

Sicherheitsanforderung

Web Service Gateway

Deutsche Telekom Gruppe

Version	8.1
Datum	01.12.2025
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 3.13	Dokumententyp Sicherheitsanforderung
Version 8.1	Stand 01.12.2025	Status Freigegeben
Fachlicher Ansprechpartner Telekom Security psa.telekom.de	Gültigkeitsdauer 01.12.2025 - 30.11.2030	Freigegeben von Stefan Pütz, Leiter SEC-T-TST

Zusammenfassung

Dieses Dokument wurde auf Basis der Vorgaben aus den im Konzern gültigen Sicherheitsrichtlinien erstellt und richtet sich an alle Systeme, die Gateways für Web Service Dienste bereitstellen.

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	4
2.	Allgemein	5
2.1.	Systemhärtung	5
2.2.	Systemaktualisierung	9
3.	Zugriffsteuerung	12
3.1.	Zugriffsteuerung von Konsumenten und Anbietern	12
3.2.	Zugriffsteuerung: Token	14
3.3.	Zugriffskontrolle auf Komponenten des Web Service Gateway	16
4.	Vertraulichkeit	20
4.1.	Transportverschlüsselung	20
4.2.	Zwischenspeichern von Web Service Anfragen und Antworten	25
5.	Sicherheits- und Compliancefunktionalitäten	27
5.1.	API-Katalog	27
5.2.	Anbieter und Konsumenten	28
5.3.	Web Service Gateway	30
6.	Aufnahme von Anbietern und Konsumenten	32
7.	Architektur	33
8.	Spezialfall: API-Gateway - Human to Maschine (H2M) Szenario	34
9.	Protokollierung	36

1. Einleitung

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.

2. Allgemein

Dank Camara und TM Forum haben sich die HTTP-Rest APIs als Quasi-Industriestandard etabliert, und es ist zu erwarten, dass sie die bestehenden älteren Technologien ersetzen werden. Aus diesem Grund liegt der Fokus dieses Dokuments auf der HTTP-REST-Technologie, wenn von APIs die Rede ist.

Die grundsätzlichen Sicherheitsprinzipien gelten für alle API-Technologien (Schnittstellen) und können auf eine Event-basierte (bspw. Kafka) oder eine ältere (legacy) Technologie wie SOAP verallgemeinert werden. Die PSA-Anforderungen sprechen allgemein von Web-Service. Ergeben sich für eine Technologie spezifische Anforderungen sind diese in einem Kapitel zusammengefasst.

Das Dokument leitet sich aus der Konzernrichtlinie "API Security @ DTAG" ab und beschreibt Konkretisierungen der Richtlinie.

2.1. Systemhärtung

Req 1	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 abubrechen 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 2 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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-3/9.0

Req 3 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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-4/9.0

Req 4 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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-5/9.0

Req 5 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 entsprechende 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. Eben-

falls 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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-9/9.0

2.2. Systemaktualisierung

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 Einsatz 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. Ana-

log 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.

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

3. Zugriffsteuerung

3.1. Zugriffsteuerung von Konsumenten und Anbietern

Req 8	Das Web Service Gateway muss Konsumenten und Anbieter eindeutig authentifizieren und autorisieren.
-------	--

Das Web Services Gateway muss sicherstellen, dass der Konsument berechtigt ist auf diesen Web Service zuzugreifen und diese Daten erhalten bzw. diese Funktionen verwenden darf.
Das Web Service Gateway muss ebenfalls die Anbieter eines Web Services authentifizieren.

Motivation: Nur die ordnungsgemäße Authentifizierung kann gewährleisten, dass die bereitgestellten Informationen ausschließlich vom designierten Benutzer gelesen werden bzw. Anbieter erzeugt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-8/8.1

Req 9	Für die Authentisierung und Autorisierung der Consumer und Provider muss ein anerkanntes, Industriestandard-Protokoll (bspw. OAuth 2) verwendet werden.
-------	---

Die Implementierung robuster und anpassungsfähiger Authentifizierungsmechanismen ist von entscheidender Bedeutung. Genutzt werden Industriestandard-Protokolle wie OAuth 2.0 zur Validierung der Identität von M2M- und menschlichen Akteuren, die versuchen, auf APIs zuzugreifen. Dabei muss der gewählte Mechanismus auf das erforderliche Schutzniveau der Daten und die Schutzwürdigkeit der API innerhalb unserer Systemarchitektur abgestimmt sein. Statische API-Tokens sollten niemals eine Option für interne Dienste sein.

In Verbindung mit der Authentifizierung sind zuverlässige Autorisierungskontrollen unverzichtbar. Es muss die Möglichkeit bestehen, fein abgestufte Zugriffskontrollen zu definieren und durchzusetzen, um sicherzustellen, dass authentifizierte Entitäten nur auf die Ressourcen und Funktionalitäten Zugriff haben, die für ihre Rollen und Verantwortlichkeiten relevant sind. Dies bedingt eine präzise Zuweisung von Benutzerrollen zu bestimmten API-Berechtigungen sowie eine präzise abgestimmte Kontrollebene.

Ein effizientes Management von kurzlebigen Tokens spielt für die sichere (konzerninterne) API-Interaktion eine entscheidende Rolle. Die Lebensdauer der Tokens ist auf der Grundlage der Schutzwürdigkeit der Daten und der Geschäftsanforderungen zu definieren. Es müssen Mechanismen für den Widerruf und die Erneuerung von Tokens eingerichtet werden, um die Kontrolle über die Zugriffsberechtigungen zu wahren.

Die Verwendung eines (förderierten) zentralisierten IDP (Identity Providers) als einziger vertrauenswürdiger Identitätsanbieter ist ein wesentlicher Teil des Konzepts eines API-Gateways.

Umsetzungsbeispiel: TARDIS mit seinem Zusammenspiel von IRIS und Gateway. Beispiel: https://developer.telekom.de/docs/src/tardis_customer_handbook/stargate/

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-9/8.1

Req 10	Das Web Service Gateway muss sich gegenüber dem Web Service Anbieter authentisieren.
--------	--

Der Anbieter eines Web Service muss das Web Service Gateway authentifizieren. Dafür muss das Web Service Gateway geeignete Authentifizierungsmerkmale bereitstellen.

Motivation: Damit der Web Service Anbieter sicher sein kann, dass die Kommunikation über das Web Service Gateway läuft und damit alle Sicherheitsfunktionalitäten gewährleistet sind muss das Web Service Gateway dem Web Ser-

vice Anbieter geeignete Authentifizierungsmerkmale bereitstellen.

Umsetzungsbeispiel: Eine gängige Methode dazu sind Gateway Access Tokens oder ein übermitteltes Clientzertifikat (mTLS).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.13-10/8.1

Req 11	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, TLS-Client-Zertifikate oder Tokens mit geeigneten Algorithmen wie ECDSA/RSA-Signaturen.
Im Restful-API Kontext sind das Protokolle wie OIDC, OAuth2.0, JWT-Token (RFC 7519) mit den dazugehörigen starken Krypto-Parametern.

Motivation: Schwache Algorithmen können von Angreifern kompromittiert und Identitäten vorgetäuscht werden.

Umsetzungsbeispiel: Zertifikate gemäß der Zertifikatsanforderungen für HTTPS
Beispiele für Tokens sind JW-, OAuth- und STS-Tokens, sofern über TLS übertragen.

Tokens:

- Ausschließlich Signatur Tokens basierend auf asymmetrischer Kryptografie (ES,RS)
- möglichst kurze Lebenszeiten

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.13-11/8.1

Req 12	Das Web Service Gateway muss die Zugriffe von Konsumenten auf Anbieter mit Hilfe nachgelagerter Servicerepositorys oder Autorisierer validieren.
--------	--

Konsumenten eines Web Service Gateways müssen auf ihre Berechtigung hin überprüft werden.
Diese Informationen sind üblicherweise in einem Dienstverzeichnis (Servicerepository oder Autorisierer) hinterlegt.

Motivation: Um Überlastungen und Fehlfunktionen in der Perimeterschicht zu verhindern, die sich auf die Zugriffsteuerung auswirken sollten Komponenten die Berechtigungen erteilen oder überprüfen von dem Web Service Gateway getrennt gehalten werden.

Umsetzungsbeispiel: Ein Umsetzungsbeispiel ist ein Keycloak das basierend auf dem enthaltenen Rollen- und Berechtigungskonzept Tokens ausstellt die durch das Gateway verarbeitet werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.13-12/8.1

Req 13 Das API-Gateway muss nicht plausible Verbindungsversuche, beispielsweise durch IP-Adressenfilterung blockieren.

IP-Adressenfilterung: Ein API-Provider sollte in der Lage sein, optional eine Whitelist gültiger Quell-IP-Adressen auf der Steuerungsebene zu definieren. Nicht übereinstimmender Verkehr muss auf Ebene des API-Gateways blockiert werden.

Motivation: Durch eine möglichst frühe Blockierung von Verbindungsversuchen, werden die nachfolgenden Schutzmechanismen entlastet.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-13/8.1

Req 14 Das Gateway darf keinen nicht authentifizierten Verkehr weiterleiten.

Kein nicht authentifizierter Verkehr: Es darf generell kein nicht authentifizierter Verkehr über das Gateway weitergeleitet werden. Ggf. gibt es Ausnahmen, also Fälle, die nicht authentifiziert sind – etwa den Authentifizierungs-Handshake.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.13-14/8.1

Req 15 Das API-Gateway muss eine Funktion zur Token-Deaktivierung einer Provider- oder Consumer-Identität für alle bereitstellen (Not-Stop).

Token-Deaktivierung: Eine Methode zum Deaktivieren einer Provider- oder Consumer-Identität für alle API-Provider oder -Konsumenten. Diese Funktion kann vom Cyber Defense Center-Team benutzt werden, um einen berechtigten Teilnehmer, dessen Datensicherheit kompromittiert wurde, zu deaktivieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-15/8.1

3.2. Zugriffsteuerung: Token

Token-Deaktivierung: Eine Methode zum Deaktivieren einer Provider- oder Consumer-Identität für alle API-Provider oder -Konsumenten. Diese Funktion kann vom Cyber Defense Center-Team benutzt werden, um einen berechtigten Teilnehmer, dessen Datensicherheit kompromittiert wurde, zu deaktivieren.

Req 16 Das API-Gateway muss zur Authentisierung und Autorisierung Self Contained Token nutzen.

Self Contained Token sind zu verwenden, um die Komplexität in Multi-Service-Umgebungen zu reduzieren und eine Grundlage für künftige Zero-Trust-orientierte Umgebungen zu schaffen.

- Nur sichere und asymmetrische Funktionen sind zur Erstellung der Signatur erlaubt.
- Die Consumer- oder Provider-Identität muss Teil des Tokens sein.

"Self Contained" meint hier, dass das Token mittels einer Signatur überprüfbar ist und dadurch der Aussteller und weitere Attribute für den Prüfenden verfügbar sind.

Umsetzungsbeispiel: Beispiele sind JWT-Token nach RFC 7519.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.13-16/8.1

Req 17	Das Gateway muss auf dem Egress die Client-Autorisierungstoken durch ein generisches Gateway-Token ersetzen.
--------	--

Ersetzen von Token: Auf dem Egress (Ausgangs)-Gateway-Baustein werden die Client-Autorisierungstoken durch ein generisches Gateway-Token ersetzt, das zum Schutz gegen Replay-Angriffe dieselbe Payload aufweist. Dies bedeutet, dass das neue Gateway-Token niemals auf einem Ingress (Eingangs)-Gateway zugelassen ist.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-17/8.1

Req 18	Token müssen kurzlebig sein.
--------	------------------------------

Kurzelebige Token: Das Authentifizierungs- und Autorisierungs-Token darf nur für ganz kurze Zeit gültig sein, maximal für 5 Minuten. Bei Aufgaben, die länger in Anspruch nehmen, kann ein Nachrichten-Broker genutzt werden.

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: 3.13-18/8.1

Req 19	Das API-Gateway muss die Validität und Signatur des Authentifizierungstokens geprüft.
--------	---

Token-Validierung: Die Validität und Signatur eines (Gateway-) Authentifizierungstokens muss immer durch die Provider-Anwendung und durch die Gateway-Bausteine geprüft werden.

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: 3.13-19/8.1

Req 20	Das API-Gateway bzw. der Identity Provider (IDP) muss auf Anforderung durch den Consumer "Scopes" im Autorisierungstoken bereitstellen.
--------	---

Unterstützung von Scopes: Eine Consumer-Anwendung kann einen oder mehrere Scopes anfordern. Diese Informati-

on wird dann im Autorisierungstoken präsentiert und kann vom Providerservice validiert werden, um den Zugriff auf bestimmte Ressourcen zu erlauben oder zu verbieten.

Motivation: Zur einfachen Umsetzung eines Rollen- und Berechtigungsmodells kann der Provider "Scopes" definieren, die dann den Consumern durch den IDP zugeordnet werden.

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
- Abstreiten von durchgeführten Aktionen

ID: 3.13-20/8.1

Req 21	In besonderen Fällen (bspw. externer SaaS-Service) darf das API-Gateway statt dem Self-Contained Token alternative Authentisierungsmerkmale, wie beispielsweise API-Schlüssel anbieten.
--------	---

Es gibt berechtigte Gründe für die Nutzung von API-Schlüsseln, jedoch nur im Zusammenhang mit externen Kunden. In einem solchen Szenario ist dies im Privacy and Security Assessment zu dokumentieren und vom Project Security Manager zu genehmigen.

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
- Abstreiten von durchgeführten Aktionen

ID: 3.13-21/8.1

3.3. Zugriffskontrolle auf Komponenten des Web Service Gateway

Req 22	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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-7/9.0

Req 23 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.

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
- Abstreiten von durchgeführten Aktionen

ID: 3.01-22/9.0

Req 24 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).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-8/9.0

Req 25 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.

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
- Abstreiten von durchgeführten Aktionen

ID: 3.01-23/9.0

Req 26 Benutzerkonten müssen mit mindestens zwei Authentisierungsmerkmalen geschützt werden.

Um einen höheren Schutz zu erreichen, müssen mehrere voneinander unabhängige Authentisierungsmerkmale parallel genutzt werden.

Dieser Ansatz wird allgemein als MFA (Multi-Faktor-Authentisierung) bezeichnet.

Eine spezifische Form ist die 2FA (2-Faktor-Authentisierung), die exakt zwei Authentisierungsmerkmale kombiniert.

Motivation: Benutzerkonten mit erhöhten Rechten, wie sie für die Wartung und Administration von Systemen genutzt werden, stellen ein erhöhtes Risiko für die Sicherheit eines Systems dar. Wenn ein Angreifer ein solches Benutzerkonto kompromittieren kann, verfügt er über weitreichende Berechtigungen, um auf weite Teile des Systems und der darauf gespeicherte Daten zu zugreifen.

Umsetzungsbeispiel: Sehr verbreitet ist 2FA in einer Variante bestehend aus einem Merkmal, das der Benutzer kennt und einem Merkmal, das der Benutzer besitzt.

Beispiele für eine solche 2FA sind:

- Smartcard (z. B. MyCard) mit PIN
- Privater Schlüssel mit Passphrase
- Hardware Token zur Generierung von Einmal-Passwörtern

In Unternehmen der Deutschen Telekom Gruppe, in denen die MyCard oder eine vergleichbare Smartcard verfügbar ist, ist diese Lösung zu favorisieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.13-26/8.1

Req 27	Der administrative Zugang zu den Komponenten des Web Service Gateway muss über Jump-Hosts oder dedizierte Netzwerke erfolgen.
--------	---

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.13-27/8.1

4. Vertraulichkeit

4.1. Transportverschlüsselung

Req 28 Es muss TLS in der Version 1.2 oder 1.3 verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

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

Quellen:

[RFC-5246] The Transport Layer Security (TLS) Protocol Version 1.2

[RFC-8446] The Transport Layer Security (TLS) Protocol Version 1.3

Motivation: Die aktuelle Version von TLS schließt bisherige Sicherheitsschwachstellen und Angriffsflächen auf den TLS-Protokoll-Handshake.

Umsetzungsbeispiel: OpenSSL> protocol = tlsv1_3

ID: 3.13-29/8.1

Req 29 Für TLS müssen die Diffie-Hellman Gruppen mit IANA-Nr. 23-33 verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Die Diffie-Hellman Gruppe wird zum Schlüsselaustausch mit Perfect Forward Secrecy (PFS) verwendet.

Die folgende Tabelle beinhaltet die zulässigen Diffie-Hellman Gruppen:

Diffie-Hellman Gruppe	IANA-Nr.	Referenzspezifikation	Hinweis
secp256r1	23	[RFC 8422]	
secp384r1	24	[RFC 8422]	
secp521r1	25	[RFC 8422]	
brainpoolP256r1	26	[RFC 7027]	TLS 1.2
brainpoolP384r1	27	[RFC 7027]	TLS 1.2
brainpoolP512r1	28	[RFC 7027]	TLS 1.2
x25519	29	[RFC-8422]	
x448	30	[RFC-8422]	
brainpoolP256r1tls13	31	[RFC 8734]	TLS 1.3
brainpoolP384r1tls13	32	[RFC 8734]	TLS 1.3
brainpoolP512r1tls13	33	[RFC 8734]	TLS 1.3
ffdhe3072	257	[RFC 7919]	
ffdhe4096	258	[RFC 7919]	

Hinweis zu x448 und x25519:

x448 und x25519 werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Hinweis zu Brainpool-Kurven:

Für die Brainpool-Kurven in TLS v1.2 [RFC-7027] wurden die IANA-Nummern 26, 27 und 28 reserviert, in TLS v1.3 [RFC-8446] die IANA-Nummern 31, 32 und 33.

Hinweis zu DH Gruppe 256:

Die Diffie-Hellman Gruppe 256 (IANA-Nr. 256) besitzt eine Schlüssellänge von 2048 Bit und darf nicht mehr eingesetzt werden.

Hinweis zu DH Gruppen 257 und 258:

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

Hinweis zu PQC:

Wenn eine hybride Implementierung von klassischer und Post-Quanten-Kryptographie verwendet werden (erfordert TLS 1.3), dann muss der klassische Teil dieser Anforderung entsprechen.

Quellen:

[RFC-7027] Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS)

[RFC-7919] Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)

[RFC-8422] Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier

[RFC-8734] Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS) Version 1.3

[TR-02102-2] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 2 TLS, Version 2025-01

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

ID: 3.13-31/8.1

Req 30	Es müssen die folgenden Cipher Suites mit Perfect Forward Secrecy (PFS) verwendet werden: für TLS v1.3: IANA-Nr. "0x13,0x01", "0x13,0x02", "0x13,0x03" or "0x13,0x04", sowie für TLS v1.2: IANA-Nr. "0xC0,0x2B", "0xC0,0x2C", "0xC0,0xAC", "0xC0,0xAD", "0xC0,0x2F", "0xC0,0x30", "0xCC,0xA9" oder "0xCC,0xA8".
--------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

Cipher Suites spezifizieren die kryptographischen Verfahren einer TLS-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 Schlüsseleinigungsverfahren 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.

Cipher Suite	IANA-Nr.	Referenzspezifikation
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xC0,0x2B	[RFC-5289]
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	0xC0,0x2C	[RFC-5289]
TLS_ECDHE_ECDSA_WITH_AES_128_CCM	0xC0,0xAC	[RFC-7251]
TLS_ECDHE_ECDSA_WITH_AES_256_CCM	0xC0,0xAD	[RFC-7251]
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	0xC0,0x2F	[RFC-5289]
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	0xC0,0x30	[RFC-5289]
TLS_ECDHE_ECDSA_WITH_CHACHA20POLY1305_SHA256	0xCC,0xA9	[RFC-7905]
TLS_ECDHE_RSA_WITH_CHACHA20POLY1305_SHA256	0xCC,0xA8	[RFC-7905]

Die folgenden Cipher Suites mit PFS in TLS v1.2 mit AES-CBC (CBC-Modus) und / oder DHE Cipher Suites sind nur in Bestandsystemen erlaubt:

Cipher Suite	IANA-Nr.	Referenzspezifikation	Hinweis
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	0xC0,0x23	[RFC-5289]	CBC
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	0xC0,0x24	[RFC-5289]	CBC
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	0xC0,0x27	[RFC-5289]	CBC
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	0xC0,0x28	[RFC-5289]	CBC
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	0x00,0xA2	[RFC-5288]	DHE
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	0x00,0xA3	[RFC-5288]	DHE
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	0x00,0x9E	[RFC-5288]	DHE
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	0x00,0x9F	[RFC-5288]	DHE
TLS_DHE_RSA_WITH_AES_128_CCM	0xC0,0x9E	[RFC-6655]	DHE

TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	0xC0,0x9F	[RFC-6655]	DHE
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	0xCC,0xAA	[RFC-7905]	DHE
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	0x00,0x40	[RFC-5246]	CBC / DHE
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256	0x00,0x6A	[RFC-5246]	CBC / DHE
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	0x00,0x67	[RFC-5246]	CBC / DHE
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	0x00,0x6B	[RFC-5246]	CBC / DHE

Hinweis zu DHE Cipher Suites:

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). Sie werden voraussichtlich vom IETF abgekündigt und vom BSI [TR-02102-2] nur bis zum Jahr 2029 empfohlen.

Hinweis zu Cipher Suites mit AES-CBC:

Cipher Suites mit AES-CBC sind nur mit der TLS-Erweiterung "Encrypt-then-MAC" vom BSI empfohlen ([TR-02102-2]). Die Erweiterung ist aber in den meisten Implementierungen nicht aktivierbar bzw. eine Überprüfung, ob sie aktiv ist, ist nicht möglich.

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.

Cipher Suite	IANA-Nr.	Referenzspezifikation
TLS_AES_128_GCM_SHA256	0x13,0x01	[RFC-8446]
TLS_AES_256_GCM_SHA384	0x13,0x02	[RFC-8446]
TLS_CHACHA20_POLY1305_SHA256	0x13,0x03	[RFC-8446]
TLS_AES_128_CCM_SHA256	0x13,0x04	[RFC-8446]

Hinweis zu CHACHA20 Cipher Suites:

CHACHA20 Cipher Suites: werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Hinweis zu PQC:

Wenn eine hybride Implementierung von klassischer und Post-Quanten-Kryptographie verwendet werden (erfordert TLS 1.3), dann muss der klassische Teil dieser Anforderung entsprechen.

Alle zukünftig spezifizierten Cipher Suites, die den in diesem Dokument definierten Anforderungen entsprechen, können ebenfalls verwendet werden. Dies gilt beispielsweise für Cipher Suites, die eine Hash-Funktion aus der SHA-3-Familie verwenden.

Quellen:

[RFC-5246] The Transport Layer Security (TLS) Protocol Version 1.2
[RFC-5288] AES Galois Counter Mode (GCM) Cipher Suites for TLS
[RFC-5289] TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)
[RFC-6655] AES-CCM Cipher Suites for Transport Layer Security (TLS)
[RFC-7251] AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS
[RFC-7905] ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)
[RFC-8446] The Transport Layer Security (TLS) Protocol Version 1.3
[TR-02102-2] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 2
TLS, Version 2025-01

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

Umsetzungsbeispiel: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

ID: 3.13-33/8.1

Req 31 Für TLS müssen digitale Zertifikate von einer Zertifizierungsstelle mit ausreichender Schlüssellänge und begrenzter Gültigkeit verwenden sowie korrekt validiert werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Digitale Zertifikate werden in TLS zur Authentisierung während des TLS-Handshakes verwendet. TLS-Server müssen ein TLS-Zertifikat verwenden, Clients benötigen ein Zertifikat falls gegenseitige Authentisierung gefordert ist (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/produkte/serverpass/ueberblick>).

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

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 Zertifikatslaufzeit von öffentlichen TLS-Serverzertifikaten (ausgestellt von einer Zertifizierungsstelle, die Zertifikate entsprechend der Vorgaben des [CA/Browser Forum] ausstellt) darf maximal 397 Tage betragen. Für andere, interne TLS-Zertifikate sollte eine Laufzeit von maximal 3 Jahren nicht überschritten werden.

Die Zertifikatslaufzeit für automatisch ausgestellte und automatisch erneuerbare Zertifikate, z.B. bei ACME (Automatic Certificate Management Environment), sollte möglichst kurz gewählt werden. Zu empfehlen ist eine Laufzeit von maximal 90 Tagen.

Zertifikats-Validierung (vgl. [RFC-5280]):

- Die Zertifikate müssen gegen eine wohldefinierte Trust-Anchor-Liste validiert werden, die auf die nötigen Trust-Anchors reduziert sind. Dabei können Intermediate Zertifikate als Trust-Anchor verwendet werden.
- Bei der Validierung von Serverzertifikaten muss eine Hostname Validation durchgeführt werden.
- Die Zertifikate müssen gültig sein.
- Wenn in der Zertifikatskette „Name Constraints“ definiert sind, müssen alle nachfolgenden Zertifikate in der Kette diese „Name Constraints“ einhalten.

Quellen:

[CA/Browser Forum] <https://cabforum.org/working-groups/server/baseline-requirements/documents/>

[RFC5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, chapter 6

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

ID: 3.13-35/8.1

4.2. Zwischenspeichern von Web Service Anfragen und Antworten

Req 32	Falls Web Service Anfragen oder Antworten zwischengespeichert werden, muss das Web Service Gateway die Integrität der Nachrichten sicherstellen.
--------	--

Motivation: Um die Integrität der gespeicherten Nachrichten sicherzustellen muss eine Prüfsumme über die Nachrichten gebildet werden.

Umsetzungsbeispiel: Um die Integrität der Kommunikation zwischen Konsument und Anbieter zu gewährleisten empfehlen sich durchgängige Verfahren wie z. B. XML Signaturen oder "JSON web signatures" (RFC 7515). Alternativ können einfache Prüfsummen wie z. B. CRC32 durch das Web Service Gateway gebildet werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-36/8.1

Req 33	Das Gateway darf nicht auf die Payload der Nachrichten zugreifen.
--------	---

Das bedeutet insbesondere, dass das Gateway niemals Entscheidungen anhand der Nachrichten-Payload treffen darf.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-37/8.1

Req 34	Falls Nachrichten auf persistenten Datenspeichern zwischengespeichert werden, müssen die Speichermedien verschlüsselt sein.
--------	---

Die Verschlüsselung muss nach aktuellem Stand der Technik erfolgen.

Motivation: Auf nicht verschlüsselten Speichermedien können Datenreste entstehen, die bei der Entsorgung der Datenträger problematisch werden können.

Auch kann im Falle einer Kompromittierung des Systemes Daten früherer Web Service Kommunikationsdaten wiederhergestellt werden.

Umsetzungsbeispiel: Hierfür empfiehlt sich eine Datenbankverschlüsselung oder komplette Dateisystem Verschlüsselung.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-38/8.1

Req 35	Falls Web Service Antworten oder Anfragen zwischengespeichert werden, muss die Gültigkeitszeit durch den Anbieter des Web Services gewählt werden können.
--------	---

Für die Mehrfachverwendung von Web Service Antworten darf im standardmäßig keine Gültigkeitszeit vorgesehen sein.

Die maximale Gültigkeitszeit muss durch das Gateway beschränkt werden.

Motivation: Um eine ungewollte Mehrfachverwendung von Web Service Antworten zu verhindern dürfen diese Standard gemäß nicht zwischengespeichert werden.

Der Anbieter muss sicherstellen, dass seine Web Service Antworten zur Mehrfachverwendung geeignet sind und darauf gezielt diese Funktionalität aktivieren.

Die maximale Gültigkeitszeit muss vom Gateway beschränkt werden um überlaufende Speicher zu verhindern.

Umsetzungsbeispiel: Zur Umsetzung empfiehlt sich der Einsatz vom Time-to-live (TTL) Header

ID: 3.13-39/8.1

5. Sicherheits- und Compliancefunktionalitäten

5.1. API-Katalog

Mechanismen und Funktionen zur Verwaltung und Bestandsführung der angebotenen APIs (Provider) und deren Nutzer (Consumer).

Req 36 Jede API (M2M & H2M) ist zentral zu dokumentieren.

Jede API (M2M & H2M) ist zentral zu dokumentieren (entweder im zentralen [DTAG Konzern-Katalog](#) oder im Nat-Co-Katalog). Spezifikation und Details zur Dokumentation finden sich in den PSA-Anforderungen bzw. in den Regularien der Geschäftseinheit, beispielsweise den [G-API-Standards and Conventions](#).

Umsetzungsbeispiel: Beispiel API-Katalog - TARDIS: [DTAG Konzern-Katalog](#)

Beispiel Spezifikation und Details zur Dokumentation als lokale Regelung [G-API-Standards and Conventions](#).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.13-40/8.1

Req 37 Die Dokumentation muss eine technische Spezifikation umfassen, z.B. OpenAPI-Spezifikation (OAS).

Die Dokumentation muss eine technische Dokumentation umfassen, z.B. OpenAPI-Spezifikation (OAS) oder Blueprint. Sämtliche Parameter, Funktionen und Methoden müssen eindeutig definiert sein. Der Grad der Vollständigkeit obliegt der Verantwortung des Eigentümers des API-Providers.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.13-41/8.1

Req 38 Die Dokumentation muss in maschinenlesbarem (OAS) und parallel dazu in von Menschen lesbarem Format vorliegen.

Die Dokumentation muss in maschinenlesbarem (OAS) und parallel dazu in von Menschen lesbarem Format vorliegen, um die Implementierung von automatisierten Sicherheitskontrollen und die Durchsetzung des Standards zu erleichtern.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.13-42/8.1

Req 39 Jedem API-Provider muss in Bezug auf seine Exposition mindestens eine Kategorie (bspw. Internet, Backend, ...) zugewiesen werden.

Veröffentlichungskategorie (Exposure): Jeder API-Provider muss in Bezug auf seine Exposition einer Kategorie zugewiesen werden. Dabei sind mindestens die Kategorien *Internet* und *Backend* oder ähnliche Nomenklaturen zu verwenden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

Req 40	Die Vollständigkeit des API-Katalogs muss mindestens alle 180 Tage überprüft werden.
--------	--

Bei jeder LBU und NatCo müssen Verfahren implementiert sein, welche die Vollständigkeit ihres API-Katalogs mindestens alle 180 Tage überprüfen.

Motivation: API-Security@DTAG Richtlinie.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

Req 41	Die technische Dokumentation muss die folgenden Informationen, Felder und Methoden bezüglich der verfügbaren API-Endpunkte enthalten:
--------	---

- Spezifikationsstandard
- API-Name
- API-Beschreibung
- API-Version
- Verantwortliche Org.- Einheit
- Kontakt-E-Mail-Adresse des API-Verantwortlichen
- Sicherheits-Scopes (bezüglich Autorisierung) für die Endpunkte
- Kundensegment
- IT-Anwendung, z.B. ICTO-Nummer
- Version des API-Dienstes
- Ggf. Restlaufzeithinweise (deprecation indicator)
- Endpunkt-Spezifikationen
- Datenspezifikationen
- Genehmigte und bereitgestellte Media-Typs
- HTTP-Status-Codes
- Error Response Object
- Differenzierte Fehlerfälle
- Datenschutzklasse (DSK) oder Informationsschutzklasse (VS)

Umsetzungsbeispiel: Nähere Erläuterungen und weitere Informationen zur allgemeinen API-Standardisierung und -Dokumentation finden sich hier:

https://developer.telekom.de/docs/src/api_standards_conventions/2_basics/documentation/

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

5.2. Anbieter und Konsumenten

Req 42	Alle Web Service Anfragen müssen durch das Web Service Gateway 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.

Motivation: Die detaillierte Spezifikation (einschließlich formelle Spezifikation) ermöglicht eine weitaus bessere Beschreibung der Daten, die in Datenfeldern eines Web Services erwartet werden.

Umsetzungsbeispiel: Eine komplette Web Service Beschreibung kann durch z. B. WSDL 2.0 oder OpenAPI erreicht werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-46/8.1

Req 43	Das Web Service Gateway muss die maximale Anzahl an Verbindungen zu Web Service Anbietern limitieren können.
--------	--

Die Limits sollten durch den Anbieter frei wählbar sein und dürfen einen Burstmodus vorsehen.

Motivation: Um die Web Service Anbieter vor Überlast zu schützen muss das Web Service Gateway Funktionalitäten zur Verbindungslimitierung bereitstellen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.13-47/8.1

Req 44	Validierungsfehler und überschrittene Verbindungslimits müssen durch das Web Service Gateway protokolliert werden.
--------	--

Motivation: Zur Fehlerbehebung und Erkennung von Angriffen auf einen Web Service Anbieter, muss das Gateway Validierungsfehler und überschrittene Verbindungslimits protokollieren und den Anbietern eines Web Services bereitstellen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.13-48/8.1

Req 45	Konfigurierte Anbieter und Konsumenten müssen jederzeit einsehbar sein.
--------	---

Anbietern von Web Service Schnittstellen muss jederzeit eine Liste aller Konsumenten bereitgestellt werden können. Diese Liste sollte automatisiert bereitgestellt werden können.

Motivation: Um eine Auditierung und Compliance Validierung zu gewährleisten müssen alle Konsumenten und Anbieter jederzeit einsehbar 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 46	Web Service Konsumenten und Anbietern muss eine Liste der durch das Gateway ausgewerteten und protokollierten Daten bereitgestellt werden.
--------	--

Motivation: Um Datensicherheit und Datenschutz sicherzustellen muss den Nutzern vom Web Service Gateway transparent gemacht werden welche Daten ausgewertet und protokolliert werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

5.3. Web Service Gateway

Req 47	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 Verminderung 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 Schutzaufgabe für nachgelagerte Systeme.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

Req 48	Die Komponenten des Web Service Gateway müssen gegen unberechtigte Veränderung geschützt werden.
--------	--

Der Schutz sollte sich nicht nur auf die Software des Web Service Gateway beschränken sondern das komplette Betriebssystem abdecken.

Motivation: Um die Software und das Betriebssystem manipulationsresistent zu machen sind Mechanismen, die die Dateintegrität sicherstellen die einzige Möglichkeit Manipulationen zu verhindern.

Umsetzungsbeispiel: In Container Architekturen sollten die kompletten Container signiert werden und vor dem aus-

führen überprüft werden.

Bei z. B. Tomcat Applikationen kann die JAR Datei signiert werden und durch den Tomcat Server überprüft werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.13-52/8.1

Req 49	Die Konfiguration des Web Service Gateway muss regelmäßig validiert werden.
--------	---

Um die Konfiguration vor unberechtigten und ungewollten Veränderungen zu schützen muss die ausgerollte Konfiguration regelmäßig mit dem Soll Zustand überprüft werden.

Das betrifft neben der Servicekonfiguration ebenfalls die Schnittstellenkonfiguration von Konsumenten und Anbietern.

Motivation: Die Konfigurationen der Web Service Gateways werden durch z. B. Servicerepertoires bereitgestellt und durch automatische Mechanismen an oft dezentrale Web Service Gateways ausgerollt.

Um die Funktion dieser Mechanismen und ungewollte Änderungen an diesen Konfigurationen zu verhindern muss der Ist-Zustand der Konfigurationen regelmäßig mit dem Soll-Zustand abgeglichen werden.

Hierdurch werden Fehlfunktionen verhindert indem die Aktualität der Konfiguration garantiert wird.

Umsetzungsbeispiel: Konfigurationsdateien können Signiert und versioniert werden.

Die verwendete Version kann als Web Service bereitgestellt und durch den Verteilungsdienst automatisiert überprüft werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.13-53/8.1

Req 50	Änderungen an der Konfiguration vom Web Service Gateway müssen fortlaufend protokolliert werden.
--------	--

Motivation: Um Änderungen, insbesondere an der Schnittstellenkonfiguration, nachvollziehen zu können muss ein fortlaufendes Protokoll über die Änderungen, sowie denjenigen der sie durchgeführt hat vorgehalten werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen

ID: 3.13-54/8.1

Req 51	Wird ein Web Service-Request mit Signaturen geschützt, dürfen die Signaturdaten nicht aus dem Request entfernt oder ersetzt 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 Bedrohungen relevant:

- Unberechtigte Modifikation von Daten

ID: 3.13-55/8.1

6. Aufnahme von Anbietern und Konsumenten

Req 52 Für jeden Anbieter und Konsumenten eines Web Service muss ein Hauptverantwortlicher benannt sein.

Der Hauptverantwortliche darf die Schnittstellenkonfiguration ändern. Beispielsweise URL Endpunkte, Konsumenten akzeptieren oder auch zusätzliche Verantwortliche benennen oder entfernen.

Motivation: Um klare Verantwortlichkeiten zu haben muss es pro Anbieter und Konsumenten eines Web Services einen Hauptverantwortlichen geben.

Dieser ist in erster Linie Verantwortlich für alle aufkommenden Themen z. B. Payload Protokollierung, Änderungen an der Verschlüsselung, Aufnahme neuer Konsumenten, rezertifizierung von Konsumenten etc.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.13-56/8.1

Req 53 Die Autorisierung neuer Konsumenten muss durch den Anbieter des Web Services genehmigt werden.

Änderungen an den Autorisierten Konsumenten müssen pro Anbieter fortlaufend protokolliert werden. Im Protokoll muss vorhanden sein wer welche Änderungen wann durchgeführt hat.

- 4-Augen-Prinzip: Eine Consumer-Provider-Verbindung ist nicht erlaubt, es sei denn, es liegt die Genehmigung von zwei Personen vor, welche verantwortlich für den entsprechenden Dienst (Consumer und Provider) sind.
- Neubewertung: Die Vereinbarung muss auf eine angemessene Zeit beschränkt sein, mit der Option einer periodischen Überprüfung.

Motivation: Ausschließlich der Anbieter eines Web Services verantwortet wer diesen Service nutzen oder diese Daten erhalten darf deshalb muss jeder neue Konsument durch den Anbieter genehmigt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.13-57/8.1

Req 54 Die Standardrichtlinie für neue Anbieter von Web Services muss alle Konsumenten ablehnen.

Die Zugriffsteuerung muss dem Prinzip „Default Deny“ folgen, d.h. der Zugang zu einem Service darf nicht ohne ausdrückliche Erlaubnis gewährt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Unbemerkt durchführbare Angriffe

ID: 3.13-58/8.1

7. Architektur

Req 55 Gateways die für interne Netzwerke zuständig sind, dürfen nicht direkt aus dem Internet erreichbar sein.

Auf Gateways die direkt im Internet erreichbar sind dürfen nur Anbieter provisioniert werden die auch aus dem Internet erreichbar sein sollen.

Motivation: Um die Anzahl von außen kompromittierbarer Web Services zu verringern und die Auswirkungen von "Denial of Service" Angriffen zu verringern müssen die internen Komponenten von externen getrennt werden.

Umsetzungsbeispiel: Für Gateways die in internen und externen Netzwerken verfügbar sind sollte ein Edge Gateway eingesetzt werden. Das Edge Gateway leitet die Anfragen aus dem Internet an das interne Gateway weiter.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.13-59/8.1

Req 56 Der Netzwerkverkehr für Web Services muss von Verkehr für Web Anwendungen getrennt sein.

Motivation: Die Gateways meist nicht dafür ausgelegt Web Anwendungen auszuliefern auch wenn sie grundsätzlich dazu in der Lage sind. Über Web Anwendungen werden meist Konfigurations- und Fehleranalysetools bereitgestellt die im Fehlerfall erreichbar bleiben müssen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.13-60/8.1

8. Spezialfall: API-Gateway - Human to Maschine (H2M) Szenario

Benutzerorientierte APIs: Je nach NatCo oder LBU verlangen die Architekturrichtlinien die Nutzung eines Backend-for-Frontend (BFF) als eine Abstraktionsebene anstelle eines API-Gateways (Erläuterung von BFF in DTAG).

<https://wiki.telekom.de/pages/viewpage.action?pageId=886833714>

Req 57	Backend-Dienste hinter dem BFF müssen über einen API-Gateway vernetzt sein.
--------	---

Backend-Dienste hinter dem BFF müssen über einen API-Gateway vernetzt sein. Die Backend-M2M-Tokens dürfen niemals den Benutzern vorgelegt werden.

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: 3.13-61/8.1

Req 58	Das BFF muss ein Rate limiting umsetzen.
--------	--

Rate limiting: Benutzer können pro Minute nur eine beschränkte Anzahl an Anfragen stellen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.13-62/8.1

Req 59	Das BFF darf nicht den (Benutzer-) Eingabe vertrauen bzw. muss diese validieren.
--------	--

Validierung: Niemals der (Benutzer-) Eingabe vertrauen, alles überprüfen. Vgl. die PSA-Anforderungen

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.13-63/8.1

Req 60	Das BFF muss die benutzerorientierten APIs (zum Browser hin) besonders auf sicherheitsrelevante Events überwachen.
--------	--

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.13-64/8.1

Req 61	Das BFF muss die Authentifizierung und Autorisierung zu jedem Request durchführen. Selbst anonymen Benutzern muss eine Rolle zugewiesen sein.
--------	---

Durchsetzen von Authentifizierung und Autorisierung: Jede Anfrage muss authentifiziert werden. Selbst anonymen Benutzern muss eine Rolle zugewiesen sein, um ein striktes Autorisierungs- und Berechtigungsmanagement durchzusetzen.

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
- Abstreiten von durchgeführten Aktionen

ID: 3.13-65/8.1

9. Protokollierung

Req 62	Payload darf nicht geloggt werden.
--------	------------------------------------

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.13-66/8.1

Req 63	Das API Gateway muss sowohl die erfolgreichen, aber noch mehr die erfolglosen Kommunikationsversuche protokollieren.
--------	--

Allgemeine Verkehrsprotokollierung: Sowohl die erfolgreichen, aber noch mehr die erfolglosen Kommunikationsversuche müssen protokolliert werden.

Minimale Protokollinformationen:

- Header,
- Pfadvariablen,
- Parameter und HTTP-Methoden,
- Zeitstempel.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.13-67/8.1

Req 64	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 Zeitzonen-Konfiguration bei der Ausgabe von Zeitinformationen in die Ortszeit transformiert werden, sofern die so stattfindende Zeitzonenanpassung 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:

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

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen

- Unbemerkt durchführbare Angriffe

ID: 3.01-32/9.0

Req 65	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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.01-33/9.0

Req 66	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

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.01-34/9.0

Req 67	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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.01-35/9.0

Req 68	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 ge-

nannten 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

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.01-36/9.0

Req 69	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.

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.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.01-37/9.0