- Systementwurf: Sicherheitskonzeption -

Sicherheitskonzept für die Neuentwicklung GDI-DE Testsuite

Version: 1

Projektbezeichnung	Neuentwicklung GDI-DE Testsuite	
Projektleiter	Steffi Forberig	
Verantwortlich	Informationssicherheitsverantwortlicher	
Erstellt am		
Zuletzt geändert	11.07.2019 14:46	
Bearbeitungszustand		in Bearbeitung
		vorgelegt
	x	fertig gestellt
Dokumentablage	/media/terrestris_projekte/bkg_gdide-testsuite/dokumente/V-Modell-Unterlagen/Testsuite3_NextCloud-Kopie/Systemspezifikation/20190711_Sicherheitskonzeption_MS2.odt	
V-Modell-XT Version	2.2	

Weitere Produktinformationen

Mitwirkend	[nicht beteiligt]	SW-Architekt
	[nicht beteiligt]	Systemarchitekt
	[nicht beteiligt]	Datenschutzverantwortlicher
	[nicht beteiligt]	Betriebsverantwortlicher
Erzeugung	Erstellung einer Sicherheitskonzeption auf Ebene des Gesamtsystems Erstellung einer Sicherheitskonzeption auf Ebene des Systems Erstellung einer Sicherheitskonzeption auf Ebene der SW-Einheit	

Änderungsverzeichnis

Änderungen			Geänderte	Beschreibung der		
Nr.	**Datum**	**Version**	**Kapitel**	**Änderungen**	**Autor**	**Zustand**
1	03.06.2019	0.1	Alle	Initiale Produkterstellung	Verena Diewald	prüffähig
2	11.06.2019	0.2	Alle	Ergänzungen	Verena Diewald	Fertig gestellt
3	26.06.2019	0.3	1	Ergänzung aktueller Link	Verena Diewald	Fertig gestellt
4	03.07.2019	0.4	4	Ergänzungen Kapitel 4 Einleitung, Kapitel 4.1	Verena Diewald	Fertig gestellt
5	11.07.2019	1	Deckblatt	Finale Versionsnummer MS2	Verena Diewald	Fertig gestellt

Prüfverzeichnis

Die folgende Tabelle zeigt einen Überblick über alle Prüfungen – sowohl Eigenprüfungen wie auch Prüfungen durch eigenständige Qualitätssicherung – des vorliegenden Dokumentes.

Datum	Geprüfte Version	Anmerkungen	Prüfer	Neuer Produktzustand
11.06.2019	0.1		EG	vorgelegt

INHALTSVERZEICHNIS

1 Einleitung....4
2 Vorgaben....5
3 Bedrohungen....6
4 Informationssicherheitsmaßnahmen....7
4.1 APP.3.1.A1 Authentisierung bei Webanwendungen....7
4.2 APP.3.1.A2 Zugriffskontrolle bei Webanwendungen....8
4.3 APP.3.1.A3 Sicheres Session-Management....8
4.4 APP.3.1.A4 Kontrolliertes Einbinden von Daten und Inhalten bei Webanwendungen....8
4.5 APP.3.1.A5 Protokollierung sicherheitsrelevanter Ereignisse von Webanwendungen....9
4.6 APP.3.1.A6 Zeitnahes Einspielen sicherheitsrelevanter Patches und Updates 9
4.7 APP.3.1.A7 Schutz vor unerlaubter automatisierter Nutzung von Webanwendungen....9
4.8 APP.3.1.A8 Systemarchitektur einer Webanwendung....9
4.9 APP.3.1.A9 Beschaffung, Entwicklung und Erweiterung von Webanwendungen....10
4.10 APP.3.1.A10 Test und Freigabe von Webanwendungen....10
4.11 APP.3.1.A11 Sichere Anbindung von Hintergrundsystemen....10
4.12 APP.3.1.A12 Sichere Konfiguration von Webanwendungen....10
4.13 APP.3.1.A13 Restriktive Herausgabe sicherheitsrelevanter Informationen…11
4.14 APP.3.1.A14 Schutz vertraulicher Daten....11
4.15 APP.3.1.A15 Verifikation essenzieller Änderungen....11
4.16 APP.3.1.A16 Umfassende Eingabevalidierung und Ausgabekodierung....11
4.17 APP.3.1.A17 Fehlerbehandlung....11
4.18 APP.3.1.A18 Kontrolle der Protokollierungsdaten....11
4.19 APP.3.1.A19 Schutz vor SQL-Injection....12
4.20 APP.3.1.A21 Sichere HTTP-Konfiguration bei Webanwendungen....12
4.21 APP.3.1.A22 Überprüfung von Webanwendungen....12
4.22 APP.3.1.A23 Verhinderung von Cross-Site Request Forgery....12

ABBILDUNGSVERZEICHNIS

Abbildung 1: Systemarchitektur....10

1 EINLEITUNG

Diese Sicherheitskonzeption beschreibt für das Projekt „Neuentwicklung der GDI-DE Testsuite“ vorhandene und umzusetzende Informationssicherheits-Maßnahmen und bewertet deren Fähigkeit zur Reduzierung von Bedrohungen.

Für die Sicherheitskonzeption wird festgelegt, dass der Systemkontext irrelevant ist und lediglich die zu implementierende Software hier Relevanz besitzt. Daher sind für das System in der Sicherheitskonzeption ausschließlich die Basis- und Standardanforderungen des IT-Grundschutz-Kompendiums für den Baustein App 3.1 Webanwendungen zu betrachten. Die konkreten Maßnahmen orientieren sich an den Handreichungen der folgenden Dokumente:

- IT-Grundschutz-Kompendium, Umsetzungshinweise:
  - https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/Umsetzungshinweise_Kompendium_CD_2019.pdf?__blob=publicationFile&v=10
- OWASP Top 10 – 2017: Die 10 kritischsten Sicherheitsrisiken für Webanwendungen:
  - https://www.owasp.org/images/9/90/OWASP_Top_10-2017_de_V1.0.pdf
- OWASP Top 10: Kritischer Blick auf die Charts:
  - https://www.heise.de/developer/artikel/OWASP-Top-10-Kritischer-Blick-auf-die-Charts-3953648.html?seite=all

Zu einzelnen Begrifflichkeiten und deren Definition sei auf das Glossar und die Begriffsdefinitionen im Komplex IT-Grundschutz des BSI verwiesen: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html.

Eine Risikoabschätzung entfällt an dieser Stelle, da das Sicherheitskonzept ausschließlich die Webanwendung abdeckt und der übergeordnete Systemkontext somit fehlt.

2 VORGABEN

Das System hat den Schutzbedarf "normal", gemäß der durch den Auftraggeber erfolgten Schutzbedarfsfeststellung. Es sind die entsprechenden Bausteine und Maßnahmen gem. zum Ausschreibungszeitpunkt aktuellem BSI IT-Grundschutz umzusetzen.

https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/bausteine/APP/APP_3_1_Webanwendungen.html

Relevant für die Webanwendung sind hierbei die Basis- und Standardanforderungen. Auf diese Anforderung wird in Kapitel 4 zu den Informationssicherheitsmaßnahmen genauer eingegangen.

3 BEDROHUNGEN

Webanwendungen unterliegen spezifischen Bedrohungen und Schwachstellen, die unter folgendem Link näher umschrieben sind:

https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/bausteine/APP/APP_3_1_Webanwendungen.html?nn=10134826#doc10095904bodyText5

4 INFORMATIONSSICHERHEITSMASSNAHMEN

Für die neu zu entwickelnde Webanwendung „GDI-DE Testsuite“ greifen Sicherheitsmaßnahmen, die hier im Kontext zu den Anforderungen aus Kapitel 2 Vorgaben erläutert sind. Umgesetzt werden die im folgenden erläuterten Maßnahmen größtenteils im Kontexts des „Security Layer“ (siehe Abbildung 1: Systemarchitektur).

4.1 APP.3.1.A1 Authentisierung bei Webanwendungen

Da der größte Teil der Webanwendung zur GDI-Testsuite nur von einem eingeschränkten Benutzerkreis genutzt werden darf, müssen sich registrierte Benutzer sowie Fach- und Systemadministratoren gegenüber der Anwendung authentisieren.

Beim Anlegen eines Passworts greift eine Passwort-Richtlinie, die den Benutzer zur Wahl eines sicheren Passworts zwingt. Die Passwortstärke wird dem Benutzer beim Anlegen angezeigt.

Bei der Registrierung legt der Nutzer ein Passwort an, das mithilfe der von Spring Security implementierten Bcrypt-Methode serverseitig verschlüsselt und danach in der Datenbank gespeichert wird. Damit ist sichergestellt, dass Passwörter nie im Klartext gespeichert werden. Die Bcrypt-Implementierung versieht den Hash außerdem automatisch mit einem Salt, einer zufällig generierten Zeichenfolge. Dieses Vorgehen erhöht die Entropie der Passwörter und verringert die Chance auf erfolgreiche Angriffe mit sogenannten Rainbow Tables.

Die Registrierung erfordert die Lösung eines Captchas zur Unterscheidung zwischen Mensch und Maschine, um massenhafte automatisierte Anmeldungen zu vermeiden. Außerdem ist ein neuer Account zunächst inaktiv und wird erst freigeschaltet, wenn die Anmeldung über einen per E-Mail versendeten Link bestätigt wird (Double Opt-In Verfahren). Die Integration eines Captchas zwecks Verhinderung automatisierter Massenaufrufe beim Schnelltest ist ebenso denkbar.

Fachadministratoren und Systemadministratoren müssen beim Login zwingend eine Multifaktor-Authentifizierung benutzen. Beim Login wird neben dem Passwort ein zusätzliches, nur einmal gültiges Token benötigt. Zur Generierung dieses Tokens wird eine Mobile App verwendet. Die Kommunikation zwischen Mobile App und Applikation läuft dabei über einen Time-Based One-Time Passwort Algorithmus (TOTP).

Versucht ein Benutzer sich mehrfach in kurzen Zeitabständen erfolglos an der Webanwendung anzumelden, so wird dies als Angriff gewertet. Wenn die Zahl der fehlgeschlagenen Versuche den festgelegten Wert von 5 Fehlversuchen überschreitet, so wird das Benutzerkonto für ein definiertes Zeitintervall von 10 Sekunden gesperrt.

Benutzer werden per E-Mail darüber informiert, sofern sich ihr Passwort geändert hat.

Authentifizierungsdaten werden nicht auf dem Client gespeichert.

Die Authentisierungskomponente ist ein zentraler Baustein der Webanwendung und wird über das Spring Security Framework implementiert. Die Authentisierung läuft dabei formularbasiert über TLS.

4.2 APP.3.1.A2 Zugriffskontrolle bei Webanwendungen

Es wird eine zentrale Sicherheitsschicht als Security-Layer implementiert. Der Zugriff von außen ist nur über einen Nginx-Container möglich, der über erreichbare HTTPS-Schnittstellen via Reverse Proxy Zugriff auf die einzelnen internen Komponenten ermöglicht, d. h. ein direkter Zugriff auf interne Endpunkte z. B. die Datenbank ist architektonisch nicht möglich. Jegliche Kommunikation erfolgt somit verschlüsselt über den HTTPS-Port 443 oder den SMTP-Port 25. Weiterhin realisiert der Security Layer die Absicherung der REST-Schnittstellen, die Authentifizierung von Nutzern und die Zugriffskontrolle als Umsetzung des Rechte- und Rollenkonzeptes. Zugriffe auf Ressourcen ohne erfolgreiche Zugriffskontrolle werden abgelehnt.

Das System unterscheidet zwischen vier Rollen, die Zugriff auf unterschiedliche Funktionen haben: Anonymer Benutzer (ROLE_ANONYMOUS), registrierter Benutzer (ROLE_USER), Fachadministrator (ROLE_TECHADMIN) und Systemadministrator (ROLE_SYSADMIN). Wird eine Funktion über eine Schnittstelle aufgerufen, wird zunächst geprüft, ob die Rolle des aktuellen Benutzers die notwendigen Privilegien besitzt, die aufgerufene Funktion auszuführen. Der Security Layer stellt somit sicher, dass Benutzer nur Aktionen ausführen können, zu denen sie berechtigt sind. Durch die Verknüpfung von Funktionen und den Endpoints der API-Schnittstellen wird sichergestellt, dass das System den Nutzern nur die jeweils für den Anwendungsfall notwendigen Daten zur Verfügung stellt (Minimal-Prinzip). Änderungen der spezifischen für einen Benutzer zugewiesenen Rollen können nur durch einen Systemadministrator durchgeführt werden. Das System sieht vor, dass Rollen sowohl hinzugefügt als auch entfernt werden können.

Für diese Aufgaben wird das Modul Spring Security 10 des Spring Frameworks 11 verwendet. Dies bietet Schutz gegen geläufige Angriffe wie zum Beispiel XSRF-, XSS- und Rainbow Table Angriffe.

4.3 APP.3.1.A3 Sicheres Session-Management

Session Management ist notwendig, da das zustandslose HTTP-Protokoll ansonsten keine Zuordnung einzelner HTTP-Requests zu konkreten Benutzern vornehmen kann. Das Session-Management erfolgt über das bewährte Spring Framework und durch den Apache Tomcat Webserver. Durch die entsprechende sichere Konfiguration der Komponenten wird ein Missbrauch von Session-IDs verhindert. Session-IDs unterliegen einem ausreichenden Schutz, da sie nicht in der URL, sondern im Post-body oder als Cookie im Header übergeben werden.

Die Webanwendung bietet dem Benutzer die Möglichkeit, sich explizit von der Anwendung abzumelden. Meldet sich ein Benutzer neu an, wird eine neue Session-ID erzeugt. Sitzungen unterliegen einer genügend kleinen maximalen Gültigkeitsdauer und werden bei Überschreitung dieser Dauer automatisch ungültig. Nachdem die Sitzung ungültig ist, werden alle Sitzungsdaten (sowohl server- als auch clientseitig) gelöscht. Session Cookies beim Client setzen die folgenden Flags: Path, Secure und HttpOnly, um den Zugriff clientseitig möglichst stark einzuschränken.

4.4 APP.3.1.A4 Kontrolliertes Einbinden von Daten und Inhalten bei Webanwendungen

Laden Benutzer lokale Daten in die Webanwendung (z.B. Testressourcen, Testklassen) hoch, so werden diese Daten in vorgegebene und nicht durch Benutzer zu beeinflussende Pfade kopiert. Hochgeladene Dateien werden nicht im Wurzelverzeichnis des Webserver-Dienstes gespeichert, sondern in fest definierten Verzeichnissen.

Die hochgeladenen Daten werden auf die Dateiendungen geprüft. Nur für die Anwendung relevante Dateien können erfolgreich hochgeladen werden. In der Regel wird für hochgeladene Dateien nur ein lesender Zugriff vorgesehen.

Die Webanwendung sieht keine notwendige Weiterleitung auf Ziele außerhalb der Anwendung vor.

4.5 APP.3.1.A5 Protokollierung sicherheitsrelevanter Ereignisse von Webanwendungen

Die Webanwendung protokolliert alle sicherheitsrelevanten Ereignisse und speichert sie serverseitig in einheitlichem Format, welches vom ELK-Stack (Elasticsearch, Logstash, Kibana) verarbeitet werden kann.

Der Zugriff auf die Protokolldaten ist reglementiert und nur der Systemadministrator hat den entsprechenden Zugriff auf die Daten.

4.6 APP.3.1.A6 Zeitnahes Einspielen sicherheitsrelevanter Patches und Updates

Patches und Updates werden aus vertrauenswürdigen Quellen bezogen. Diese Quellen sind Maven-Repositories (Etwa https://central.sonatype.org/, https://mvnrepository.com/repos/central & https://nexus.terrestris.de/) und npm-js (https://www.npmjs.com/).

In regelmäßigen Abständen werden zentrale Basis-Docker-Images neugebaut, um von publizierten Sicherheitspatches auf der Betriebssystem-Ebene zu partizipieren. Jene Image Änderungen werden regelmäßig in die Projekt-Sourcen (nach entsprechender Sichtung) überführt.

4.7 APP.3.1.A7 Schutz vor unerlaubter automatisierter Nutzung von Webanwendungen

Mittels des Spring Frameworks wird die Anwendung gegen automatisierte Angriffe geschützt. Nach 5 erfolglosen Anmeldeversuchen an der Anwendung wird das Benutzerkonto für 10 Sekunden gesperrt. Dies erschwert Brute-Force-Angriffe.

Die Registrierung an der Anwendung ist durch zusätzliche Eingabe eines Captchas abgesichert, um zwischen Mensch und Maschine unterscheiden zu können. Es existieren keine explizit zur automatischen Nutzung vorgesehenen Funktionen wie etwa RSS-Feeds o.ä.

4.8 APP.3.1.A8 Systemarchitektur einer Webanwendung

Die Systemarchitektur der Webanwendung verfolgt einen mehrschichtigen Ansatz. Die Architektur ist in folgender Abbildung ersichtlich:

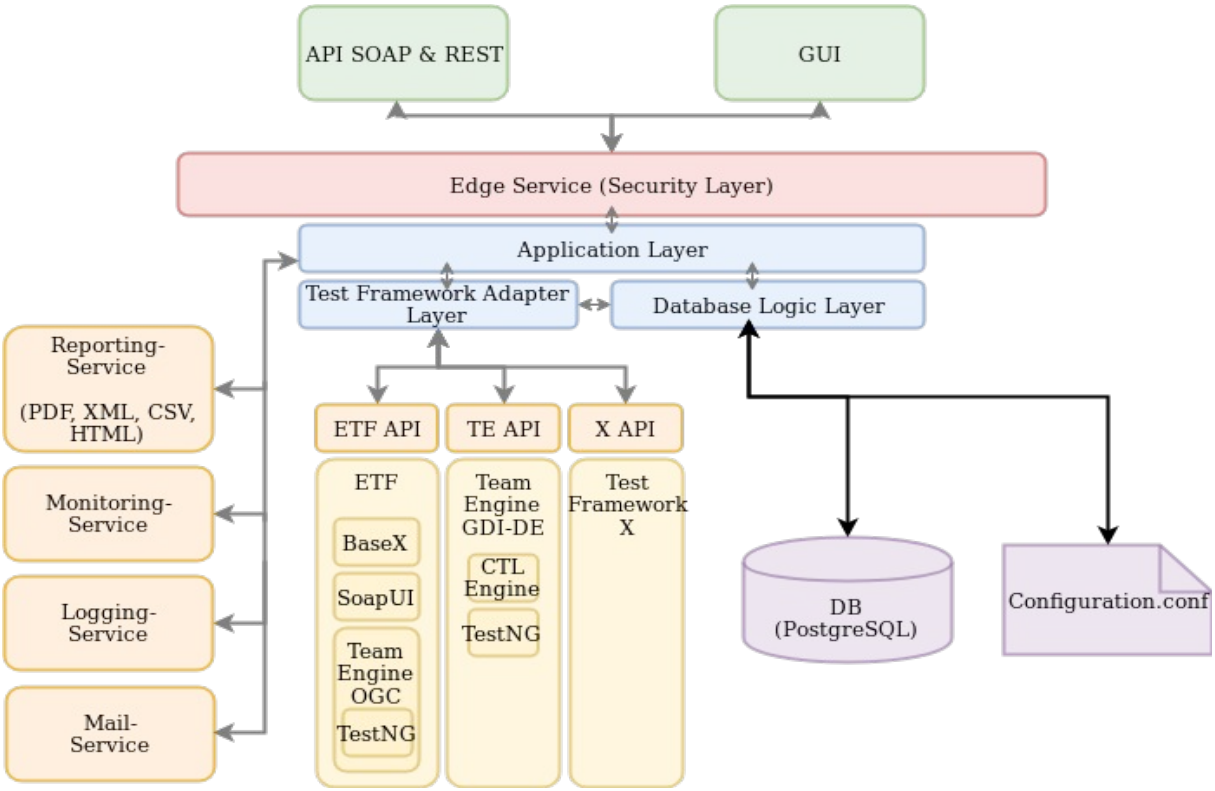

Abbildung 1: Systemarchitektur

Das System baut sich auf aus Webschicht (grün), Anwendungsschicht (blau/gelb) und Datenschicht (violett).

Durch die Dockerisierung der Komponenten wird ermöglicht, einzelne Benutzerkonten für die jeweiligen Serverprozesse zu nutzen.

Die in den Kubernetes-Cluster zu deployenden Docker-Container werden im Betriebshandbuch beschrieben.

4.9 APP.3.1.A9 Beschaffung, Entwicklung und Erweiterung von Webanwendungen

Die Webanwendung wird auf Basis von Open-Source-Software entwickelt. Es wird bewährte Standard-Software eingesetzt.

Die Anwendungsentwicklung erfolgt in getrennten Systemen für die Entwicklungs-, Test- und Produktivumgebung.

4.10 APP.3.1.A10 Test und Freigabe von Webanwendungen

Die abzunehmenden Testfälle werden auf Basis der Anwendungsfälle laut Leistungsbeschreibung umschrieben, im Betriebshandbuch dokumentiert und auf dieser Basis getestet.

4.11 APP.3.1.A11 Sichere Anbindung von Hintergrundsystemen

Die Webanwendung läuft innerhalb eines Kubernetes-Cluster, so dass die Anwendung selbst nicht für die sichere Anbindung an Hintergrundsysteme zuständig ist.

4.12 APP.3.1.A12 Sichere Konfiguration von Webanwendungen

Die sichere Konfiguration der Webanwendung wird durch das Spring Framework gewährleistet.

Die Webanwendung übermittelt Daten durchgehend in UTF-8-Kodierung.

Eine zentrale Funktion (Autorisierungsmethodik, s.o.) wird wie beschrieben bei der Überschreitung von gescheiterten Login-Versuchen die Möglichkeit des Logins für den konkret anfragenden Client unterbinden. Die Einbindung in ein Kubernetes-Cluster sorgt für den abgesicherten Zugriff auf die Anwendung von außen.

4.13 APP.3.1.A13 Restriktive Herausgabe sicherheitsrelevanter Informationen

Die Webanwendung wird so umgesetzt, dass sie nur neutrale Fehlermeldungen ausgibt und keine sicherheitsrelevanten Informationen enthält. Nicht benötigte Dateien werden regelmäßig gelöscht. Es wird nur mit relativen Pfadangaben gearbeitet.

Kommentare aus JavaScript-Quellcode (welche theoretisch sicherheitsrelevante Informationen enthalten könnten) werden automatisiert im Rahmen des build Prozesses entfernt und nicht an den Client ausgeliefert,

Konfigurationsdateien der Webanwendung werden außerhalb des Web-Root-Verzeichnisses gespeichert.

Eine konkrete und restriktive robots.txt (Instruktionsanweisung für Webcrawler, z.B. von Suchmaschinen) wird im Wurzelverzeichnis der Webresource verfügbar sein.

4.14 APP.3.1.A14 Schutz vertraulicher Daten

Vertrauliche Daten werden ausschließlich über die POST-Methode übertragen.

Clientseitige Ressourcen, die vertrauensvolle Daten enthalten, werden mit Caching verhindernden Headern ausgegeben.

Session Cookies verwenden Flags (siehe 4.3).

Konfigurationsdateien mit vertraulichen / schutzwürdigen Daten werden nicht im Webserver-Wurzelverzeichnis gespeichert.

4.15 APP.3.1.A15 Verifikation essenzieller Änderungen

Ändert sich das Passwort eines Benutzers, so wird dieser per E-Mail darüber informiert.

4.16 APP.3.1.A16 Umfassende Eingabevalidierung und Ausgabekodierung

Eingabedaten werden sowohl client- wie auch serverseitig validiert. Das Framework Hibernate unterbindet SQL Injektionen, Session-IDs werden über CSRF-Tokens validiert.

Jedwede Benutzereingabe, die in einen anderen Kontext überführt wird (HTML, E-Mail usw.), wird für den jeweiligen Zielkontext entsprechend enkodiert (Kontextsensitive Maskierung).

4.17 APP.3.1.A17 Fehlerbehandlung

Die Webanwendung wird so umgesetzt, dass ein sauberer Betrieb auch im Fehlerfall gewährleistet ist. Fehler werden bei Auftreten entsprechend abgefangen und behandelt sowie zur Nachvollziehbarkeit protokolliert.

4.18 APP.3.1.A18 Kontrolle der Protokollierungsdaten

Die Kontrolle der Protokolldateien obliegt dem Systemadministrator.

4.19 APP.3.1.A19 Schutz vor SQL-Injection

Das eingebundene Framework Hibernate unterbindet SQL Injektionen.

4.20 APP.3.1.A21 Sichere HTTP-Konfiguration bei Webanwendungen

Für die Webanwendung erfolgt eine sichere HTTP-Konfiguration im Apache Tomcat Webserver.

4.21 APP.3.1.A22 Überprüfung von Webanwendungen

Die Überprüfung der Webanwendung obliegt dem Betreiber.

4.22 APP.3.1.A23 Verhinderung von Cross-Site Request Forgery

Die Webanwendung nutzt das Modul Spring Security des Spring Frameworks. Session-Ids werden über CSRF-Tokens validiert. Die Anwendung ist somit geschützt gegen CSRF-Angriffe.