiederum auch zur Sicherheit des Systems beiträgt.

Filtermechanismus	ACL-Dateien	Authentication Handler	HTTP Rewrite-Handler
Verwendung	Verwenden Sie ACL-Dateien, um Zugriffe auf bestimmte Client-IP-Adressen oder Client-IP-Adressbereiche einzuschränken, wenn die Beschränkung nicht vom Inhalt des HTTP-Requests (also auch nicht von der URL) abhängt, und keine HTTP-Fehlerseite gewünscht ist.	Verwenden Sie den Authentication Handler, um URL-Filter einzurichten (als „White List“ oder „Black List“). Regeln im Authentication Handler können sich ebenfalls auf bestimmte Client-IP-Adressen, oder auf IP-Adressen des Servers beziehen.	Verwenden Sie den HTTP Rewrite-Handler für Filter, die sich nicht durch ACL-Dateien oder den Authentication Handler abbilden lassen. Der Rewrite Handler ist ein mächtiges Werkzeug für unterschiedliche Filtermechanismen. Er ermöglicht es, anhand eines Regelwerks zahlreiche Daten eines HTTP-Request abzufragen und miteinander zu verknüpfen.
Dynamisches Nachladen der Konfigurationsdatei	Ist möglich	Ist möglich	Ist möglich

Positiv- bzw. Negativlisten	• Ja, beide • Auch gemischt möglich	• Ja, beide • Auch gemischt möglich	• Ja, beide • Auch gemischt möglich
Filterung auf URLs, Behandlung von Groß- und Kleinschreibung	Nein	• Ja • Standardeinstellung ist case-insensitiv • Ist konfigurierbar	• Ja • Groß- und Kleinschreibung ist pro Filterregel konfigurierbar
Security Logging	Ja	Ja	Nein
Filterung auf Client IP Adressen, inklusive Netzmasken	Ja	Ja	Ja
Parameter	icm/server_port_<xx> mit Option ACLFILE	icm/HTTP/auth_<xx> mit Option PERMFILE	icm/HTTP/mod_<xx> mit Option FILE

Motivation: Verhindern Sie unberechtigten Webzugriff auf Ihr Backendsystem, um Daten- und Vertraulichkeitsverlust vorzubeugen.

Umsetzungsbeispiel: Für welche Filtertechnik (ACL Datei, Authentication Handler und / oder HTTP Rewrite-Handler) Sie sich auch entscheiden, beachten Sie folgendes:

- Werten Sie das Connection-Log inhaltlich aus (s. das geschriebene LOGFILE über den Parameter `icm/HTTP/logging_<xx>`), um die relevanten URLs zu ermitteln und diese mit genauen Regeln in Ihrer URL-Permission-Tabelle zu beschreiben (White List).
- Filtern / sperren Sie auf alle Fälle folgende Pfade in Ihrer URL-Permission-Tabelle:
 - `/sap/public/icman/*`
 - `/sap/public/icf_info/*`
 - `/sap/wdisp/info`
- Wenn Sie Negativlisten (Deny-Einträge) in URL-Filtern angeben, verwenden Sie case-insensitive Filter, da der Application Server ABAP URLs case-insensitiv behandelt.
- Anstatt der Filterung der URLs kann - wenn möglich - alternativ eine Deaktivierung von URLs vorgenommen werden.

1) ACL Dateien / Filter für das System einstellen

Die Option ACLFILE gibt die Datei an, die als Zugriffskontrollliste verwendet wird (ACL = Access Control List). Damit kann eine Zugriffskontrollliste eingerichtet werden, welche Verbindungen das SAP Web Dispatcher akzeptiert und welche nicht. Für jeden Port des ICMS kann eine eigene ACL-Datei angelegt werden. Die Zeichenkette gehorcht folgender Syntax:

```
icm/server_port_<xx> = PROT=<Protokollname>, PORT=<Port oder Servicen-
ame>[, TIMEOUT=<timeout>, PROCTIMEOUT=<proctimeout>, EXTBIND=1,
HOST=<Rechnername>, VCLIENT=<SSL Client Verifikation>, SSLCON-
FIG=ssl_config_<xx>, ACLFILE=<Pfad zur ACL-Datei>]
```

Beschreibung der Zugangspunkte

```
icm/server_port_0 = PROT=HTTP , PORT=0 , TIMEOUT=30 , PROCTIMEOUT=60,
ACLFILE=<Pfad zur ACL-Datei>
icm/server_port_1 = PROT=HTTPS , PORT=1443, TIMEOUT=30 , PROCTI-
MEOUT=60, ACLFILE=<Pfad zur ACL-Datei>
```

Eine Zeile der ACL Datei muss der folgenden Syntax entsprechen:

```
<permit | deny> <ip-address> [tracelevel] [# comment]
```

Die Regeln werden sequentiell von oben beginnend geprüft („top down“). Die erste zutreffende Regel bestimmt das Ergebnis („first match“). Wenn keine Regel zutrifft, wird die Verbindung abgelehnt. Zum eindeutigen Verständnis sollte dennoch ein explizites deny (deny 0.0.0.0/0) als letzte Regel eingetragen werden.

Beispiel Inhalt ACL-Datei:

```
permit 10.1.2.0/24 # permit client network
...
```

```
deny 0.0.0.0/0 # deny the rest
```

2) Authentication Handler für das System einstellen.

Um die Zugriffsbeschränkungen über Authentication Handler einzurichten, verwenden Sie den Parameter mit folgender Syntax:

```
icm/HTTP/auth_<xx> = PREFIX=<URL-Präfix>
[,PERMFILE=<permission file>, AUTHFILE=<authentication file>,
FILTER=<name>]
```

Beispiel für die Setzung des Parameters:

```
i c m / H T T P / a u t h _ 0      =      P R E F I X = / ,      P E R M F I -
L E = D : \ u s r \ s a p \ W D P \ S Y S \ p r o f i l e \ p e r m _ f i l t e r . t x t
```

Die Suboption PERMFILE wird die Art des Zugriffsschutzes mit der Permission-Datei bestimmt. Definieren Sie eine Permission-Datei und legen Sie die Regeln an. Die Regeln werden in der Reihenfolge angewendet, in der sie in der Datei erscheinen, von oben nach unten, und die erste Regel, die auf die Anfrage zutrifft, wird angewendet. Es gibt eine implizite "deny all"-Regel, die am Ende der Datei zur Laufzeit hinzugefügt wird. Es wird daher empfohlen, "positive Regeln" zu verwenden (erstellen Sie nur S- oder P-Regeln, da alles andere standardmäßig verweigert wird).

Die Permission-Datei hat den folgenden Aufbau:

- Kommentarzeilen beginnen mit einem # und werden ignoriert
- Andere Zeilen haben die Form: P/D/S <URI-pattern> <USER> <GROUP> <CLIENT-IP> <SERVER-IP>
- Der Buchstabe am Zeilenanfang hat folgende Bedeutung:
 - P (Permit) lässt den Request durch
 - D (Deny) lehnt den Request ab und schickt dem Client eine entsprechende Nachricht
 - S (Secure) erlaubt für den URL-Präfix nur sichere Verbindungen (HTTPS)
 - <URI-pattern> ist der Abschnitt der URL, der im Abschnitt Cache-Key als translated path bezeichnet ist
- Für das URI-Pattern können Sie das Wildcard-Zeichen * verwenden, allerdings nur am Anfang oder am Ende des <URI-pattern> -Strings.
- Für die Client- und Server-IP-Adresse können Sie einen genauen Vergleich, das Wildcard-Zeichen * oder die Netzmasken Syntax verwenden.

Beispiel einer White List mit Erzwingung von HTTPS durch Setzung des "S" am Zeilenanfang:

```
# This is the "permission file" used by the ICM/SAP Web Dispatcher Au-
thentication handler
# webadmin and ping only from Admin-workstations <IP-Admin workstati-
on> allowed
S /sap/bc/ping * * <IP-Admin workstation/net class> *
S /sap/wdisp/admin/* * * <IP-Admin workstation/net class> *
# Access from Webapps and WebGUI (https only) from internal IPs
S /sap/bc/webdynpro/* * * <internal IP/net class> *
S /sap/bc/gui/sap/its/webgui * * <internal IP/net class> *
# Access from Webapps and WebGUI (https only) from Internet
S /sap/bc/srt/rfc/meinwebservice * * * * *
# Everything not listed above will be denied because of the final im-
plicit rule "D * * * * *"
```

3) HTTP Rewrite-Handler für das System einstellen

Der HTTP-Rewrite-Handler ist ein sehr leistungsfähiges Werkzeug. Sie können ihn für verschiedene Aktionen verwenden:

- Löschen, Hinzufügen und Erweitern von HTTP-Header-Feldern:
Sie können HTTP-Header-Felder löschen oder hinzufügen oder sie mit zusätzlichen Werten erweitern.
- URLs umschreiben (rewrite):
Sie können sowohl den URL-Pfad als auch den Query-String einer eingehenden HTTP-Anfrage in einen anderen, zuvor definierten URL-Pfad oder Query-String umschreiben. Sie können die Ausführung von URL-Änderungen mit einer oder mehreren Bedingungen verknüpfen.
- URLs umleiten (redirect) oder filtern:
Sie können die URL einer eingehenden HTTP-Anfrage an einen anderen Anwendungsserver oder eine andere URL umleiten. Sie können eingehende HTTP-Anfragen auch nach bestimmten Mustern filtern.
- Keine Änderungsaktion / nop-Aktion (no operation):
Modifikatoren (Optionen) wie break, restart und noescape können nach einer Änderungsaktion (Regel) auftreten. Damit Sie auch Modifikatoren ohne Definitionen verwenden können, gibt es die nop-Aktion.

Um die Zugriffsbeschränkungen über HTTP Rewrite-Handlers einzurichten, verwenden Sie den Parameter mit folgender Syntax:

```
icm/HTTP/mod_<xx> = PREFIX=<URL-Präfix>[, FILE=<Aktionsdatei>]
```

Empfehlung von SAP: Um auf allen Instanzen die gleichen Filterregeln zu haben, legen Sie die Datei mit den Filterregeln in einem globalen Verzeichnis ab und setzen den Parameter im Default-Profil DEFAULT.PFL. Die Filterregeln stehen in der Datei icm_filter_rules.txt im Verzeichnis \$(DIR_GLOBAL)/security/data:

```
i c m / H T T P / m o d _ 0 = P R E -  
FIX=/, FILE=$(DIR_GLOBAL)/security/data/icm_filter_rules.txt
```

Die Aktivierung des HTTP Rewrite-Handlers erfolgt über eine Aktionsdatei (durch Option FILE). Die Aktionsdatei ist eine Textdatei mit jeweils einer Aktion (Regel) pro Zeile. Kommentarzeilen beginnen mit einem Gatter (#). Die Reihenfolge in der Aktionsdatei beschreibt auch die Ausführungsreihenfolge. Sie können mehrere URL-Umschreibungen hintereinander ausführen. Die Ausführung der HTTP-Headerfeld-Modifikationen können Sie mit einer bzw. mehreren Bedingungen verknüpfen. Anders als bei der Verwendung des Authentication Handlers wird am Ende keine implizite "deny all"-Regel hinzugefügt. Beachten Sie zudem, dass eine "Negativliste" erstellt werden muss, statt einer "Positivliste" wie im Authentication Handler.

Verwenden Sie die folgende Syntax, um eine Regel in der Aktionsdatei zu definieren (s. SAP Help):

```
<operation> <pattern> <dest> [<option>]
```

Erzwingen Sie HTTPS durch Nutzung des Rewrite-Handler:

Um die Verwendung von unverschlüsseltem HTTP zu verhindern, müssen Sie das Protokoll des Rewrite-Handlers von HTTP auf HTTPS ändern. Dies verhindert Fehlermeldungen im Browser, wenn Benutzer versehentlich mit einer HTTP-URL auf das System zugreifen. Beachten Sie, dass nicht alle HTTP-Clients diese Umleitung befolgen. Während die Redirect-Konfiguration sicherstellt, dass kein HTTP-Zugriff auf das System möglich ist, ist es möglich, dass einzelne Benutzer des Systems, z. B. Web-Service-Endpunkte, von HTTP auf HTTPS umgestellt werden müssen.

Beispiel für strikte Nutzung von HTTPS:

```
# We allow access to the WEB GUI, but only through HTTPS  
# The indentation is optional  
if %{SERVER_PROTOCOL} !stricmp "HTTPS" [and]  
if %{PATH} regimatch ^/sap/bc/gui/sap/its/webgui(.*)  
RegForbiddenURL ^/sap/bc/gui/sap/its/webgui(.*) - [break]
```

Beispiel Verbot Ausführen eines Ping:

```
# We deny access to the "ping" service  
RegForbiddenURL ^/sap/public/ping(.*) - [break]
```

ID: 3.22-171/13.0

Req 172	Zwischen dem SAP Web Dispatcher und den Backendsystemen/Applikationen muss ein sicheres Protokoll (HTTPS) verwendet werden.
---------	---

Der Parameter `wdisp/ssl_encrypt` legt fest, wie der SAP Web Dispatcher mit eingehenden HTTP(S)-Requests verfährt. Folgende Werte sind möglich:

0: Request unverschlüsselt weitergeben.

1: Request wieder SSL-verschlüsseln, falls der Request über das HTTPS-Protokoll hereinkam.

2: auf jeden Fall wieder SSL-verschlüsselt weitergeben. Sie können den SAP Web Dispatcher auch für End-to-End SSL konfigurieren, indem Sie bei der Einstellung von `icm/server_port_<xx>` das Protokoll ROUTER angeben.

Motivation: Der Webdispatcher regelt die Verbindungen zwischen Client und Applikation in Bezug auf das HTTP-Protokoll.

Umsetzungsbeispiel: Verwenden Sie den Profilparameter `wdisp/ssl_encrypt = 2`.

ID: 3.22-172/13.0

Req 173 Der SAP Web Dispatcher Server (Port) muss sicher konfiguriert werden.

Der SAP Web Dispatcher muss so konfiguriert sein, dass nur sichere Protokolle Zugriff haben. Dieses geschieht durch das Setzen des Parameters `icm/server_port_<xx>` zur Nutzung des HTTPS-Protokolls. Mit den Optionen PROT und PORT spezifizieren Sie das Protokoll und den dazugehörigen Port durch die Portnummer oder den Servicenamen. Auf einen Port kann genau nur ein Service gebunden werden.

Motivation: Eine unverschlüsselte Kommunikation erlaubt das Mitlesen dieser mit gefährlichem Potenzial. User-IDs mit den zugehörigen Passwörtern können abgegriffen werden und für eine unberechtigte Anmeldung missbraucht werden, welches wiederum einen unberechtigten Datenzugriff zur Folge hat.

Umsetzungsbeispiel: Durch Setzung des Parameters `icm/server_port_<xx>` wird die Nutzung von HTTPS erzwungen und Zugriffe können kontrolliert werden. Dieser Profilparameter des SAP Web Dispatchers konfiguriert den Server Port. Die Zeichenkette gehorcht folgender Syntax:

```
icm/server_port_<xx> = PROT=<Protokollname>, PORT=<Port oder  
Servicename>[, TIMEOUT=<timeout>, PROCTIMEOUT=<proctimeout>,  
EXTBIND=1, HOST=<Rechnername>, VCLIENT=<SSL Client Verifikation>,  
SSLCONFIG=ssl_config_<xx>, ACLFILE=<Pfad zur ACL-Datei>]
```

Beschreibung der Zugangspunkte (`_0` und `_1` sind im AS ABAP vordefinierte Werte für ausgehende Verbindungen, `_2` ist ein Beispiel für eine eingehende HTTPS Verbindung, `_3` ist ein Beispiel für eine ausgehende HTTPS Verbindung):

```
icm/server_port_0 = PROT=HTTP,PORT=0,TIMEOUT=30,PROCTIMEOUT=60  
icm/server_port_1 = PROT=SMTP,PORT=0,TIMEOUT=120,PROCTIMEOUT=120  
icm/server_port_2 = PROT=HTTPS , PORT=443, TIMEOUT=30 , PROCTI-  
MEOUT=60, ACLFILE=<Pfad zur ACL-Datei>  
icm/server_port_3 = PROT=HTTPS,PORT=0,TIMEOUT=30,PROCTIMEOUT=60,  
ACLFILE=<Pfad zur ACL-Datei>
```

Bitte beachten Sie, dass der Wert 0 bei PORTS bedeutet, dass für das angegebene Protokoll kein Port für eingehende Verbindungen geöffnet wird. Ports für eingehende Verbindungen müssen aus Sicherheitsgründen explizit konfiguriert werden.

Achtung: Mit dem SAP Web Dispatcher werden von SAP vorgelegte Werte ausgeliefert. Diese müssen auf Gültigkeit überprüft werden und ggf. mit " deaktiviert werden z. B.:

```
icm/server_port_0 ="  
icm/server_port_1 ="
```

ID: 3.22-173/13.0

Req 174 Die Administration des SAP Web Dispatchers über die webbasierte Administrationsoberfläche muss gesichert sein.

Sie müssen den Parameter `icm/HTTP/admin_<xx>` setzen, wenn Sie den SAP Web Dispatcher über die webbasierte Administrationsoberfläche (Browser) überwachen und administrieren wollen.

Motivation: Die Weboberfläche zur Administration des SAP Web Dispatchers darf nicht für alle Benutzer erreichbar sein. Diese muss geschützt sein, um einen unberechtigten Zugriff zu verhindern.

Ebenfalls ist die Anzeige von Systeminformationen zu verhindern, da diese Informationen zu einem Angriff auf das System missbraucht werden könnten.

Umsetzungsbeispiel: Der Parameter `icm/HTTP/admin_<xx>` hat folgende Syntax:

```
icm/HTTP/admin_<xx> = PREFIX=<URL-Präfix>, DOCROOT=<Wurzelverzeichnis  
der Administrationsdateien> [, AUTHFILE=<Dateiname der Authentifizie-  
rungsdatei>, PORT=<TCP/IP-Port, auf dem Adminrequests akzeptiert wer-  
den>, HOST=<Rechnername oder IP-Adresse, auf dem Admin-Requests akzep-  
tiert werden>, CLIENTHOST=<Rechnername oder IP Adresse, von dem Admin-  
requests akzeptiert werden>, ALLOWPUB=<Wert> ]
```

Beispiel zur Setzung des Parameters im Profil:

```
i c m / H T T P / a d m i n _ 0 = P R E -  
F I X = / s a p / a d m i n , D O C R O O T = $ ( D I R _ I C M A N _ R O O T ) /  
a d m i n , P O R T = 1 4 4 3 , H O S T = l o c a l h o s t ; < I P   S A P   W e b   D i s p a t -  
c h e r > , C L I E N T H O S T = l o c a l h o s t ; < H o s t n a m e n   v o n   A d m i n i s t r a t o r   C l i -  
e n t s > , [ A L L O W P U B = F A L S E ]
```

Setzung des Wertes für PREFIX: URL-Präfix, für den dieser HTTP-Subhandler aufgerufen werden soll. Im Standard wird der URL-Präfix für die Verwaltung auf `/sap/admin` gesetzt.

Setzung des Wertes für DOCROOT: Wurzelverzeichnis der Administrationsdateien.

Setzung des Wertes für AUTHFILE: Dateiname der Authentifizierungsdatei (Defaultwert: `icmauth.txt`), in der die Hash-Werte der Passwörter für die Admin-Benutzer liegen. Wenn keine Benutzerauthentifizierung erfolgen muss (z.B. weil diese im Authorisation Handler bereits erfolgt), so kann dies mit der Angabe von `AUTHFILE=none` erreicht werden.

Setzung des Wertes für PORT (MUSS): Im Default werden Admin-Requests auf allen lokalen TCP/IP-Ports angenommen. Das ist nicht erlaubt. Eine Einschränkung auf einen TCP/IP-Port, der nur lokal erreichbar ist und auf denen Admin-Requests angenommen werden, muss umgesetzt werden. So kann die Verwendung von HTTPS leicht umgesetzt werden.

! Hinweis: Dieser Port muss über den Parameter `icm/server_port_<xx>` angegeben worden sein. Dafür muss ein HTTPS-Port eingerichtet werden, sonst werden die Administrator Passwörter bei der Anmeldung im Klartext übertragen. Unsichere Protokolle sind nicht erlaubt!

Beispiel:

```
icm/server_port_<xx>=.... PROT=HTTPS, PORT=1443, ....
```

Setzung der Werte für HOST (MUSS): Wird hier nichts angegeben, werden Admin-Requests von allen Rechnernamen akzeptiert. Die Einschränkung des Zugriffs auf den lokalen Host muss umgesetzt werden, damit können nur die lokalen Benutzer die webbasierte Oberfläche benutzen. Geben Sie dafür den Wert `localhost` oder `127.0.0.1` und die `<IP des SAP Web Dispatchers>` an.

Setzung der Werte für CLIENTHOST (optional): Wird hier nichts angegeben, werden Admin-Requests von jedem Clientrechner akzeptiert. Dies ist nicht erlaubt. Die Administration muss auf bestimmte Clientrechner (Rechner der SAP Web Dispatcher Administratoren) eingeschränkt werden. Geben Sie dafür den Wert `localhost` und die `<Rechnernamen oder IP-Adressen der Administratoren>` ein.

Setzung des Wertes für ALLOWPUB: Ist `ALLOWPUB=TRUE` gesetzt, so werden unter dem Pfad `"public/index.html"` öffentliche und lesende Zugriffe auf bestimmte Administrationsseiten ohne Anmeldung zugelassen (z.B. "Monitor", "Aktive Services", "Core Thread Status", "Hostname Buffer", "Release Information" und "MPI Status"). Der Zugriff auf diese Seiten ohne Anmeldung muss verhindert werden. Auch wenn der Zugriff mit den Subparametern `HOST` und `CLIENTHOST` eingeschränkt ist (s. oben), dürfen Systeminformationen, die für einen Angriff auf das System genutzt werden könnten, nur über eine Anmeldung am System einsehbar sein. `ALLOWPUB=FALSE` muss gesetzt werden.

ID: 3.22-174/13.0

Req 175 Die Autorisierungsdatei "icmauth.txt" muss vor unberechtigten Zugriffen geschützt sein.

In der Autorisierungsdatei (Defaultwert: icmauth.txt) stehen die Benutzer und ihre Gruppenzugehörigkeit im Klartext, die Passwörter sind verschlüsselt. Der Dateipfad ist entsprechend den Konventionen des zugrunde liegenden Betriebssystems. Diese Datei muss aus Sicherheitsgründen nur für den Administrations-Benutzer des SAP-Systems (<sid>adm) lesbar und änderbar sein.

Motivation: Das "icmauth.txt"-File ist gegen Fremdzugriffe zu schützen, um den Einfallsvektor für einen Angriff möglichst klein zu halten. Das Verschlüsselte Passwort ist leicht zu knacken und für einen unberechtigten Angriff zu nutzen.

Umsetzungsbeispiel: Setzen von restriktiven Zugriffsrechten auf die Autorisierungsdatei icmauth.txt mit folgenden Befehlen (UNIX):

```
chmod 700 <autorisierungsdatei>
```

Der Ablageort der Datei sowie deren Namen ist im Parameter icm/authfile spezifiziert.

ID: 3.22-175/13.0

Req 176 Der Zugriff auf das WebDispatcher Administrationsverzeichnis muss restriktiv gesetzt sein.

Das Administrationsverzeichnis des ICM muss restriktiv eingestellt sein. Das Administrationsverzeichnis ist im Standard mit <DIR_ICMAN_ROOT>/admin definiert und enthält die Dateien für die Administration über die Weboberfläche.

Motivation: In dem Administrationsverzeichnis befinden sich Daten (z. B. Datei icmauth.txt) mit einem hohen Schutzbedarf.

Umsetzungsbeispiel: Anpassen der Zugriffsrechte für das Administrationsverzeichnis mit folgenden Befehlen:

```
chown <sid>adm:sapsys <Verzeichnisname>
```

```
chmod 750 <Verzeichnisname> oder wenn möglich restriktiver =>  
700
```

ID: 3.22-176/13.0

Req 177 Die Web Administration erfolgt personifiziert und nicht über den User "icmadm".

Um den ICM von einem Browser aus zu überwachen und zu administrieren, benötigen Sie einen Benutzer.

Der User icmadm ist der Standard User für die Webadministrationsoberfläche und wird in der Datei icmauth.txt hinterlegt, wenn die Einrichtung des Administrationsuser durchgeführt wird.

Für die nachvollziehbare Administration werden personifizierte Administratoren angelegt.

Motivation: Unberechtigter Administrationszugriff kann zur Dateneinsicht und -änderung mit einem Systemausfall führen.

Umsetzungsbeispiel: Bei der Einrichtung der Administrationsuser müssen folgende Schritte ausgeführt werden:

1. Legen Sie als <sid>adm den icmadm Administrationsbenutzer mit wdispmon oder sapwebdisp -bootstrap an.
Hierdurch wird die Berechtigungsdatei icmauth.txt generiert.
2. Legen Sie personifizierte Administrationsuser mit icmon an.
3. Probieren Sie den Zugriff mit den personifizierten Usern.
4. Ändern Sie das Passwort des Standardusers icmadm gemäß den Passworrichtlinien.

ID: 3.22-177/13.0

Req 178 Der Zugriff auf das Kommando icmbnd zum Binden von Ports muss restriktive vergeben werden.

Der Zugriff auf das Kommando icmbnd zum Binden von Ports muss restriktive vergeben werden. Im Standardfall bindet der SAP Web Dispatcher die Ports selbst (empfohlen). Um die privilegierten Ports mit Portnummer 0 bis einschließlich 1023 („well known ports“) vom SAP Web Dispatcher binden zu lassen, muss das externe Bindeprogramm icmbnd genutzt werden. Durch Setzen der Option EXTBIND=1 bei der Parameterdefinition von icm/server_port_<xx> wird das Programm icmbnd vom SAP Web Dispatcher aufgerufen, um die Ports < 1023 zu setzen.

Das Programm icmbnd kann auch direkt aufgerufen werden, um Ports zu konfigurieren (nicht empfohlen).

Das Programm ist unter Pfad, der mit folgenden Parametern im Profil definiert wird, zu finden:

- exe/icmbnd
- DIR_EXECUTABL

Motivation: Unberechtigtes Ausführen des Programmes icmbnd kann zu einer Fehlkonfiguration. Damit können Angriffe über den SAP Web Dispatcher auf die nachgelagerten Systeme ausgeführt werden. Dies kann zu unberechtigten Zugriffen auf die Daten oder zu Systemausfällen führen.

Umsetzungsbeispiel: Setzen von eingeschränkten Zugriffsrechten auf die Datei icmbnd:

```
chown root:sapsys icmbnd
```

```
chmod 4750 icmbnd
```

ID: 3.22-178/13.0

Req 179 Es muss verhindert werden, dass im Standard erzeugte HTTP-Fehlermeldungen vom Server für den Client sichtbar werden.

Aus Sicherheitsgründen sollten Informationen über das System und/oder die Anwendung nicht an den Client weitergegeben werden.

Motivation: Über Fehlermeldung und Versionsstände, die angezeigt werden, kann festgestellt werden, welche Version einer Software genutzt wird. Mit einer einfachen Suche mit dem Versionsstand kann z. B. eine Exploit für eine Version einfach gefunden werden.

Umsetzungsbeispiel: Der Parameter is/HTTP/show_detailed_errors sollte auf "false" gesetzt werden. Überprüfung des SAP Web Dispatcher Profils:

```
is/HTTP/show_detailed_errors = FALSE
```

Weitere Informationen dazu finden Sie in der SAP Online Hilfe im Kapitel "Fehlerbehandlung durch den SAP Web Dispatcher".

ID: 3.22-179/13.0

Req 180 Die Informationen zum Server in der HTTP Antwort an den Client müssen limitiert werden.

Der Parameter steuert, ob das Server-Headerfeld bei HTTP-Antworten des Servers an den Client eingefügt werden soll oder nicht. Das Server-Header-Feld umfasst die Werte der Parameter is/server_name und is/server_version. Aus Sicherheitsgründen empfiehlt SAP diese Informationen nicht an den Client weiterzugeben.

Motivation: Es dürfen keine Informationen an den Client weitergegeben werden, die zu einem Angriff ausgenutzt werden könnten.

Umsetzungsbeispiel: Setzen Sie dafür den Parameter is/HTTP/show_server_header wie folgt:

is/HTTP/show_server_header = FALSE

ID: 3.22-180/13.0

Req 181 Das Sicherheitslogging muss für den SAP Web Dispatcher aktiviert sein

Mit dem Parameter `icm/security_log` können Sie die Ausgabe des Sicherheits-Protokolls aus dem ICM und SAP Web Dispatcher steuern.

Administratoren können das Sicherheitsprotokoll (`icm/security_log`) verwenden, um mögliche unberechtigte Zugriffe auf das System zu erkennen. Die folgenden Unregelmäßigkeiten werden protokolliert:

- Daten mit ungültiger Syntax
- Versuchter Zugriff auf Objekte, die nicht existieren (NOT found)
- Zugriff auf Objekte, die aufgrund von Filterregeln nicht erlaubt sind (permission denied)
- Anmeldefehler bei der Webadministration (im ICM und Web Dispatcher)

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Umsetzungsbeispiel: Setzen Sie den Parameter `icm/security_log` für die Aktivierung der Security Protokollierung.

Syntax des Parameters:

`icm/security_log = LOGFILE=<file name>, LEVEL=<security level>, MAXSIZEKB=<max size in KB>, SWITCHTF=<options>, FILEWRAP=on`

LOGFILE: Name der Datei. Durch Zeitangaben wird der Name eindeutig.

LEVEL: Gibt den Grad (1 - 3) der Protokollierung an. Mit Level 1 wird nur der Grund für den Eintrag protokolliert. Mit Level 2 werden zusätzliche Informationen über den Zustand der Verbindung und der Beginn der Daten, die den Eintrag verursachen, protokolliert (Standard, wenn LEVEL nicht angegeben wird). Mit Level 3 werden alle Daten, die den Eintrag verursachen, protokolliert.

MAXSIZEKB: Gibt die Größe der Protokolldatei in KB an. Wird diese Größe überschritten, wird die aktuelle Datei geschlossen und eine neue Datei (mit einem neuen Namen) geöffnet. Durch Definition von LOGFILE wird der Name eindeutig und damit wird ein Überschreiben verhindert.

SWITCHTF: Eine neue Protokolldatei kann nicht nur angelegt werden, wenn die Datei eine bestimmte Größe erreicht hat, sondern auch, wenn sich die Zeitdaten ändern (eine neue Datei stündlich, täglich oder monatlich möglich).

FILEWRAP: Wenn FILEWRAP=on aktiv ist, wird bei jedem Öffnen einer neuen Datei die bestehende Protokolldatei zurückgesetzt und überschrieben (!). Es gibt also immer nur eine Protokolldatei mit den aktuellen Protokolldaten. Wenn Sie diese Option weglassen, wird nach Überschreiten der Größe eine neue Datei geschrieben (siehe MAXSIZEKB und SWITCHTF).

Beispiel Setzung des Parameters (Option FILEWRAP wird **nicht** gesetzt!):

`icm/security_log = LOGFILE=dev_icm_seclog-%d-%m-%y_%h:%t:%s, LEVEL=2, MAXSIZEKB=10000, SWITCHTF=day`

ID: 3.22-181/13.0

Req 182 Das HTTP Logging muss aktiviert sein.

Mit den Parametern `icm/HTTP/logging_<xx>` (eingehende Requests) und `icm/HTTP/logging_client_<xx>` (ausgehende Requests) können Sie die Ausgabe des Verbindungs-Protokolls aus dem ICM und SAP Web Dispatcher steuern und protokollieren.

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Umsetzungsbeispiel: Setzen Sie folgende zwei Parameter:

1) Protokollierung eingehender Verbindungen

Syntax des Parameters:

icm/HTTP/logging_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file name> [, LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBytes>, SWITCHTF=<options>, FILEWRAP=on]

PREFIX: URL-Präfix, der für diesen HTTP-Subhandler aufgerufen wird (z. B. /).

LOGFILE: Name der Ausgabedatei im Dateisystem. Der ICM / Web Dispatcher prüft bei jedem Neustart, ob die als Logdatei angegebene Datei existiert. Ist dies der Fall, schreibt er weiter in diese Datei, ist dies nicht der Fall, legt er eine neue Datei an. An den eigentlichen Dateinamen können Sie einen anderen Zeitstempel anhängen, welches die Datei eindeutig macht.

LOGFORMAT: Es gibt die verschiedene Formate für Protokolldateien. Sie können das "NCSA Combined Log Format", das anderen Webservern bekannt ist, durch Verwendung der folgenden Formatzeichenfolge reproduzieren: %h %l %u %t "%r" %s %b "%{referer}i" "%{user-agent}i"

FILTER: Die Filtereigenschaft legt fest, dass eine HTTP-Anfrage nur dann protokolliert wird, wenn ein bestimmtes Header-Feld (z. B. ein HTTP-Header-Feld) in der Anfrage oder Antwort enthalten ist.

MAXSIZEKB: Gibt die Größe der Protokolldatei in KB an. Wird diese Größe überschritten, wird die aktuelle Datei geschlossen und eine neue Datei (mit einem neuen Namen) geöffnet. Durch Definition von LOGFILE wird der Name eindeutig und damit wird ein Überschreiben verhindert.

SWITCHTF: Eine neue Protokolldatei kann nicht nur angelegt werden, wenn die Datei eine bestimmte Größe erreicht hat, sondern auch, wenn sich die Zeitdaten ändern (eine neue Datei stündlich, täglich oder monatlich möglich).

FILEWRAP: Wenn FILEWRAP=on aktiv ist, wird bei jedem Öffnen einer neuen Datei die bestehende Protokolldatei zurückgesetzt und überschrieben (!). Es gibt also immer nur eine Protokolldatei mit den aktuellen Protokolldaten. Wenn Sie diese Option weglassen, wird nach Überschreiten der Größe eine neue Datei geschrieben (siehe MAXSIZEKB und SWITCHTF).

Beispiel Setzung des Parameters (Option FILEWRAP wird **nicht** gesetzt!):

icm/HTTP/logging_0 = PREFIX=/, LOGFILE=incoming_icm_log-%d-%m-%y_%h:%t:%s, LOGFORMAT=%h %l %u %t "%r" %s %b "%{referer}i" "%{user-agent}i", MAXSIZEKB=10000, SWITCHTF=day

2) Protokollierung ausgehender Verbindungen

Syntax des Parameters:

icm/HTTP/logging_client_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file name> [, LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBytes>, SWITCHTF=<options>, FILEWRAP=on]

Die Optionen sind dieselben wie bei icm/HTTP/logging_<xx>.

Beispiel Setzung des Parameters (Option FILEWRAP wird **nicht** gesetzt!):

icm/HTTP/logging_client_<xx> = PREFIX=/, LOGFILE=outgoing_icm_log-%d-%m-%y_%h:%t:%s, LOGFORMAT=%h %l %u %t "%r" %s %b "%{referer}i" "%{user-agent}i", MAXSIZEKB=10000, SWITCHTF=day

Achtung bezüglich der Parameter icm/HTTP/logging_<xx> und icm/HTTP/logging_client_<xx>: Die Protokolldateien für eingehende und ausgehende Anfragen müssen unterschiedliche Namen haben, d.h. Sie können nicht beide Richtungen in eine Datei schreiben.

ID: 3.22-182/13.0

2.12. Allgemeingültige Vorgaben für Datenbanken

Req 183	Der Zugriff auf die SAP BR*Tools muss eingeschränkt werden (gültig für Oracle und SAP HANA Datenbanken).
---------	--

SAP liefert die Tools BRBACKUP, BRARCHIVE, BRRESTORE, BRRECOVER, BRSPACE, BRCONNECT und BRTOOLS zur Verwaltung und zum Schutz der Daten in der eingesetzten Oracle und SAP HANA Datenbank aus. Weitere Information erhalten Sie über die SAP Hinweise 8523, 113747 und in der SAP Hilfe Dokumentation.

Motivation: Durch einen unberechtigten Zugriff auf die Dateien können die Dateiinhalte verändert werden. Die Änderungen können zu einem fehlerhaften Verhalten und/oder zu einem Ausfall führen.

Umsetzungsbeispiel: Die BR*Tools müssen vor unberechtigten Zugriff geschützt werden:

Tool Name	Verzeichnis	Berechtigung	Besitzer/Gruppe	Funktion
BRBACKUP	/sapmnt/<SID>/exe	4775	ora<dbsid>/sapsys	SAP Tool für Oracle/ HANA Datenbanken um die Datenbankda- teiensicherung einzu- planen
BRARCHIVE	/sapmnt/<SID>/exe	4775	ora<dbsid>/sapsys	Archivieren von Offli- ne Redo Log Dateien
BRCONNECT	/sapmnt/<SID>/exe	4775	ora<dbsid>/sapsys	Datenbankadministra- tions Tool (Säubern der Logdateien, Up- date der Statistiken, Anfügen von Next Ex- tents, Änderung des Passwortes der Da- tenbankbenutzer, ...)
BRRESTORE	/sapmnt/<SID>/exe	755	<sid>adm/sapsys	Restore der gesam- ten Datenbanksiche- rung oder nur Teile davon
BRRECOVER	/sapmnt/<SID>/exe	755	<sid>adm/sapsys	Wiederherstellung der Datenbank
BRSPACE	/sapmnt/<SID>/exe	755	<sid>adm/sapsys	Verwaltung des Plat- zes der Datenbank
BRTTOOLS	/sapmnt/<SID>/exe	755	<sid>adm/sapsys	User Interface: Samm- lung von Befehlen im aus dem Bereich DBA-Cockpit

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.22-183/13.0

Req 184 Datenbankfunktionen, welche einen Zugriffe auf Betriebssystemebene und/oder Netzwerkdienste ermöglichen, muss das Zugriffsrecht der Rollen/Gruppen „Public“ und/oder „Everyone“ entzogen werden.

Motivation: Motivation: Durch den Entzug der Rechte lässt sich ein unautorisierter Zugriff auf Netzwerkdienste und die Betriebssystemebene durch die Nutzung von Datenbankfunktionen verhindern. Funktionen zur Ausführung von Betriebssystembefehlen und zur Nutzung von Netzwerkdiensten steht somit nicht allen, sondern nur autorisierten Benut-

gern zur Verfügung.

ID: 3.22-184/13.0

Req 185	Nicht benötigte (Default-) Datenbanken auf dem Datenbanksystem müssen gelöscht werden
---------	---

Nicht benötigte (Default-) Datenbanken auf dem Datenbanksystem müssen gelöscht werden.

Motivation: Bei der Installation von Datenbanksystemen werden oftmals Test- oder Übungsdatenbanken installiert, die im Wirkbetrieb der Datenbank nicht mehr erforderlich sind. In der Vergangenheit zeigten diese Testdatenbanken Schwachstellen, die es einem Angreifer ermöglichen, privilegierte Rechte auf dem Datenbanksystem zu erlangen.

ID: 3.22-185/13.0

Req 186	Auf einer Betriebssysteminstanz dürfen nur Datenbankinstanzen mit gleichen Schutzbedarf betrieben werden.
---------	---

Auf einer Betriebssysteminstanz (mit physikalischer oder virtualisierter Hardware) dürfen nur Datenbankinstanzen betrieben werden, die

- den gleichen Schutzbedarf (Datenschutzklasse) haben,
- unter einer einzigen Kundenhoheit stehen und
- administrativ vom gleichen Personenkreis betrieben werden.

Motivation: Bei der Mehrfachausnutzung von Serverhardware durch mehrere Datenbanksysteme steigt die Gefahr, dass ein großer Personenkreis unberechtigterweise Zugriff auf Systeme hat, für die er weder fachlich noch administrativ zuständig ist.

Umsetzungsbeispiel: Zur Trennung der einzelnen Datenbankinstanzen können Administratoren physikalische Serverhardware, wie auch Virtualisierungslösungen einsetzen.

ID: 3.22-186/13.0

Req 187	Nicht benötigte SQL-Funktionen müssen gelöscht werden.
---------	--

Nicht benötigte, erweiterte SQL-Funktionen (z. B. T-SQL, PL/SQL, SQL PL, extended stored procedures) und/oder Pakete aus dem Datenbanksystem müssen gelöscht werden.

Motivation: Vielfach nutzt ein Angreifer solche Funktionen, um Schadcode auf das System zu transferieren oder Kommandos mit privilegierten Rechten auszuführen.

ID: 3.22-187/13.0

Req 188	Datenbankfunktionen, die den Zugriff auf Dateien des Betriebssystems erlauben, müssen gelöscht oder deaktiviert werden.
---------	---

Datenbankfunktionen, die den Zugriff auf Dateien des Betriebssystems erlauben, müssen gelöscht oder deaktiviert werden.

Motivation: Einige Datenbanksysteme erlauben den Zugriff auf die Betriebssystemebene über spezielle stored procedures oder SQL-Funktionen. Es sind zahlreiche Schwachstellen dieser Prozeduren bekannt, die es einem Angreifer erlauben, unauthorisiert Zugriff zu einem System zu erlangen oder Schadcode auf dem Zielsystem auszuführen. Häufig sind diese Prozeduren mit unnötig hohen, zumeist privilegierten Rechten versehen oder erlauben jedem die Ausführung ohne Authentisierung. Malware nutzt zum Beispiel die Prozeduren MS-SQL: xp_cmdshell und Oracle: utl_tcp.

ID: 3.22-188/13.0

Req 189	Datenbankfunktionen, die den Zugriff auf andere Netzdienste erlauben, müssen gelöscht oder deaktiviert werden.
---------	--

Datenbankfunktionen, die den Zugriff auf andere Netzdienste (z. B. SMTP, HTTP, SNMP, FTP etc.) erlauben, müssen gelöscht oder deaktiviert werden.

Motivation: Datenbanksysteme stellen zum Teil Funktionen zur Verfügung, die üblicherweise ein Applikationsserver anbietet. So ist es zum Beispiel möglich, über spezielle Funktionen E-Mails zu versenden oder Webanfragen an externe Systeme zu starten.

ID: 3.22-189/13.0

Req 190	Datenbankfunktionen die einen Zugriff auf das Betriebssystem oder Netzwerkdienst ermöglichen, muss das Zugriffsrecht entzogen werden.
---------	---

Datenbankfunktionen die einen Zugriff auf Betriebssystemebene oder Netzwerkdienste ermöglichen, muss das Zugriffsrecht der Rollen/Gruppen „Public“ und/oder „Everyone“ entzogen werden.

Motivation: Durch den Entzug der Rechte lässt sich ein unautorisierter Zugriff auf Netzwerkdienste und die Betriebssystemebene durch die Nutzung von Datenbankfunktionen verhindern. Funktionen zur Ausführung von Betriebssystembefehlen und zur Nutzung von Netzwerkdiensten steht somit nicht allen, sondern nur autorisierten Benutzern zur Verfügung.

ID: 3.22-190/13.0

Req 191	Datenbankzugriffe zwischen verschiedenen Datenbanksystemen müssen nach dem least privilege principle erfolgen.
---------	--

Die Zugriffsrechte müssen mit minimalen Privilegien ausgestattet sein. Der Benutzer darf nur die notwendigen Rechte zum Abfragen oder Extrahieren der erforderlichen Daten auf dem anderen Datenbanksystem haben. Nicht notwendige Rechte müssen dem Benutzer entzogen werden, damit er z. B. nicht unberechtigt andere Daten lesen oder verändern kann.

Motivation: Eine Beschränkung der Zugriffsrechte verkleinert die Angriffsfläche.

ID: 3.22-191/13.0

Req 192	Datenbankzugriffe zwischen verschiedenen Datenbanksystemen müssen über einen individuellen Benutzeraccount erfolgen.
---------	--

Datenbankzugriffe zwischen verschiedenen Datenbanksystemen müssen über einen individuellen Benutzeraccount erfolgen.

Motivation: Die Zugriffe müssen über einen speziellen Benutzer erfolgen. Nur so lassen sich die Zugriffsrechte individuell auf das unbedingt notwendige Maß begrenzen.

ID: 3.22-192/13.0

Req 193	Zugriffe auf Datenbanksysteme sowie kritische "stored procedure" und Datenbankinhalte müssen geloggt werden.
---------	--

Zugriffe auf Datenbanksysteme sowie kritische Datenbank-Procedures und Datenbankinhalte müssen geloggt werden.

Motivation: Für einen sicheren und nachvollziehbaren Betrieb einer Datenbank ist das Loggen wichtiger Betriebsinfor-

mationen notwendig. Hierzu zählt zum Beispiel das Erfassen von erfolglosen Loginversuchen, um mögliche Einbruchversuche zu entdecken. Die Protokollierung sicherheitsrelevanter Benutzeraktionen muss in Übereinstimmung mit dem jeweils gelten- den nationalen Recht erfolgen. Bei der Umsetzung von Maßnahmen, die sich aus dieser Anforderung ergeben, sind die jeweils geltenden Beteiligungsrechte der zuständigen Sozialpartner sowie die jeweils bestehenden betrieblichen oder tariflichen Regelungen zu beachten.

ID: 3.22-193/13.0

Req 194 Die Datenbankverschlüsselung muss aktiviert sein.

Die sensiblen Datenbankdaten müssen vor Zugriffen von Betriebssystemebene geschützt werden. Dies lässt sich über eine Datenbankverschlüsselung erreichen.

Motivation: Ein unberechtigter Zugriff über die Betriebssystemebene auf die Daten der Datenbank muss verhindert werden.

Umsetzungsbeispiel: Beispiel SAP HANA DB:

Um die auf der Festplatte gespeicherten Daten vor unberechtigtem Zugriff auf Betriebssystemebene zu schützen, unterstützt die SAP HANA-Datenbank die Datenverschlüsselung in der Persistenzschicht. Die Data-Volume-Verschlüsselung schützt den Datenbereich auf der Festplatte, während die Redo-Log-Verschlüsselung den Protokollbereich auf der Festplatte schützt.

Die Verschlüsselung von Daten- und Log-Volumes ist nicht standardmäßig aktiviert. Im Idealfall wird die Verschlüsselung in der Datenbank sofort bei der Erstellung aktiviert.

Richten Sie die Verschlüsselung in der Konfigurationsdatei global.ini in der Systemdatenbank ein:

Sektion [database_initial_encryption]

- persistence_encryption = on
- log_encryption = on

Wer die Verschlüsselung von Daten und Log-Volumes in der Tenant Datenbank aktivieren oder deaktivieren kann, hängt initial davon ab, wie der folgende Parameter eingestellt ist:

Sektion [database_initial_encryption]

- encryption_config_control = local_database (default)
kann der Administrator des Tenants die Verschlüsselung in der Tenant Datenbank mit der ALTER SY-STEM-Anweisung aktivieren oder deaktivieren.
- encryption_config_control = system_database
nur der Administrator der Systemdatenbank kann die Verschlüsselung Tenant Datenbank(en) mit der Anweisung ALTER DATABASE aktivieren oder deaktivieren.

Achtung: Bei einer nachträglichen Übergabe der Verantwortung der Verschlüsselung der Daten- und Log-Volumes von dem System Datenbankadministrators an den Tenant Datenbankadministrator oder umgekehrt muss über ein zusätzliches ALTER Statement erfolgen. Ein einfaches Umsetzen der Parameter reicht nicht!

Requirement-ID: 3e1d6282

ID: 3.22-194/13.0

2.13. Oracle

Req 195 Die Oracle-Datenbankrolle PUBLIC darf nicht geändert werden.

'PUBLIC' ist eine spezielle Datenbankrolle. Sie wird beim Anlegen der Datenbank erstellt und konfiguriert. Die Rolle wird nicht in DBA_ROLES oder SESSION_ROLES angezeigt.

Da PUBLIC für jeden Datenbankbenutzer zugänglich ist, sind alle Privilegien und Rollen, die PUBLIC erteilt wurden, für jeden Datenbankbenutzer zugänglich. Es ist daher nicht ratsam, PUBLIC zusätzliche Privilegien und Rollen zu er-

teilen.

Inhalt SAP Hinweis 2553347 erläutert die Probleme, wenn die Rolle PUBLIC geändert wird:

- Sie sollten der Datenbankrolle PUBLIC (in 11g und höher) keine Privilegien/Rollen entziehen.
- Sie sollten der Datenbankrolle PUBLIC keine Privilegien/Rollen erteilen.
- Wenn die Konfiguration der Rolle PUBLIC in Ihrer SAP-Datenbank geändert und PUBLIC Privilegien entzogen wurden, kann das Patching oder Upgrade Ihrer SAP-Datenbank fehlschlagen. Sie sollten die ursprüngliche Standardkonfiguration wiederherstellen, bevor Sie ein Patch für Ihre Datenbank einspielen oder ein Upgrade auf ein neueres Release durchführen.

Motivation: Unberechtigter Zugriff auf Datenbankinhalte muss verhindert werden, in dem die Rolle PUBLIC keine weiteren Berechtigungen erhält (Einhaltung Need-to-do- und Need-to-know-Prinzip).

Umsetzungsbeispiel: Standardmäßig hat PUBLIC keine erteilten Rollen oder Systemprivilegien. Überprüfen Sie das mit:

```
• SQL> select granted_role from dba_role_privs where grantee =  
      'PUBLIC';
```

=> Keine Zeilen ausgewählt

```
• SQL> select PRIVILEGE from dba_sys_privs where grantee =  
      'PUBLIC';
```

=> Keine Zeilen ausgewählt

Der Rolle PUBLIC werden nur Objektprivilegien erteilt. Welche Objektberechtigungen erteilt werden, hängt von Oracle-Release und installierten Datenbankkomponenten ab. In 12.2 werden PUBLIC mehr als 3000 Privilegien erteilt. Überprüfung möglich mit:

```
• SQL> select count(*) from dba_tab_privs where grantee = 'PUBLIC';
```

COUNT(*)

Weitere Hilfen sind dem SAP Hinweis 2553347 zu entnehmen.

ID: 3.22-195/13.0

Req 196	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss ein Schutz gegen Online-Angriffe wie Wörterbuch- und Brute-Force-Attacken vorhanden sein, der das Erraten von Passwörtern stark erschwert.
---------	--

Online Wörterbuch- und Brute-Force Angriffe haben zum Ziel, durch automatisiertes Ausprobieren an einer regulären Zugangsschnittstelle des Systems die Passwörter von Benutzerkonten zu erraten.

Um dies zu verhindern, muss eine der folgenden Maßnahmen oder eine Kombination aus diesen umgesetzt werden:

- mittels technischer Maßnahmen abgebildete Erzwingung einer Wartezeit nach einem fehlgeschlagenen Anmeldeversuch, bevor eine erneute Anmeldung ermöglicht wird. Die Wartezeit sollte sich zudem mit jedem weiteren fehlgeschlagenen Anmeldeversuch automatisch deutlich erhöhen (z. B. durch Verdopplung).
- automatisiertes Sperren des Benutzerkontos nach einer vorgegebenen Anzahl von aufeinanderfolgenden fehlgeschlagenen Anmeldeversuchen (typischerweise 5). Allerdings ist bei dieser Lösung zu bedenken, dass diese einen Entsperrungs-Prozess erforderlich macht und ein Angreifer dies gezielt nutzen kann, um Konten zu sperren und somit vorübergehend unbrauchbar zu machen.
- Verwendung von CAPTCHA ("Completely Automated Public Turing test to tell Computers and Humans Apart"), um automatisierte Anmeldeversuche von Maschinen ("Robots" oder "Bots") bestmöglich zu unterbinden. Ein CAPTCHA stellt dazu eine kleine Aufgabe, i. d. R. basierend auf grafischen oder akustischen Elementen, die von einer Maschine nur schwer zu lösen ist. Es ist entsprechend zu berücksichtigen, dass CAPTCHA proble-

matisch bzgl. der Barrierefreiheit sein können.

Um eine höhere Sicherheit zu erreichen, ist es oftmals sinnvoll, zwei oder mehrere der genannten Maßnahmen miteinander zu kombinieren. Dies ist abhängig vom Einzelfall zu prüfen und entsprechend umzusetzen.

Motivation: Ohne entsprechenden Schutz kann ein Angreifer durch bloßes Durchprobieren von Wörterbuchlisten oder automatisch erzeugten Zeichenkombinationen versuchen ein Passwort zu ermitteln, um so das entsprechende Benutzerkonto missbräuchlich zu nutzen.

Umsetzungsbeispiel: Setzen Sie folgende Parameter für die angegebenen Userprofile:

- Userprofil SAPUPROF (Profil des SAP Schemausers):
FAILED_LOGIN_ATTEMPTS = unlimited
PASSWORD_LOCK_TIME = unlimited

Damit soll ein Systemstillstand durch einen gesperrten SAP-Schema-User verhindert werden s.SAP Hinweise 1519872 und 951167.

- Alle anderen Userprofile z. B. DEFAULT-Profil:
FAILED_LOGIN_ATTEMPTS = 5
PASSWORD_LOCK_TIME = unlimited

ID: 3.22-196/13.0

Req 197	Falls Passwörter als Authentisierungsmerkmal für technische Benutzerkonten genutzt werden, müssen diese mindestens 30 Zeichen lang sein und drei der folgenden Zeichenkategorien beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.
---------	--

Technische Benutzerkonten zeichnen sich dadurch aus, dass sie nicht von Personen verwendet werden. Stattdessen werden sie für die Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt.

Ein System darf für technische Benutzerkonten nur Passwörter verwenden, welche die folgende Komplexität erfüllen:

- Mindestlänge 30 Zeichen
- Bestehend aus mindestens drei der folgenden vier Zeichenkategorien:
 - Kleinbuchstaben
 - Großbuchstaben
 - Ziffern
 - Sonderzeichen

Motivation: Technische Benutzerkonten sind aufgrund ihrer Verwendung in Machine-to-Machine (M2M) Kommunikations-Szenarien oft mit Berechtigungen ausgestattet, die für einen Angreifer von hohem Interesse sein können, um Infrastrukturen zu kompromittieren. Ohne Mechanismen einer umfassenden Kompromittierungserkennung kann dann das Risiko, dass ein Passwort von einem Angreifer ermittelt oder gebrochen wird, über die Zeit deutlich ansteigen. Eine signifikante Erhöhung der Passwort-Länge wirkt diesen Risiken entgegen und lässt sich zudem in M2M Szenarien besonders leicht umsetzen, da die Handhabung eines sehr langen Passworts für eine Maschine (im Gegensatz zu einer Person) keine besondere Herausforderung darstellt.

Umsetzungsbeispiel: Im SAP-Umfeld gibt es keine personifizierte DB-User. Auf Oracle-Ebene existieren einzig der SAP-Schema-User (DB-User der SAP-Anwendung) und die Oracle Standarduser. Personifizierter Zugriff auf die Datenbank erfolgt gesteuert über den Datenbank-OPSS-Mechanismus.

Setzen Sie folgende Parameter für alle Userprofile z. B. SAPUPROF und DEFAULT:

- PASSWORD_VERIFY_FUNCTION = ora12c_stig_verify_function

- Zudem muss bei der Passwortvergabe darauf geachtet werden, eine Passwortlänge von 30 Zeichen zu setzen!

Hinweis: Mit der "ora12c_stig_verify_function" wird beim Setzen eines Passwortes folgendes erzwungen:

- Passwortlänge mindestens 15 Zeichen => 30 Zeichen müssen verwandt werden!
- Jeweils ein Klein- und Großbuchstabe
- Eine Zahl
- Ein Sonderzeichen
- Das neue Passwort muss sich außerdem in mindestens 8 Zeichen vom alten unterscheiden.

ID: 3.22-197/13.0

Req 198	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss nach spätestens 12 Monaten ein Passwort-Wechsel erfolgen.
---------	---

Die maximal zulässige Nutzungsdauer für Passwörter beträgt 12 Monate.
Erreicht ein Passwort die maximal zulässige Nutzungsdauer, muss es geändert werden.

Das System muss hierzu den Benutzer automatisch bei dessen nächster Systemanmeldung auf die abgelaufene Nutzungsdauer hinweisen und ihn unmittelbar durch einen Passwort-Änderungsdialog führen. Der Zugriff auf das System darf ohne eine erfolgreich abgeschlossene Passwort-Änderung nicht weiter zugelassen werden.
Für technische Benutzerkonten (M2M bzw. Machine-2-Machine), die für die Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt werden, müssen ebenfalls automatisierte Lösungen zur Einhaltung der zulässigen Nutzungsdauer für Passwörter implementiert werden.

Falls eine automatisierte Abbildung des Prozesses zur Passwort-Änderung nicht umsetzbar ist, muss ersatzweise eine wirksame organisatorische Maßnahme angewendet werden, die eine verbindliche manuelle Passwort-Änderung bei Ablauf der zulässigen Nutzungsdauer sicherstellt.

Motivation: Passwörter sind im Gegensatz zu moderneren Authentisierungsmerkmalen leichter angreifbar. Ohne spezifische Maßnahmen einer verlässlichen technisch automatisierten Kompromittierungserkennung kann das Risiko, dass ein Passwort von einem Angreifer ermittelt oder gebrochen wird, über die Zeit deutlich steigen.

Umsetzungsbeispiel: Setzen Sie folgende Parameter für die angegebenen Userprofile:

- Userprofil SAPUPROF (Profil des SAP Schema-Users):
PASSWORD_LIFE_TIME = unlimited
PASSWORD_GRACE_TIME = 0

Damit soll ein Systemstillstand durch einen gesperrten SAP-Schema-User verhindert werden s.SAP Hinweise 1519872 und 951167.

- Alle anderen Userprofile z. B. DEFAULT-Profil:
PASSWORD_LIFE_TIME = 365
PASSWORD_GRACE_TIME = 0

ID: 3.22-198/13.0

2.14. HANA

Req 199	Der Zugriff von der Applikation auf die Datenbank muss verschlüsselt erfolgen.
---------	--

Die Verbindung zwischen der Anwendung und der SAP Datenbank muss verschlüsselt sein.

Motivation: In der Cloud kann nicht sichergestellt werden, dass eine unverschlüsselte Verbindung ausreichend sicher

ist.

Umsetzungsbeispiel:

- Verschlüsselung Anwendung - HANA DB (ODBC Verbindung): Erfolgt über einen von SAP ausgelieferten Treiber.
- Verschlüsselung JAVA-Anwendung - HANA DB (JDBC Verbindung): Erfolgt über einen von SAP ausgelieferten Treiber.

Die JAVA-Anwendung ermöglicht, sich über das JDBC Application Programming Interface (API) mit der SAP HANA Datenbank zu verbinden. Diese JDBC Verbindung ist im Default nicht verschlüsselt. Aktivieren Sie die JDBC TLS/SSL Verbindung zwischen dem JAVA-Anwendungsserver und der SAP HANA Datenbank in allen Anwendungen der SAP HANA Datenbank.

- <https://help.sap.com/viewer/b3ee5778bc2e4a089d3299b82ec762a7/2.0.04/en-US/0ae2b75266df44499d8fed8035e024ad.html?q=ODBC>
- <https://help.sap.com/viewer/b3ee5778bc2e4a089d3299b82ec762a7/2.0.04/en-US/dbd3d887bb571014bf05ca887f897b99.html?q=ODBC>
- <https://help.sap.com/viewer/b3ee5778bc2e4a089d3299b82ec762a7/2.0.04/en-US/64065bc94ccb4f64bfb544686db23e60.html?q=ODBC>

ID: 3.22-199/13.0

Req 200 Für die Anwendungen darf nur der PROD Space oder ein neuer Space genutzt werden.

Um eine Trennung umzusetzen, dürfen Anwendungen, die nicht zur Netweaver-Familie gehören, nicht im SAP Space deployed werden. Zudem darf die SpaceDeveloper Rolle nicht den Plattform Benutzern in dem SAP Space zugewiesen werden.

Motivation: Um den Angriffsvektor auf die SAP-Applikation gering zu halten, muss eine Trennung zwischen SAP-Netweaver-Systemen und anderen Applikation erfolgen.

Umsetzungsbeispiel: https://help.sap.com/docs/SAP_HANA_PLATFORM/742945a940f240f4a2a0e39f93d3e2d4/a752216352b14d1e90bc09f1f2107217.html?locale=en-US

ID: 3.22-200/13.0

Req 201 SAP-HANA-Studio muss in einem separaten VLAN installiert werden.

SAP-HANA-Studio muss in einem separaten VLAN installiert werden, damit sowohl die Basisadministration als auch die Applikationsadministration über das Werkzeug auf die Applikation zugreifen kann.

Motivation: HANA-Studio ist ein Administrationswerkzeug, dass nur für Administratoren erreichbar sein darf, um das System nicht zu gefährden.

ID: 3.22-201/13.0

Req 202 Anwendungen außerhalb der Multi-Target-Anwendungen (MTA) müssen streng getrennt von deren MTA Ressourcen sein.

Anwendungen außerhalb der Multi-Target-Anwendungen (MTA) müssen streng getrennt von deren MTA Ressourcen sein, damit jeder Space seinen eigenen dedizierten Betriebssystembenutzer hat.

Motivation: Einhaltung der Trennung bezüglich der Daten und Rollen

Umsetzungsbeispiel: Bei der Umsetzung des Konzeptes von Organisationen und Spaces in der Kombination mit einem separaten (dedizierten) Betriebssystembenutzer. Typischerweise besteht eine Mikroservice-getriebene Architektur einer Lösung aus einer Kooperation von verschiedenen Service Instanzen, die ihre dedizierte Aufgabe erfüllen. Nur im Zusammenspiel der Mikroservices kann die Lösung die nötigen Anforderungen erfüllen. Im XS Advanced Kontext arbeiten verschiedene Anwendungen Hand-in-Hand zusammen und bilden eine sogenannte Multi-Target-Anwendung (MTA). Wenn Ihre MTA zum Beispiel aus zwei Anwendungen besteht, einer User-Interface-basierten Anwendung, die auf Datenbanktabellen des Schemas A lesend zugreifen muss, und einer Anwendung die ebenfalls auf Datenbanktabellen des Schemas A lesend zugreifen muss. Beide Anwendungen müssen also auf Datenbanktabellen des Schemas A lesend zugreifen. Damit sollten beide Anwendungen einem gemeinsamen Berechtigungskonzept angehören mit demselben Benutzerpool. Alle anderen Anwendungen außerhalb des Raums dieser MTA sollten von den MTA Ressourcen ausgeschlossen bzw. isoliert sein, diese sollten keinen Zugriff auf diese MTA Daten oder MTA HTTP Endpunkte haben. Die Isolation des Raums der MTA ist abhängig von unterschiedlichen Betriebssystembenutzern. Die Räume verschiedener MTAs sollten unterschiedlichen Betriebssystembenutzer haben, denn nur durch unterschiedliche Betriebssystembenutzer sind die Räume der MTAs wirklich von einander getrennt bzw. isoliert und es ist gesichert, dass nur die Ressourcen eines Raums (MTA) sich zum Beispiel Storage, Userberechtigungen und Userpassworte teilen.

1. Benutzen Sie nicht den Benutzer <sid>adm oder keinen anderen hoch privilegierten Betriebssystembenutzer als Benutzer für einen Raum.
2. Schränken Sie privilegierte Benutzer des Raums soweit wie möglich ein.

Organisation	Raum	Raum (Space) Betriebssystembenutzer	Beschreibung
Initiale Organisation	SAP	sap<sid>xsa	Beinhaltet vorinstallierte SAP Anwendungen wie zum Beispiel: • Deploy-Service für MTAs • Produkt-Installer und Komponenten-Registry für die Anwendungsinstallation und Update(s) Dieser Raum ist Ort für andere System-relevante SAP Anwendungen wie das optionale Administrations-UI oder die Anwendung Rollen Builder.
Initiale Organisation	PROD	<sid>xsa	Empty after installation The space PROD can be used to deploy your custom applications.

ID: 3.22-202/13.0

Req 203 SAP HANA Sicherungen müssen verschlüsselt sein.

Überlegungen zur Verschlüsselung von Backups:

- Die Erstellung verschlüsselter Sicherungen dauert länger als die unverschlüsselter Sicherungen.
- Es dauert länger, eine Datenbank mit verschlüsselten Sicherungen wiederherzustellen als mit unverschlüsselten Sicherungen.

- Wenn die Verschlüsselung von Sicherungen aktiviert ist, werden sowohl Datensicherungen als auch Log-Sicherungen verschlüsselt.

! Wichtiger Hinweis: Gültig für Anwendungen auf Plattformen in T-Systems Rechenzentren:

Verschlüsselte Sicherungen können die Backupmenge und damit die Kosten massiv erhöhen (da keine Komprimierung/Dedublizierung mehr möglich ist). Für Anwendungen, welche auf einer Plattform in T-Systems Rechenzentren betrieben werden, ist diese Anforderung auf "nicht zutreffend" zu stellen, da jeglicher TSI-Storage verschlüsselt betrieben werden muss.

Eine Verschlüsselung der Sicherung und der Log-Sicherung hat in diesem Fall keinen zusätzlichen Effekt, da der T-Systems OS- und der SAP Basis Betrieb Zugriff auf die laufende Datenbank haben und ebenfalls Zugriff auf die Schlüssel der Sicherungsverschlüsselung.

Motivation: Die Verschlüsselung von Backups schützt die Privatsphäre der SAP HANA-Geschäftsdaten, indem sie verhindert, dass Unbefugte den Inhalt von Backups lesen können.

Umsetzungsbeispiel: Die Verschlüsselung von Sicherungen ist nicht standardmäßig aktiviert. Im Idealfall wird die Verschlüsselung der Datenbank Sicherungen in der Datenbank sofort bei der Erstellung aktiviert.

Die Verschlüsselung der Sicherungen wird durch den folgenden Parameter in der Konfigurationsdatei global.ini in der Systemdatenbank gesetzt:

Sektion [database_initial_encryption]

- backup_encryption = on

Wer die Verschlüsselung von Sicherungen in der Tenant Datenbank aktivieren oder deaktivieren kann, hängt initial davon ab, wie der folgende Parameter eingestellt ist:

Sektion [database_initial_encryption]

- encryption_config_control = local_database (default)
kann der Administrator des Tenants die Verschlüsselung der Sicherungen in der Tenant Datenbank mit der ALTER SYSTEM-Anweisung aktivieren oder deaktivieren.
- encryption_config_control = system_database
nur der Administrator der Systemdatenbank kann die Verschlüsselung der Sicherungen der Tenant Datenbank(en) mit der Anweisung ALTER DATABASE aktivieren oder deaktivieren.

Achtung: Bei einer nachträglichen Übergabe der Verantwortung der Verschlüsselung der Sicherungen von dem System Datenbankadministrators an den Tenant Datenbankadministrator oder umgekehrt muss über ein zusätzliches ALTER Statement erfolgen. Ein einfaches Umsetzen des Parameters reicht nicht!

ID: 3.22-203/13.0

2.14.1. HANA: Datenbank User Rollen und Berechtigungen

Req 204 Der Datenbankadministrator SYSTEM muss nach der Installation deaktiviert werden.

Verwenden Sie den Datenbankadministrator SYSTEM, um einen für Ihre Aufgaben erforderlichen Benutzer mit den Mindestberechtigungen anzulegen. Danach muss der Datenbankbenutzer SYSTEM deaktiviert werden.

Motivation: Der Datenbankadministrator SYSTEM ist der leistungsfähigste Datenbankbenutzer mit unwiderruflichen Systemrechten. Der Datenbankadministrator "SYSTEM" ist nach der Installation aktiv und muss für den Regelbetrieb deaktiviert sein.

ID: 3.22-204/13.0

Req 205 Auf einem Produktionssystem darf die Standardrolle DEVELOPMENT nicht vergeben werden, da Sie "ALTER SYSTEM"-Kommandos enthält.

Auf einem Produktionssystem darf die Standardrolle DEVELOPMENT nicht vergeben werden, da Sie "ALTER SYSTEM"-Kommandos enthält. Der "ALTER-SYSTEMS" Befehl ist ein Statement, die Objekte verändern können. In einem Produktionssystem dürfen nur die User SYSTEM und _SYS_REP diese Berechtigung besitzen.

Motivation: Mit ALTER SYSTEM-Statement können weitreichende Veränderungen in einem System gemacht werden und darf deshalb nur restriktiv vergeben werden.

ID: 3.22-205/13.0

Req 206 Die Rolle "DATA ADMIN" darf keinem Benutzer in einem Produktionssystem zur Verfügung stehen.

Das Systemprivileg DATA ADMIN ist ein mächtiges Privileg und darf nicht in produktiven Umgebung als Rolle verwendet werden. Es berechtigt einen Benutzer, alle Daten in Systemsichten zu lesen und alle DDL-Befehle (Data Definition Language) in der SAP-HANA-Datenbank auszuführen. Standardmäßig haben nur die Benutzer SYSTEM und _SYS_REPO dieses Privileg.

ID: 3.22-206/13.0

Req 207 Repository Berechtigungen müssen restriktiv behandelt und regelmäßig verifiziert werden.

Repository Berechtigungen müssen restriktiv behandelt und regelmäßig verifiziert werden. Der Standardbenutzer SYS_REPO hat automatisch alle diese Rollen. Je nach Installation können auch Repository-Berechtigungen an Standardbenutzer SYSTEM vergeben sein.

Motivation: Anwendungsspezifische Repository-Rollen sollten nur an Anwendungsb Benutzer vergeben werden.

Umsetzungsbeispiel: SAP HANA wird mit einem Satz vorinstallierter Softwarekomponenten ausgeliefert, die als SAP HANA Web-Anwendungen, Bibliotheken und Konfigurationsdaten implementiert sind. Die für die Verwendung dieser Komponenten erforderlichen Berechtigungen sind in Repository Rollen enthalten. Die mit der Komponente selbst ausgeliefert werden.

Der Standardbenutzer _SYS_REPO hat automatisch alle diese Rollen. Einige können auch automatisch an den Standardbenutzer SYSTEM vergeben werden, damit Werkzeuge wie das SAP-HANA-Cockpit sofort nach der Installation verwendet werden können.

Repository Rollen können sich bei einer neuen ausgelieferten Version des Paketes ändern. Sie sollten nicht genutzt werden, sondern sollten nur als Vorlage zur Erstellung eigener Rollen verwendet werden. Sie sollten über einen Review Prozess regelmäßig verifiziert werden, ob diese nur die nötigen Berechtigungen besitzen.

Falls Repository Paket Berechtigungen einer eigenen Rolle zugewiesen wurden, können diese Berechtigungen restriktiver und besser an die Anforderungen der Organisation angepasst werden. Das ist viel besser als würde das gesamte Repository Paket als Vorlage für Rollen verwendet. Deshalb sollte für jede Repository Berechtigung (REPO.*), die einer Rollenvorlage angehört und dem Paket .REPO_PACKAGE_ROOT zugewiesen ist, überprüft werden, ob diese zu einem einzelnen Paket zugewiesen sein darf oder zugewiesen sein sollte oder lieber eine kleine Anzahl spezifischer Pakete lieber als das gesamte Repository.

ID: 3.22-207/13.0

Req 208 Es muss sichergestellt sein, dass keinem Benutzer DEBUG-Berechtigungen zugewiesen sind.

Die Berechtigungen DEBBUG und ATTACH DEBUGGER dürfen keinem Nutzer des Systems zugewiesen sein.

Motivation: Debug und Attach Debug sind Funktionen (Rollen), die für Entwickler wichtige Informationen ausgeben. Auf Produktionssystemen sollten Funktionen nicht verwendet werden, neben den erweiterten Rechten, kann es zu erheblichen Performanceeinbußen kommen.

ID: 3.22-208/13.0

Req 209 Die Rolle SAP_INTERNAL_HANA_SUPPORT darf nur an den SAP Support vergeben werden.

Die Rolle SAP_INTERNAL_HANA_SUPPORT darf nur vom Benutzer:System verwendet werden oder muss einem anderen Benutzer zugewiesen sein, er nicht regelmäßig im System verwendet wird.

Motivation: Mit der Rolle SAP_Internal_Hana_Support werden automatisch administrative Berechtigungen für das System vergeben.

Umsetzungsbeispiel: Die Rolle wird Beispielsweise im Supportfall von SAP angefordert.

ID: 3.22-209/13.0

2.14.2. HANA: Netzwerkschnittstelle

Req 210 Nur die benutzten SQL- und HTTP Ports dürfen geöffnet werden.

Es muss sichergestellt sein, dass die Ports nur für die entsprechende Instanz geöffnet sind. Während der Installation werden Ports wie SQL 3<instance_no>15 und HTTP 80<instance_no> standardmäßig geöffnet.

Motivation: Nur die Ports, die für die Ausführung Ihres SAP-HANA-Szenarios benötigt werden, sind zu öffnen.

ID: 3.22-210/13.0

Req 211 Systemreplikationen und Backups müssen über einen separaten, internen Netzwerkkanal konfigurieren sein.

Systemreplikation und Backups müssen über einen separaten interne Netzwerkkanal konfigurieren sein [system_replication_communication - Listeninterface-Parameter .internal] abgesichert sein. Alternative ist es möglich den Listeninterface-Parameter global (.global) zu setzen. Wenn der Listenparameter verwendet wird muss die Kommunikation (über TSL/SSL) abgesichert sein.

Motivation: Die empfohlene Einstellung hängt davon ab, ob ein separates Netzwerk für die interne Kommunikation definiert ist oder nicht:

- Wenn ein separater interner Netzwerkkanal für die Systemreplikation konfiguriert ist, sollte der Parameter [system_replication_communication] Listeninterface-Parameter .internal sein. Außerdem müssen Sie Schlüssel-Wert-Paare für die IP-Adressen der Netzwerkadapter für die Systemreplikation im Abschnitt [system_replication_communication] internal_hostname_resolution hinzufügen.
- Wenn kein separates Netzwerk für die Systemreplikation konfiguriert ist, sollte der Parameter [system_replication_communication] listeninterface-Parameter auf .global gesetzt werden. In diesem Fall ist es jedoch wichtig, die Kommunikation über TSL/SSL zu sichern und/oder die SAP-HANA-Landschaft mit einer Firewall zu schützen. Setzen Sie außerdem den Parameter [system_replication_communication] allowed_sender, um die mögliche Kommunikation auf bestimmte Hosts einzuschränken. Der Parameterwert muss eine Liste der fremden Hosts enthalten, die Teil der SAP-HANA-Systemreplikationslandschaft sind.

ID: 3.22-211/13.0

2.14.3. HANA: Applikation und File System

Req 212 Um Dateien zu schützen, die auf dem Server exportiert werden, muss der Parameter "import_export" in der indexserver.ini angepasst werden.

Die Zugriffsberechtigung von Dateien, die auf den SAP-HANA-Server exportiert werden, kann mit dem Parameter "import_export" in der Konfigurationsdatei indexserver.ini konfiguriert werden. Die Standardberechtigung ist 640

([import_export] file_securityparameter=medium).

Motivation: Keine Anzeige von Dateien im Verzeichnis.

Umsetzungsbeispiel: Setzen des Parameters file_security in der indexserver.ini Konfigurationsdatei.
file_security=<Wert>

ID: 3.22-212/13.0

Req 213 Der Log-Output der "audit trail" funktion muss ins Syslog schreiben.

SAP schreibt in unterschiedlichste Log-Verzeichnisse. Es muss sichergestellt sein, dass mindestens der Log-Output vom "Audit Trail" ins Syslog des Systems schreibt.

Motivation: SAP schreibt in unterschiedlichste Log-Verzeichnisse. Es muss sichergestellt sein, dass die Informationen die für den Betrieb benötigt werden, diesem auch zur Verfügung stehen.

ID: 3.22-213/13.0

Req 214 Die vom System automatisch erzeugten Dumpfiles, müssen in einem Bereich in das File-System geschrieben werden, in dem nur privilegierte Benutzer Zugriff haben.

Die vom System automatisch erzeugten Dumpfiles, müssen in einem Bereich im File-System geschrieben werden, in dem nur privilegierte Benutzer Zugriff haben. Dumpfiles können schützenswerte Informationen enthalten. Gleiches gilt für Dumpfiles, die manuell für die Fehlersuche erzeugt werden.

Motivation: Die automatisch erzeugten Dumpfiles können Informationen enthalten die schützenswert sind.

Umsetzungsbeispiel: RTE dump files must be generated by the <sid>adm user. To create RTE dump files in a running system as part of a full system information dump in the SAP HANA studio, a user requires the EXECUTE privilege on procedure SYS.FULL_SYSTEM_INFO_DUMP_CREATE.

Dump files are stored in the trace directory and have the same access permissions as other trace files (see above).

Runtime dump files created as part of a full system information dump can be retrieved by users with the EXECUTE privilege on the procedure SYS.FULL_SYSTEM_INFO_DUMP_RETRIEVE using the SAP HANA studio. At operating system level, any user in the SAPSYS group can access their storage location: /usr/sap/SID/SYS/global/sapcontrol/snapshots

ID: 3.22-214/13.0

Req 215 Es muss sichergestellt sein, dass alle Daten in einer Hana-Datenbank (inclusive der Tenants), dem Betriebsmodel und den damit verbunden Schutzbedarf entsprechen.

Es muss sichergestellt sein, dass alle Daten in einer Hana-Datenbank (inclusive der Tenants), dem Betriebsmodel und den damit verbunden Schutzbedarf entsprechen. Über den Master-Token können auch Daten in den Tenants gelesen werden.

Motivation: Sicherstellen das alle Daten in einer Datenbank den Schutzbedarf entsprechend verarbeitet werden.

ID: 3.22-215/13.0

Req 216 Datenbankfunktionen, die einen direkten Zugriff auf das Dateisystem oder Netzwerkschnittstelle ermöglichen, müssen deaktiviert werden.

Um das System zu schützen, müssen Sie bestimmte Datenbankfunktionen deaktivieren. Insbesondere den direkten Zugriff auf das Dateisystem, das Netzwerk oder auf andere Ressourcen müssen deaktiviert sein. Der richtige Weg ist,

Import- und Exportvorgänge dafür zu nutzen.

Motivation: Überprüfen Sie die Liste der Funktionen. Deaktivieren Sie diejenigen, die in Ihrem Implementierungsszenario nicht erforderlich sind.

ID: 3.22-216/13.0

Req 217 Auf Applikationsebene muss der Zugriff für Betriebssystem-Accounts gesperrt sein.

Bei der Installation wird ein Betriebssystem-Account für SAP-HANA angelegt, der für alle weiteren Prozesse auf Betriebssystemebene benutzt wird. Dieser Benutzer darf nicht im Regelbetrieb genutzt werden.

Motivation: Der Benutzer <sid> adm verfügt über sehr hohe Berechtigungen und muss geschützt werden.

Umsetzungsbeispiel: https://help.sap.com/viewer/666e73444beb4f1582ec605e3ea95aaa/CURRENT_VERSION/en-US/f35e144d63224704831653b1eff76ddd.html

ID: 3.22-217/13.0

Req 218 Für Aufgaben auf Betriebssystemebene müssen ausschließlich personenbezogene Benutzer eingesetzt werden.

Es muss unterschiedliche Rollen und Berechtigungen geben, um Aufgaben auf der Applikations- oder Betriebssystemebene durchzuführen.

Motivation: Es muss sichergestellt sein, dass die Administration des Betriebssystems und der Applikation unterschiedlichen Accounts benutzen und somit unterschiedlichen Betriebsmodelle abgebildet werden können.

ID: 3.22-218/13.0

Req 219 Der Inhalt der HANA-DB Verzeichnisse muss vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Der Inhalt der HANA-DB Verzeichnisse muss vor nicht autorisiertem Zugriff auf Betriebssystemebene geschützt werden.

Motivation: Ein Zugriff auf Dateien und Verzeichnisse innerhalb des Verzeichnisses können dazu führen, dass Information und Daten verändert werden können oder die Applikation und Datenbank sich fehlerhaft verhalten.

Umsetzungsbeispiel:

OS-Verzeichnisname	OS-Zugriffsberechtigung	OS-User/OS-Gruppe
/hdb/<(DB)SID>/	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/sapdata*	750	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/sapdata*/mnt0000<m>	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/sapdata*/mnt0000<m>/hdb0000<n>/datavolume 0000.dat	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/saplog	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/saplog/mnt0000<m>	755	<(db)sid>adm/sapsys

/hdb/<(DB)SID>/saplog/ mnt0000<m>/hdb0000<n>	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/saplog/ mnt0000<m>/hdb0000<n>/logseg- ment 000.dat	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/hdbicm	755	root/root
/hdb/<(DB)SID>/HDB<instancexx>	750	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/profile	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/global	750	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/exe	550	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/hdbstudio	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/hdbclient	755	root/root
/hdb/<(DB)SID>/Im structure	755	<(db)sid>adm/sapsys
/hdb/<(DB)SID>/federation	750	<(db)sid>adm/sapsys
/usr/sap/<(DB)SID>	755	<(db)sid>adm/sapsys
/usr/sap/hostctrl	755	root/sapsys
/usr/sap/SMDAgent	775	<(smdagent)sid>adm/sapsys

OS-Kommandos zum Setzen der OS-Zugriffsberechtigungen und des OS-Users und der OS-Gruppe:

```
chown <OS-User>:<OS-Gruppe> <Verzeichnisname>
chmod <OS-Berechtigung> <Verzeichnisname>
```

Beispiel:

```
chown <(db)sid>adm:sapsys /usr/sap/<(DB)SID>
chmod 755 /usr/sap/<(DB)SID>
```

ID: 3.22-219/13.0

2.14.4. HANA: Multitenant

Req 220 Für jeden Mandant (im Sinne eines Kunden/Auftraggebers) muss auf der SAP HANA Instanz ein Tenant angelegt und eingerichtet werden.

Um auf Datenbankebene die Mandantentrennung bei HANA einzuhalten, muss jeder Mandant einen eigenen Tenant benutzen. Eine Mandantentrennung über Berechtigungen ist auf Datenbankebene nicht zulässig.

Motivation: Im Standard laufen alle Datenbankprozesse unter dem Standard Betriebssystembenutzer <sid>adm bei einem geringen Isolations Modus. In diesem Modus sind Cross-Tenant Attacks über das Betriebssystem möglich. Um die Sicherheit in einem Multi Tenant Datenbank Container zu erhöhen, ist der Modus High Isolation zu konfigurieren. Mit dem Modus der High Isolation laufen die Betriebssystemprozesse einer individuellen Tenant Datenbank unter dem individuellen Betriebssystembenutzer mit einer dedizierten Gruppe. Die Prozesse der System Datenbank laufen weiterhin unter dem <sid>adm Benutzer.

U m s e t z u n g s b e i s p i e l : <https://help.sap.com/viewer/78209c1d3a9b41cd8624338e42a12bf6/2.0.04/en-US/a92f66e5a6564c1bafd33244b94558dc.html>

ID: 3.22-220/13.0

Req 221 Wenn mehr als ein Tenant genutzt wird, muss der "High-Isolation-Modus" eingesetzt werden, um zu verhindern das Cross-Datenbank Attacken über Betriebssystemmechanismen erfolgreich sind.

Um die Sicherheit in einem Multi Tenant Datenbank Container zu erhöhen, muss der "High Isolation"-Modus benutzt werden. In dem Modus laufen die Betriebssystemprozesse einer individuellen Tenant Datenbank unter dem individuellen Betriebssystembenutzer mit einer dedizierten Gruppe.

Motivation: Durch Migration von anderen Datenbanken in einen SAP HANA Multi Tenant Datenbank Container (MDC) können die Kosten für Infrastruktur und das Datenbankmanagement reduziert werden. Das SAP HANA Multi Tenant Datenbank Konzept erlaubt das Betreiben von multiplen Tenant Datenbanken in einem SAP HANA System. Im Standard laufen alle Datenbankprozesse unter dem Standard Betriebssystembenutzer <sid>adm bei einem geringen Isolations Modus. In diesem Modus sind Cross-Tenant Attacken über das Betriebssystem möglich. Um die Sicherheit in einem Multi Tenant Datenbank Container zu erhöhen, ist der Modus High Isolation zu konfigurieren. Mit dem Modus der High Isolation laufen die Betriebssystemprozesse einer individuellen Tenant Datenbank unter der individuellen Betriebssystembenutzer mit einer dedizierten Gruppe. Die Prozesse der System Datenbank laufen weiterhin unter dem <sid>adm Benutzer.

Umsetzungsbeispiel: Um den Modus "High Isolation" zu aktivieren, müssen verschiedene Schritte ausgeführt werden. Neben der Erstellung Tenant-dedizierter Betriebssystembenutzer und -gruppen muss das Python-Skript "convertMDC.py" ausgeführt werden. Außerdem muss eine HANA interne Verschlüsselung aufgesetzt werden. Für den genauen Ablauf der Aktivierung von dem Modus "High Isolation" gehen Sie bitte auf die SAP Hilfe in den "SAP HANA Tenant Databases".

<https://help.sap.com/viewer/78209c1d3a9b41cd8624338e42a12bf6/2.0.04/en-US/a92f66e5a6564c1bafd33244b94558dc.html>

ID: 3.22-221/13.0

Req 222 Zugriffe von einem Tenant auf andere Tenants müssen verboten sein.

Zugriffe zwischen zwei unterschiedlichen Tenants müssen verboten sein.

Motivation: Einhaltung und Umsetzung der Mandantentrennung (Mandant = Tenant) zum Schutz vor Ausspähung und Veränderung von Daten eines anderen Mandanten.

Umsetzungsbeispiel: Öffnen Sie auf der Übersichtsseite der System Datenbank in dem SAP HANA Cockpit die Konfiguration der Systemeigenschaften, in durch Klicken auf den korrespondierenden Administration Link. Wählen Sie die Konfigurationsdatei "global.ini" aus und gehen Sie zur Section "cross_database_access" zum Parameter "enable". Setzen Sie die Einstellung von "TRUE" auf "FALSE".

ID: 3.22-222/13.0

Req 223 Im Umfeld des Datenbank-Tenants und des Redo-Logs muss die Berechtigung zur Änderung von systemrelevanten Schlüsseln in separaten Berechtigung/Rollen liegen.

Die Berechtigung zur Änderung der Schlüssel, die im Umfeld des Datenbank-Tenants und des Redo-Logs verwendet werden, müssen in separaten Rollen/Berechtigungen abgelegt werden. Die Berechtigungen sind schützenswert und dürfen nicht an Benutzer gebunden werden, die im Regelbetrieb benutzt werden.

ID: 3.22-223/13.0

Req 224 Bei Multi-Tenant Verwendung muss der Port 17 geschützt sein.

Beim Einsatz einer MDC (Multi Tenant Datenbank Container) ist Port 17 auf dem System (Host der Datenbank) zu schützen, anderenfalls ist die komplette Datenbank incl. der Tenants zu erreichen.

Motivation: Über Port 17 wird die Datenbank via SQL angesprochen.

2.14.5. HANA: Extended Services (XS) Engine

Req 225	Der XS-Server-Port muss vor direktem Zugriff geschützt werden.
---------	--

Der XS-Server muss geschützt werden, da über ihn die Datenbank via SQL direkt angesprochen werden.

Motivation: Durch die Öffnung des XS-Ports ist das System angreifbar. Über das http-Protokoll ist es möglich auf die Datenbank zu zugreifen.

ID: 3.22-225/13.0

Req 226	Der XS-Dienst darf keine selbst signierten Zertifikate nutzen.
---------	--

Standardmäßig läuft der XS-Server mit selbstsigniertem Zertifikat für alle Domänen. Es ist nicht erlaubt den XS-Service im "Standard" zu betreiben.

Motivation: Der XS-Server wird von Usern genutzt, um auf eine SAP HANA Datenbank zuzugreifen. Durch die Verwendung von Zertifikaten die vom Browser überprüfen werden können, werden Fehlermeldung beim Benutzer vermieden und eine Akzeptanz sowie das Gefühl von Sicherheit für den Dienst aufgebaut.

Umsetzungsbeispiel: Einsatz von vertrauenswürdigen Zertifikaten.

ID: 3.22-226/13.0

Req 227	Bei Verwendung von SAP HANA und dem XS-Port, darf das Berechtigungskonzept des ABAP-Stacks nicht umgangen werden.
---------	---

Es ist sicherzustellen, dass in der SAP HANA Datenbank die Berechtigungen, die im ABAP-Stack festgelegt wurden, weitergeführt werden.

Motivation: Es ist möglich, ein in ABAP aufgesetztes Berechtigungskonzept über die XS-Engine zu umgehen. Das Berechtigungskonzept der XS-Engine ist unabhängig vom ABAP-Berechtigungskonzept.

ID: 3.22-227/13.0

2.14.5.1. HANA: Benutzer

Req 228	Die Standardpasswörter der Datenbankstandardbenutzer müssen entsprechend der Passwortrichtlinie gesetzt sein.
---------	---

Vordefinierte XS Systemuser sind (ein Login ist möglich):

XSA_ADMIN
HDI_BROKER_CONTROLLER
sap_sb

Vordefinierte technische SAP HANA XS Benutzer sind (ein Login ist nicht möglich):

SYS_XS_RUNTIME
SYS_XS_UAA
SYS_XS_UAA_SEC
SYS_XS_HANA_BROKER
SYS_XS_SBSS
_SYS_DI
_SYS_DI_*_CATALOG
_SYS_DI_SU
_SYS_DI_TO

Die vordefinierten XS Systembenutzer werden automatisch während der Installation generiert. Die vordefinierten technischen Benutzer werden während der Installation und Konfiguration erstellt.

Motivation: Unkontrolliertes und unberechtigtes Arbeiten mit Standardbenutzern ist nicht erlaubt. Es kann zu Dateneinsicht und -Datenveränderungen und zu Systemausfällen führen.

Umsetzungsbeispiel: Das Kommando zum Ändern der Passwörter lautet:

```
ALTER USER <user> {FORCE_FIRST_PASSWORD_CHANGE ::= PASSWORD <password>}
```

Um den Benutzerstatus zu überprüfen, können Sie folgendes Kommando (listet alle auf HANA existierenden Benutzer) ausführen:

```
select * from "SYS"."USERS";
```

Der Status der Benutzer kann auch mit dem SAP HANA Studio kontrolliert werden.

ID: 3.22-228/13.0

Req 229	Für ein HANA System muss eine Minimalkonfiguration umgesetzt sein.
---------	--

Für eine Hana System muss die Mindestkonfiguration Umgesetzt sein:

Installieren Sie eigene SSL Zertifikate (XS Trust-Zertifikate und XS setzen der Zertifikat Kommandos)
Bestimmen Sie mindestens einen XS Advanced User als OrgManager für jede einzelne Organisation
Registrierung aller benötigten Service Brokers
Erstellung aller benötigten Shared Domänen
Erstellung aller benötigten Custom Buildpacks
Erstellung aller benötigten Laufzeitumgebungen
Konfigurieren Sie logische Datenbanken
Setzen Sie die globalen Umgebungsvariablen (xs set_running | staging_environment_variable_groups command)

Motivation: Verhinderung von unerlaubten Zugriffen durch nicht benötigte Komponenten.

Umsetzungsbeispiel: Überprüfen Sie den Status der XS Engine auf OS Level:

```
xs service-brokers  
xs domains  
xs domain-certificates  
xs buildpacks  
xs runtimes  
xs tenant-databases
```

Überprüfen Sie die die Setzung der Parameter mit folgendem SQL Kommando:

```
SELECT KEY, VALUE FROM M_INIFILE_CONTENTS WHERE KEY = '<parameter>'
```

ID: 3.22-229/13.0

Req 230	Es muss ein Benutzer- und Berechtigungskonzept für die XS Engine umgesetzt sein.
---------	--

Es muss ein Benutzer- und Berechtigungskonzept für die XS Engine unter der Berücksichtigung der Aufgabentrennung umgesetzt sein. Dabei muss darauf geachtet werden, dass eine Benutzertrennung zwischen folgenden Rollen besteht:

- XS_AUTHORIZATION_ADMIN (Rollenmanagement, Rollensammlungsmanagement)
- XS_USER_ADMIN (Zuweisung der Rollensammlung zu Benutzern)

Motivation: Es muss verhindert werden, dass die Policies (need-to-know, need-to-do, segregation of duties) umgangen werden.

Umsetzungsbeispiel: Überprüfen Sie, ob die Rollen nur verschiedenen Benutzern zugewiesen wurden:

```
SELECT DISTINCT USER_NAME FROM USER_PARAMETERS WHERE PARAMETER =  
'XS_RC_XS_CONTROLLER_ADMIN'
```

ID: 3.22-230/13.0

Req 231	Der Benutzer XSA_ADMIN muss gesperrt sein.
---------	--

Der XSA_ADMIN Benutzer kann den Controller ohne jede Einschränkung benutzen. Es ist der einzige Benutzer, der das initiale Set Up der Models durchführen kann und muss aus diesen Gründen gesperrt sein.

Motivation: Verhinderung des Missbrauchs des XSA_ADMIN Benutzers.

Umsetzungsbeispiel: Sperren Sie den Benutzer XSA_ADMIN mit folgenden SQL Statement:

```
ALTER USER XSA_ADMIN DEACTIVATE USER NOW
```

Im Notfall kann der Benutzer mit dem folgenden SQL Statement wieder entsperrt werden:

```
ALTER USER XSA_ADMIN ACTIVATE USER NOW
```

Hinweis: Nur Benutzer mit der Systemberechtigung USER ADMIN können diese Aktion durchführen.

ID: 3.22-231/13.0

Req 232	Die Anmeldung an XS Advanced-Service muss mit einem personalisierten Betriebssystembenutzer erfolgen.
---------	---

Die Anmeldung an XS Advanced-Service muss mit einem personalisierten Betriebssystembenutzer erfolgen. Dieser muss ein HOME-Verzeichnis besitzen. Diese darf nicht für andere Betriebssystembenutzer lesbar sein.

Motivation: Verhinderung der Zugriffe über anonyme Defaultbenutzer.

Umsetzungsbeispiel: Ändern der Zugriffsrechte auf das HOME-Verzeichnisses für XS advanced Benutzer mit folgenden Befehlen:

```
chmod 700 <verzeichnisname>
```

ID: 3.22-232/13.0

2.14.5.2. HANA: Extended Services (XS) EngineBenutzer auf Betriebssystemebene

Req 233	Auf die Standardbenutzer auf Betriebssystemebene darf nicht direkt zugegriffen werden.
---------	--

Mit dem XS Engine Set Up werden zwei OS Benutzer <sid>xsa und sap<sid>xsa erstellt.

<sid>xsa: Benutzer wird für das "staging and running applications" für den vordefinierten "PROD space" genutzt

sap<sid>xsa: Benutzer wird für das "staging and running applications" in den vordefinierten "SAP space" genutzt

ID: 3.22-233/13.0

2.15. Anbindung von externen Webservices

Req 234	Grundlegende Sicherheitsfunktionen für extern zugängliche Web Services müssen in der DMZ implementiert werden und je nach Kritikalität mit einem Web Service Gateway geschützt werden.
---------	--

Eine DMZ ist eine Zone, in der sämtlicher externer Verkehr terminiert wird.

Datenbanken und Applikationen werden in der MZ aufgebaut, das heißt in einer Zone in der kein direkter externer Zugriff möglich ist.

Folgende Funktionen müssen auf der Zugangs- und Validierungsebene (DMZ) realisiert werden:

- Authentisierung und Autorisierung für den Zugriff (d.h., ist der Konsument des Web Services zum Zugriff berechtigt?)
- grundsätzliche Validierung auf Basis der vorliegenden Definition wie z. B. WSDL, JSON Schemata oder XSD
- Vor- und Nachbearbeitungsaufgaben (z. B. Terminierung einer TLS Verschlüsselung oder sicherheitsrelevante Protokollierung)
- Terminierung von TCP-Verbindungen und Blockieren aller direkten Zugriffe auf Web Service-Endpunkte in der MZ
- Limitieren der maximalen Verbindungen

Um diese Funktionalitäten zu gewährleisten sollte immer ein Web Service Gateway eingesetzt werden.

Motivation: In der DMZ müssen alle Maßnahmen ergriffen werden, um die Services in der MZ gegen Angriffe abzusichern.

Umsetzungsbeispiel:

- Authentisierung und Autorisierung für den Zugriff (d.h., ist der Konsument des Web Services zum Zugriff berechtigt)
- grundsätzliche Validierung auf Basis der vorliegenden Definition wie z. B. WSDL, JSON Schemata oder XSD
- Vor- und Nachbearbeitungsaufgaben (z. B. Terminierung einer TLS Verschlüsselung oder sicherheitsrelevante Protokollierung)
- Terminierung von TCP-Verbindungen und Blockieren aller direkten Zugriffe auf Web Service-Endpunkte in der MZ
- limitieren der maximalen Verbindungen

ID: 3.22-234/13.0

Req 235	Werden bei einem Web Service besonders schützenswerte Daten verarbeitet, so müssen diese Daten durch Einsatz von durchgängigen Mechanismen auf der Anwendungsebene (End-to-End), wie z. B. XML-Verschlüsselung, individuell geschützt werden.
---------	---

Da Transportverschlüsselungen bereits auf äußeren Schichten terminiert werden, sowie möglicher Payloadprotokollierung und "Hop by Hop" Kommunikation bietet eine reine Transportverschlüsselung in den meisten Fällen keinen ausreichenden Schutz für besonders vertrauliche Daten.

Beispiele für besonders schützenswerte Daten sind Gesundheitsdaten, Führungszeugnis, Kontodaten von Personen, Quartalszahlen vor Veröffentlichung und Vertragsentwürfe mit sehr hohem Volumen.

Motivation: Aufgrund des besonderen Schutzbedarfes ist für bestimmte Daten selbst dann Vertraulichkeitsschutz erforderlich, wenn sie transportverschlüsselt oder über sichere Netze übertragen werden.

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

- Vertraulichkeit

ID: 3.22-235/13.0

Req 236 Der Öffentlichkeit darf nur die kleinstmögliche Anzahl an Web Services zugänglich gemacht werden.

Insbesondere bei der Weitergabe eines internen Web Services an einen externen Partner oder bei der Erweiterung einer Legacy Anwendung um ein Web Service Frontend muss darauf geachtet werden, dass nur erforderliche Web Services weiter gegeben werden.

Motivation: Indem nur die minimale erforderliche Anzahl an Web Services angeboten wird, wird die Angriffsfläche reduziert.

Umsetzungsbeispiel: Auf Internet Web Service Gateways dürfen nur die Web Services verfügbar gemacht werden, die aus dem Internet heraus erreichbar sein müssen.

Gleiches gilt z. B. für die ausgehenden Verbindungen eines Container Namespaces.

ID: 3.22-236/13.0

Req 237 Anbieter und Konsumenten eines Web Service müssen sich gegenseitig authentifizieren und autorisieren, wenn schutzbedürftige Daten übertragen werden.

Der Anbieter eines Web Services muss sicherstellen, dass der Konsument berechtigt ist auf diesen Web Service zuzugreifen und diese Daten erhalten bzw. diese Funktionen verwenden darf.

Der Konsument muss sicherstellen, dass er mit dem korrekten Anbieter kommuniziert.

Berechtigungsprüfungen müssen in jeder Funktion berücksichtigt werden. Es darf nicht auf einen besonderen Vertrauensstatus aufgrund der Authentisierung eines Gerätes auf Ebene des Netzwerkes oder von Verbindungen aus z. B. dem gleichen Netzwerk oder Namespace vertraut werden.

Die Berechtigungsprüfung muss sicherstellen, dass der Konsument nur die Daten erhält, die er erhalten darf

Motivation: Um eine unerlaubte Nutzung von Ressourcen oder Ausgabe von Daten zu verhindern, ist eine angemessene Authentisierung und Autorisierung erforderlich.

Ein Zugriff auf einen nicht schutzbedürftigen Web Service muss nicht authentifiziert oder autorisiert werden.

Umsetzungsbeispiel: Um generelle Berechtigungen für den Zugriff auf einen Web Service zu erteilen, eignet sich z. B. eine zertifikatsbasierte Authentifizierung / Autorisierung.

Um eine Berechtigung auf Datenfeldebene zu erteilen eignen sich z. B. JW-Tokens mit individuellen "Claims", in Verbindung mit Filtern, um nur die für den Konsumenten berechnete Datenfelder herauszugeben.

Der JW-Token oder das Zertifikat muss durch jede Funktion / Container / Microservice überprüft werden.

ID: 3.22-237/13.0

Req 238 Der Mechanismus zur Authentifizierung und Autorisierung muss auf starken kryptografischen Algorithmen / Frameworks basieren.

Starke kryptografische Frameworks/Algorithmen in diesem Sinne sind zum Beispiel XML-Signaturen oder TLS-ClientZertifikate mit geeigneten Algorithmen wie SHA3- / SHA2 Hashes und ECDSA/RSA-Signaturen.

Weitere Beispiele sind JW-, OAuth- und STS-Tokens, sofern über TLS übertragen.

Motivation: Schwache Algorithmen können von Angreifern kompromittiert und Identitäten vorgetäuscht werden

Umsetzungsbeispiel: Zertifikate gemäß der Zertifikatsanforderungen für HTTPS.

JW Tokens:

- Hash Verfahren SHA 256 oder höher
- keine "none" Algorithmen ({"alg":"none"})
- ES, RS Algorithmen gegenüber HS bevorzugen

- möglichst kurze Lebenszeiten

ID: 3.22-238/13.0

Req 239 Alle Web Service Anfragen und Antworten müssen durch den Web Service Anbieter und Konsumenten gegen eine detaillierte Spezifikation validiert werden.

Jeder durch einen Web Service übermittelte Datensatz muss mit den erwarteten Datenfeldern, der erwarteten Länge und / oder dem erwarteten Bereich übereinstimmen, sowie ggf. mit einer formellen Spezifikation zur Beschreibung der akzeptablen Daten.

Sollten die erwarteten Datenfelder nicht immer fest definierbar sein, dürfen nur erwartete Datenfelder verarbeitet werden.

Motivation: Die detaillierte Spezifikation (einschließlich formelle Spezifikation) ermöglicht eine weitaus bessere Beschreibung der Daten, die in Datenfelder eines Web Services erwartet werden.

Umsetzungsbeispiel: Eine komplette Web Service Beschreibung kann durch z. B. WSDL 2.0 oder Swagger erreicht werden.

Beispiele für Datenformatbeschreibungen sind JSON Schemata oder XML Schemata Definitions.

Ein Beispiel für eine formelle Spezifikation sind reguläre Angaben wie z. B. "[A-Za-z0-9]+[. _+&#%/=~])*[A-Za-z0-9]+@[A-Za-z0-9]+[.]+[A-Za-z]{2,6}" zur Spezifizierung einer gültigen E-Mail-Adresse.

ID: 3.22-239/13.0

Req 240 Enthält der Web Service keine formelle Spezifikation der Eingabedaten, muss mittels Black- / Whitelists verhindert werden, dass illegale Zeichen akzeptiert werden.

Daten, die von den nachfolgenden Teilen der Anwendung nicht erwartet werden, können unerwartete und unerwünschte Auswirkungen haben.

Wenn die Eingabedaten in der DMZ ordnungsgemäß gefiltert werden, werden solche Angriffe erheblich erschwert.

Motivation: Durch Black- / Whitelisting kann effektiv vor gewissen Angriffsarten, wie z. B. SQL-, LDAP-, XML-, XPath-, XQuery-, Code-, Kommandoinjektion, geschützt werden.

Der Einsatz von Whitelists ist Blacklists vorzuziehen, da Blacklists im Laufe der Zeit tendenziell obsolet werden.

Umsetzungsbeispiel: Für Web Services, die über einen Webserver geführt werden stellt das Open-Source-Tool mod_security eine gute Lösung dar.

ID: 3.22-240/13.0

Req 241 Antworten von REST Web Services müssen auf HTTP Status Codes basieren.

Das HTTP Protokoll liefert für jeden Status einer Verbindung standardisierte Codes.

Weitere zusätzliche Informationen wie z. B. die empfangenen Daten (Payload) oder technische Details zu dem Fehler dürfen nicht ausgegeben werden, sondern sind mit einer eindeutigen Fehlerreferenz im Protokoll abzuspeichern. Die Fehlerreferenz darf ausgegeben werden.

Kategorie

1xx: Informational

2xx: Success

Beschreibung

Transfer Prokoll Informationen

Zeigt an, dass die Anfrage erfolgreich angenommen wurde

3xx: Redirection	Zeigt an, dass der Anfragende zusätzliche Maßnahmen ergreifen muss, um seine Anfrage zu vervollständigen
4xx: Client Error	Der Client scheint einen Fehler zu haben
5xx: Server Error	Der Server scheint einen Fehler zu haben

Motivation: Das HTTP Protokoll liefert bereits für jeden Status einer Verbindung standardisierte Codes. Durch das Verwenden dieser Status Codes wird verhindert, dass schutzbedürftige Informationen, z. B. Fehler- und Systemmeldungen, in Ausgaben enthalten sind. Ebenso kann der Konsument des Web Services anhand dieser standardisierten Codes generisch feststellen, woran der Fehler liegt. Durch das Widerspiegeln der empfangen Daten (Payload) können Sicherheitslücken entstehen (besonders bei Webanwendungen).

Umsetzungsbeispiel:

Ein Beispiel einiger HTTP Status codes und deren Bedeutung:

HTTP Status Code	Nachricht	Bedeutung
200	OK	Die Anfrage wurde erfolgreich bearbeitet
400	Bad Request	Die Anfrage-Nachricht war fehlerhaft aufgebaut
406	Not Acceptable	Der vom Client angefragte Inhaltstyp im Accept-Header wird vom Web Service nicht angeboten
413	Payload to large	Die gestellte Anfrage war zu groß

Alle HTTP Status Codes und deren Verwendung sind im RFC7231 spezifiziert.

ID: 3.22-241/13.0

Req 242 Funktionalitäten von REST Web Services müssen auf HTTP Methoden basieren.

Das HTTP Protokoll liefert standardisiert alle Methoden, die für sämtliche Funktionalitäten benötigt werden. Durch das Verwenden dieser Methoden wird eine einheitliche und transparente Berechtigungssteuerung ermöglicht,

die durch Web Server, Reverseproxys und Web Service Gateways unterstützt wird.

Motivation: Durch das Verwenden standardisierter Methoden wird vorgelagerten Schichten wie z. B. Web Servern, Reverseproxys und Web Service Gateways ermöglicht grundsätzliche Validierungen der Anfragen durchzuführen. Ebenso sorgt dies für ein einheitliches Verständnis für Konsumenten, wie ein Web Service zu verwenden ist.

Umsetzungsbeispiel: Beispiele für die gängigsten HTTP Methoden:

HTTP Methode	Beschreibung
GET	Die GET-Methode fordert die Übertragung der aktuell ausgewählten Ressource an
HEAD	Die HEAD-Methode ist identisch mit GET, außer dass der Server nicht den HTTP-BODY der Antwort sendet
POST	Die POST-Methode fordert die Verarbeitung der übermittelten Daten an
PUT	Die PUT-Methode fordert die Erstellung oder den Ersatz vom Zustand der Zielressource
DELETE	Die DELETE-Methode fordert das Entfernen der Zielressource an
PATCH	Die PATCH-Methode fordert eine Reihe von Änderungen an, die in der Anfrage beschrieben sind

ID: 3.22-242/13.0

Req 243 Vertrauliche Daten dürfen nicht in der URL übertragen werden.

Vertrauliche Daten dürfen ausschließlich außerhalb der URL übertragen werden (z. B. HTTP-BODY, HTTP-HEADER). Bei der Verwendung von HTTP-Headern dürfen nur Felder verwendet werden, die nicht in Protokolldateien erfasst werden.

Beispiele für vertrauliche Daten sind:

- API Keys
- Security Tokens (z. B. OAuth-, JW-Token)
- Passwörter

Motivation: Sämtliche in der URL enthaltenen Daten erscheinen in Protokolldateien von z. B. Web Server, Reverse Proxy und Web Service Gateway. Protokolldateien werden oft an zentraler Stelle gesammelt oder auch an SIEM Systeme zur Überwachung übertragen. Um dies zu ermöglichen und gleichzeitig vertrauliche Daten zu schützen, dürfen diese nicht in der URL übertragen werden.

Umsetzungsbeispiel: Protokolldateien erfasst werden. Beispiele sind:

- API Keys

- Security Tokens (z. B. OAuth, JW-Token)
- Passwörter

ID: 3.22-243/13.0

Req 244 Inhaltstypen (Content-Type) müssen validiert werden.

Eine REST Abfrage und Antwort muss mit dem beabsichtigten Inhaltstyp in der Kopfzeile übereinstimmen. Anderenfalls könnte dies zu Fehlinterpretationen auf der Konsumenten- / Anbieterseite führen und eine Code-Injektion / Code-Ausführung zur Folge haben.

Motivation: REST Web Services erlauben oft mehrere Datenformate zur Anfrage und Antwort.

Der Consumer muss das Datenformat seiner Anfrage, sowie das gewünschte Datenformat für die Antwort des Web Services in den "Content-Type" und "Accept" Headern bestimmen.

Der Provider darf die Anfrage nur verarbeiten, sofern er das Datenformat für die Anfrage und für die Antwort unterstützt.

Umsetzungsbeispiel: Anfragen mit unerwarteten oder fehlenden Inhaltstyp-Headern mit HTTP Code 415 "Nicht unterstützter Medientyp" ablehnen.

ID: 3.22-244/13.0

Req 245 Wird ein Web Service-Request mit Signaturen geschützt, dürfen die Signaturdaten nicht von einem Intermediate Processor aus dem Request entfernt werden.

Die Signatur muss über den ganzen Übertragungsweg hinweg validiert werden, um die Integrität und Authentizität der Web Service Anfragen / Antworten zu garantieren.

Motivation: Wird diese Signatur entfernt, ist die Integrität und Authentizität des Web Services nicht mehr gewährleistet.

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

- Integrität

ID: 3.22-245/13.0

Req 246 Ereignisse müssen mit genauem Zeitstempel protokolliert werden und je nach ihrer Kritikalität Alar-
me auslösen.

Die Protokollierung von Benutzern / Benutzeraktionen, sowie die Payloads sind mit dem Datenschutz und der Datensicherheit abzustimmen.

Ereignisse müssen entsprechend ihrer Ereignisart eine Ereignis-Kennzeichnung erhalten.

Sicherheitsrelevante Ereignisse müssen besonders gekennzeichnet werden um bedarfsgerecht Alarme auslösen zu können.

Als Ereignis-Kennzeichnung eignen sich z. B. „info“, „error“, „warning“, „alert“, „emergency“.

Anhand dieser Ereignis-Kennzeichnungen können automatisiert Alarme ausgelöst werden und Fehlerbehebungen erleichtert werden.

Ebenso ist es möglich, anhand dieser Ereignis-Kennzeichnungen Löschfristen und Zugriffsberechtigungen zu steuern.

Motivation: Damit ein sicherer Betrieb sowie eine Fehleranalyse möglich ist, muss ein "Logging, Monitoring und Alarming" Konzept erstellt werden.

Sicherheitsrelevante Ereignisse müssen sofort an einen geeigneten Betrieb weitergeleitet werden, der das Problem analysieren und Gegenmaßnahmen ergreifen kann.

Umsetzungsbeispiel: Die Ereignis-Kennzeichnung sowie die Notwendigkeit eines Alarms richtet sich nach dem Schutzbedarf und den Schnittstellen, über die der Web Service erreichbar ist.

Bei einem Web Service, der aus dem Internet erreichbar ist, muss eher mit Validierung und Autorisierungsfehlern gerechnet werden. Deshalb ist hier ein Alarm nicht immer zielführend. Sollte dies jedoch bei einem Webservice auftreten, der nur aus vertrauenswürdigen Bereichen erreichbar ist, z. B. aus dem eigenen Namespace, Administrative Sondernetze oder M2M Verbindungen, kann das ein Indiz für einen vorgedrungenen Angreifer sein.

Ebenfalls kann die Notwendigkeit für einen Alarm sich nach der Häufigkeit des Ereignis richten.

Ein Zugriffsversuch auf eine Ressource ohne entsprechende Berechtigung oder ein Validierungsfehler kann bei einmaligen Auftreten irrelevant sein. Sollte dies gehäuft auftreten, kann das ein Indiz für einen gezielten Angriff auf die Applikation sein oder es liegen Fehlfunktionen der Applikation vor.

Die folgende Tabelle gibt Beispiele für Ereignisse, Ereignis-Kennzeichnungen und auch den zu speichernden Logkontext.

Kategorie	Ereignis	Kennzeichnung	Logkontext
Zugriff	Zugriff auf den Web Service	info	Applikations-, Container-ID, Authentizität, IP
Validierungsfehler	falsches Datenformat	error, debug	Applikations-, Container-ID, Authentizität, IP, Payload
	Falsches Encoding	error, debug	Applikations-, Container-ID, Authentizität, IP, Payload
	Falsche Datenfeldnamen	error, debug	Applikations-, Container-ID, Authentizität, IP, Payload
	Prüfung gegen formelle Spezifikation schlägt fehl	error, debug	Applikations-, Container-ID, Authentizität, IP, Payload
	Falscher Wertebereich	error, debug	Applikations-, Container-ID, Authentizität, IP, Payload
Ausgabefehler	Datenbank / zuliefernde Systeme nicht erreichbar	error, debug	Applikations-, Container-ID, Authentizität, Anfrage-ID, IP
	Fehler in den Datensätzen der Datenbank	error, debug	Applikations-, Container-ID, Authentizität, Anfrage-ID, IP
	Ausgabe unvollständig	error, debug	Applikations-, Container-ID, Authentizität, Anfrage-ID, IP
Autorisierungsfehler	Zugriff auf Ressourcen ohne entsprechende Autorisierung	warning	Applikations-, Container-ID, Authentizität, IP, Payload
	Signaturvalidierungsfehler bei JW Token	alert	Applikations-, Container-ID, Authentizität, IP, Payload
	Durchführen von Aktionen die nicht auf die Rolle / ACL passen	alert	Applikations-, Container-ID, Authentizität, IP, Payload
Authentifizierung	Fehlschläge	warning	Applikations-, Container-ID, IP

	Erfolg	info	Applikations-, Container-ID, IP
Privilegierte Aktionen	Hinzufügen / Löschen von Accounts	trace	Applikations-, Container-ID, Authentizität, IP
	Ändern von Privilegien	trace	Applikations-, Container-ID, Authentizität, IP
	Exportieren von schützenden Daten	trace	Applikations-, Container-ID, Authentizität, IP
	Import von Daten	trace	Applikations-, Container-ID, Authentizität, IP
Laufzeitfehler	Ändern von Konfigurationen	trace	Applikations-, Container-ID, Authentizität, IP
	Absturz / Neustart der Applikation / Container	emergency	Applikations-, Container-ID, Stacktrace
	Konnektivitätsprobleme	emergency	Applikations-, Container-ID
	Performance Probleme	alert	Applikations-, Container-ID
	Viren Erkennung z. B. File Upload	alert	Applikations-, Container-ID, Authentizität, IP

ID: 3.22-246/13.0

Req 247 Ein SOAP Request der nicht sofort abgewickelt werden kann, muss die Vertraulichkeit und Integrität der zwischengespeicherten Nachricht sicherstellen.

Generell kann eine Warteschlangenbildung aus verschiedenen Gründen stattfinden, z. B. wenn die Systeme in der MZ nicht schnell genug antworten.

Im Allgemeinen sollte eine Warteschlangenbildung nach Möglichkeit in der MZ erfolgen. Falls ein SOAP Request oder Antwort nicht sofort abgewickelt werden kann und in der DMZ zwischengespeichert wird, muss die Vertraulichkeit und Integrität der zwischengespeicherten Nachricht sichergestellt werden.

Motivation: Ohne zusätzliche Verschlüsselung kann ein Angreifer Zugriff auf den Inhalt der zwischengespeicherten Nachrichten erhalten.

Umsetzungsbeispiel: Die Nachrichten können AES verschlüsselt in einer Datenbank zwischengespeichert werden. Um die Integrität zu gewährleisten muss zusätzlich ein Hashwert über die Nachricht gebildet werden.

ID: 3.22-247/13.0

2.16. Der Netweaver als Webapplikation

Req 248 Der Webserver-Dienst darf nur an eine Schnittstelle gebunden werden.

Der Webserver-Dienst darf nur an die Schnittstellen gebunden werden, über die er erreichbar sein muss. In den meisten Fällen muss der Webserver-Dienst nur an eine Schnittstelle gebunden werden.

Motivation: Je mehr Schnittstellen Zugriff zum Webserver erlauben, desto höher ist ein Angriffsrisiko

ID: 3.22-248/13.0

Motivation: Die Technik „Server Side Includes (SSI)“, die als zusätzliches ladbares Modul in den meisten Webserverprodukten implementiert ist, kann potenziell von Angreifern verwendet werden. Insbesondere die „exec“-Funktion von SSI könnte zur Ausführung von Systembefehlen verwendet werden, was ein Risiko darstellt.

ID: 3.22-252/13.0

Req 253	Wenn WebDAV für das Schreiben von Dateien verwendet wird, darf Zugriff nicht ohne erfolgreiche Authentifizierung gewährt werden.
---------	--

Falls WebDAV für das Schreiben von Dateien verwendet wird, darf Zugriff nicht ohne erfolgreiche Authentifizierung gewährt werden.

Motivation: WebDav ermöglicht den Online-Update von Content, der durch den Webserver zur Verfügung gestellt wird. Diese Funktion könnte daher zur Veränderung von Website-Inhalten missbraucht werden.

Umsetzungsbeispiel: Die Zugriffsrechte auf alle Dateien, auf die über WebDAV zugegriffen werden kann, müssen möglichst restriktiv gesetzt werden. Zusätzlich muss der WebDAV-Zugriff auf die benötigten Verzeichnisse beschränkt werden.

ABAP: ICM Baum: Überprüfung, ob Service "dav" aktiv.
TA SICF => Service ausführen => Baum Default host => sap => bc => ecm

ID: 3.22-253/13.0

Req 254	Falls WebDAV verwendet wird, muss der Zugriff auf erforderliche Verzeichnisse und bezüglich der berechtigten Anwender beschränkt werden.
---------	--

Die Zugriffsrechte auf alle Dateien, auf die über WebDAV zugegriffen werden kann, müssen möglichst restriktiv gesetzt werden. Zusätzlich muss der WebDAV-Zugriff auf die benötigten Verzeichnisse beschränkt werden

Motivation: WebDav ermöglicht den Online-Update von Content, der durch den Webserver zur Verfügung gestellt wird. Diese Funktion könnte daher zur Veränderung von Website-Inhalten missbraucht werden.

ID: 3.22-254/13.0

Req 255	Default-Inhalte müssen entfernt werden.
---------	---

Default-Inhalte (Beispiele, Hilfedateien, Dokumentation, Aliase), die mit der Standardinstallation bereitgestellt werden, müssen entfernt werden.

Motivation: Durch die Verwendung von Beispielen könnten Informationen über die installierte Software (Version) erlangt werden oder ausführbare Beispiele Sicherheitsschwachstellen enthalten.

ID: 3.22-255/13.0

Req 256	Die Erstellung von Verzeichnislisten (Indizierung) muss deaktiviert werden.
---------	---

Die Erstellung von Verzeichnislisten (Indizierung) muss deaktiviert werden.

Motivation: Verzeichnislisten enthalten Informationen über Dateien und Verzeichnisstrukturen, die missbraucht werden könnten.

ID: 3.22-256/13.0

Req 257 Der HTTP-Header darf keine Informationen zur Version des Webserverns enthalten.

Informationen über den Webserver in HTTP-Headern müssen auf ein Mindestmaß beschränkt werden. Der HTTP-Header darf keine Informationen zur Version des Webserverns und verwendeter Module/Erweiterungen enthalten.

Motivation: Alle Informationen über den Webserver könnten über Sicherheitsschwachstellen Aufschluss geben.

Umsetzungsbeispiel: Use the following settings to reduce the server information to a minimum:

- Set the parameter `is/HTTP/show_server_header` to FALSE. When you change this, the "Server:" header field is no longer set in HTTP responses.
- Set the parameter `is/HTTP/show_detailed_errors` to FALSE. After you do this, the system does not return any details about the error to the client.

ID: 3.22-257/13.0

Req 258 Informationen über den Webserver in Fehlerseiten, die durch den Webserver ausgeliefert werden, müssen entfernt werden.

Informationen über den Webserver in Fehlerseiten, die durch den Webserver ausgeliefert werden, müssen entfernt werden. Die Standardfehlerseiten müssen durch benutzerspezifisch definierte Fehlerseiten ersetzt werden. Benutzerspezifisch definierte Fehlerseiten dürfen keine Versionsinformationen des Webserverns und verwendeter Module/Erweiterungen enthalten. Die Fehlermeldungen dürfen keine internen Informationen, wie z. B. interne Servernamen, Fehlercodes usw. enthalten.

Motivation: Alle Informationen über den Webserver könnten Aufschluss über Sicherheitsschwachstellen geben.

ID: 3.22-258/13.0

Req 259 Das unautorisierte Überschreiben der Webserver-Konfiguration muss verhindert werden.

Das unautorisierte Überschreiben der Webserver-Konfiguration muss verhindert werden. Die Webserver-Konfiguration kann beispielsweise durch das Anlegen zusätzlicher Config-Dateien überschrieben werden.

Motivation: Änderung an sicherheitsrelevanter Einstellungen muss verhindert werden.

ID: 3.22-259/13.0

Req 260 Der Webserver darf nur Dateien ausliefern, die für die Auslieferung bestimmt sind.

Der Webserver darf nur Dateien ausliefern, die für die Auslieferung bestimmt sind. Alle Dateien, die sich direkt oder indirekt (beispielsweise über Links oder in virtuellen Verzeichnissen) im Dokumentenverzeichnis des Webserverns befinden, müssen mit restriktiven Zugriffsrechten versehen werden. Insbesondere darf der Webserver nur auf Dateien zugreifen können, die für die Auslieferung bestimmt sind.

Motivation: Falls zusätzliche Dateien oder Verzeichnisse über Links oder virtuelle Verzeichnisse in das Dokumentenverzeichnis des Webserverns eingebunden werden besteht die Möglichkeit, dass ein Benutzer über den Webserver auf Dateien zugreifen kann, die er nicht einsehen dürfte. Dies muss durch eine sorgfältige Konfiguration verhindert werden.

ID: 3.22-260/13.0

Req 261 ECC Zertifikate müssen eine Schlüssellänge von mindestens 3072 Bit umsetzen

Zertifikate müssen eine Schlüssellänge von mindestens 3072 Bit bei Verwendung von RSA oder mindestens 256 Bit bei Verwendung von ECC aufweisen.

Motivation: Um die Sicherheit der Zertifikate über den Gültigkeitszeitraum zu gewährleisten, müssen die kryptographischen Schlüssel eine entsprechende Länge aufweisen. Nach allgemeiner Einschätzung bietet eine Schlüssellänge von 3072 Bit einen ausreichenden Schutz für die nächsten Jahre. Bei ECC-Algorithmen bieten kürzere Schlüssellängen bereits dasselbe Maß an Sicherheit.

ID: 3.22-261/13.0

Req 262	Der Zugriff auf den Webserver muss protokolliert werden.
---------	--

Der Zugriff auf den Webserver muss protokolliert werden und muss die folgenden Informationen enthalten:

- Zugriffszeitpunkt
- Quelle (IP-Adresse)
- Konto (falls bekannt)
- URL
- Status-Code der Antwort des Webservers

Motivation: Für die Analyse von Sicherheitsvorfällen ist es sehr wichtig grundlegende Informationen darüber zu besitzen, wie ein Angriff erfolgt ist. Da ein Webserver eine Außenschnittstelle darstellt liegen gewisse Informationen auch bei dem Angriff auf ein nachgelagertes System lediglich auf dem Webserver vor und müssen dort daher zwingend protokolliert werden.

ID: 3.22-262/13.0

Req 263	Die Web-Anwendung muss mittels HTTP Strict Transport Security (HSTS) erzwingen, dass alle zukünftigen Verbindungen vom Browser ausschließlich verschlüsselt aufgebaut werden. Hierzu muss die Web-Anwendung den HTTP-Response-Header „Strict-Transport-Security“ setzen.
---------	--

Die Web-Anwendung muss mittels HTTP Strict Transport Security (HSTS) erzwingen, dass alle zukünftigen Verbindungen vom Browser ausschließlich verschlüsselt aufgebaut werden. Hierzu muss die Web-Anwendung den HTTP-Response-Header „Strict-Transport-Security“ setzen.

Motivation: Wenn ein Browser von einer Web-Anwendung den HSTS-Header empfängt, wird der Browser für die im Header angegebene Zeit ausschließlich verschlüsselt mit dieser Web-Anwendung kommunizieren. Wenn der Benutzer danach die Web-Anwendung mit einer HTTP-Adresse aufruft, wandelt der Browser diese automatisch in die entsprechende HTTPS-Adresse um. Falls die verschlüsselte Übertragung irgendeinen Fehlercode bewirkt, bricht der Browser die Verbindung mit einer Fehlermeldung ab. Das schließt auch Zertifikatsfehler ein, die ein Benutzer dann nicht mehr übergehen kann.

HSTS adressiert in erster Linie das Problem, dass viele Benutzer selten beispielsweise `https://www.anwendung.de` als Adresse eintippen - sondern eher `anwendung.de` bzw. `www.anwendung.de` oder die Anwendung aus ihren Bookmarks heraus unverschlüsselt aufrufen. Üblicherweise erfolgt dann ein Redirect von HTTP auf HTTPS. Aber dieser erste unverschlüsselte Request inkl. Redirect bietet Angreifern schon Möglichkeiten, Man-in-the-Middle-Angriffe durchzuführen. Wenn eine Web-Anwendung den HSTS-Header setzt, wird der erste unverschlüsselte Request vermieden. Ebenso ruft der Browser auch keine anderen unverschlüsselten Ressourcen auf, wenn beispielsweise versehentlich gemischte HTTP- und HTTPS-Ressourcen in der Web-Anwendung eingestellt wurden

Umsetzungsbeispiel: ABAP, Java und SAP Web Dispatcher:

Ab Release 7.53 Profilparameter setzen:

`icm/HTTP/strict_transport_security = max-age=31536000; includeSubDomains`

HSTS funktioniert nicht mit Servern, die auf nicht standardmäßigen HTTP/HTTPS-Ports ausgeführt werden, da die Benutzeragenten lediglich das Protokoll von HTTP in HTTPS ändern, ohne die nicht standardmäßigen Ports zu beachten.

Alternativ und ohne Einschränkung auf die standardmäßigen HTTP/HTTPS-Ports (s. SAP-Hinweis 2042819):

Pflegen Sie folgendes mit dem Parameter `icm/HTTP/mod_<xx>` option FILE in die Datei:

Legen Sie zuerst eine Modifikationsregel für die Umleitung von HTTP-Requests an HTTPS an:

```
if %{SERVER_PROTOCOL} !stricmp "https"  
RegRedirectUrl /(.* ) https://<host>:<https_port>/$1
```

<host> ist dabei der Web Dispatcher oder ICM-Host, und <https_port> ist der HTTPS-Serverport.

Legen Sie dann für HTTPS-Requests eine Modifikationsregel fest, um den Response-Header für Strict Transport Security zu setzen, z.B.:

```
if %{SERVER_PROTOCOL} stricmp "https"  
SetResponseHeader Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

Sie können den Header-Wert an Ihre spezifischen Anforderungen anpassen.

S. SAP Hinweis 2202116

ID: 3.22-263/13.0

Req 264	Um schutzbedürftige Daten zu übertragen, darf die Web-Anwendung keine URL-Parameter oder andere Felder verwenden, die in Protokolldateien erfasst werden.
---------	---

Um schutzbedürftige Daten zu übertragen, darf die Web-Anwendung keine URL-Parameter oder andere Felder verwenden, die in Protokolldateien erfasst werden

Motivation: Bei der GET-Methode werden Parameter in der URL übertragen und dann üblicherweise von Webservern und Clients protokolliert. Falls die Web-Anwendung Verknüpfungen zu anderen Web-Anwendungen enthält, besteht zudem die Möglichkeit, dass URL-Parameter über den HTTP-Header „Referer“ an diese anderen Web-Anwendungen übertragen werden und dort in den Log-Dateien erscheinen. Diesen Teil der Problematik könnte man auch (für die meisten Browser) durch Verwendung des HTTP-Response-Headers „Referrer-Policy“ (zum Beispiel mit dem Wert „no-referrer“) adressieren. Zudem kann nicht ausgeschlossen werden, dass ein Nutzer eine verwendete URL inklusive schutzbedürftiger Daten kopiert und weitergibt. Bei der POST-Methode beispielsweise werden Parameter im Request-Body übertragen.

ID: 3.22-264/13.0

Req 265	Die Web-Anwendung darf clientseitig keine schutzbedürftigen Daten speichern, weder temporär noch persistent.
---------	--

Dies gilt sowohl für Cookies als auch für die verschiedenen Arten von lokalen Speichern des Browsers. Stattdessen müssen schutzbedürftige Daten serverseitig gespeichert werden.

Als Ausnahme hiervon darf die Web-Anwendung Session-Identifizierer oder (Session-/Access-) Token in nichtpersistenten Cookies übertragen oder im Session Storage des Browsers temporär speichern.

Als weitere Ausnahme darf die Web-Anwendung Daten zur Identifikation eines Benutzers, beispielsweise den Benutzernamen oder eine ID, in persistenten Cookies oder anderen lokalen Speichern hinterlegen, falls personenbezogene Daten verschlüsselt gespeichert werden und falls der Nutzer eingewilligt hat.

Die Web-Anwendung muss zudem verhindern, dass schutzbedürftige Daten, die der Benutzer in ein HTML-Formular eingibt, vom Browser gespeichert und bei der nächsten Verwendung automatisch vervollständigt werden. Hierzu muss die Web-Anwendung das Attribut „autocomplete = \"off\"“ setzen. Dies gilt insbesondere für Kreditkartendaten und andere Konto- oder Zahlungsinformationen. Es wird aber empfohlen, dies grundsätzlich für schutzbedürftige Daten umzusetzen.

Motivation: Alle auf dem Client gespeicherten Daten können grundsätzlich analysiert und manipuliert werden, entweder lokal durch direkten Zugriff oder ggf. durch einen erfolgreichen Angriff von außen. Für Cookies kann eine Web-Anwendung über das Attribut „expires“ oder das Attribut „max-age“ ein Ablaufdatum setzen. In dem Fall handelt es sich aber um ein „persistentes Cookie“. Der Browser speichert persistente Cookies bis zum Ablaufdatum auf dem Endgerät des Benutzers. Jede Person mit Zugang zum jeweiligen Endgerät kann die persistenten Cookies auslesen. Dies kann in einem Internet-Café, bei anderen Arten von gemeinsamer Nutzung oder durch einen erfolgreichen Angriff geschehen. Cookies ohne Ablaufdatum sind nicht persistent, sie werden nicht dauerhaft gespeichert. Der Browser löscht diese Cookies, sobald er geschlossen wird. Bis dahin sind jedoch auch diese Daten für einen Angreifer zugänglich. Daher dürfen auch Cookies, die nicht persistent sind, keine schutzbedürftigen Daten enthalten.

Die verschiedenen Technologien, die unter anderem für HTML5-Anwendungen verwendet werden, bieten weitere Mechanismen, um Daten lokal auf dem Client zu speichern. Aber auch für diese lokalen Speicher gilt, dass sie auf dem Client analysiert bzw. manipuliert werden können. Die DOM-Objekte „localStorage“ und „sessionStorage“ sind solch eine Alternative, mit der Daten in einem Webbrowser gespeichert werden können. Im Local Storage werden die Daten jedoch dauerhaft gespeichert, während sie im Session Storage mit Ablauf der Sitzung gelöscht werden. Beim Session Storage muss aber beachtet werden, dass das Öffnen einer Webseite in einem neuen Tab oder Browserfenster eine neue Sitzung erzeugt. Alle Skripte einer Domain, von der aus die Daten gespeichert wurden, können auf die gespeicherten Daten (des gleichen lokalen Benutzerprofils bei Local Storage bzw. des jeweiligen Tabs/Browserfensters bei Session Storage) zugreifen. Anders als bei Cookies über das „HttpOnly“-Flag gibt es bei dieser Speichervariante somit keinen inhärenten Schutzmechanismus gegen Cross-Site Scripting.

Durch die Deaktivierung des Autocomplete wird erreicht, dass unabhängig von der Konfiguration des Browsers eingegebene Daten nicht gespeichert werden. Es ist hierbei zu beachten, dass seit einiger Zeit die meisten aktuellen Browser das Attribut „autocomplete“ für „password“-Eingabefelder ignorieren.

ID: 3.22-265/13.0

Req 266	Die Web-Anwendung darf keine Eingabedaten von Benutzern für den direkten Zugriff auf Dateien und Verzeichnisse sowie auf andere serverseitige Ressourcen verwenden.
---------	---

Stattdessen ist eine indirekte Referenzierung der serverseitigen Ressourcen zu implementieren.

Motivation: Falls eine Web-Anwendung Eingabedaten für den direkten Zugriff auf Dateien verwendet, kann sich ein Angreifer unter Umständen auch Zugriff auf solche Dateien verschaffen, auf die er normalerweise keinen Zugriff hat. Falls zum Beispiel eine Web-Anwendung die Hilfedatei Deutsch.txt anzeigt, wenn ein Benutzer „Deutsch“ eingibt und zudem die Web-Anwendung die Benutzereingaben nicht ausreichend validiert, kann sich ein Angreifer unter Umständen durch Eingabe von „.././../etc/passwd%00“ auch die Benutzerdatei anzeigen lassen (Path Traversal/Local File Inclusion). Gefährliche Zeichen in diesem Zusammenhang sind die Zeichen zum Wechseln des Verzeichnisses und das Nullbyte. Das Nullbyte-Zeichen terminiert in diesem Beispiel den String und verhindert so, dass „.txt“ angehängen wird. Der Angreifer kann zudem versuchen, diese gefährlichen Zeichen in verschiedenen Codierungen zu verwenden. Darüber hinaus kann ein Angreifer bei einer entsprechenden Sicherheitslücke sogar externe Dateien in ein Script einbinden lassen und so Code zur Ausführung bringen (Remote File Inclusion).

ID: 3.22-266/13.0

Req 267	Die Web-Anwendung darf keine Eingabedaten verwenden, um Shell-Befehle oder Befehle für das laufende Programm (zum Beispiel über „eval“) zu erstellen.
---------	---

Die Web-Anwendung darf keine Eingabedaten verwenden, um Shell-Befehle oder Befehle für das laufende Programm (zum Beispiel über „eval“) zu erstellen.

Motivation: Hiermit wird OS Command Injection und Code Injection verhindert.

Zu OS Command Injection kann es kommen, wenn Eingabedaten in einen Befehl integriert werden, der dann zur Ausführung an das Betriebssystem weitergegeben wird. Dies ermöglicht es einem Angreifer unter Umständen, beliebige Betriebssystembefehle auf dem Server auszuführen, auf dem die Web-Anwendung läuft. In dem Fall kann er in der Regel das System vollständig kompromittieren. Sehr oft kann ein Angreifer solch eine Schwachstelle sogar dazu ausnutzen, um andere Teile der Hosting-Infrastruktur zu kompromittieren, indem er Vertrauensbeziehungen ausnutzt und seinen Angriff auf andere Systeme ausweitet.

Bei Code Injection schleust ein Angreifer dagegen ausführbaren Code ein. Dies kann serverbasiert erfolgen (zum Beispiel über php) oder clientbasiert (zum Beispiel über JavaScript). Ist der Angreifer erfolgreich, wird dieser Code ausgeführt und kann ebenfalls großen Schaden anrichten. Code Injection kann ein Angreifer aber auch nutzen, um XSS- oder SQL-Injection-Angriffe einzuschleusen.

Beispielsweise könnte dieser Code Teil einer Web-Anwendung zur Administration eines Webservers sein:

```
string dirName = "C:\\filestore\\" + Directory.text;  
ProcessStartInfo psInfo = new ProcessStartInfo ("cmd", "/c dir " + dirName);
```

```
...  
Process proc = Process.Start(psInfo);
```

Mit dieser Funktion wird der Inhalt eines Verzeichnisses angezeigt. Das Script fügt den Wert des vom Benutzer übergebenen Parameters „Directory“ in einen festgelegten Befehl ein. Ein Angreifer könnte jedoch Shell-Metazeichen verwenden, um eigene Befehle einzuschleusen und ausführen zu lassen: Verzeichnisname && rm -rf.

Wenn die PHP-Funktion eval() verwendet wird und ihr nicht vertrauenswürdige Daten übergeben werden, die ein Angreifer beeinflussen kann

```
$username = $_POST['username'];
```

```
eval("echo $username");
```

ist Code Injection möglich, zum Beispiel durch Eingabe von mustermann; phpinfo().

Wenn andererseits mit JavaScript ein JSON-Dokument per „eval“ interpretiert wird

```
eval({"menu":{"address":{"line1":addressLine1,"line2":"","line3":""}}});
```

und dann die Variable „addressLine1“ aus dem Request mit dem Wert

```
„,[code],arbitrary:alert('executed!'),continue:“
```

belegt wird, wird hier ebenfalls eingeschleuster Code ausgeführt.

Sämtliche Benutzereingaben stellen hier eine mögliche Angriffsfläche dar. Die direkte Ausführung per „eval“ (oder vergleichbarer Funktionen) ist sehr problematisch, da die Daten direkt interpretiert werden ohne weitere Prüfungen hinsichtlich Sicherheit bzw. Gültigkeit. Dies gilt sowohl auf Serverseite als auch auf Clientseite.

Umsetzungsbeispiel: Beispielsweise könnte dieser Code Teil einer Web-Anwendung zur Administration eines Webservers sein:

```
string dirName = "C:\\filestore\\" + Directory.text;  
ProcessStartInfo psInfo = new ProcessStartInfo ("cmd", "/c dir " + dirName);
```

```
...  
Process proc = Process.Start(psInfo);
```

Mit dieser Funktion wird der Inhalt eines Verzeichnisses angezeigt. Das Script fügt den Wert des vom Benutzer übergebenen Parameters „Directory“ in einen festgelegten Befehl ein. Ein Angreifer könnte jedoch Shell-Metazeichen verwenden, um eigene Befehle einzuschleusen und ausführen zu lassen: Verzeichnisname && rm -rf.

Wenn die PHP-Funktion eval() verwendet wird und ihr nicht vertrauenswürdige Daten übergeben werden, die ein Angreifer beeinflussen kann

```
$username = $_POST['username'];
```

```
eval("echo $username");
```

ist Code Injection möglich, zum Beispiel durch Eingabe von mustermann; phpinfo().

Wenn andererseits mit JavaScript ein JSON-Dokument per „eval“ interpretiert wird

```
eval({"menu":{"address":{"line1":addressLine1,"line2":"","line3":""}}});
```

und dann die Variable „addressLine1“ aus dem Request mit dem Wert

```
„,arbitrary:alert('executed!'),continue:“
```

belegt wird, wird hier ebenfalls eingeschleuster Code ausgeführt.

ID: 3.22-267/13.0

Req 268	Die Länge der Session-Identifizier der Web-Anwendung muss mindestens 120 Bit betragen. Alle hierbei relevanten Zeichen müssen nach dem Zufallsprinzip generiert werden.
---------	---

Zeichenketten von mehr als 36 Dezimalziffern [0...9] oder Zeichenketten, die aus mehr als 20 Großbuchstaben [A...Z], Kleinbuchstaben [a...z] und Dezimalziffern [0...9] bestehen, erfüllen beispielsweise diese Anforderung.

Motivation: Durch statistische Analysen oder Brute-Force-Angriffe kann ein Angreifer versuchen, gültige Sessi-

on-Identifizierung zu ermitteln. Hat er Erfolg, kann er die Session der Opfer übernehmen. Dies kann verhindert werden, indem für die Web-Anwendung Session-Identifizierung verwendet werden, die zufällig und komplex sind.

Umsetzungsbeispiel: Überprüfung: Melden Sie sich mit dem Chrome-Browser an der SAP-Anwendung an und öffnen Sie die Entwicklertools (Strg + Umschalt + i). Markieren Sie in der Menüleiste "Application" im Kontextmenü "Cookies" die URL der Webseite. Lesen Sie in der Zeile mit *session* die Spalte "Size" [bytes]. Je nach verwendetem ASCII Code muss die Zahl in der Spalte "Größe" mit 7 oder 8 multipliziert werden. Das Ergebnis muss größer 120 sein.

ID: 3.22-268/13.0

Req 269	Falls stateless Session-/Access-Token (wie zum Beispiel JSON Web Token, JWT) eingesetzt werden, muss die Web-Anwendung Manipulationen, Replay-Angriffe oder andere Arten unrechtmäßiger Verwendung durch geeignete Maßnahmen verhindern.
---------	--

Im Falle von JWT müssen diese beispielsweise über sichere kryptographische Methoden und Algorithmen integritätsgeschützt sein, vorzugsweise mittels kryptographischer Signaturen. Der Empfänger der JWT muss die Integrität entsprechend seiner eigenen Konfiguration/Anwendungslogik überprüfen (also nicht basierend auf den Header-Informationen des JWT) und insbesondere keine ungesicherten JWT mit „{“alg”:“none”}“ akzeptieren. Die Gültigkeitsdauer muss zudem so kurz wie möglich sein. Und die Token müssen grundsätzlich durch TLS geschützt sein.

Motivation: Es muss verhindert werden, dass ein Angreifer derartige Token missbräuchlich verwenden kann. Daher ist es notwendig, dass die Token sicher übertragen und restriktiv verifiziert werden. Jedoch sollte in diesem Zusammenhang grundsätzlich beachtet werden, dass Lösungen basierend auf JWT zwar möglicherweise vergleichsweise einfach zu realisieren sind, vor allem um „stateless“ Services zu realisieren, aber nicht für alle Arten von Web-Anwendungen geeignet sind, insbesondere weil sie mit einigen Problemstellungen verbunden sind, wie zum Beispiel mit der Frage der Speicherung der Token im Browser oder der Realisierung eines restriktiven serverseitigen Timeout- bzw. Logout-Mechanismus.

ID: 3.22-269/13.0

Req 270	Um Session-Identifizierung oder stateful/stateless (Session-)Token zu übertragen, darf die Web-Anwendung keine URL-Parameter verwenden.
---------	---

Solche Identifizierung bzw. Token dürfen ausschließlich außerhalb der URL übertragen werden, also in einem (Session-)Cookie oder alternativ beispielsweise in einem anderen HTTP-Header.

Motivation: Session-Cookies bieten den Vorteil, dass wichtige sicherheitsrelevante Eigenschaften durch die Web-Anwendung einfach definiert werden können und von den Browsern entsprechend gewährleistet werden (keine dauerhafte Speicherung, sichere Übertragung, kein Script-Zugriff, eingeschränkter Gültigkeitsbereich). Bei Verwendung anderer Mechanismen müssen diese Eigenschaften durch die jeweilige Implementierung sichergestellt werden.

Nicht zulässig ist die Verwendung von URL-Parametern: In dem Fall kann nicht ausgeschlossen werden, dass ein Nutzer eine verwendete URL inklusive gültigem Identifizierung kopiert und weitergibt. Zudem besteht die Gefahr, dass die Identifizierung über den „Referer“-Header an andere Web-Anwendungen gesendet werden. Und nicht zuletzt wird durch die Verwendung von Cookies oder HTTP-Headern die Gefahr minimiert, dass eine Web-Anwendung mittels Session-Fixation angegriffen werden kann: Bei einem solchen Angriff erstellt ein Angreifer eine Session und schiebt den entsprechenden Identifizierung einem Opfer unter. Dies erfolgt in der Regel durch einen Link, in dem der Identifizierung als URL-Parameter enthalten ist. Verwendet das Opfer diesen Identifizierung und authentifiziert sich dann gegenüber der Web-Anwendung, kann der Angreifer in dieser laufenden Session dessen Identität annehmen.

ID: 3.22-270/13.0

Req 271	Ein Session-Identifizierung oder stateful/stateless (Session-)Token darf nicht persistent im Browser gespeichert werden.
---------	--

Ein Session-Identifizierung oder stateful/stateless (Session-)Token darf nicht persistent im Browser gespeichert werden.

Motivation: Im Falle von Session-Cookies kann eine Web-Anwendung über das Attribut „expires“ oder das Attribut „max-age“ ein Ablaufdatum für ein Cookie setzen. In dem Fall handelt es sich aber um ein „persistentes Cookie“. Der Browser speichert persistente Cookies bis zum Ablaufdatum auf dem Endgerät des Benutzers. Jede Person mit Zugang zum jeweiligen Endgerät kann die persistenten Cookies auslesen. Dies kann in einem Internet-Café, bei anderen Arten von gemeinsamer Nutzung oder durch einen erfolgreichen Angriff geschehen. Cookies ohne Ablaufdatum sind nicht persistent, sie werden nicht dauerhaft gespeichert. Der Browser löscht diese Cookies, sobald er geschlossen wird.

Werden keine Session-Cookies verwendet, kann beispielsweise der Session Storage des Browsers verwendet werden, da dieser ebenfalls nicht persistent ist und die Inhalte beim Schließen des Browsers gelöscht werden. Hierbei muss aber beachtet werden, dass das Öffnen einer Webseite in einem neuen Tab oder Browserfenster eine neue Sitzung erzeugt. Im Local Storage des Browsers würden die Identifier dagegen persistent gespeichert werden.

ID: 3.22-271/13.0

Req 272 Die Web-Anwendung muss im Session-Cookie das Attribut „secure“ setzen.

Zusätzlich wird zum Schutz der Integrität eines Cookies empfohlen, Cookie-Namen mit dem Prefix „__Host-“ (oder alternativ „__Secure-“) zu verwenden.

Wenn die Web-Anwendung durchgängig den HTTP-Response-Header „Strict-Transport-Security“ setzt und somit sicherstellt, dass der Browser ausschließlich verschlüsselte Requests verschickt, ist es nicht mehr zwingend notwendig, das Attribut „secure“ zu setzen. Es wird aber dennoch als zusätzliche Sicherheitsmaßnahme empfohlen.

Motivation: Das Attribut „secure“ verhindert, dass der Browser Cookies unverschlüsselt versendet. Dies geschieht zum Beispiel bei unverschlüsselten Inhalten einer Web-Anwendung. Dies kann aber auch durch einen aktiven Angriff geschehen, bei dem ein Angreifer unverschlüsselte Links bzw. Referenzen injiziert oder präsentiert. Selbst wenn also eine Seite ausschließlich über HTTPS erreichbar ist, könnte eine vom Angreifer kontrollierte Seite den Browser des Opfers dazu veranlassen, einen unverschlüsselten Request zu stellen. Ein Angreifer könnte diesen Request abfangen und hätte Zugriff auf den Cookie. Wird das „secure“-Attribut gesetzt, ist dieser Angriff nicht erfolgreich. Unverschlüsselte HTTP-Requests fügt der Browser entsprechend gesicherte Cookies gar nicht erst hinzu.

Die Cookie-Prefixes sollen vor Cookie Injection bzw. Cookie Clobbering schützen: Auch eine unverschlüsselte HTTP-Response kann einen „secure“-Cookie setzen. Beispielsweise kann deshalb ein Man-in-the-Middle-Angreifer „secure“-Cookies setzen oder verändern. Oder ein Angreifer könnte Probleme bei Subdomains ausnutzen, wenn er zum Beispiel böse.beispiel.de kontrolliert und bei Requests auf diese Subdomain einen Cookie mit „domain=beispiel.de“ setzt oder wenn er eine XSS auf unsicher.beispiel.de findet und hierüber einen Cookie mit „domain=beispiel.de“ setzt. Über einen Prefix beim Cookie-Namen können zusätzliche Regeln zum Browserverhalten bestimmt werden (wenn der Browser diese Funktion unterstützt): Hat der Cookie-Namen einen Prefix „__Secure-“, wird ein kompatibler Browser diesen Cookie nur setzen, falls er von einer HTTPS-Response gesetzt wird und das „secure“-Flag gesetzt ist (Beispiel: Set-Cookie: __Secure-ID=123; Secure; Domain=example.com). Hat der Cookie-Namen einen Prefix „__Host-“, wird ein kompatibler Browser diesen Cookie nur setzen, falls er von einer HTTPS-Response gesetzt wird, das „secure“-Flag gesetzt ist, das „path“-Flag auf root („/“) gesetzt ist und das „domain“-Flag nicht gesetzt ist (Beispiel: Set-Cookie: __Host-ID=123; Secure; Path=/).

Umsetzungsbeispiel: ABAP:

Setzen des Parameters:

login/ticket_only_by_https = 1

Überprüfung: Melden Sie sich mit dem Chrome-Browser an der SAP-Anwendung an und öffnen Sie die Entwickler-tools (Strg + Umschalt + i). Markieren Sie in der Menüleiste "Application" im Kontextmenü "Cookies" die URL der Webseite. Prüfen Sie in der Zeile mit *session* die Spalte "Secure", ob ein Haken gesetzt ist.

ID: 3.22-272/13.0

Req 273 Die Web-Anwendung muss im Session-Cookie das Attribut „HttpOnly“ setzen.

Die Web-Anwendung muss im Session-Cookie das Attribut „HttpOnly“ setzen.

Motivation: Das Attribut „HttpOnly“ erzwingt nicht (wie es der Name suggeriert), dass ein Cookie nur über HTTP über-

tragen wird. Vielmehr verhindert es den Zugriff für JavaScript über die „document.cookie“-API. Deshalb verhindert es XSS-Angriffe auf den Cookie.

Umsetzungsbeispiel: ABAP:

Setzen des Parameters:

icf/set_HTTPOnly_flag_on_cookies = 0

Überprüfung: Melden Sie sich mit dem Chrome-Browser an der SAP-Anwendung an und öffnen Sie die Entwickler-tools (Strg + Umschalt + i). Markieren Sie in der Menüleiste "Application" im Kontextmenü "Cookies" die URL der Webseite. Prüfen Sie in der Zeile mit *session* die Spalte "HttpOnly", ob ein Haken gesetzt ist.

ID: 3.22-273/13.0

Req 274 Die Web-Anwendung darf das Attribut „domain“ im Session-Cookie nicht setzen.

Das Attribut „domain“ gibt den Domain-Namen an, für den der Cookie gilt. Der Browser schickt den Cookie in allen Requests mit, die der Browser an diese Domain und deren Subdomains sendet. Dieses Attribut nicht zu setzen, ist die restriktivste Einstellung: In dem Fall wird als Default-Wert der Host-Name des Servers verwendet, der den Cookie gesetzt hat.

Das Attribut „domain“ kann jedoch gesetzt werden, wenn auf der angegebenen Domain und allen deren Subdomains keine anderen Web-Anwendungen laufen, so dass ausgeschlossen ist, dass der Session-Cookie auch an fremde Web-Anwendungen gesendet wird.

Darüber hinaus wird generell empfohlen, dass Web-Anwendungen unterschiedlicher Kritikalität oder mit unterschiedlichem Sicherheitsniveau auch unter unterschiedlichen Domains betrieben werden. Ansonsten gefährden Schwachstellen der einen Web-Anwendung unter Umständen auch die Sicherheit der anderen Web-Anwendung.

Motivation: Eine restriktive Verwendung des „domain“-Attributs verhindert, dass der Session-Cookie an andere Web-Anwendungen geschickt wird.

Wenn beispielsweise die Web-Anwendung xyz.telekom.net einen Cookie ohne „domain“-Attribut setzt, wird der Cookie danach in allen Requests an xyz.telekom.net und an deren Subdomains (wie abc.xyz.telekom.net) gesendet. Wenn aber die Web-Anwendung das „domain“-Attribut auf „telekom.net“ setzt, wird jede Web-Anwendung in einer Sub-Domain von telekom.net den Cookie erhalten

Umsetzungsbeispiel: AS Java (Cookies):

1. Session Cookie:

Der Name des Session-Cookies lautet JSESSIONID (gemäß der Java Servlet 2.5-Spezifikation). Der Wert des JSESSIONID-Cookies hat ein bestimmtes Format, das Informationen enthält, die für den Prozess der Sitzungsverfolgung erforderlich sind. Wenn das mit der Anfrage gesendete Cookie nicht mit diesem Format konform ist, wird es vom Web Container ignoriert und daher nicht für das Session-Tracking verwendet.

Aus Sicht eines Anwendungsentwicklers sind die Generierung des Cookies und die Details des Formats nicht von Interesse. Das alles wird vom Web Container abgewickelt. Sie müssen sich nur darüber im Klaren sein, dass ein JSESSIONID-Cookie bei der ersten Anforderung an die Anwendung und Sitzung auf Ihren HTTP-Client gesetzt und eingerichtet wurde.

2. Anwendungs-Cookie:

Das Anwendungs-Cookie wird nur vom SAP Web Dispatcher (oder einer Load-Balancing-Lösung eines Drittanbieters) für den instanzbasierten Load-Balancing verwendet.

Der Cookie-Name setzt sich aus der saplb_Zeichenfolge und einer Zeichenfolge zusammen, die eine Anmeldegruppe darstellt. Eine Anmeldegruppe wird durch eine Gruppe von Java EE-Instanzen mit bestimmten Anwendungsaliasen definiert, die auf ihnen aktiv sind. Der Wert enthält Informationen über den Serverprozess, auf dem die Anwendung ausgeführt wird, sowie über die Web AS Java-Instanz, in der sich der Server befindet.

Setzen des Attributes "Domain":

Das Attribut "Domain" muss sowohl für das Session- als auch das Anwendungscookie auf "NONE" gesetzt sein.

Setzen benutzerdefinierter Werte für SAP-spezifische Cookie-Attribute werden lokal (mit dem SAP NetWeaver Administrator) und global (mit dem Config Tool) durchgeführt. Sie können die Cookies auch lokal über den Application De-

ployment Descriptor web-j2ee-engine.xml konfigurieren.

Weitere Informationen über Java Cookies und das Setzen dieser:

https://help.sap.com/docs/SAP_NETWEAVER_731/c591e2679e104fcd8dc8e77771ff524/49c9fa1002c8087be10000000a42189d.html?locale=en-US

https://help.sap.com/docs/SAP_NETWEAVER_731/0c333adb55cd4dbf8e92a5175703224c/4975556b9bea200ce10000000a42189c.html?locale=en-US

https://help.sap.com/saphelp_em92/helpdata/de/49/75556b9bea200ce10000000a42189c/content.htm?no_cache=true

ID: 3.22-274/13.0

Req 275	Die Web-Anwendung muss das Attribut „path“ im Session-Cookie so restriktiv setzen, dass der Cookie nicht an andere Web-Anwendungen auf dem gleichen Host gesendet wird
---------	--

Die Web-Anwendung muss das Attribut „path“ im Session-Cookie so restriktiv setzen, dass der Cookie nicht an andere Web-Anwendungen auf dem gleichen Host gesendet wird

Motivation: Web-Anwendungen können über den gleichen Host-Namen verfügen, sich aber in unterschiedlichen Verzeichnissen befinden. Es muss daher sichergestellt werden, dass unterschiedliche Web-Anwendungen auf dem gleichen Host nicht die Cookies der jeweils anderen Anwendungen erhalten. Wird beim Setzen eines Cookies das Attribut „path“ angegeben, so ist er nur in diesem Verzeichnis und allen Unterverzeichnissen gültig. Das Attribut „path“ muss daher so gesetzt sein, dass keine andere Web-Anwendung den Session-Cookie erhält.

Wenn beispielsweise eine Web-Anwendung unter `telekom.net/meineAnwendung/index.jsp` einen Cookie mit der Pfadangabe „;path=/“ setzt, wird der Cookie danach in allen Requests an die Domain `telekom.net` mitgesendet, ggf. auch an andere, weniger vertrauenswürdige Anwendungen, die im root oder in beliebigen anderen Verzeichnissen platziert wurden.

Wird die Pfadangabe jedoch auf „;path=/meineAnwendung/“ gesetzt, wird der Cookie nur bei Requests an `telekom.net/meineAnwendung/` gesendet (und auch an alle hierunter liegenden Unterverzeichnisse, aber nicht an darüber liegende Verzeichnisse). Das abschließende Slash-Zeichen darf nicht weggelassen werden, da der Cookie ansonsten auch an andere Verzeichnisse mit übereinstimmenden Namen gesendet wird, zum Beispiel an `telekom.net/meineAnwendung-exploited`.

Wenn keine Pfadangabe gemacht wird, verwendet der Browser als Default den Pfad des aktuellen HTTP-Requests, aufgrund dessen der Cookie gesetzt wurde.

Umsetzungsbeispiel: Wenn beispielsweise eine Web-Anwendung unter `telekom.net/meineAnwendung/index.jsp` einen Cookie mit der Pfadangabe „;path=/“ setzt, wird der Cookie danach in allen Requests an die Domain `telekom.net` mitgesendet, ggf. auch an andere, weniger vertrauenswürdige Anwendungen, die im root oder in beliebigen anderen Verzeichnissen platziert wurden.

Wird die Pfadangabe jedoch auf „;path=/meineAnwendung/“ gesetzt, wird der Cookie nur bei Requests an `telekom.net/meineAnwendung/` gesendet (und auch an alle hierunter liegenden Unterverzeichnisse, aber nicht an darüber liegende Verzeichnisse). Das abschließende Slash-Zeichen darf nicht weggelassen werden, da der Cookie ansonsten auch an andere Verzeichnisse mit übereinstimmenden Namen gesendet wird, zum Beispiel an `telekom.net/meineAnwendung-exploited`.

ID: 3.22-275/13.0

Req 276	Für ein Benutzerkonto darf nur eine Session zur selben Zeit aktiv sein.
---------	---

Eine zweite Anmeldung mit dem gleichen Benutzerkonto muss verhindert werden. Alternativ kann aber auch die zweite Anmeldung zugelassen werden; in dem Fall muss die ursprüngliche Session beendet werden. Diese Variante kann sinnvoll sein, um zu verhindern, dass ein Benutzerkonto zeitweilig blockiert wird, zum Beispiel durch Schließen oder Absturz des Browsers ohne vorherigen Logout.

Es wird empfohlen, dass die Web-Anwendung dem Benutzer eine Warnmeldung anzeigt, wenn er sich anmeldet, gleichzeitig aber bereits eine Session für sein Benutzerkonto aktiv ist. Dies erhöht die Wahrscheinlichkeit, dass Angriffe auf Accounts entdeckt werden.

Es gibt jedoch Web-Anwendungen, die explizit den Zugriff über diverse Kanäle (Web, Mobile, TV) vorsehen oder aus anderen Gründen Mehrfachanmeldungen erlauben. Es wird in solchen Ausnahmefällen empfohlen, dass der Benutzer dann die Möglichkeit hat, die anderen parallelen Sessions über eine entsprechende Funktion bewusst zu beenden, zum Beispiel bei einem Passwortwechsel.

Motivation: Wenn für ein Benutzerkonto mehrere Sessions gleichzeitig aktiv sind, kann das bedeuten, dass das Konto von unterschiedlichen Benutzern gleichzeitig verwendet wird oder dass hier ein erfolgreicher Angriff vorliegt.

ID: 3.22-276/13.0

Req 277	Die Web-Anwendung muss über eine Funktion verfügen, die es dem angemeldeten Benutzer ermöglicht, sich jederzeit abzumelden.
---------	---

Die Web-Anwendung muss über eine Funktion verfügen, die es dem angemeldeten Benutzer ermöglicht, sich jederzeit abzumelden.

Motivation: Ein Benutzer muss die Möglichkeit haben, seine Session und somit seine Daten vor fremden Zugriff zu schützen. Eine gezielte Abmeldung dient dazu, eine Session zu beenden, um sicherzustellen, dass diese Session nicht von einem unberechtigten Benutzer fortgesetzt werden kann.

Umsetzungsbeispiel: Die Web-Anwendung muss eine Möglichkeit der Abmeldung zur Verfügung stellen.

ID: 3.22-277/13.0

Req 278	Falls ein Benutzer über ein Single-Sign-on-Verfahren (SSO) Zugang zu der Web-Anwendung erhalten hat und sich in der Web-Anwendung abmeldet, müssen beide Sessions beendet werden: sowohl die Session mit der Web-Anwendung als auch die Session mit dem SSO-Portal.
---------	---

Falls die Session mit dem SSO-Portal nicht automatisch beendet werden kann, muss alternativ nach der Abmeldung von der Web-Anwendung die SSO-Session im Browser (wieder) sichtbar sein. Dies kann beispielsweise realisiert werden, indem nach der Abmeldung ein Redirect auf das SSO-Portal erfolgt.

Motivation: SSO bedeutet, dass ein Benutzer nach einer einmaligen Authentisierung an einem SSO-Portal weitere Anwendungen nutzen kann, ohne sich gegenüber diesen Anwendungen erneut authentisieren zu müssen. Falls bei einer Abmeldung von einer Anwendung nur die Session mit der Anwendung invalidiert wird und eine weiterhin gültige Session mit dem SSO-Portal nicht mehr sichtbar ist, wird ein Benutzer sich in der Regel auch nicht vom SSO-Portal abmelden. Die Session mit dem SSO-Portal bleibt gültig und könnte unter Umständen von einem Angreifer unbemerkt übernommen werden.

ID: 3.22-278/13.0

Req 279	Falls ein Benutzer über ein SSO-Verfahren Zugang zu der Web-Anwendung erhalten hat und sich vom SSO-Portal abmeldet, muss auch die Session mit der Web-Anwendung automatisch beendet werden.
---------	--

Hierfür muss das SSO-Portal initiieren, dass auch die Session mit der Web-Anwendung beendet wird.

Motivation: Wenn der Benutzer sich vom SSO-Portal abmeldet, ist ihm ggf. nicht bewusst, dass er andere Anwendungen weiterhin nutzt. Werden die Sessions mit diesen Anwendungen nicht automatisch beendet, könnten sie unter Umständen von einem Angreifer unbemerkt übernommen werden

ID: 3.22-279/13.0

Req 280	Nach einer definierten Zeit der Inaktivität des Benutzers muss in der Web-Anwendung ein Timeout erfolgen und der Benutzer automatisch von der Web-Anwendung abgemeldet werden.
---------	--

Die genaue Zeitdauer, nach der ein Timeout erfolgt, muss individuell für jede Anwendung festgelegt werden. Sie ist abhängig von deren Sensibilität und Einsatzzweck. Für kritische, aus dem Internet erreichbare Web-Anwendungen mit Zugriff auf schutzbedürftige Daten (bei denen beispielsweise der Zugriff eines Benutzers auf personenbezogene Daten unterschiedlicher Kunden vorgesehen ist) wird empfohlen, dass der Timeout spätestens nach 60 Minuten erfolgt, während bei internen Web-Anwendungen ohne Zugriff auf schutzbedürftige Daten dieser Zeitraum deutlich länger gewählt werden könnte. Ferner wird empfohlen, dass diese Zeitdauer ein konfigurierbarer Systemparameter ist.

Motivation: Der Timeout schützt den Benutzer, wenn er vergisst sich abzumelden und inaktive Sessions von einem Angreifer (durch Schwachstellen in der Web-Anwendung oder auch durch Mitbenutzung des Browsers) unbemerkt übernommen werden können. Durch einen Timeout von 30 Minuten wird der Bedienungskomfort in der Regel nicht reduziert, die Gefahr von Session-Übernahmen aber signifikant reduziert. Die Gründe für die Festlegung der jeweiligen Timeout-Dauer können sich im laufend

Umsetzungsbeispiel: **AS ABAP:**

Setzen des Parameters:

`http/security_session_timeout = 1800`

Der Wert 1800 entspricht 30 Minuten.

AS Java (Cookies):

1. Session Cookie:

Der Name des Session-Cookies lautet JSESSIONID (gemäß der Java Servlet 2.5-Spezifikation). Der Wert des JSESSIONID-Cookies hat ein bestimmtes Format, das Informationen enthält, die für den Prozess der Sitzungsverfolgung erforderlich sind. Wenn das mit der Anfrage gesendete Cookie nicht mit diesem Format konform ist, wird es vom Web Container ignoriert und daher nicht für das Session-Tracking verwendet.

Aus Sicht eines Anwendungsentwicklers sind die Generierung des Cookies und die Details des Formats nicht von Interesse. Das alles wird vom Web Container abgewickelt. Sie müssen sich nur darüber im Klaren sein, dass ein JSESSIONID-Cookie bei der ersten Anforderung an die Anwendung und Sitzung auf Ihren HTTP-Client gesetzt und eingerichtet wurde.

2. Anwendungs-Cookie:

Das Anwendungs-Cookie wird nur vom SAP Web Dispatcher (oder einer Load-Balancing-Lösung eines Drittanbieters) für den instanzbasierten Load-Balancing verwendet.

Der Cookie Name setzt sich aus der saplb_ Zeichenfolge und einer Zeichenfolge zusammen, die eine Anmeldegruppe darstellt. Eine Anmeldegruppe wird durch eine Gruppe von Java EE-Instanzen mit bestimmten Anwendungsaliases definiert, die auf ihnen aktiv sind. Der Wert enthält Informationen über den Serverprozess, auf dem die Anwendung ausgeführt wird, sowie über die Web AS Java-Instanz, in der sich der Server befindet.

Setzen des Attributes "Maximum age":

Das Attribut "Maximum age" muss sowohl für das Session- als auch das Anwendungscookie auf "**1800**" gesetzt sein.

Setzen benutzerdefinierter Werte für SAP-spezifische Cookie-Attribute werden lokal (mit dem SAP NetWeaver Administrator) und global (mit dem Config Tool) durchgeführt. Sie können die Cookies auch lokal über den Application Deployment Descriptor web-j2ee-engine.xml konfigurieren.

Weitere Informationen über Java Cookies und das Setzen dieser:

https://help.sap.com/docs/SAP_NETWEAVER_731/c591e2679e104fcd8dc8e77771ff524/49c9fa1002c8087be1000000a42189d.html?locale=en-US

https://help.sap.com/docs/SAP_NETWEAVER_731/0c333adb55cd4dbf8e92a5175703224c/4975556b9bea200ce1000000a42189c.html?locale=en-US

ID: 3.22-280/13.0

Req 281	Bei einer Abmeldung oder einem Timeout muss die Web-Anwendung den dazugehörigen Session-Identifizierer oder stateful/stateless (Session-)Token serverseitig ungültig machen.
---------	--

Bei einer Abmeldung oder einem Timeout muss die Web-Anwendung den dazugehörigen SessionIdentifizierer oder stateful/stateless (Session-)Token serverseitig ungültig machen.

Motivation: Ohne eine vollständige serverseitige Invalidierung der Session kann ein Angreifer offene Sessions fortset-

zen, wenn er Zugriff auf diese Sessions erhält. Dies ist beispielsweise möglich, wenn er den Rechner des Opfers mitbenutzt oder aber wenn er über einen anderen Angriff die Session-ID ermittelt hat

ID: 3.22-281/13.0

Req 282	Falls das Session-Management der Web-Anwendung (ausschließlich) auf Session-Cookies basiert, muss die Web-Anwendung einen Mechanismus gegen Angriffe mittels Cross-Site Request Forgery (CSRF) einsetzen. Dieser Mechanismus muss verhindern, dass ein Angreifer Requests auf einer von ihm kontrollierten Webseite hinterlegen kann, die (bei einem Aufruf jener Webseite durch den Benutzer) automatisch Aktionen des Benutzers in der zu schützenden Web-Anwendung auslösen.
---------	--

Insbesondere die Requests in der Web-Anwendung, die eine Datenänderung oder Zustandsänderung verursachen können, sind entsprechend zu schützen. Requests, über die lediglich Informationen angefordert werden, benötigen keinen Schutz vor CSRF.

In der Regel werden hierzu „CSRF-Token“ (auch „Synchronizer-Token“ genannt) als Hidden-Field eingebunden. Ein CSRF-Token muss serverseitig generiert werden, darf nicht vorhersagbar sein und muss individuell mindestens für jede Session sein. Die Web-Anwendung muss bei allen relevanten Requests serverseitig prüfen, ob der Token übertragen wurde und dessen Wert mit dem erwarteten Wert übereinstimmt, bevor die angeforderte Aktion ausgeführt wird. Viele Frameworks stellen entsprechende Schutzmechanismen zur Verfügung. In dem Fall ist es meistens ratsam, diese zu nutzen, statt eine entsprechende Lösung selbst zu implementieren. Wenn die Realisierung als Hidden-Field nicht möglich oder sinnvoll ist, kann der CSRF-Token in einem spezifischen HTTP-Header übertragen werden (zum Beispiel „X-CSRF-Token“). Informationen, die vom Browser automatisch mitgesendet werden, wie beispielsweise Cookies, bieten jedoch keinerlei Schutz.

Als zusätzlicher Schutzmechanismus wird empfohlen, im Session-Cookie das Attribut „SameSite“ (in der Regel mit dem Wert „Lax“) zu setzen.

Motivation: Requests, die nicht auf diese Weise geschützt werden, sind anfällig für CSRF (auch als XSRF oder Session Riding bekannt). Hierbei wird ein Opfer dazu gebracht, unwissentlich einen präparierten HTTP-Request abzuschicken. Dies geschieht beispielsweise beim Besuch einer bösartigen Webseite, in der ein entsprechender Link auf eine andere Web-Anwendung enthalten ist (zum Beispiel als „img“, „script“- oder „iframe“-Tag). Dieser Link ist für das Opfer jedoch nicht zu erkennen. Aber der Browser folgt im Hintergrund diesem Link und schickt auch den Session-Cookie automatisch mit, sofern das Opfer aktuell eine gültige Session für diese andere Web-Anwendung hat.

Wenn zum Beispiel in einer Web-Anwendung ein Benutzer eine Überweisung durchführen kann

`https://example.com/app/transferFunds ?amount=1500 &destinationAccount=4673243243`

und der entsprechende Request keinen geheimen Token enthält, kann ein Angreifer einen Request vorbereiten, der Geld vom Konto des Opfers auf das Konto des Angreifers transferiert. Diesen Request bettet er in ein Image-Tag ein und hinterlegt ihn in einer beliebigen Webseite, die er kontrolliert: [bad URL at `img=https://example.com/app/transferFunds ?amount=1500 &destinationAccount=attackersAcct#`]

[bad URL at `img=https://example.com/app/transferFunds ?amount=1500 &destinationAccount=attackersAcct#`]

``

Wenn das Opfer diese präparierte Seite besucht, während es gleichzeitig bei `example.com` angemeldet ist, wird der untergeschobene Request den aktuell gültigen Cookie mitsenden und damit auch ausgeführt werden.

Wenn aber jeder HTTP-Request einen unvorhersagbaren Token enthalten muss, kann ein Angreifer keinen gültigen Request präparieren. Somit ist ein CSRF-Angriff nicht mehr möglich.

Für einen vollumfänglichen Schutz gegen CSRF ist für jeden einzelnen Request in einer Web-Anwendung, der eine Datenänderung verursacht, ein unterschiedlicher CSRF-Token zu verwenden. Dies verhindert, dass abgefangene Token für einen CSRF-Angriff verwendet werden können. CSRF-Token, die individuell für jeden Request sind, werden jedoch von manchen Frameworks nicht unterstützt. Diese Variante kann ggf. auch bei der Benutzung der WebAnwendung zu Problemen führen, zum Beispiel bei parallelem Arbeiten in mehreren Tabs oder Fenstern.

Mit dem Setzen des „SameSite“-Attributs im Session-Cookie wird der Browser angewiesen, bei einem Cross-Site-Request den Session-Cookie nicht mitzusenden. Dies unterbindet CSRF-Angriffe grundsätzlich (wenn der Browser diese Funktion unterstützt). Wird hierbei der Wert „Strict“ gesetzt, wird der Cookie bei keinem derartigen Request mitgesendet; dies kann jedoch ggf. unerwünschte Auswirkungen bzgl. des Browserverhaltens haben, zum Beispiel wenn der Nutzer einen Link auf eine andere Web-Anwendung auswählt und der Session-Cookie für diese Webanwendung folglich nicht mitgesendet wird, so dass ein ursprünglich angemeldeter Nutzer sich erneut anmelden muss. Wird der Wert „Lax“ gesetzt, wird der Cookie im Fall von Cross-Site-Requests ausschließlich bei als sicher geltenden HTTP-Methoden (GET) sowie für die Top-Level-Navigation (also nicht bei Inlineframes, Image-Tags usw.) weiterhin mitgesendet. Nur für eine bzgl. der HTTP-Methoden stringent implementierte Web-Anwendung wird daher auch der Wert „Lax“ einen ausreichenden Schutz gegen CSRF-Angriffe bieten.

Bei einer anderen CSRF-Variante, Login-CSRF, muss das Opfer keine aktuell gültige Session besitzen. Vielmehr fälscht der Angreifer in dem Fall einen Login-Request für die verwundbare Web-Anwendung, wobei in dieser gefälschten Anfrage die Zugangsdaten des Angreifers verwendet werden. Ist der Angriff erfolgreich, wird das Opfer mit dem Account des Angreifers angemeldet. Alle nachfolgenden Requests erfolgen dann mit dem Account des Angreifers. Ein Szenario, wo ein derartiger Angriff interessant sein kann, ist die ungewollte Anmeldung an einer Suchmaschine mit der Konsequenz, dass der Angreifer die nachfolgenden Suchanfragen des Opfers nachvollziehen kann.

Ein Logout-CSRF-Angriff könnte dagegen beispielsweise zu DoS-Szenarien führen.

ID: 3.22-282/13.0

Req 283	Die Web-Anwendung muss einen Mechanismus gegen Clickjacking-Angriffe einsetzen, der verhindert, dass die Web-Anwendung von anderen unberechtigten Web-Anwendungen eingebunden wird.
---------	---

Dies geschieht in der Regel durch Ausgabe des HTTP-Headers „X-Frame-Options“ in den Responses der Web-Anwendung. Hiermit wird festgelegt, ob die Inhalte der Web-Anwendung innerhalb eines Frames (also eines „frame“- oder „iframe“-Elements) einer anderen Web-Anwendung dargestellt werden dürfen. Der Header wird dabei auf „DENY“ oder „SAMEORIGIN“ gesetzt.

Wenn die Verwendung dieses Headers nicht möglich ist, kann auch über die Content Security Policy (CSP) und deren Direktive „frame-ancestors“ Clickjacking verhindert werden. Bei Verwendung des CSP-Headers ist neben den Werten 'none' und 'self' auch die Angabe bestimmter Domains (inkl. Wildcards) möglich, um dedizierten Domains zu erlauben, die Web-Anwendung einzubinden.

Insbesondere die Formulare und Funktionen in der Web-Anwendung, die eine Datenänderung oder Zustandsänderung verursachen können, sind entsprechend zu schützen. Es wird aber empfohlen, auch Seiten, die lediglich Informationen darstellen, davor zu schützen, dass sie innerhalb eines Frames dargestellt werden, da dies mit Seiteneffekten verbunden ist, die zu weiteren Angriffsszenarien führen können.

Motivation: Clickjacking (kurz für Click-Hijacking, auch als UI-Redressing bekannt) ist eine Angriffstechnik, bei der ein Angreifer ein Opfer dazu bringt, auf Elemente zu klicken, auf die es gar nicht klicken will. Der Angreifer fügt Teile der legitimen Web-Anwendung in seine Seite ein, in der Regel mithilfe eines Inlineframes, aber getarnt bzw. überdeckt durch andere Elemente seiner eigenen Seite. Das Opfer wird dann dazu verführt, einen Klick vermeintlich auf Inhalte der Seite des Angreifers auszuführen. Tatsächlich führt es aber in dem Moment einen Klick und eine Aktion auf seiner legitimen Web-Anwendung aus.

Weitere Angriffsszenarien bei Web-Anwendungen, die innerhalb eines Frames dargestellt werden können, ergeben sich beispielsweise dadurch, dass im Falle des Internet Explorers der „Document Mode“ an die Seite im Frame vererbt wird. Durch so einen vom Angreifer erzwungenen „Downgrade“ der Rendering-Engine können aber auch Schwachstellen (vor allem für XSS), die lediglich in neueren Versionen des Internet Explorers behoben wurden, wieder reaktiviert werden.

Mit dem HTTP-Header „X-Frame-Options“ wird die Darstellung in einem Frame verhindert. Wird das Element auf

„DENY“ gesetzt, wird das Darstellen der Inhalte in einem Frame generell verhindert. Der Wert „SAMEORIGIN“ beschränkt die Darstellung in Frames auf Seiten innerhalb derselben Domain, in der sich die Web-Anwendung befindet.

Mit dem HTTP-Header „Content-Security-Policy“ kann ebenfalls die Darstellung in einem Frame verhindert werden. Bei Angabe der Direktive „frame-ancestors 'none'“ wird das Darstellen der Inhalte in einem Frame generell verhindert. Die Direktive „frame-ancestors 'self'“ beschränkt die Darstellung in Frames auf Seiten innerhalb derselben Domain, in der sich die Web-Anwendung befindet. Da aber auch eine oder mehrere Domains (inkl. Wildcards) in der Direktive angegeben werden können, die dann die Web-Anwendung innerhalb eines Frames darstellen können, ist diese Variante flexibler als der HTTP-Header „X-Frame-Options“. Allerdings wird die Direktive noch nicht von allen gängigen Browsern unterstützt. Mittels JavaScript-Code könnte zwar auch überprüft werden, ob die Seite, in der sich der Code befindet, wirklich die oberste Seite in der Frame-Hierarchie ist, um dann die Darstellung innerhalb eines Frames zu verhindern. Leider sind verschiedene Techniken bekannt, die die gängigen Varianten dieser JavaScript-Mechanismen wieder aushebeln.

ID: 3.22-283/13.0

Req 284 Berechtigungen für Cross-Domain-Zugriffe müssen restriktiv definiert werden.

Die Web-Anwendung darf Cross-Domain-Zugriffe ausschließlich solchen Domains erlauben, für die dies explizit vorgesehen und zwingend erforderlich ist. Cross-Domain-Zugriffe auf schutzbedürftige Inhalte der Web-Anwendung dürfen nicht ermöglicht werden. Insbesondere dürfen keine Cross-Domain-Zugriffe auf die Inhalte von authentifizierten Sessions ermöglicht werden. Intranet-Anwendungen dürfen keine Berechtigungen für Cross-Domain-Zugriffe gewähren für Domains, die nicht zum Intranet gehören.

Cross-Origin Resource Sharing (CORS) ist ein Mechanismus, um Ausnahmen von den Regeln der Same Origin Policy (SOP) zuzulassen und Kommunikation zwischen Web-Anwendungen über den Browser des Benutzers zu realisieren. Mittels CORS wird ermöglicht, dass der Browser Cross-Origin-Requests initiiert und es zulässt, dass die Response ausgelesen wird. Diese Requests erfolgen zum Beispiel mittels XMLHttpRequests (XHR). Der Mechanismus basiert auf dem Austausch von Header-Informationen. Browser senden den „Origin“-Header, der die Domain der Anwendung angibt, die den Request abschickt. Server senden ihrerseits den „Access-Control-Allow-Origin“-Header zurück, um auszudrücken, welche Domain auf ihre Ressourcen zugreifen darf. Dies wird dann durch die Browser entsprechend durchgeführt.

Eine Berechtigung für einen solchen Cross-Domain-Zugriff muss daher restriktiv definiert werden ohne Verwendung von Wildcards im „Access-Control-Allow-Origin“-Header. Falls der Wert des „Access-Control-Allow-Origin“-Headers dynamisch generiert wird (ggf. abhängig vom jeweiligen „Origin“-Header des Requests), muss der „Origin“-Header validiert werden, müssen zugelassene Domains restriktiv definiert werden und zudem muss zusätzlich der „Vary: Origin“-Header gesetzt werden, um Angriffe basierend auf Cache-Poisoning zu unterbinden. Die Header dürfen zudem vom Server nur bei ausgewählten URLs/Ressourcen gesendet werden, für die ein Cross-Domain-Zugriff vorgesehen ist. Sie dürfen nicht pauschal für die ganze Anwendung gesendet werden.

Die „window.postMessage“-API erweitert jedes Fenster bzw. jeden Frame um eine neue Methode, die es erlaubt, Textnachrichten von einem Fenster an ein anderes zu versenden. In dem Fall muss der „Empfänger“ der Daten explizit definiert werden, um zu vermeiden, dass die Daten an ein falsches Ziel geschickt werden. Insbesondere darf der Empfänger nicht auf „*“ gesetzt werden. Der „Absender“ („origin“-Attribut) muss verifiziert werden. Die Daten („data“-Attribut) müssen validiert werden. Die ausgetauschten Daten müssen als Daten ausgewertet werden, niemals als Code (zum Beispiel mittels „eval“). Um DOM-basierte XSS-Angriffe zu vermeiden, darf das DOM einer Seite nicht basierend auf den ausgetauschten Daten angepasst werden (zum Beispiel mittels „innerHTML“).

Die Web-Anwendung darf keine RIA-Services nutzen, die die SOP umgehen. Hierzu zählen unter anderem AJAX-Service-Bridges, HTML-Bridges, AJAX-Proxys sowie Aggregate-Sites. Behelfslösungen und veraltete Funktionen dürfen ebenfalls nicht verwendet werden, um Cross-Domain-Zugriffe zu ermöglichen: Dies gilt insbesondere für Fragmentbezeichner (dem Teil einer URL nach dem Hash-Zeichen), für JSONP (JSON with Padding) und auch für die Property „document.domain“, mit der eine Web-Anwendung die eigene Origin (auf eine übergeordnete Domain) ändern könnte. Grundsätzlich sind standardisierte Technologien proprietären Technologien oder Behelfslösungen vorzuziehen.

Alle Vorgaben hinsichtlich Cross-Domain-Interaktion gelten gleichermaßen für Fenster, Dialogfelder, Frames usw.

Motivation: Die SOP ist ein in den Browsern implementiertes Sicherheitskonzept. Es soll sicherstellen, dass jede Ressource einer Webseite nur mit dem Server kommunizieren darf, von dem sie geladen wurde, und nur auf Objekte zugreifen darf, die ebenfalls vom gleichen Server geladen wurden. Domain-übergreifende Zugriffe werden somit unterbunden.

Für einzelne Web-Techniken gibt es jedoch Konfigurationsmöglichkeiten, die SOP aufzuweichen und Cross-DomainZugriffe zu erlauben. Dies ist aber gefährlich, da dadurch andere Web-Anwendungen in der Lage sind, Code im Kontext der Domain der eigenen Web-Anwendung auszuführen. Eine restriktive Definition von ggf. notwendigen Berechtigungen kann das damit verbundene Risiko reduzieren. Falls Cross-Domain-Zugriffe zu freizügig gewährt werden, kann hierüber unter Umständen sogar auf schutzbedürftige Daten zugegriffen werden, die ansonsten durch eine Authentifizierung geschützt sind. Denn wenn Cross-Domain-Requests verschickt werden, senden Browser diese Requests zusammen mit den Cookies für die Domain der Web-Anwendung, an die diese Requests gehen. Deren Responses können clientseitig ausgewertet und so ggf. auf schutzbedürftige Inhalte zugegriffen werden. Und auch der grundlegende Schutz vor Zugriff aus dem Internet wird ausgehebelt, wenn eine Anwendung, die ansonsten nur aus dem Intranet erreichbar ist, Cross-Domain-Zugriffe aus anderen Domains heraus erlaubt.

ID: 3.22-284/13.0

Req 285	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss der Registrierungsprozess der Web-Anwendung sicherstellen, dass der Benutzer sein Passwort selber wählt: Entweder setzt der Benutzer das Passwort direkt bei der Registrierung; oder er erhält ein individuelles Initialpasswort, das er bei der ersten Anmeldung sofort ändern muss.
---------	---

Es wird empfohlen, dass die Web-Anwendung den Benutzer über die Kriterien für Passwörter informiert. Darüber hinaus kann die Web-Anwendung ggf. eine Anzeige mit einer Indikation der Passwortstärke („Passwort-Meter“) einblenden. Alternativ kann die Web-Anwendung auch visualisieren, welche Kriterien der Passwort-Policy durch das eingegebene Passwort erfüllt werden.

Motivation: Vordefinierte Passwörter werden von den Benutzern oftmals nicht sorgfältig behandelt, möglicherweise sogar notiert, da sie in der Regel nicht gleichzeitig sicher und einfach zu erinnern sind. Initialpasswörter werden zudem häufig unverschlüsselt übertragen oder von Dritten im Rahmen von Support-Prozessen gesetzt. Um die Gefahr eines Missbrauchs zu reduzieren, müssen neue Benutzer diese Passwörter umgehend ändern. Transparente Kriterien für die Wahl von starken Passwörtern erhöhen die Akzeptanz und das Bewusstsein des Benutzers für Sicherheit

ID: 3.22-285/13.0

Req 286	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss die Web-Anwendung über eine gegen Missbrauch geschützte Funktion verfügen, über die der Benutzer das Passwort zurücksetzen kann.
---------	--

Ein Missbrauch, der dazu führt, dass über diese Funktion die vorhandenen sicheren Authentisierungsmechanismen umgangen werden können, muss verhindert werden.

Es wird daher empfohlen, dass der Benutzer hierfür eine „Sicherheitsfrage“ auswählen und die dazugehörige Antwort eingeben muss (für kritische Anwendungen ggf. auch mehrere Fragen und Antworten). Diese Daten muss die Web-Anwendung im Rahmen des Registrierungsprozesses oder nach der initialen Anmeldung des Benutzers abfragen. Wenn ein Benutzer sein Passwort vergessen hat, kann er bei der Anmeldung eine entsprechende Funktion auswählen. Nachdem er seine Sicherheitsfrage korrekt beantwortet hat, wird sein Passwort zurückgesetzt: Die WebAnwendung erzeugt ein neues Initialpasswort oder einen Link mit einem Aktivierungstoken. Sie kann das Initialpasswort bzw. Aktivierungstoken dem Benutzer per Push-Notification oder SMS zusenden, alternativ per E-Mail, an entsprechend vorab hinterlegte Kontaktinformationen.

Diese Funktion darf jedoch auf keinen Fall vorsehen, dass sich der Benutzer allein durch Beantwortung der Sicherheitsfrage erfolgreich an der Web-Anmeldung anmelden kann oder auch direkt ein neues Passwort vergeben kann. Es ist auch zu verhindern, dass diese Funktion einem Angreifer die Möglichkeit bietet, automatisiert valide Benutzer-namen zu ermitteln. Hierzu können Captchas abgefragt oder Schwellwerte für ungültige Eingaben/Versuche vorgesehen werden, die zu einer (temporären) Sperrung dieser Funktionalität (zum Beispiel für eine IP-Adresse) führen. Für den Fall, dass ein Benutzer auch die Antwort seiner Sicherheitsfrage vergessen hat, kann ggf. ein Fallback-Mechanismus vorgesehen werden. Hierfür wird dem Benutzer dann ein weiterer Prozess angeboten, der aber in der Regel nicht automatisiert durchgeführt werden kann, zum Beispiel über den Anruf bei einer Hotline. Auch bei einem solchen Fallback-Mechanismus muss sichergestellt sein, dass dieser nicht missbraucht werden kann.

Motivation: Wenn ein Benutzer sein Passwort vergessen hat, muss er die Möglichkeit haben, sein Passwort zurücksetzen zu lassen. Hierdurch darf aber nicht das vorhandene Sicherheitsniveau der Web-Anwendung reduziert werden. So

muss verhindert werden, dass ein Angreifer die Möglichkeit erhält, die (Initial-)Passwörter anderer Benutzer zu ermitteln und deren Benutzerkonten zu übernehmen. Ebenso ist zu verhindern, dass ein Passwortrücksetzungsprozess (automatisiert) missbraucht wird, zum Beispiel indem massenhaft Passwörter zurückgesetzt und damit andere Benutzer belästigt werden.

ID: 3.22-286/13.0

Req 287	Falls die Web-Anwendung Initialpasswörter oder Aktivierungstoken verwendet, müssen diese gegen missbräuchliche Verwendung sowie Brute-Force-Angriffe abgesichert werden.
---------	--

Insbesondere müssen sie nach einer angemessenen Zeit ihre Gültigkeit verlieren. Die genaue Gültigkeitsdauer muss individuell für jede Anwendung festgelegt werden. Sie ist abhängig von deren Sensibilität und Einsatzzweck sowie von dem Verfahren der Passwortvergabe. Falls zum Beispiel bei einer aus dem Internet erreichbaren Web-Anwendung der Benutzer die Registrierung selber startet und die Initialpasswörter (oder Aktivierungstoken) auf elektronischem Weg versendet werden, müssen sie wenige Stunden nach ihrer Ausstellung ihre Gültigkeit verlieren. Bei einer internen Anwendung und Versand ausschließlich über interne Netze kann dieser Zeitraum deutlich länger gewählt werden (bis hin zu mehreren Tagen).

Initialpasswörter müssen die gleichen Komplexitätsanforderungen erfüllen wie Passwörter, die die Benutzer wählen.

Die Länge der Aktivierungstoken muss (analog zu Session-Identifiern) mindestens 120 Bit betragen. Alle hierbei relevanten Zeichen müssen nach dem Zufallsprinzip generiert werden.

Motivation: Falls ein neuer Benutzer sein Initialpasswort nach längerer Zeit noch nicht verwendet hat, kann davon ausgegangen werden, dass ein Fehlerfall vorliegt oder der Benutzer die Registrierung nicht abschließen will. Zudem ist das Initialpasswort oftmals noch zugreifbar (zum Beispiel in Posteingangsfächern). Es kann daher von Unberechtigten ermittelt und missbraucht werden. Deshalb muss es zeitnah seine Gültigkeit verlieren.

Ohne entsprechenden Schutz könnte ein Angreifer zudem durch bloßes Durchprobieren versuchen, ein Initialpasswort oder einen Aktivierungstoken zu ermitteln.

ID: 3.22-287/13.0

Req 288	Bei einem fehlgeschlagenen Anmeldeversuch eines Benutzers darf die Web-Anwendung keine Information ausgeben, welche der eingegebenen Anmeldeinformationen nicht korrekt war
---------	---

Bei einem fehlgeschlagenen Anmeldeversuch eines Benutzers darf die Web-Anwendung keine Information ausgeben, welche der eingegebenen Anmeldeinformationen nicht korrekt war

Motivation: Durch eine allgemeine Fehlermeldung wird es einem Angreifer erschwert, gültige Benutzernamen zu ermitteln.

ID: 3.22-288/13.0

2.16.1. Content-Management-Systeme (CMS)

Req 289	Der Zugang zur Redaktionsumgebung des Content-Management-Systems (CMS) darf nicht aus dem Internet erfolgen.
---------	--

Entweder ist die Redaktionsumgebung nur aus internen Netzen erreichbar, oder eine Zugangsbeschränkung wird über technische Lösungen realisiert, zum Beispiel VPNs.

Ist ein Schutz über eine Zugangsbeschränkung nicht realisierbar, muss der Zugang zur Redaktionsumgebung durch eine andere zusätzliche Sicherungsebene geschützt werden, beispielsweise durch Verwendung einer Zweifaktorauthentisierung.

Motivation: So wird das Risiko minimiert, dass Angreifer sich unberechtigt Zugang zum CMS verschaffen (zum Beispiel über bekannt gewordene Schwachstellen des CMS) und Inhalte modifizieren oder unveröffentlichte Informationen auslesen.

Req 290	Das CMS muss es ermöglichen, Berechtigungen für unterschiedliche Tätigkeiten des Content-Managements verschiedenen Nutzern/Nutzergruppen zuzuweisen, so dass ein mehrstufiger Veröffentlichungsprozess realisiert werden kann.
---------	--

Mögliche unterschiedliche Nutzergruppen sind „Reader“, „Editor“, „Chief Editor“, „Administrator“, „Approver“ und „Publisher“. Die Veröffentlichung von Inhalten muss minimal im Vieraugenprinzip durch einen „Editor“ und einen „Approver“ bzw. „Publisher“ erfolgen.

Motivation: Durch einen geeigneten Veröffentlichungsprozess kann das Risiko minimiert werden, dass (durch böse Absicht oder aufgrund eines Fehlers) unerwünschte oder fehlerhafte Inhalte veröffentlicht werden.

Req 291	Das CMS muss es ermöglichen, bestimmte Inhalte exklusiv bestimmten Redakteuren oder einer Gruppe von Redakteuren zuzuweisen.
---------	--

Die entsprechenden Inhalte dürfen dann nicht durch andere, nicht berechnigte Redakteure/Gruppen einsehbar oder änderbar sein. Dies kann realisiert werden, indem mehrere Mandanten vom CMS bedient werden können, oder durch ein entsprechendes Rollenkonzept.

Motivation: So wird ermöglicht, unveröffentlichte (geheime) Inhalte einzupflegen, ohne dass sie direkt allen Redakteursgruppen bekannt werden. Außerdem wird so das Risiko unberechtigt modifizierter Inhalte minimiert.

Req 292	Falls das CMS Funktionen bereitstellt, um Inhalte zu erstellen, die erst zu einem späteren Zeitpunkt veröffentlicht werden sollen, dürfen diese Inhalte vorher in der Web-Anwendung nicht einsehbar oder auffindbar sind.
---------	---

Diese unveröffentlichten Inhalte dürfen auch nicht durch Suchfunktionen der Web-Anwendung, geheime URLs oder aktive Manipulationen auffindbar sein.

Motivation: So wird ermöglicht, unveröffentlichte (geheime) Inhalte einzupflegen, ohne dass sie bereits von Nutzern der Web-Anwendung einsehbar sind.

Req 293	Das CMS muss Funktionen bereitstellen, die es erlauben, die Verwendung von aktivem Content und Scripting innerhalb der erstellten Inhalte einzuschränken (auf bestimmte Inhalte bzw. für bestimmte Benutzergruppen) und ggf. ganz zu unterbinden.
---------	---

Motivation: Inhalte, die potenziell eine Gefährdung für die Nutzer der Web-Anwendung darstellen können, dürfen nur bei entsprechender Notwendigkeit verwendet werden und nur von Redakteuren erstellt werden, die hierfür explizit verantwortlich sind. Beispielsweise könnten durch die Verwendung von Scriptsprachen Inhalte mit XSS-Angriffen erstellt werden.

Req 294	Preview-Funktionalitäten des CMS müssen vor unberechtigtem Zugriff geschützt werden: Nur erfolgreich authentifizierte Benutzer des CMS dürfen die Inhalte des Previews einsehen; sie dürfen hierbei auch nur auf die Inhalte zugreifen, für die sie mindestens Leserechte haben.
---------	--

Motivation: So wird ermöglicht, unveröffentlichte oder unvollendete Inhalte einzupflegen und zu prüfen, ohne dass sie bereits von Nutzern der Web-Anwendung oder von anderen, nicht berechtigten Redakteuren einsehbar sind.

ID: 3.22-294/13.0

2.17. Proxy-Server

Req 295	Der Zugriff auf administrative Schnittstellen eines Proxyservers aus dem Internet muss verhindert werden.
---------	---

Administrative Schnittstellen eines Proxyservers, beispielsweise ein SSH-Zugang oder ein Webinterface, dürfen aus dem Internet nicht ansprechbar sein. Der Zugang darf nur aus und über ein Netz erfolgen, das sich vollständig unter der Kontrolle der Deutschen Telekom Gruppe befindet.

Motivation: Über administrative Schnittstellen eines Proxyservers lassen sich alle Konfigurationseinstellungen eines Proxyservers ändern. Durch eine entsprechende Änderung der Konfiguration kann sich ein Angreifer daher Zugriff auf weitere Systeme verschaffen, Sicherheitsregeln außer Kraft setzen oder sogar direkt in den über den Proxyserver laufenden Datenverkehr Einblick nehmen. Es ist daher wichtig, den Zugriff auf administrative Schnittstellen so weit wie möglich zu beschränken.

ID: 3.22-295/13.0

Req 296	HTTP-Proxyserver müssen sich an das in einem Request oder einer Response geforderte Caching-Verhalten halten.
---------	---

Ein HTTP-Proxyserver darf nicht so konfiguriert werden, dass eine Datenspeicherung länger erfolgt, als dies in Request oder Response angegeben ist. Insbesondere darf bei einem vorliegenden HTTP-Header „cache-control: no-store“ nicht gespeichert werden.

Motivation: Sensible Daten lassen sich durch entsprechend gesetzte Cache-Control-Header vor unerwünschter Speicherung schützen. Daher ist es wichtig, dass sich jede am Datenaustausch beteiligte Komponente an das vorgegebene Caching-Verhalten hält.

ID: 3.22-296/13.0

Req 297	Auf ausgehenden Proxyservern muss ein Verfahren eingerichtet sein, um unerwünschte Zieladressen oder Content zu sperren.
---------	--

Auf ausgehenden Proxyservern muss ein Verfahren eingerichtet sein, mittels dessen sich der Zugriff auf unerwünschte Zieladressen oder unerwünschten Content sperren lässt (Filterung).

Motivation: Nicht jeden Inhalt darf das System an einen Client-Rechner ausliefern. Neben Inhalten, die für den Client oder das Unternehmen schädlichen Programmcode enthalten, ist insbesondere auch auf rechtswidrige Inhalte zu achten.

ID: 3.22-297/13.0

Req 298	Bei M2M-Verbindungen über ausgehende Proxyserver muss der Zugriff auf nicht erforderliche URLs gesperrt werden.
---------	---

Bei M2M-Verbindungen über ausgehende Proxyserver muss der Zugriff auf nicht erforderliche URLs gesperrt werden.

Motivation: Die Begrenzung von aufrufbaren URLs reduziert das Risiko des Downloads von Malware über diese Verbindung und verringert die nutzbare Angriffsfläche.

Umsetzungsbeispiel: Beispielsweise könnte ein Windows-Server-System für Updates Zugriff über den Proxyserver auf den Microsoft Update-Server benötigen. Weitere Verbindungen benötigt der Server zur Durchführung der Updates allerdings nicht. Der Proxyserver muss sie daher unterbinden.

ID: 3.22-298/13.0

Req 299	Die HTTP-Methode „CONNECT“ darf nicht verwendet werden.
---------	---

In manchen Fällen erfolgt ein Öffnen der Verbindungen über ausgehende HTTP-Proxyserver mittels der HTTP-Methode „CONNECT“. In einem solchen Fall muss eine Einschränkung der Protokolle vorgenommen werden, die über die HTTP-Verbindung getunnelt werden können.

Motivation: Die HTTP-Methode „CONNECT“ erlaubt es, beliebige Protokolle über eine HTTP-Verbindung zu einem Proxyserver zu tunneln. Damit sind auch prinzipiell unerwünschte Verbindungen möglich, etwa eine VPN-Verbindung aus dem Intranet zu einem externen Netzwerk. Eine getunnelte Verbindung lässt sich von einem Proxyserver zudem nur eingeschränkt kontrollieren und überprüfen.

ID: 3.22-299/13.0

Req 300	Ausgehende Proxyserver dürfen die Netzwerkadressen der Clients nicht außerhalb des Intranets weitergeben.
---------	---

Ausgehende Proxyserver dürfen die Netzwerkadressen der Clients nicht außerhalb des Intranets weitergeben.

Motivation: Der Zielrechner benötigt die Netzwerkadresse des Clients üblicherweise nicht. Leitet das System die Netzwerkadresse eines Clients weiter, besteht für einen Angreifer die Möglichkeit, sich wertvolle Informationen über die Netzwerkstruktur des Intranets zu verschaffen.

ID: 3.22-300/13.0

Req 301	Zu jeder Verbindung muss der Proxyserver Informationen in einer Log-Datei ablegen.
---------	--

Zu jeder Verbindung muss der Proxyserver Informationen in einer Log-Datei ablegen.

Motivation: Erst das Logging ermöglicht die Nachverfolgung sicherheitsrelevanter Aktionen (beispielsweise der Angriff auf eine Webanwendung über einen Proxyserver) sowie eventuell die zukünftige Unterbindung dieser Aktionen.

Umsetzungsbeispiel: Insbesondere muss der Proxyserver die folgenden Informationen ablegen:

- Datum und Uhrzeit
- Quell-IP-Adresse oder eindeutiger Rechnername
- Ziel-IP-Adresse oder FQDN
- verwendetes Protokoll

Bei HTTP-Verbindungen muss der Proxyserver ebenfalls ablegen:

- Ziel-URL
- HTTP-Methode
- HTTP-Status-Code

ID: 3.22-301/13.0

Req 302 Jeder Anwendung, die über einen Reverse Proxyserver angesprochen wird, muss auf dem Proxyserver eine eigene Netzwerkadresse (IP Adresse) oder ein eigener Port zugeordnet sein.

Wenn zu einer Anwendung mehrere vollständige Domain-Namen (FQDN) gehören, können diese auch an dieselbe IP-Adresse gebunden sein (gegebenenfalls per Alias).

Motivation: Um zu verhindern, dass auf eine Anwendung (gegebenenfalls auch nur auf Teile einer Anwendung) unberechtigter Zugriff besteht, muss eine (logische) Trennung von anderen Anwendungen erfolgen.

ID: 3.22-302/13.0

Req 303 Netzwerkadresse und Port müssen bei einem Reverse-Proxy eindeutig auf eine Applikation zeigen.

Über eine eingehende Netzwerkadresse (IP-Adresse) und einen Port darf nur der Zugriff auf die zugehörigen Anwendungen möglich sein.

Motivation: Bei eingehenden Verbindungen muss sichergestellt sein, dass der Proxyserver diese an die richtige Anwendung weiterleitet. Andernfalls könnte sich ein Angreifer über einen Proxyserver unbefugten Zugriff zu anderen Anwendungen verschaffen.

ID: 3.22-303/13.0

Req 304 Ein Reverse HTTP-Proxyserver muss die HTTP-Methode die nicht zulässig sind ablehnen.

Ein Reverse HTTP-Proxyserver muss Requests mit nicht erforderlichen HTTP-Methoden zurückweisen. Insbesondere darf ein Reverse HTTP-Proxyserver die HTTP-Methode „CONNECT“ nicht unterstützen. Üblicherweise werden nur die Methoden „GET“ und „POST“ benötigt.

Motivation: Durch die Nutzung von nicht erforderlichen HTTP-Methoden kann sich ein Angreifer eventuell unberechtigten Zugriff auf einen nachgelagerten Web- oder Applikationsserver verschaffen.

Umsetzungsbeispiel: Es wird empfohlen, dass der Proxyserver Requests mit nicht erforderlichen Methoden mit dem HTTP Status Code 405 (Method Not Allowed) beantwortet. Er sagt aus, dass die Methode der Anfrage für das gewünschte Ziel nicht erlaubt ist .

ID: 3.22-304/13.0

Req 305 Eingehenden Netzwerkverbindungen müssen auf dem Reverse Proxyserver terminiert werden

Alle eingehenden Netzwerkverbindungen müssen auf dem Reverse Proxyserver terminiert werden. Ein Reverse Proxyserver muss eingehende Verbindungen mit Netzwerkprotokollen ablehnen, die nicht auf dem Proxyserver terminiert werden können.

Motivation: Wenn eine Verbindung nicht auf dem Proxyserver terminiert wird, ermöglicht dies den direkten Zugriff über den Proxyserver auf ein nachgelagertes System. Der Proxyserver gewährt somit bestenfalls eine minimale Sicherheit. Ein nachgelagertes System befindet sich zudem eventuell in einem nicht entsprechend geschützten Netzbereich, so dass sich für einen Angreifer verschiedene Angriffsvektoren ergeben.

ID: 3.22-305/13.0

Req 306 Reverse Proxyserver müssen Requests ablehnen, die nicht den entsprechenden Protokoll-Spezifikationen genügen.

HTTP-Requests müssen RFC2616 entsprechen. Wenn beispielsweise ein inkorrektter Content-Length-Header gesendet wird, muss der Proxyserver den Request ablehnen. Er muss den Request zudem mit einem HTTP Status Code 403 ohne eine nähere Fehlerbeschreibung beantworten.

FTP Requests müssen RFC 959 genügen, für weitere Protokolle gilt der entsprechende RFC.

Motivation: Das RFC 2616 behandelt das Hypertext Transfer Protocol 1.1 , welches als Standard festgelegt ist. Wenn ein Request nicht der Spezifikation gemäß RFC genügt, ist dies möglicherweise ein Hinweis auf einen Angriffsversuch.

ID: 3.22-306/13.0

Req 307	Ein Reverse Proxyserver muss die URL eingehender Requests auf Konformität zu einem vorgegebenen Muster überprüft und gegebenenfalls abzuweisen.
---------	---

Ein Reverse Proxyserver muss die URL eingehender Requests auf Konformität zu einem vorgegebenen Muster überprüfen. Das Muster ist von der über den Reverse Proxyserver angesprochenen Anwendung abhängig und daher meistens individuell zu definieren. Unpassende Muster müssen abgelehnt werden.

Motivation: URLs, die nicht einem durch eine nachgelagerte Anwendung definierten und von der Anwendung erwarteten Muster entsprechen, deuten auf einen Angriff hin. Sie könnten in der Anwendung ein unerwartetes Verhalten verursachen oder einem Angreifer unerlaubten Zugriff auf die Anwendung ermöglichen.

Umsetzungsbeispiel: Requests, bei denen die URL nicht dem Muster entspricht, insbesondere solche Requests, bei denen die URL (für die Anwendung) ungültige Zeichen enthält sollten der Proxy-Server ablehnen.

ID: 3.22-307/13.0

Req 308	Verbindungen über ausgehende Proxyserver müssen authentifiziert und autorisiert werden.
---------	---

Die Identität eines Benutzers oder einer Maschine muss eindeutig festgestellt werden. Die Authentifizierung einer Maschine darf dabei über die Netzwerkadresse (IP), einen Verzeichnisdienst oder ein Zertifikat erfolgen. Es muss überprüft werden, ob der Benutzer oder die Maschine für den Zugriff auf die Zielressource autorisiert ist.

Motivation: Die Authentifizierung und Autorisierung von Verbindungen schränken die Wahrscheinlichkeit des Missbrauchs eines Proxyservers deutlich ein. Wenn ein Sicherheitsvorfall eintritt, bei dem eine Verbindung über den Proxyserver erfolgte, lassen sich zudem die Authentifizierungsinformationen mit Hilfe der Protokollierung zur Nachverfolgung nutzen.

ID: 3.22-308/13.0

2.18. SAP Fiori Applikation Server

Req 309	Es muss sichergestellt sein, dass die Netztrennung zwischen dem SAP-Netweaver-Server und den Fiori-Apps (UI5-Apps) besteht.
---------	---

Es muss sichergestellt sein, dass eine Netztrennung zwischen dem SAP-Netweaver-Server und den UI5 /Fiori-Apps besteht.

Motivation: Schutz der SAP Applikation vor unberechtigten Zugriff.

Umsetzungsbeispiel: Fiori-Apps müssen immer einen Netweaver-Server erreichen. Dieser wird in der Regel in der DMZ aufgebaut. Alternativ zum Netweaver-Server für Fiori-Apps, kann ein Gateway, welches in der DMZ aufgebaut wird, verwendet werden.

ID: 3.22-309/13.0

2.19. SAP Fiori-Apps

Vorgaben zur Entwicklung von SAP-Fiori-Applikationen und anderen Client-Applikationen.

Req 310 Bei der Verwendung von Fiori-Applikationen aus der SAP Cloud Platform muss immer der SAP Cloud Connector zum Einsatz kommen.

Bei der Verwendung von Fiori-Applikationen aus der SAP Cloud Platform muss immer der zentrale SAP Cloud Connector zum Einsatz kommen (vergleichbar mit der Landing Zone der AWS).

Motivation: Ohne ein trennendes Gateway, wie der SAP Cloud Connector oder ein Proxy für das entsprechende Protokoll, ist ein Zugriff nicht erlaubt.

Umsetzungsbeispiel: Bei Zugriffen von Fiori-Apps aus der SAP-Cloud heraus muss der SAP Cloud Connector verwendet werden.

ID: 3.22-310/13.0

Req 311 Bei der Verwendung von Fiori muss eine personalisierte Anmeldung erfolgen.

Bei der Nutzung von Fiori muss eine oder auch mehrere Rollen auf dem Backendsystem explizit für Fiori vorhanden sein. Für die Benutzung der Applikation über Fiori, muss sich jeder Benutzer mit seiner eigenen Kennung autorisieren. Es darf keine Sammelbenutzer geben.

Motivation: Eine eindeutige Identifizierung macht eine Nachvollziehbarkeit der Änderungen und Einsichten auf dem System sichtbar.

Umsetzungsbeispiel: Anmeldung via Username/Passwort oder MyCard.

ID: 3.22-311/13.0

Req 312 Bei der Entwicklung von Fiori-Apps darf es nicht zu einer Vermischung von unterschiedlichen Web-Technologien für den SAP Systemzugriff kommen.

Bei der Entwicklung von Fiori-Apps darf es nicht zu einer Vermischung von unterschiedlichen Web-Technologien für den SAP Systemzugriff kommen.

Motivation: Schutz des Systems vor unterschiedlichen Technologien und deren Eigenarten.

Umsetzungsbeispiel: SAP liefert zwei Ausführungen seines Development Toolkits aus: SAPUI5 und die Open-Source Variante OpenUI5. Das JavaScript-Framework OpenUI5 hat SAP unter einer Apache-2.0-Lizenz entwickelt, um betriebssystemunabhängige Geschäftsanwendungen zu erstellen.

Da OpenUI5 auf CSS3, HTML5 und der JavaScript-API von ECMA-Script 5 (ES5) basiert, werden nur Browser mit HTML5-Fähigkeiten unterstützt. Mit iOS-, Android-, MacOS- und Windows befinden sich darunter die relevanten Plattformen.

Auf der Client-UI-Technologie OpenUI5 entwickelte Apps laufen im Browser und somit auf jedem Endgerät. Greift der Anwender auf eine App zu, wird eine Anfrage an den jeweiligen Server gesendet, um die Anwendung in den Browser zu laden.

ID: 3.22-312/13.0

Req 313 Für die Entwicklung einer Fiori-App muss ein Framework benutzt werden.

Für die Entwicklung einer Fiori-App muss eine Entwicklungsumgebung benutzt werden.

Motivation: Die Benutzung einer Entwicklungsumgebung hilft, ein Mindestmaß an Qualität sicherzustellen. Eclipse wird zur Entwicklung diversen Programmierungen genutzt (insbesondere für Java). Für SAP-Fiori-Applikationen sollte der native Runtime-Container aus dem AppStores verwendet werden.

Umsetzungsbeispiel: SAP liefert zwei Ausführungen seines Development Toolkits aus: SAPUI5 und die Open-Source Variante OpenUI5. Das JavaScript-Framework OpenUI5 hat SAP unter einer Apache-2.0-Lizenz entwickelt, um betrie-

bssystemunabhängige Geschäftsanwendungen zu erstellen.

Da OpenUI5 auf CSS3, HTML5 und der JavaScript-API von ECMA-Script 5 (ES5) basiert, werden nur Browser mit HTML5-Fähigkeiten unterstützt. Mit iOS-, Android-, MacOS- und Windows befinden sich darunter die relevanten Plattformen.

Auf der Client-UI-Technologie OpenUI5 entwickelte Apps laufen im Browser und somit auf jedem Endgerät. Greift der Anwender auf eine App zu, wird eine Anfrage an den jeweiligen Server gesendet, um die Anwendung in den Browser zu laden.

ID: 3.22-313/13.0

Req 314 Der Datenaustausch (z. B. OData, XML) muss verschlüsselt stattfinden.

Der Datenaustausch zwischen der Applikation und dem Backend insbesondere bei der Verwendung von ODATA und XML darf nur über verschlüsselte Verbindungen erfolgen.

Motivation: Beispielsweise werden über das Open Data (OData) Protokoll die Datenanfragen und Daten zwischen der Applikation und dem Backend übertragen. Durch eine Verschlüsselung werden die Daten vor der Einsicht Dritter geschützt.

ID: 3.22-314/13.0

Req 315 Es muss vermieden werden, dass Daten via JSON oder XML vom Server abgerufen werden.

Es muss vermieden werden, dass Daten via JSON oder XML vom Server abgerufen werden. Die Protokolle können nur komplette Datensätze vom Server abrufen und skalieren dadurch nicht so gut wie OData.

Motivation: Um bei großen Datenmengen performant zu sein, ist es sinnvoll, nur die benötigten Daten abzurufen. Im Gegensatz dazu wird der komplette Datensatz bei Verwendung von JSON oder XML geladen.

ID: 3.22-315/13.0

Req 316 Bei der Verwendung von Fiori-Apps muss sichergestellt sein, dass bekannte Web-Angriffe durch entsprechende Gegenmaßnahmen abgefangen werden.

Bei der Verwendung von Fiori-Apps muss sichergestellt sein, dass bekannte Web-Angriffe durch entsprechende Gegenmaßnahmen abgefangen werden.

Motivation: SAP Fiori ist eine in HTML-geschriebene Web-Applikation und ist dementsprechend genauso anfällig gegen die gängigen Angriffe wie eine klassische Webapplikation. Um einen unberechtigten Zugriff durch Dritte zu verhindern, muss SAP Fiori gehärtet werden.

Umsetzungsbeispiel: Cross-Site Scripting
Clickjacking framing protection

ID: 3.22-316/13.0

Req 317 Es dürfen keine "Offline-App's" zur Anwendung kommen.

Bei Offline-Apps handelt es sich um Apps, die Daten auf dem mobilen Gerät speichern und wiedergeben können, auch wenn sie keine Verbindung zum Server haben. Wenn die Daten einen Schutzbedarf der Klasse "intern" oder höher haben, dürfen keine "Offline-Apps" zur Anwendung kommen.

Motivation: Insbesondere ist bei Daten mit einem Schutzbedarf "intern" oder höher darauf zu achten, dass diese nicht gespeichert werden. Gespeicherte Daten könnten bei einem Verlust des Gerätes von unberechtigten Dritten eingese-

hen werden.

ID: 3.22-317/13.0

2.20. Cloudnutzung

Req 318	Vor dem Onboarding eines Projekts oder eines Systems auf einer Public Cloud / SAP Cloud Service muss eine klare Exit-Strategie existieren.
---------	--

Trotz guter Absichten und Verträge ist es möglich, dass der CSP nicht in der Lage sein wird, die Bedürfnisse der DTAG und der jeweiligen Systeme auch in Zukunft zu erfüllen. Um mögliche rechtliche, finanzielle oder technische Probleme zu berücksichtigen ist es notwendig einen Plan zu haben, um Dienste und Daten von einem bestimmten Cloud-Anbieter und zu einem bestimmten Zeitpunkt runter migrieren zu können.

In der traditionellen Welt wurde dieses Problem durch eine Dual- (oder Multi-) Vendor-Strategie gelöst, um Vendor-Lock-ins zu vermeiden. Der Übergang von einem Anbieter zum Anderen war möglich, weil die Geräte von der DTAG gekauft wurden und ihr gehörten. Dies ist nicht der Fall bei der Nutzung von Public Cloud-Anbietern. Zwei oder mehr CSPs zur Verfügung zu haben, ist nicht ausreichend für eine Dual- (oder Multi-) Vendor Strategie. Es ist auch notwendig, die betroffenen Systeme und Daten zu verlagern, entweder vom CSP zu einem anderen CSP oder vom CSP in die eigenen Räumlichkeiten.

Wenn möglich, sollten Cloud-Konsumenten in Erwägung ziehen, mehr generische Cloud-Dienste zu nutzen, so dass der Übergang in die Public Cloud von einem anderen CSP (oder in die eigenen Räumlichkeiten) mit minimalen Modifikationen (oder im besten Fall ohne Modifikationen) erfolgen kann.

Verträge und Gerichtsverfahren können nicht als brauchbarer Ersatz für eine Ausstiegsstrategie angesehen werden.

Motivation: Mit der Nutzung der Cloud, begibt man sich in eine Abhängigkeit. Mit der Exit stratigie soll versucht werden, sich nicht komplett abhängig zu machen.

ID: 3.22-318/13.0

Req 319	Die Verschlüsselung aller Daten bei der Übertragung, im Ruhezustand oder während der Verarbeitung muss entsprechend dem Erfordernis des Datenschutzes erfolgen.
---------	---

In einer Public-Cloud-Umgebung müssen alle gespeicherten Daten (im Ruhezustand) und alle nicht offenen Daten bei der Übertragung verschlüsselt werden, um einen unberechtigten Zugriff auf Daten zu verhindern, da eine Public Cloud von einem externen Unternehmen betrieben wird und es sich um eine mandantenfähige Umgebung handelt. Die obligatorische Verschlüsselung gilt für die Nutzung von Cloud-APIs und für alle Dienste, die innerhalb der Cloud laufen (z. B. die Kommunikation zwischen virtuellen Maschinen).

Aufstrebende Technologien wie die homomorphe Verschlüsselung oder die VM-Arbeitsspeicherverschlüsselung ermöglichen die Verschlüsselung von Daten während der Verarbeitung, aber sie ist noch nicht weit verbreitet. Wenn eine solche Technologie verfügbar ist, muss sie eingesetzt werden. Die Verfügbarkeit und Verwendung von solcher Technologie kann die Anzahl der Anwendungsfälle für Anwendungen, die in der Cloud laufen, erweitern.

Datenverschlüsselung muss für alle IaaS-Dienste und für PaaS-Dienste verwendet werden, wann immer dies möglich ist. SaaS-Dienste bieten typischerweise keine direkte Verschlüsselung an, sondern bieten Konfigurationsoptionen für die zugrundeliegende Verschlüsselung, die sinnvoll genutzt werden müssen (d. h. die Verschlüsselung muss eingeschaltet sein).

Es ist nicht notwendig, öffentliche Daten zu verschlüsseln (z. B. bei der Verarbeitung von rohem Internetverkehr). Obwohl Verschlüsselung eine Best Practice ist, ist es aus Performance-Gründen erlaubt, die Verschlüsselung zwischen verschiedenen Diensten innerhalb einer Einheit wegzulassen, die vollständig unter der Kontrolle der DTAG steht, z. B.:

- zwischen Containern innerhalb einer virtuellen Maschine, wenn sie immer als ganze Einheit agieren und es keine Möglichkeit gibt, dass einer von ihnen in einer anderen VM landet
- zwischen Prozessen innerhalb einer VM oder eines Containers

Wenn die verfügbaren Verschlüsselungsmethoden und Schlüsselverwaltungsmodelle es zulassen, ist die Verschlüsselung der Boot-Festplatte des Betriebssystems für virtuelle Maschinen ebenfalls notwendig.

Die Verschlüsselung von Daten ermöglicht eine einfachere Löschung von Daten, die z. B. für die Einhaltung der GDPR (General Data Protection Regulation) erforderlich ist, z.B. durch einfaches Löschen des Schlüssels, der für die Verschlüsselung der Daten im Ruhezustand verwendet wird.

Falls ein System, ein Datenspeicher oder ein Kommunikationspfad eine Mischung aus offenen und nicht-offenen (internen, vertraulichen) Daten enthält, ist es empfehlenswert, alles zu verschlüsseln.

Motivation: Schutz gegen die Einsicht der Daten von dritten. Diese kann durch Softwarefehler oder fehlerhafte Konfigurationen entstehen.

ID: 3.22-319/13.0

Req 320	Prozess für den Umgang mit Missbrauchs- und Vorfallsmeldungen vom und zum CSP muss etabliert werden.
---------	--

CSP benachrichtigen Kunden in der Regel bei festgestelltem Missbrauch und Vorfällen, manchmal auch bei festgestellten Unregelmäßigkeiten (z. B. ein sprunghafter Anstieg des Ressourcenverbrauchs). Die Landing-Zone-Betreiber müssen in Zusammenarbeit mit dem zuständigen SOC einen Prozess etablieren, der die Bearbeitung solcher Meldungen rund um die Uhr ermöglicht.

Ebenso muss es für einen Cloud-Nutzer oder Landing-Zone-Betreiber möglich sein, Vorfälle und Missbrauch an den CSP zu melden, was durch die Landing-Zone-Betreiber in Absprache mit dem zuständigen SOC organisiert werden kann.

ID: 3.22-320/13.0

Req 321	Die IAM-Lösung der DTAG muss für die Benutzerauthentifizierung verwendet werden, wenn sie vom CSP unterstützt wird. Wenn nicht, muss eine Automatisierung für die Verwaltung von Identitäten in der Cloud vorhanden sein.
---------	---

Die doppelte Verwaltung von Benutzern bringt Probleme mit sich, wie z. B. die nicht rechtzeitige Erstellung und Löschung von Konten (was zu Datenleaks führen kann), daher ist es notwendig, dass Konten in der Cloud auf Konten im IAM der eigenen DTAG abgebildet werden. In Bezug auf die Technologien: Active Directory Identity Federation, AFDS Shared (AAD wird in naher Zukunft erwartet), SAML 2.0, OpenID+OAuth (2.0) und LDAPS (nur innerhalb einer VPN-Verbindung) sind praktikable Optionen.

Für den unwahrscheinlichen Fall, dass sich die DTAG-eigene IAM-Lösung für die Verwaltung von Identitäten in der Cloud nicht integrieren lässt, ist es notwendig die Kontoverwaltung zu automatisieren. Eine solche Automatisierung zur Verwaltung von Identitäten muss das "IAM (Identity Access Management) - Framework" (3.69.) Anforderungen 01-57 und 65 erfüllen.

Die bevorzugte Lösung wäre sitzungsbasiert, da keine persönlichen Daten der Mitarbeiter mit dem Cloud-Anbieter ausgetauscht werden.

Es ist erwähnenswert, dass diese Anforderung auch für Anwendungen gilt, die in der Cloud laufen, aber das ist bereits mit anderen Sicherheitsanforderungen (für Betriebssysteme und Anwendungen) abgedeckt.

ID: 3.22-321/13.0

Req 322	Dedizierte physikalische oder dedizierte virtuelle Verbindungen müssen für die Konnektivität zwischen Applikationen auf der Public Cloud und On-Premise Ressourcen der Deutschen Telekom genutzt werden. Das gleiche gilt für Applikationen, die auf unterschiedlichen Public Clouds gehostet werden. Internetzugriff ist hierbei nur cloud-native, sowie internet-native Anwendungen vorbehalten, sofern diese eine Freigabe durch die Telekom Security erhalten haben.
---------	--

Typischerweise werden die Lokalisationen von Cloud Service Provider und DTAG via Glasfaser verbunden - dedizierte physikalische Verbindung. Dedizierte virtuelle Verbindungen werden durch einen VPN Endpunkt innerhalb der DTAG Lokalisationen mit einem VPN Endpunkt innerhalb der Public Cloud verbunden. Die Verbindung von zwei Public Clouds muss auch via VPN erfolgen.

Eine Applikation auf der Public Cloud, die eine permanente Konnektivität von/zu zu DTAG Ressourcen benötigt, unabhängig davon, ob sie innerhalb von DTAG Lokalisationen oder einer Public Cloud ausgeführt wird, muss hierfür einen VPN benutzen. Nicht jede Applikation braucht eine eigene VPN Verbindung, es kann die gleiche Verbindung genutzt werden, die schon zwischen CSP und DTAG benutzt wird.

Die möglichen Methoden sind die folgenden in Reihenfolge:

1. Dedizierte physikalische Verbindung (verlässlicher falls exklusiv benutzt)
2. Dedizierte virtuelle Verbindung
3. VPN via Internet (weniger verlässlich)
4. (Beschränkt) Einfache Internetverbindung; Benötigt Security Freigabe.

VPN Verbindungen, die ein geteiltes backbone Netzwerk mit anderen Kunden oder via Internet benutzen sind typischerweise weniger verlässlich im Falle von z.B. DDoS Angriffen.

Einfache Internetverbindung ist nur bei Cloud/Internet-native Anwendungen erlaubt. Desweiteren ist es nötig die Kommunikation mit IP ranges einzuschränken und auf ein Minimum zu begrenzen. Diese Methode ist typischerweise für Tests/Trails gedacht und kommt in der DTAG nicht häufig vor.

ID: 3.22-322/13.0

Req 323	Verbindungen zwischen Public Cloud und Ressourcen innerhalb von DTAG Netzwerken müssen durch ein unabhängiges Security Produkt unter DTAG Kontrolle geschützt werden. Ein Dienst in der Cloud darf keinen direkten Internetzugang in Kombination mit Zugriff auf DTAG interne Netzwerke haben (eine solche Funktionalität muss via Virtuelle Maschinen, Container bzw. mehrere Stufen getrennt sein).
---------	---

Verbindungen zwischen Public Clouds und internen Netzwerken der DTAG (z.B. CN-DATG) müssen durch ein Security Produkt geschützt werden. Abhängig vom Anwendungsfall sind zu benutzen:

- stateful firewall (als absolutes Minimum)
- web application firewall
- application-level gateway
- API gateway
- DoS/DDoS Schutz
- ...

Das Security Produkt in diesem Kontext kann aus mehreren physischen oder virtuellen Teilen bestehen, dies als eine Einheit fungieren. Der Schutz des eingesetzten Produkts muss bidirektional funktionieren, also DTAG Netzwerke von Angriffen aus der Cloud schützen, sowie auch Ressourcen innerhalb der Public Cloud vor Schadsoftware und böswilligen Insidern schützen. Es wird vorausgesetzt dass Anwendungen auf der Public Cloud Funktionen wie Security Gruppen und andere von der Cloud bereitgestellte Sicherheitsfunktionen nutzen. Aus diesem Grund müssen die Regeln für die Verbindung von DTAG internen Netzwerken Richtung Public Cloud nicht besonders detailliert sein. Für die andere

Richtung, Public Clouds zu DTAG internen Netzwerken muss die Implementierung vollständig durch ein geeignetes Produkt realisiert werden.

Für HTTP(S) basierte Dienste sollte das Produkt über einen Schutz gegen die OWASP Top 10 Webschwachstellen verfügen - als Minimum.

Die Konfiguration eines solchen Produkts sollte vollautomatisiert möglich sein, sodass es mit der schnellen Provisionierung im Cloud Umfeld schritt halten kann (es sollte dynamische IP-Zuweisung bewältigen können).

Interne Netzwerke der DTAG sind meistens nicht mit modernen Verfahren wie z.B. Mikrosegmentierung aufgebaut und Applikationen in der Cloud müssen sich nach diesem Aufbau richten, sollten sie sich mit solchen Systemen verbinden müssen. Applikationen dürfen nicht gleichzeitig mit dem Internet und DTAG internen Netzwerken verbunden werden, da hier sonst zwei Security Ebenen entfallen. Dies würde es Angreifern erleichtern DTAG interne Netzwerke zu infiltrieren, z.B. durch Hintertüren wissentlich oder unwissentlich in Applikationen versteckt. Für spezielle Anwendungsfälle wie z.B. das Management oder die Überwachung einer Internetapplikation muss eine Lösung im Einzelfall gefunden werden.

ID: 3.22-323/13.0

Req 324	Ressourcen müssen durch virtual private clouds (VPC) getrennt werden. Netzwerkkommunikation von Applikationen/Diensten muss durch den Einsatz von Security Gruppen beschränkt werden. Falls ein CSP erweiterte Methoden zum Schutz von Netzwerken anbietet müssen diese genutzt werden.
---------	---

Virtual private clouds sind ein Konzept zur Separierung (von Netzwerken) von Ressourcen eines Systems (oder einer Gruppe von Systemen) innerhalb der Public Cloud von anderen Ressourcen. Für Ressourcen werden Richtlinien für Konnektivität/Erreichbarkeit definiert. Diese Richtlinien sind über verschiedene Rechenzentren, Verfügbarkeitszonen/regionen der Public Cloud heraus definierbar. Um einfache und effektive Netzwerkabgrenzung zu gewährleisten, müssen VPCs verwendet werden. Wenn VPCs verwendet werden kann dies den Einsatz von VPNs für jedes einzelne Rechenzentrum des CSPs minimieren und entsprechende Routing Tabellen können implementiert werden. Das VPC Konzept erlaubt es Kunden der Cloud eigene IP-Bereiche für Ressourcen zu benutzen.

Security Gruppen sind im Zusammenspiel mit VPCs der Grundstein für das Einschränken der Kommunikationspfade innerhalb der Cloud. Security Gruppen sollten so restriktiv wie möglich sein und nur benötigte Kommunikation erlauben.

Zusätzlich zu Security Gruppen bieten CSPs meistens einen DoS/DDoS Schutz, Rate Limiting und andere Netzwerksicherheitsfeatures an - diese müssen genutzt werden.

Manche CSPs bieten den VPC Dienst nicht wörtlich an, es ist jedoch möglich das gleiche mithilfe von virtuellen Netzwerken zu implementieren.

Motivation: Einhaltung der Trennung zwischen den Applikationen

ID: 3.22-324/13.0

Req 325	Test-, Referenz- und Produktionssystem müssen netzwerktechnisch voneinander getrennt werden. Innerhalb der netzwerktechnischen Einheit muss eine Trennung via Tenants erfolgen.
---------	---

Applikationen müssen (logisch) voneinander getrennt werden. Eine Trennung wird auch für verschiedene netzwerktechnische Einheiten benötigt.

Dieser Ansatz ermöglicht eine obligatorische Aufgabentrennung zwischen den Systemen und verhindert ungewollten Zugriff auf Daten und Systeme.

Zusätzlich zur Separierung von Accounts müssen unterschiedliche Tenants/Projekte nach den gleichen Regeln geschützt werden. Es ist verständlich, dass Produktionssysteme den höchsten Schutzbedarf benötigen, jedoch müssen Test und Entwicklungssysteme gleichwertig geschützt werden.

Motivation: Einhaltung des Need-to-Know-Prinzips.

Der Zugriff zwischen den netzwerktechnischen Einheiten muss verhindert werden.

Beispielsweise dürfen die Zugangsdaten dieser Systeme nicht in Klartext auf GIT Repositories gespeichert werden.

ID: 3.22-325/13.0

Req 326 Applikationen müssen in multi-tier Modellen gebaut werden.

Dieses vorgehen ist eine bereits bewährte Methode und soll an dieser Stelle nur dazu dienen daran zu erinnern, dass dies auch im Cloud Umfeld weiterhin so ist.

Klassische Applikationen benötigen immer noch Firewalls als Separierungselement zwischen Netzwerken, speziell während der Übergangszeit zu Public Clouds.

Aus diesem Grund muss die klassische Netzwerksegmentierung mit den mitteln der Cloud nachgeahmt werden und entsprechende Sicherheitsmechanismen müssen implementiert werden. Auf der Public Cloud wird dies mit hilfe von Sicherheitsgruppen und virtuellen Netzwerken umgesetzt.

Cloud Native Applikationen müssen dem Mikrosegmentierungsansatz folgen um eine trennung zwischen verschiedenen ebenen zu realisieren. Sas resultiert in flacheren Netzwerken, in denen jede Komponente durch entsprechde Sicherheitsrichtlinien und Sicherheitsgruppen der Cloud geschützt wird.

Sicherheitsanforderungen aus anderen Dokumenten sind mit einzubieziehen, z.B. aus den Containeranforderungen :

- Container, die verschiedenen Sicherheitszonen zugeordnet sind (z. B. DMZ-Zone, Anwendungs- / Datenbank-zone) müssen auf verschiedenen Hosts / Host-Pools bereitgestellt werden. Ein Pool bezieht sich auf eine Gruppe von mindestens einer oder mehreren VMs. Im Falle der Verwendung von Container as a Service (CaaS) - Plattformen können diese Pools für mehrere Anwendungen gemeinsam genutzt werden. Für kritische Anwendungen müssen dedizierte Hosts / Host-Pools. Wenn der einzige exponierte Dienst ein Ingress-Controller (z. B. Nginx) ist, der hinsichtlich Sicherheit gehärtet ist, können Sie dies implementieren, indem Sie nur Container-spezifische Maßnahmen verwenden, wie z.B. Pod Security Policies / Network Security Policies.

ID: 3.22-326/13.0

2.21. SAP Cloud Plattform als PaaS

Req 327 Hat eine Applikation keine Verbindung zum Hitnet, muss die Applikation die passenden Schnittstellen und Anbindungen zum CDC zu Verfügung stellen.

Wird die SAP-Cloud-Plattform verwendet, ohne dass Verbindungen in lokalen Rechenzentren der Telekom (z. B. Biere) benutzt werden, so ist die Schnittstelle des Telekom Cyber-Defense-Center so einzubinden, als würde das Security-Framework zur Anwendung kommen.

Motivation: In der Regel muss eine Cloud-Applikation genauso betrieben werden wie ein On-Premise Installation. Es muss sichergestellt sein, dass Informationen wie Zugriffe (insbesondere fehlerhafte Zugriffe) ins globale Monitoring des CDCs laufen.

ID: 3.22-327/13.0

Req 328 Bei der Verwendung der SAP-Cloud-Plattform muss der verwendete Service in einem in Deutschland betriebenen Rechenzentrum der SAP erfolgen (z. B. St.Leon-Rot).

Bei der Verwendung der SAP-Cloud-Plattform muss dafür Sorge getragen werden, dass die verarbeitende Instanz in Europa (bzw. in Deutschland) liegt. Zusätzlich muss sichergestellt sein, dass Backup und andere Services ebenfalls Europa produziert werden. Außerdem müssen die Telekomprozesse befolgt werden (BIA = Business Impact Analyse, TIA = Transfer Impact Assessment sowie ein vorhandenes SOC2 Zertifikat vom Provider für den Service müssen vor-

liegen).

Motivation: Dem Schutzbedarf den Daten, Einhaltung und Sicherstellen von gesetzlichen Vorgaben.

ID: 3.22-328/13.0

Req 329 Jede Schnittstelle, die über das Security-Framework läuft, muss dokumentiert werden.

Mindestens die folgenden Punkte müssen dokumentiert sein:

- Quelle /Senke/
- Protokoll /Art der Verschlüsselung/
- Framework in der SCP/
- Schutzbedarf der Daten/
- Angelegte Benutzer in der SCP und Art der Berechtigung.

Motivation: Es muss jederzeit nachvollziehbar sein, welches System und in welcher Art, über welche Benutzer, mit der Cloud kommuniziert.

ID: 3.22-329/13.0

Req 330 Bei der Verwendung der SAP-Cloud-Plattform muss mindestens sichergestellt sein:

- Datenbank Verschlüsselung,
- Trennung der unterschiedlichen Umgebungen,
- verschlüsselte Kommunikation zwischen Applikationen zur Cloud
- Cloud-Applikationen untereinander.

Bei der Verwendung der SAP-Cloud-Plattform muss sichergestellt sein:

- dass die Datenbank verschlüsselt ist
- dass schützenswerte Datenbankfelder bei der Übertragung verschlüsselt (geschützt) sind
- unterschiedliche Umgebungen (Test, Abnahme, Produktion, ...) voneinander getrennt sind
- die Kommunikation zwischen Applikationen und Cloud über sichere Protokolle (z.B https) erfolgt.
- die Trennung der Umgebung darf nicht über Mandanten erfolgen.

Motivation: Daten in der Cloud müssen technisch sicher übertragen und verarbeitet werden. Dem Schutzbedarf der Daten entsprechend, ist eine Verschlüsselung notwendig. Ziel ist es, sich so weit wie möglich von dem Provider und dem Betriebspersonal zu schützen.

ID: 3.22-330/13.0

2.21.1. Anforderungen Cloud Connector (nur wenn ein SaaS in der SAP-Cloud benutzt wird)

Req 331 Die Übereinstimmung der Rollen und Berechtigungen für Cloud, On-Premise und Mobile Apps muss gewährleistet sein.

Bei der Benutzung von verschiedenen Komponenten (On-Premise, Cloud, App via Cloud, ...) müssen die Benutzerberechtigungen auf dem allen Systemen mit der ursprünglichen Berechtigung zusammen passen. Bei Zugriffen über eine App, muss sichergestellt sein, dass der Benutzer die gleichen Berechtigungen in der App wie auf dem lokalen System besitzt.

Motivation: Es muss dafür Sorge getragen werden, dass Benutzer nicht mehr Rechte zugeordnet werden als die höch-

sten Rechte, die er für das entsprechende System besitzt.

ID: 3.22-331/13.0

Req 332 Der Cloud-Connector darf nicht durch einen Proxy oder Reverse-Proxy ersetzt werden.

Der Cloud-Connector darf nicht durch einen Reverse-Proxy ersetzt werden. Der Cloud-Connector stellt neben dem HTTP-Protocol noch weitere Funktionalitäten zur Verfügung, die es ermöglichen auch andere SAP-typische Protokolle zwischen den Welten zu benutzen.

Motivation: Der Cloud-Connector kann einen VPN-Tunnel zur SCP (SAP Cloud Plattform) aufbauen. Falls er das nicht macht, muss sichergestellt sein, dass das System alle Verbindungen (HTTP, RFC, TCP, UDP) schützt, um einen unberechtigten Zugriff zu verhindern.

ID: 3.22-332/13.0

Req 333 Der Cloud-Connector muss so betrieben werden, dass immer eine verschlüsselte Verbindung aufgebaut wird.

Der Cloud-Connector muss so betrieben werden, dass immer eine sichere Verbindung aufgebaut wird. Der Verbindungsaufbau muss immer aus dem Hitnet kommen und die Verbindung muss jederzeit durch das Betriebsteam der Landing Zone abbrechbar sein. Falls ein VPN-Tunnel genutzt wird, muss durch die Art des Aufbaus verhindert werden, dass auf das System native und ungeschützt Zugriffe aus der SCP-Plattform möglich sind.

Motivation: Schutz der Daten beim Benutzen von Cloud-Diensten.

ID: 3.22-333/13.0

Req 334 Cloud Anwendungen dürfen keine selbst signierte Zertifikate verwenden.

Die SCP-Cloud akzeptiert die Telekom-CA als „trusted Provider“. Bei jeglicher Art der Kommunikation muss für jede Verbindung jederzeit nachvollziehbar sein, ob das genutzte Zertifikat valide und gültig ist. Die Nutzung von selbst signierten Zertifikaten ist für keine Art der Kommunikation akzeptabel.

Motivation: Für jeden Client muss prüfbar ein, ob es sich um vertrauenswürdigen System handelt.

Umsetzungsbeispiel: Benutzung der Telekom-CA zum Erstellen von Zertifikaten.

ID: 3.22-334/13.0

Req 335 Der SAP Cloud Connector muss verwendet werden. Er ist für das Protokollieren und Weiterleiten aller relevanten Daten an das Cyber-Defense-Center zuständig.

Der Cloud-Connector ist für das Protokollieren und Weiterleiten aller relevanten Log-Daten aus den Cloud Anwendungen verantwortlich. Die Daten müssen den Betrieb der Telekom sowie Cyber-Defense-Center zu Verfügung gestellt werden.

Motivation: Unabhängig des Standortes einer Applikation müssen die Log-Daten für den Betrieb und Cyber-Defense-Center zur Verfügung gestellt werden.

ID: 3.22-335/13.0

2.22. Microsoft Azure Cloud via DTIT Landing Zone und Produkt "Enterprise"

Die Sicherheitsvorgaben beziehen sich nur auf das Produkt "Enterprise" der DTIT Landing Zone. In der Version Enterprise ist die DTIT Landing Zone als trennendes und koppelndes Netz zwischen dem CN-DTAG (Hitnet) und der Azure

Cloud mit spezifizierten Schutzmechanismen aufgebaut.

Req 336	SAP-Systeme, die in der Azure Cloud betrieben werden, dürfen nur über das "Hitnet" (CN-DTAG) erreichbar sein.
---------	---

Ein Zugriff auf die SAP-Systeme, die in der Azure Cloud betrieben werden (im Produkt Enterprise), darf nur geregelt über DTIT Landing Zone mit deren Schutzmaßnahmen (z. B. Überwachung, Schutz, sichere Verbindungen, Authentifizierung) über das Hitnet erfolgen,
z. B. TPAM Tool: <https://yam-united.telekom.com/pages/t-pam/apps/content/home>.

Clients dürfen nicht aus dem Internet auf die Anwendung in der Azure Cloud zugreifen.

Motivation: Das SAP-System muss gegen unberechtigten Zugriff geschützt werden.

ID: 3.22-336/13.0

Req 337	Anwendungen aus den unterschiedlichen Azure Produkten "Discovery", "Voyager" und "Enterprise" dürfen nicht miteinander kommunizieren.
---------	---

Es wird in die Azure Produkte "Enterprise", "Voyager" und "Discovery" der DTIT Landing Zone unterschieden. Eine Kommunikation zwischen den verschiedenen Azure Produkten ist nicht erlaubt. In den verschiedenen Azure Produkten besteht ein unterschiedlicher Schutz der Applikationen in Bezug auf Patchen, Monitoring, Schnittstellenzugriffe, etc

Motivation: Ein unberechtigter Zugriff auf die Anwendung muss verhindert werden.

ID: 3.22-337/13.0

Req 338	SAP-Systeme, die zu einem Transportdomäne gehören, müssen in einer Subscription stehen und die Trennung der DEV-, QAS und PRD-Umgebungen erfolgt über Network Security Groups (NSGs).
---------	---

In einem Projekt /SAP-Systemlandschaft stehen alle Systeme (Entwicklung/Abnahme/Produktion/...), die zu einer Transportdomäne gehören, in einer Subscription. Die Trennung der Umgebungen erfolgt über NSGs.

Dies ist eine Ausnahmeregelung für die Azure Cloud gegenüber den NON-SAP-Systemen, da es ansonsten nicht möglich ist, der Arbeitsweise von SAP via Transportverzeichnis nachzukommen (keine Shares über die Grenzen von Subscriptions technisch sinnvoll möglich).

Motivation: Unberechtigte Zugriffe über Systemgrenzen hinweg müssen verhindert werden. Durch die Trennung über NSGs wird geregelt, welche Zugriffe kontrolliert möglich sind.

ID: 3.22-338/13.0

Req 339	Für eine SAP-Systemlandschaft (SAP-Projekt) in der Azure Cloud dürfen nur bestimmte Azure Services genutzt werden.
---------	--

Azure Services dürfen nicht innerhalb einer SAP-Systemlandschaft (SAP-Projekt) der Azure genutzt werden, da diese teilweise nicht über die NSG zugriffsbeschränkt werden können.

Ausnahmen für nutzbare Azure Services:

- Virtual Machine
- Storage Account
- Loadbalancer
- Key Vault
- Azure Automation Account

Motivation: In den Azure Services gibt es viele Services die Verbindungen ins Internet benötigen. Im Gegensatz zu der Subscription können diese nicht in der NSG konfiguriert werden, so dass der Schutz der Applikation gewährleistet ist.

Zudem nutzen diese Services Authentisierungsdaten, die zentral von Azure gemanagt werden.

ID: 3.22-339/13.0

Req 340	Für SAP-Systeme in der Azure Cloud muss ein SAProuter für die Verbindung zum CNDTAG aufgebaut werden.
---------	---

Für SAP-Systeme in der Azure Cloud muss ein SAProuter für die Verbindung zum CNDTAG aufgebaut werden. Als Gegenstück steht ein zentrales System im Netz der CNDTAG bereit. Der SAProuter darf nur die SAP Protokolle DIAG, RFC und SNC routen.

Motivation: Mit dem SAProuter wird die SAP-spezifische Kommunikation in das Hitnet sichergestellt.

ID: 3.22-340/13.0

Req 341	Für SAP-Systeme in der Azure Cloud muss ein SAP Web Dispatcher für die Verbindung zum CNDTAG aufgebaut werden.
---------	--

Für SAP-Systeme in der Azure Cloud muss ein SAP Web Dispatcher (oder etwas Vergleichbares) für die Verbindung zum CNDTAG aufgebaut werden. Als Gegenstück steht ein zentrales System im Netz der CNDTAG bereit. Der SAP Web Dispatcher darf nur die SAP Protokolle http und https routen.

Motivation: Mit dem SAP Web Dispatcher wird die Web-Kommunikation in das Hitnet sichergestellt.

ID: 3.22-341/13.0

Req 342	Applikationszugriffe über die Protokolle DIAG, RFC und SNC müssen über den zentralen, vorgelagerten Proxy "SAPaaS Router Service" führen.
---------	---

Applikationszugriffe aus dem CN-DTAG (Hitnet) auf SAP-Systeme, die in der Azure Cloud im Produkt „Enterprise“ stehen, müssen Ihre Kommunikation über die zentralen, vorgelagerten SAProuter des "SAPaaS Router Service" führen.

Entsprechend dem zu übertragende Protokolle (es muss ein für SAP passendes System verwendet werden):
RFC und SNC => SAProuter

Motivation: Insbesondere der SAProuter ist eine wichtige Komponente für das SAP-spezifische Protokoll RFC und SNC. In vielen Fällen wird er auch als netztrennendes Element benutzt.

Über die zentralen, vorgelagerten SAProuter des "SAPaaS Router Service" wird eine sichere Verbindung von der On-Premise- zur Azure Cloud-Welt mit dem Produkt "Enterprise" sichergestellt.

ID: 3.22-342/13.0

Req 343	Applikationszugriffe über das http-Protokoll müssen über den zentralen, vorgelagerten Proxy "SAPaaS Proxy Service" führen.
---------	--

Applikationszugriffe aus dem CN-DTAG (Hitnet) auf SAP-Systeme, die in der Azure Cloud im Produkt „Enterprise“ stehen, müssen Ihre Kommunikation über die zentralen, vorgelagerten SAP Web Dispatcher der "SAPaaS Proxy Service" führen.

Entsprechend dem zu übertragende Protokolle (es muss ein für SAP passendes System verwendet werden):
SAP-http, HTTPS, API => SAP Web Dispatcher

Motivation: Insbesondere der SAP Web Dispatcher ist eine wichtige Komponente für das Protokoll http und https. In vielen Fällen wird er auch als netztrennendes Element benutzt.

Über die zentralen, vorgelagerten SAP Web Dispatcher des "SAPaaS Proxy Service" wird eine sichere Verbindung von der OnPremise- zur Azure Cloud-Welt mit dem Produkt "Enterprise" sichergestellt.

ID: 3.22-343/13.0

Req 344	Applikationstraffic zwischen Subscriptions muss immer über das netztrennende zentrale Element ("SAPaaS Router Service" oder "SAPaaS Proxy Service") geführt werden.
---------	---

Applikationstraffic zwischen Subscriptions muss immer über das netztrennende zentrale Elemente in der Cloud ("SAPaaS Router Service" oder "SAPaaS Proxy Service") geführt werden. Als weitere netztrennende Elemente in der Azure Cloud gelten Firewalls und Network Security Groups.

Motivation: Um eine Netztrennung zwischen den unterschiedlichen Umgebungen umzusetzen, müssen alle Verbindungen über passende Proxys bzw. Router laufen.

Des weiteren stellen wir über die Proxy und Router sicher, dass die Kommunikation mindestens auf Applikationsbasis verschlüsselt ist.

Es wird eine Entkopplung und ein Schutz der Umgebungen umgesetzt.

ID: 3.22-344/13.0

Req 345	Alle Applikationsverbindungen müssen auf Applikationsebene verschlüsselt sein.
---------	--

Eine Ende-zu-Ende-Verschlüsselung auf Applikationsebene muss umgesetzt sein (auf Applikationsebene via SNC oder HTTPS).

Motivation: Schutz der Anmeldeinformation und Daten auf Applikations-Ebene.

ID: 3.22-345/13.0

Req 346	Alle Verbindung innerhalb in einer Network Security Group (NSG) müssen verschlüsselt sein.
---------	--

Um die Daten gegenüber dritten sicher zu stellen, muss auch innerhalb einer NSG zwischen den Applikationen verschlüsselt kommuniziert werden.

Kann aus technischen Gründen innerhalb der NSG nicht verschlüsselt kommuniziert werden, so ist dies als Maßnahme zu dokumentieren.

Motivation: Sichere Kommunikation im Sinne einer Ende-zu-Ende Sicht. Kleinstmögliches Risiko gegenüber dritten (z.B Cloud-Betreiber).

ID: 3.22-346/13.0

Req 347	Ein Backup der Daten außerhalb der Cloud muss vorhanden sein.
---------	---

Falls die Daten nur noch in der Cloud gespeichert werden, muss sichergestellt sein, dass ein akzeptabler Datensatz (Datenmenge, Aktualität) auch außerhalb der Cloud gespeichert wird.

Motivation: Es muss abgestimmt sein, falls die Cloud nicht mehr genutzt werden kann, dass es eine Möglichkeit gibt, unabhängig des Cloudzugriffs ein Backup mit Daten zu beschaffen. Welches Zeitfenster genutzt wird, bzw. wie viele Daten verloren gehen dürfen, wenn die Cloud über lange Zeit nicht mehr nutzbar ist, muss im Einzelfall bestimmt werden.

ID: 3.22-347/13.0

Req 348	Die Jobsteuerung über Automic Automation (UC4) muss für Anwendungen in der Cloud über einen sicheren Zugriff realisiert werden.
---------	---

Bei der Verwendung von Automic Automation (UC4) muss ein separater Tenant/Mandant für die Applikation in der Cloud genutzt werden, sodass eine Trennung zwischen der OnPrem- und der Cloud-Welt in Automic Automation (UC4) hergestellt ist.

Des weiteren dürfen nur noch verschlüsselte Verbindungen genutzt werden.

Motivation: Falls Automic Automation (UC4) zum Einsatz kommt, so muss die Kommunikation auf sicheren Protokollen und eine sichere Authentifizierung basieren.

Ein unberechtigter Zugriff durch höher berechnigte Automic Automation (UC4)-Benutzer muss über verschlüsselte Protokolle und eine sichere Authentisierung verhindert werden.

ID: 3.22-348/13.0

2.22.1. Zentrale Management Subscription "SAPaaS central Management Service"

Req 349	In der Azure Cloud im Produkt "Enterprise" der DTIT Landing Zone gibt es nur eine zentrale Subscription für den administrativen Zugriff auf die SAP Anwendungen in den Subscriptions, den "SAPaaS central Management Service".
---------	--

Aus dieser zentralen Management Subscription "SAPaaS central Management Service" werden alle SAP Azure Anwendungen administriert.
Es ist eine 2FA erforderlich.

Motivation: Unkontrollierter privilegierter Zugriff muss verhindert werden.

ID: 3.22-349/13.0

Req 350	Für den Zugriff von administrativen Benutzern muss Cyberark / T-PAM genutzt werden.
---------	---

Für den Zugriff von administrativen Benutzern (auf Kundenprojekte/Systeme) via "SAPaaS central Management Service" muss Cyberark / T-PAM Services (zentral bereitgestellt) genutzt werden sowie die Key Rotation aktiviert sein.

Motivation: Durch den Einsatz von Cyberark/T-PAM wird ist kein personalisierter Admin-Account im SAP erforderlich.

ID: 3.22-350/13.0

Req 351	In der zentralen Management Subscription "SAPaaS central Management Service" dürfen keine Anwenderzugriffe erfolgen.
---------	--

Die zentrale Management Subscription "SAPaaS central Management Service" ist ausschließlich, für betriebliche Belange zu nutzen.

Motivation: Sicherstellung von einem sicheren und einheitlichen Applikationsadministrationszugang für SAP-Systemlandschaften.

ID: 3.22-351/13.0

Req 352	In der zentralen Management Subscription "SAPaaS central Management Service" dürfen nur bestimmte Azure Services genutzt werden.
---------	--

In der zentralen Management Subscription dürfen nur bestimmte Azure Services genutzt werden, da diese teilweise nicht über die NSGs zugriffsbeschränkt werden können.

Ausnahmen für nutzbare Azure Services:

Virtual Machine
Storage Account
Loadbalancer
Key Vault
Azure Automation Account
Azure Firewall

Motivation: Es gibt Azure Services, die Verbindungen ins Internet benötigen.

Zudem nutzen diese Services Authentisierungsdaten, die zentral von Azure gemanagt werden.

ID: 3.22-352/13.0

2.23. Google Cloud via DTIT Landing Zone mit Produkt "SAP on GCP"

Die Sicherheitsvorgaben beziehen sich nur auf das Produkt "SAP on GCP" der DTIT Landing Zone. Mit dem Produkt SAP on GCP ist die DTIT Landing Zone als trennendes und koppelndes Netz zwischen dem CN-DTAG (Hitnet) und der Google Cloud mit spezifizierten Schutzmechanismen aufgebaut.

Req 353	SAP-Systeme, die in der Google Cloud betrieben werden, dürfen nur über das "Hitnet" (CN-DTAG) erreichbar sein.
---------	--

Ein Zugriff auf die SAP-Systeme, die in der Google Cloud betrieben werden (im Produkt SAP on GCP), darf nur geregelt über DTIT Landing Zone mit deren Schutzmaßnahmen (z. B. Überwachung, Schutz, sichere Verbindungen, Authentifizierung) über das Hitnet erfolgen,
z. B. TPAM Tool: <https://yam-united.telekom.com/pages/t-pam/apps/content/home>.

Clients dürfen nicht aus dem Internet auf die Anwendung in der Google Cloud zugreifen.

Motivation: Das SAP-System muss gegen unberechtigten Zugriff geschützt werden.

ID: 3.22-353/13.0

Req 354	Anwendungen aus den unterschiedlichen Google Cloud Foldern (managed und unmanaged) dürfen nicht miteinander kommunizieren.
---------	--

In der Google Cloud wird in managed und unmanaged Foldern unterschieden. Eine Kommunikation zwischen den verschiedenen Google Cloud Foldern (managed und unmanaged) ist nicht erlaubt.

Motivation: Ein unberechtigter Zugriff auf die Anwendung im managed Folder muss verhindert werden. Projekte im managed Folder haben eine Verbindung in das CNDTAG und damit verbunden werden Compliance Regeln umgesetzt.

ID: 3.22-354/13.0

Req 355 SAP-Anwendungen müssen über COSMOS in der Google Cloud via Produkt "SAP on GCP" bereitgestellt werden.

SAP-Anwendungen müssen über COSMOS in der Google Cloud via Produkt "SAP on GCP" bereitgestellt werden. COSMOS (Cloud Orchestration SAP Managed Operation Service) ist ein Orchestrierungstool für SAP Managed Services für SAP-Applikationslandschaften in der Google Cloud via Produkt "SAP on GCP".

Das Transportverzeichnis wird via COSMOS realisiert.

Eine Bereitstellung von SAP-Systemen via Cloud Shepherd ist nicht erlaubt und nicht möglich.

Motivation: Automatisierte Bereitstellung und Umsetzung von Sicherheitsvorgaben erhöhen die Effizienz und das Sicherheitsniveau.

Umsetzungsbeispiel: <https://cosmos.sap.t-systems.com>

ID: 3.22-355/13.0

Req 356 In der SAP on GCP dürfen keine zusätzlichen preapproved Services genutzt werden.

In der SAP on GCP dürfen keine zusätzlichen preapproved Services genutzt werden.

Motivation: Ein sicherer Betrieb der Applikation muss gewährleistet sein.

ID: 3.22-356/13.0

Req 357 Für SAP-Systeme in der Google Cloud wird eine zentrale Verbindung "SAPaaS Router Service" und "SAPaaS Proxy Service" zwischen CN-TAG und der Google Cloud in Verbindung mit der DTIT Landing Zone zur Verfügung gestellt.

Die zentrale Verbindung ("SAPaaS Router Service" und "SAPaaS Proxy Service") besteht aus SAProutern und SAP Web Dispatchern für die Kopplung. Diese wird zusammen mit der DTIT Landing Zone zur Verfügung gestellt.

Motivation: Mit dem "SAPaaS Router Service" and "SAPaaS Proxy Service" wird die SAP-spezifische Kommunikation sicher gestellt. Die "SAPaaS Router Service" und "SAPaaS Proxy Service" wird über ein Team in der DT-IT ähnlich wie die Landing Zone bereitgestellt.

*Bitte nutzen Sie für Anfragen unser Support-Eingangstor:
<https://jira.telekom.de/servicedesk/customer/portal/1286>*

ID: 3.22-357/13.0

Req 358 Applikationszugriffe über die Protokolle DIAG, RFC und SNC müssen über den zentralen, vorgelagerten Proxy "SAPaaS Router Service" führen.

Applikationszugriffe aus dem CN-DTAG (Hitnet) auf SAP-Systeme, die in der Google Cloud im Folder „SAP on GCP“ stehen, müssen ihre Kommunikation über die zentralen, vorgelagerten SAProuter des "SAPaaS Router Service" führen.

Entsprechend dem zu übertragenden Protokollen (es muss ein für SAP passendes System verwendet werden):
RFC und SNC => SAProuter

Motivation: Insbesondere der SAProuter ist eine wichtige Komponente für das SAP-spezifische Protokoll RFC und SNC. In vielen Fällen wird er auch als netztrennendes Element benutzt.

Über die zentralen, vorgelagerten SAProuter des "SAPaaS Router Service" wird eine sichere Verbindung von der On-Premise- zur Google Cloud-Welt mit dem Produkt "SAP on GCP" sichergestellt.

ID: 3.22-358/13.0

Req 359	Applikationszugriffe in die Google Cloud, die das http-Protokoll verwenden, müssen über einen vorgelagerten Proxy (SAP Web Dispatcher) führen.
---------	--

Applikationszugriffe aus dem CN-DTAG (Hitnet) auf SAP-Systeme, die in der Google Cloud im Folder „SAP on GCP“ stehen, müssen Ihre Kommunikation über die vorgelagerten SAP Web Dispatcher führen.

Entsprechend dem zu übertragende Protokolle (es muss ein für SAP passendes System verwendet werden):
SAP-http, HTTPS, API => SAP Web Dispatcher

Motivation: Insbesondere der SAP Web Dispatcher ist eine wichtige Komponente für das Protokoll http und https. In vielen Fällen wird er auch als netztrennendes Element benutzt.

Über die vorgelagerten SAP Web Dispatcher wird eine sichere Verbindung von der OnPremise- zur Google Cloud-Welt mit dem Produkt "SAP on GCP" sichergestellt.

ID: 3.22-359/13.0

Req 360	Ausgehende Verbindungen von der Google Cloud in das Internet müssen auf Projektebene eingerichtet werden.
---------	---

Ausgehende Verbindungen von der Google Cloud in das Internet müssen auf Projektebene eingerichtet werden. Dafür muss der managed Service "Secure Web Proxy" in der GCP genutzt.

Eine einheitliche Implementierung soll zeitnah via COSMOS bereitgestellt werden.

Motivation: Gewährleistung einer sicheren Kommunikation in das Internet.

ID: 3.22-360/13.0

Req 361	Applikationstraffics zwischen Applicationslandschaften in der Google Cloud muss immer über ein zentrales netztrennendes Element ("SAPaaS Router Service", "SAPaaS Proxy Service") geführt werden.
---------	---

Applikationstraffics zwischen Applicationslandschaften (im Umfeld der managed Folder) in der Google Cloud muss immer über ein zentrales netztrennendes Element ("SAPaaS Router Service", "SAPaaS Proxy Service") geführt werden. Als weitere netztrennende Elemente in der Google Cloud gelten gemanaged Firewalls und Security Groups.

Motivation: Um eine Netztrennung zwischen den unterschiedlichen Umgebungen umzusetzen, müssen alle Verbindungen über passende Proxys bzw. Router laufen.

Des weiteren stellen wir über die Proxy und Router sicher, dass die Kommunikation mindestens auf Applikationsbasis verschlüsselt ist.

Es wird eine Entkopplung und ein Schutz der Umgebungen umgesetzt.

ID: 3.22-361/13.0

Req 362	Alle Applikationsverbindungen müssen auf Applikationsebene verschlüsselt sein.
---------	--

Eine Ende-zu-Ende-Verschlüsselung auf Applikationsebene muss umgesetzt sein (auf Applikationsebene via SNC oder HTTPS).

ID: 3.22-362/13.0

Req 363	Ein Backup der Daten muss Geo-Redundant vorhanden sein.
---------	---

Falls die Daten nur noch in der Cloud gespeichert werden, muss sichergestellt sein, dass ein akzeptabler Datensatz (Datenmenge, Aktualität) Geo-Redundant gespeichert wird.

Motivation: Ein Geo-redundant vorliegendes Backup sichert eine hohe Verfügbarkeit im Schadenfall eines Rechenzentrums ab.

ID: 3.22-363/13.0

Req 364	Die Jobsteuerung über Automic Automation (UC4) muss für Anwendungen in der Cloud über einen sicheren Zugriff realisiert werden.
---------	---

Bei der Verwendung von Automic Automation (UC4) muss ein separater Tenant/Mandant für die Applikation in der Cloud genutzt werden, sodass eine Trennung zwischen der OnPrem- und der Cloud-Welt in Automic Automation (UC4) hergestellt ist.

Des Weiteren dürfen nur noch verschlüsselte Verbindungen genutzt werden.

Motivation: Falls Automic Automation (UC4) zum Einsatz kommt, so muss die Kommunikation auf sicheren Protokollen und einer sicheren Authentifizierung basieren.

Ein unberechtigter Zugriff durch höher berechnete Automic Automation (UC4)-Benutzer muss über verschlüsselte Protokolle und eine sichere Authentifizierung verhindert werden.

ID: 3.22-364/13.0

2.23.1. Zentrales Management Projekt "SAPaaS central Management Service"

Req 365	Der administrative Zugriff auf die SAP Applikationslandschaften in der Google Cloud im Produkt "SAP on GCP" der DTIT Landing Zone muss über den "SAPaaS central Management Service" erfolgen.
---------	---

Der administrative Zugriff auf die SAP Applikationslandschaften in der Google Cloud im Produkt "SAP on GCP" der DTIT Landing Zone muss über den "SAPaaS central Management Service" erfolgen. Aus diesem zentralen Management Projekt "SAPaaS central Management Service" werden alle SAP Anwendungen der Google Cloud administriert.

Hinweis: Es ist eine 2FA erforderlich.

Motivation: Unkontrollierter privilegierter Zugriff muss verhindert werden.

ID: 3.22-365/13.0

Req 366	In der zentralen Management Subscription "SAPaaS central Management Service" dürfen keine Anwenderzugriffe erfolgen.
---------	--

Die zentrale Management Subscription "SAPaaS central Management Service" ist ausschließlich, für betriebliche Belange zu nutzen.

Motivation: Sicherstellung von einem sicheren und einheitlichen Applikationsadministrationszugang für SAP-Systemlandschaften.

2.24. FCI Cloud via Landing Zone mit Produkt "SAP on FCI"

Die Sicherheitsvorgaben beziehen sich nur auf das Produkt „SAP on FCI“. Mit dem Produkt „SAP on FCI“ ist die DTIT Landing Zone als trennendes und koppelndes Netz zwischen dem CN-DTAG (Hitnet) und der FCI

Req 367	SAP-Systeme, die in der FCI-Cloud betrieben werden, dürfen nur über das "Hitnet" (CN-DTAG) erreichbar sein.
---------	---

Ein Zugriff auf die SAP-Systeme, die in der FCI Cloud betrieben werden (im Produkt „SAP on FCI“), darf nur geregelt über DTIT Landing Zone mit deren Schutzmaßnahmen (z. B. Überwachung, Schutz, sichere Verbindungen, Authentifizierung) über das Hitnet erfolgen.

Clients dürfen nicht aus dem Internet auf die Anwendung in der FCI Cloud zugreifen.

Motivation: Das SAP-System muss gegen unberechtigten Zugriff geschützt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Störung der Verfügbarkeit

ID: 3.22-367/13.0

Req 368	SAP-Anwendungen müssen über COSMOS in der FCI-Cloud via Produkt "SAP on FCI" bereitgestellt werden.
---------	---

SAP-Anwendungen müssen über COSMOS in der FCI-Cloud via Produkt "SAP on FCI" bereitgestellt werden. (Das Transportverzeichnis wird ebenfalls via COSMOS realisiert).

COSMOS ist der Baukasten zum automatisieren erstellen von System und Applikationslandschaften.
<https://cosmos.sap.t-systems.com>

Motivation: Einfacher Bestellweg, schneller Installation, harmonisierter Betrieb.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.22-368/13.0

Req 369	Für SAP-Systeme in der FCI Cloud wird eine zentrale Verbindung "SAPaaS Router Service" und "SAPaaS Proxy Service" zwischen CN-TAG und der FCI Cloud in Verbindung mit der DTIT Landing Zone zur Verfügung gestellt.
---------	---

Für SAP-Systeme in der FCI Cloud wird eine zentrale Verbindung "SAPaaS Router Service" und "SAPaaS Proxy Service" zwischen CN-TAG und der FCI Cloud in Verbindung mit der DTIT Landing Zone zur Verfügung gestellt.

Die zentralen Verbindungen "SAPaaS Router Service" und "SAPaaS Proxy Service" bestehen aus SAProutern, SAP Web Dispatchern (Webproxy) und Komponenten, die für das SAP-System und dessen Kopplung an unterschiedliche Netze nötig sind. Diese werden ergänzend zur DTIT Landing Zone bereitgestellt.

Motivation: Mit dem "SAPaaS Router Service" and "SAPaaS Proxy Service" wird die SAP-spezifische Kommunikation sicher gestellt. Die "SAPaaS Router Service" und "SAPaaS Proxy Service" wird über ein Team in der DT-IT, ähnlich der Landing Zone, bereitgestellt.

*Bitte nutzen Sie für Anfragen unser Support-Eingangstor:
<https://jira.telekom.de/servicedesk/customer/portal/1286>*

Req 370 Applikationstraffic zwischen Applicationslandschaften in der FCI Cloud muss immer über ein zentrales netztrennendes Element ("SAPaaS Router Service", "SAPaaS Proxy Service") geführt werden.

Applikationstraffic zwischen Applicationslandschaften in der FCI Cloud (CWD1 - CWDn) muss immer über ein zentrales netztrennendes Element ("SAPaaS Router Service", "SAPaaS Proxy Service") geführt werden.

Motivation: Umsetzung des unterschiedlichen Schutzes der Anwendungen in Bezug auf Patching, Überwachung, Schnittstellenzugriff

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten

Req 371 Applikationszugriffe über die Protokolle DIAG, RFC und SNC müssen über den zentralen, vorgelagerten Proxy "SAPaaS Router Service" führen.

Applikationszugriffe aus dem CN-DTAG (Hitnet) auf SAP-Systeme, die in der FCI Cloud im Folder „SAP on FCI“ stehen, müssen Ihre Kommunikation über die zentralen, vorgelagerten SAProuter des "SAPaaS Router Service" führen.

Entsprechend den zu übertragenden Protokollen (es muss ein für SAP passendes System verwendet werden):
RFC und SNC => SAProuter

Motivation: Insbesondere der SAProuter ist eine wichtige Komponente für das SAP-spezifische Protokoll RFC und SNC. In vielen Fällen wird er auch als netztrennendes Element benutzt.

Über die zentralen, vorgelagerten SAProuter des "SAPaaS Router Service" wird eine sichere Verbindung von der On-Premise- zur FCI Cloud mit dem Produkt "SAP on FCI" hergestellt.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten

Req 372 Applikationszugriffe in die FCI Cloud, die das http-Protokoll verwenden, müssen über einen vorgelagerten Proxy (SAP Web Dispatcher) führen.

Applikationszugriffe aus dem CN-DTAG (Hitnet) auf SAP-Systeme, die in der FCI-Cloud im Bereich „SAP on FCI“ stehen, müssen Ihre Kommunikation über die vorgelagerten SAP Web Dispatcher führen.

Entsprechend den zu übertragenden Protokollen (es muss ein für SAP passendes System verwendet werden):
SAP-http, HTTPS, API => SAP Web Dispatcher

Motivation: Insbesondere der SAP Web Dispatcher ist eine wichtige Komponente für das Protokoll http und https. In vielen Fällen wird er auch als netztrennendes Element benutzt.

Über die vorgelagerten SAP Web Dispatcher wird eine sichere Verbindung von der OnPremise- zur FCI Cloud mit dem Produkt "SAP on FCI" sichergestellt.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten

Req 373	Für ausgehende Verbindungen aus der FCI Cloud, die in das Internet erfolgen, muss ein Proxy im HitNet/CNDTAG verwendet werden.
---------	--

Für ausgehende Verbindungen aus der FCI Cloud, die in das Internet erfolgen, muss ein Proxy im HitNet/CNDTAG verwendet werden.

Motivation: Ein zentraler Proxy-Server sichert die Plattform, da nur ein zentraler Punkt ins Internet kommunizieren darf. Zudem bietet dies die Möglichkeit eine einfachere Implementation von Viren-Scanner oder Cache-Server, Filterlisten usw.

Req 374	Alle Applikationsverbindungen müssen auf Applikationsebene verschlüsselt sein.
---------	--

Alle Applikationsverbindungen müssen auf Applikationsebene verschlüsselt sein.

Eine Ende-zu-Ende-Verschlüsselung auf Applikationsebene muss umgesetzt sein (auf Applikationsebene via SNC oder HTTPS).

Motivation: Schutz der Anmeldeinformation und Daten auf Applikations-Ebene.

Req 375	Ein Backup der Daten muss redundant vorhanden sein.
---------	---

Es muss sichergestellt sein, dass ein akzeptabler Datensatz (Datenmenge, Aktualität) redundant in einer anderen Verfügbarkeitszone gespeichert wird.

Motivation: Sicherstellung eines Backups zur Verhinderung eines Datenverlustes.

Umsetzungsbeispiel: Zum Beispiel: Rechenzentrum Magdeburg / Rechenzentrum Biere

Req 376	Die Jobsteuerung über Automic Automation (UC4) muss für Anwendungen in der FCI Cloud über einen sicheren Zugriff realisiert werden.
---------	---

Die Jobsteuerung über Automic Automation (UC4) muss für Anwendungen in der FCI Cloud über einen sicheren Zugriff realisiert werden.

Bei der Verwendung von Automic Automation (UC4) muss ein separater Tenant/Mandant für die Applikation in der FCI Cloud genutzt werden, sodass eine Trennung zwischen Anwendungslandschaften und der Automic Automation (UC4) hergestellt ist.

Motivation: Sicherer Zugriff auf Systeme und Trennung der unterschiedlichen Umgebungen (OnPrem, FCI, Google, Azure, Non-SAP).

Req 377	Beim Einsatz von Automation (UC4) muss die Kommunikation auf sicheren Protokollen und eine sichere Authentifizierungen basieren.
---------	--

Beim Einsatz von Automic Automation (UC4) muss die Kommunikation auf sicheren Protokollen und eine sichere Authentifizierungen basieren.

Motivation: UC4 Benutzer haben hohe Berechtigungen und müssen geschützt werden.

Umsetzungsbeispiel: Nutzung von Zertifikaten bei den Agenten.

ID: 3.22-377/13.0

2.24.1. Zentrale Management Services (administrativer Zugriff)

Req 378	Der administrative Zugriff auf die SAP Applikationslandschaften in der FCI Cloud im Produkt "SAP on FCI" der DTIT Landing Zone muss über einen zentralen Management Service erfolgen.
---------	---

Der administrative Zugriff auf die SAP Applikationslandschaften in der FCI Cloud im Produkt "SAP on FCI" der DTIT Landing Zone muss über einen zentralen Management Service erfolgen.

Aus diesem zentralen Management Service werden alle SAP-Anwendungen administriert.
Es ist eine 2FA erforderlich.

Motivation: Unkontrollierter privilegierter Zugriff muss verhindert werden.

Umsetzungsbeispiel: Beispiele für zentrale Management Services:

- Zentrales HANA Studio
- Zentrales HANA Cockpit
- COSMOS Management Services

ID: 3.22-378/13.0

Req 379	In dem zentralen Management Service dürfen nur Mitarbeiter aus dem Operationsbereich zugreifen.
---------	---

In dem zentralen Management Service dürfen nur Mitarbeiter aus dem Operationsbereich (OS und SAP-Basis) zugreifen.

Motivation: Sicherstellung von einem sicheren und einheitlichen Applikationsadministrationszugang für SAP-Systemlandschaften.

ID: 3.22-379/13.0

Req 380	Auf die zentralen Management Services müssen die administrativen Zugriffe müssen über das Tool USP erfolgen.
---------	--

Auf die zentralen Management Services müssen die administrativen Zugriffe müssen über das Tool USP erfolgen.

Motivation: Sicherstellen einer sicheren Authentifizierung für Administratoren.

ID: 3.22-380/13.0