Informationssicherheit

Anwendungen

Regelung Nr. 03.04.02

Bereitstellen sicherer Applikationen

Klassifikation: Intern – KSU 2.1

Geltungsbereich: Die Regelungen gelten für die AUDI BRUSSELS

Version 1.0

Inhalt

I. Ziel

Ziel dieser Regelung ist die Festlegung von Sicherheitsanforderungen für Software und die Entwicklung von Applikationen.

Im Sinne dieser Regelung bezeichnet der Begriff Informationssicherheit die IT-Sicherheit als Bestandteil der ganzheitlichen Informationssicherheit.

1. Mindestanforderungen für Software

1.1. Ziel

Dieses Kapitel beschreibt Sicherheitsanforderungen, die im Konzern verwendete Software erfüllen muss. Die Anforderungen gelten sowohl für proprietär entwickelte Software als auch für erworbene Software.

1.2. Allgemeine Anforderungen

1.2.1 Softwareanforderungen

- Alle Sicherheitsanforderungen aus den jeweiligen Leitlinien, Regelungen und Sicherheitskonzepten müssen identifiziert werden.
- Alle identifizierten Anforderungen müssen dokumentiert werden.
- Die Software und aktuelle Sicherheits-Patches müssen mit dem jeweiligen Betriebssystem, auf dem diese installiert sind, kompatibel sein.
- Die Software muss den Härtungsvorgaben entsprechen[1].
- Die Software muss mit der standardmäßigen Anti-Malware-Software des Konzerns kompatibel sein.
- Die Software muss mit den standardmäßigen Verschlüsselungsprodukten des Konzerns kompatibel sein.
- Bei der Softwareentwicklung sollten Standard-Frameworks und Programmiersprachen in aktuellen Versionen verwendet werden.
- Es sollte ein sicherer Entwicklungsprozess (z. B. Microsoft SDL, OWASP SAMM, BSIMM, IT-PEP) verwendet werden.
- Sicherheitsaspekte müssen in allen Phasen des Entwicklungsprozesses berücksichtigt werden. ‘Security by Design’ ist im Entwicklungsprozess von Beginn an zu beachten.
- Datenschutzaspekte müssen in allen Phasen des Entwicklungsprozesses berücksichtigt werden. ‘Privacy by Design‘ und ‘Privacy by Default‘ sind in allen Phasen des m Entwicklungsprozesses sicherzustellen.
- Beim Erwerb von Software muss darauf geachtet werden, dass diese die Anforderungen des Kapitels Mindestanforderungen für Software erfüllt.
- Lizenzvereinbarungen, Eigentum des Codes und geistiges Eigentum müssen festgelegt sein.

[1] Siehe Informationssicherheit Regelung 03.01.01 Anti Malware & Systemschutz

1.2.2 Betriebliche Anforderungen

- Für jede Software muss in einer LifeCycle-Dokumentation festgelegt werden, welche Stellen zuständig sind für:
  - Implementierung/-bereitstellung der Software
  - Softwarekonfiguration und -Härtung
  - Betrieb der Software (inklusive des Schließens von Sicherheitslücken)
- Die oben erwähnten Stellen müssen über diese Zuständigkeit informiert werden.
- Die für die Softwareimplementierung zuständige Stelle muss die nötigen Aspekte zur Implementierung vorher dokumentieren.
- Die für das Schließen der Sicherheitslücken zuständige Stelle muss die nötigen Maßnahmen sofort oder bis zu dem Datum umsetzen, dass mit der zuständigen Stelle[2] vereinbart wurde.

1.2.3 Test

- Vor der Implementierung muss die Software auf die festgelegten Sicherheitsanforderungen hin getestet werden.
- Verwundbarkeiten, die beim Test entdeckt wurden, müssen geschlossen werden.
- Bekannte Verwundbarkeiten in verwendeten libraries oder frameworks müssen geschlossen werden bevor die Implementation stattfindet und/oder so bald möglich (in Abhängigkeit der Kritikalität und der Vereinbarung mit der verantwortlichen Stelle).
- Software muss einer Applikationssicherheitsprüfung, wie penetration tests oder security source code analysis unterzogen werden.
- Bei Kaufsoftware sollte der Hersteller ein aussagefähiges Ergebnis von penetration test und security source code analysis vorlegen.
- Es muss durch eine Risikoanalyse beurteilt werden, ob die Implementierung von neuer Software die Informationssicherheit des Konzerns oder der Gesellschaften beeinträchtigt.
- Negative Auswirkungen von Software auf die Datensicherheit müssen eliminiert werden.
- Tests und deren Ergebnisse sind detailliert zu dokumentieren.

1.2.4 Freigabe

- Vor der Implementierung muss die Software von der zuständigen Stelle freigegeben werden.
- Mindestens folgende Anforderungen müssen zur Freigabe der Software erfüllt sein:
  - Überprüfung, ob die Software auf die festgelegten Sicherheitsanforderungen hin getestet wurde
  - Überprüfung, ob während der Tests festgestellte Sicherheitslücken geschlossen wurden

[2] Siehe Anhang B.2.1

- Überprüfung, ob Anforderungen an die Datentrennung für die Verwendung durch unterschiedliche Kunden/Clients beachtet/bewertet wurden.
- Überprüfung, ob die implementierten Sicherheitsanforderungen für die Klassifizierungsstufen (Vertraulichkeit, Integrität, und Verfügbarkeit) der mit der Software zu verarbeitenden Daten angemessen sind

1.2.5 Code Signing

- Zertifikate zur Zertifizierung von Software müssen die Anforderungen für „Code Signing" erfüllen. Andere Zertifikate sind nicht zulässig. Zertifikate werden entsprechend x.509v3 definiert.
- Jede Konzerngesellschaft sollte eine certification authority benennen, die von den Architekturverantwortlichen autorisiert sein muss. Die Aufgaben der certification authority kann anderen Konzerngesellschaften übertragen werden.
- Das Ausstellen von code signing Zertifikaten wird durch die Volkswagen PKI durchgeführt. Das Beantragen von code signing Zertifikaten über öffentliche trustcenter wird von jeder Konzerngesellschaft selbst durchgeführt, kann aber anderen Konzerngesellschaften übertragen werden.

1.2.6 Dokumentation

- Die Software muss auf jedem System, auf dem sie installiert wird, eindeutig identifizierbar sein. Diese Information muss einfach zugänglich sein (CMDB-System).
- Die Software muss eindeutig benannt werden.
- Für jede Softwareinstanz muss die aktuell verwendete Version (inklusive Patch Level) dokumentiert werden.
- Für jede Softwareinstanz muss der aktuelle Anbieter der verwendeten Version (inklusive Kontaktinformationen) dokumentiert werden.
- Der Applikationseigentümer muss die Kritikalität der Daten inkl. Datenklassifikation klar definieren.
- Der Applikationseigentümer muss deutlich beschreiben, welche Daten gesichert werden müssen und wie diese wiederhergestellt werden können.
- Alle eingesetzten Produkte, Libraries oder Teile von Produkten und Frameworks sollten (bei Open Source: müssen), inklusive der eingesetzten Versionen und entsprechenden Lizenzen dokumentiert sein.

1.2.7 Sicherheitsaktualisierungen und Support

- Der Softwarehersteller oder die Stelle, die die Software entwickelt hat, müssen Aktualisierungen oder Support zur Schließung von bekannten Sicherheitslücken zur Verfügung stellen.
- Der Softwarehersteller oder die Stelle, die die Software entwickelt hat, müssen (entsprechend der Kritikalität) Benutzer über bekannte Sicherheitslücken informieren und auf Aktualisierungen zur Schließung der Sicherheitslücken verweisen.
- Für jede Softwareinstanz muss bekannt sein, ob für die Software ein Support vorhanden ist und ob Patches regelmäßig oder auf Anfrage durch den Kunden zur Verfügung gestellt werden.

1.3. Technische Anforderungen

- Für den Betrieb der Software dürfen keine sehr hohen Betriebssystemrechte benötigt werden.
- Für reguläre Benutzeraktivitäten dürfen keine speziellen Benutzerkonten erforderlich sein, deren Berechtigungen nicht konfiguriert werden können (z. B. Administratorkonto). Stattdessen muss bei regulären Benutzeraktivitäten die Verwendung von Benutzerkonten mit niedrigen Zugriffsrechten von der Software unterstützt werden.
- Die Software muss eine Benutzer-, Rechte- und Rollenverwaltung gemäß den funktionellen/betrieblichen Anforderungen zur Verfügung stellen.
- Die Software muss die Bereitstellung von personalisierten Benutzerkonten unterstützen.
- Die Benutzerverwaltung muss Authentifizierungsmechanismen gemäß den Sicherheitsanforderungen der verarbeiteten Daten zur Verfügung stellen[3][4][5].
- Benutzerkonten müssen deaktiviert werden können.
- Alle Softwarekomponenten müssen über eine Protokollierungsfunktion verfügen. Das Protokoll muss über standardisierte Softwareschnittstellen zur Verfügung gestellt werden.
- Bei Software, mit der Daten ausgetauscht werden können, muss Malwareschutz implementiert werden, oder es muss sichergestellt werden, dass die Daten vor dem Austausch durch Malwareschutz-Mechanismen bereinigt wurden.
- Die Software darf keine Internetverbindung ohne Interaktion des Benutzers herstellen. Dies beinhaltet z.B. "nach Hause telefonieren" für Lizenzprüfungen oder Softwareupdate-Anfragen.
- Die Software darf keine Dokumente oder andere Informationen über das Internet senden oder innerhalb des Internets (Cloud) speichern, es sei denn, dies ist eine explizite Geschäftsanforderung, der Benutzer wird zur Freigabe aufgefordert und es entspricht den Anforderungen des Sicherheitsregelwerks.
- Die Software darf keine Peer-to-Peer-Netzwerke erstellen.
- Es darf nicht möglich sein, Verbindungen aus dem Internet zum konzerneigenen Intranet unter Verwendung der Software zu initiieren und dadurch die Sicherheitsumgebung des Netzwerks zu umgehen.

[3] Siehe Anhang A.1.5

[4] Siehe Anhang A.1.2

[5] Siehe Anhang A.1.3

2. Sichere Entwicklung von Anwendungen

2.1. Ziel

Ziel dieses Kapitels ist, das Auftreten von Sicherheitslücken in Anwendungen zu verhindern, die durch das Design und die Implementierung bei der Softwareentwicklung für oder durch den Audi-Konzern entstehen.

2.2. Softwareentwicklungsprozess

Zur Entwicklung von sicheren Anwendungen müssen in allen Phasen des Entwicklungsprozesses die Sicherheitsaspekte beachtet werden. Sicherheit muss Teil der funktionalen und technischen Anforderungen einer Anwendung sein. Die funktionalen und technischen Anforderungen müssen vorher festgelegt werden. Im Anschluss müssen Sicherheitsanforderungen in der Analyse- und Gestaltungsphase modelliert werden. Es muss eine sichere Methode zur Entwicklung des Codes[6] angewandt werden, um die Entwicklung von sicheren Anwendungen zu gewährleisten. Die folgende Grafik gibt einen Überblick über ein sicheres Softwareentwicklungsverfahren.

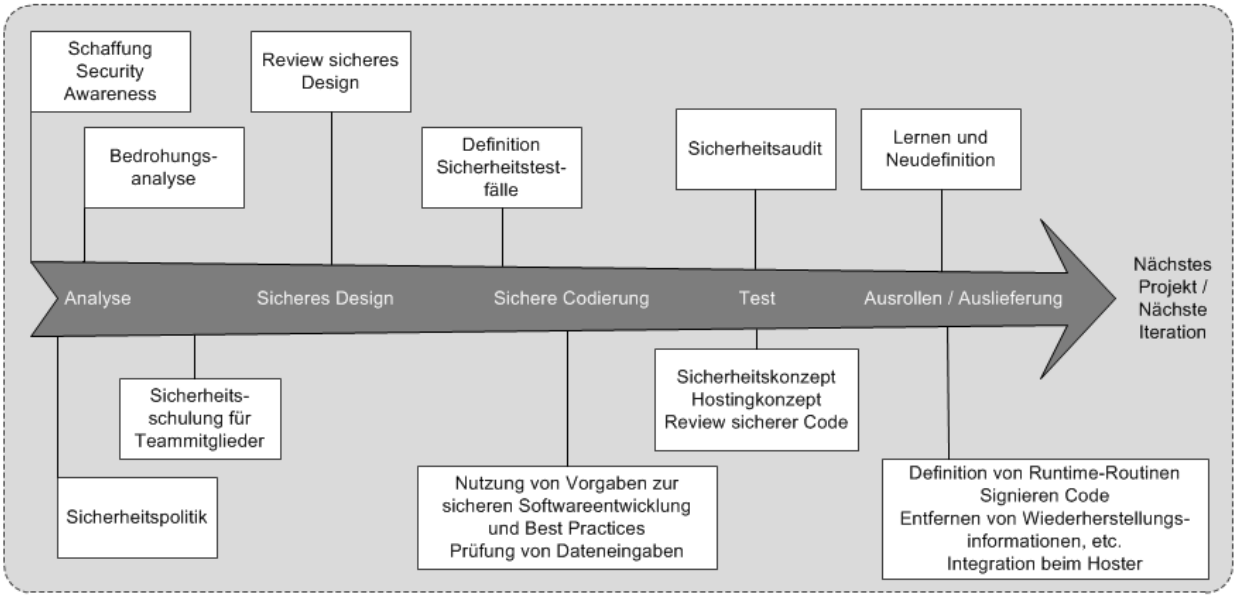

Abbildung1: Sicheres Softwareentwicklungsverfahren

Jedes Softwareentwicklungsverfahren, wie „Waterfall“, „Extreme Programming“ und ähnliches, enthält Aufgaben, die den in der Grafik dargestellten Phasen zugeordnet werden können. Für die unterschiedlichen Verfahren gibt es unterschiedliche Phasenbezeichnungen, und auch die Anzahl der Phasen kann variieren, die Funktionen bleiben jedoch gleich. Es muss für jede Phase eine verantwortliche Person oder Stelle festgelegt werden. Die folgenden Anforderungen müssen eingehalten werden:

[6] Wie z. B. durch OWASP festgelegt, siehe Anhang A.3

2.2.1 Analyse

- Eine Risikoanalyse zur Untersuchung von bestehenden Bedrohungen muss durchgeführt werden. Es müssen angemessene Sicherheitsanforderungen identifiziert werden.
- Alle Sicherheitsanforderungen aus den relevanten Regelungen und Sicherheitskonzepten,[7] sowie anwendungsspezifische Anforderungen müssen identifiziert werden.
- Alle identifizierten Anforderungen müssen dokumentiert werden.
- Alle Mitarbeiter, die am Softwareentwicklungsprojekt beteiligt sind, müssen bezüglich der Sicherheitsaspekte in ihrem Bereich geschult werden. Die Schulung muss eine angemessene Verwendung von Softwareentwicklungstools, Technologien, Frameworks und Programmiersprachen beinhalten.

2.2.2 Sicheres Design

- Ausgehend von den identifizierten Sicherheitsanforderungen muss eine sichere Architektur entwickelt werden. Detaillierte Sicherheitsmaßnahmen müssen unter Berücksichtigung der folgende Aspekte festgelegt werden:
  - Zugriffskontrolle durch geeignete Authentifizierungs- und Autorisierungsmethoden[8]
  - Input- und Output-Validierung, um Risiken durch falsche oder manipulierte Daten zu vermeiden
  - Protokollierung[9] von sicherheitsrelevanten Aktivitäten und Fehlern
  - Fehlerbehandlung
  - Sichere Kommunikation
  - Datensicherheit und Datenschutz
- Ein Sicherheitskonzept muss erstellt werden. Darin müssen die zu implementierenden Sicherheitsmaßnahmen beschrieben werden.
- Bei der Auswahl von Technologien, Softwareentwicklungstools, Frameworks und Programmiersprachen ist darauf zu achten, dass damit alle in der Analysephase[10] festgelegten Sicherheitsanforderungen eingehalten werden können.
- Die Sicherheit der Architektur muss durch eine Architekturprüfung kontrolliert werden. Die Architekturprüfung muss unter Verwendung des Vier-Augen-Prinzips von Entwicklern, die über das notwendige Wissen verfügen, durchgeführt werden. Es muss überprüft werden, ob der Architekturverantwortliche die in der Analysephase festgelegten Sicherheitsanforderungen[11] implementiert hat.
- Erkannte Sicherheitslücken müssen geschlossen werden. Ist dies nicht möglich, müssen Sicherheitslücken durch zusätzliche Sicherheitsmaßnahmen ausgeglichen

[7] Siehe Kapitel 1.2.1

[8] Siehe Anhang A.1.5

[9] Siehe Anhang A.1.6

[10] Siehe Kapitel 2.2.1

[11] Siehe Kapitel 2.2.1

werden (z. B. Leitfaden für Sicherheitsoptimierung, Benutzerhandbuch für sichere Anwendung).

2.2.3 Sichere Codierung

- Die Programmierung muss entsprechend der festgelegten Architektur und den entsprechenden Spezifikationen der Sicherheitsanforderungen erfolgen.
- Abweichungen von der Architektur und den festgelegten Sicherheitsanforderungen müssen von der zuständigen Stelle[12] freigegeben werden.
- Der Code muss entsprechend der Anforderungen aus Kapitel 1.2.3 und 2.2.4 überprüft werden und ist im Bedarfsfall zu korrigieren. Feststellungen müssen dokumentiert werden. Die Code-Prüfung muss unter Verwendung des Vier-Augen-Prinzips von Entwicklern, die über das notwendige Wissen verfügen, durchgeführt werden. Für Anwendungen, die sensible Daten verarbeiten, müssen spezielle Methoden/Standards zur Code-Prüfung verwendet werden (z. B. „Extreme Programming“).
- Die Code-Prüfung muss mindestens folgende Aspekte abdecken:
  - Maßnahmen zur Validierung von Input- und Output-Daten
  - Maßnahmen zur Sicherstellung der korrekten Verarbeitung von Daten in der Anwendung
  - Maßnahmen zur Sicherstellung von Authentizität und Schutz der Integrität von Nachrichten
- Code-Änderungen im Rahmen der Überprüfung müssen umfassend dokumentiert werden (z. B. durch ein Versionsverwaltungs-Tool).

2.2.4 Test

- Tests der Sicherheitsmaßnahmen müssen durchgeführt werden.
- Erkannte Sicherheitslücken müssen dokumentiert und geschlossen werden.
- Zur Überprüfung der Effektivität der Korrekturmaßnahmen ist eine weitere Prüfung durchzuführen.
- Im Falle von Designfehlern muss das Sicherheitskonzept auf Basis der Testergebnisse aktualisiert werden, bevor es produktiv betrieben wird.

2.2.5 Code Signing

Code Signing ermöglicht, dass signierte Software zur Laufzeit überprüft werden kann. Dies kann sicherstellen, dass nur überprüfter Programmcode für dedizierte Umgebungen verwendet wird.

- Programmcode sollte signiert werden und nach der Validierung einer Software erfolgen.
- Ausführbare Inhalte/Dateien und Scripte sollten digital signiert werden um sicherzustellen, dass der Programmcode nicht verändert wurde.
- Für code signing muss ein zentraler Dienst verwendet werden.

[12] Siehe Anhang B.1.1

2.2.6 Ausrollen/Auslieferung

- Ein Instandhaltungsplan muss ausgearbeitet werden.

2.3. Entwicklungsumgebung

Die Anforderungen der Handlungsleitlinien für Systementwickler[13] müssen beachtet werden. Im Speziellen sind dies:

- Trennung von Entwicklungs-, Test- und Produktionsumgebung
- Daten in Testumgebungen sollten bereinigt werden und sich von den produktiven Daten unterscheiden
- Zugriff auf Quellcode muss eingeschränkt sein
- Standardisierte Versionsverwaltung (z.B. SVN) muss für Änderungen und Quellcodereview implementiert werden.

2.4. Softwareentwicklung durch externe Dienstleister

Verträge müssen angemessene Sicherheitsanforderungen[14] beinhalten. Es müssen mindestens folgende Aspekte enthalten sein:

- Verpflichtung des externen Dienstleisters, die Anforderungen der Informationssicherheitsregelungen und die Regelungen für sichere Entwicklung des Auftraggebers einzuhalten.
- Verpflichtung des externen Dienstleisters, die Sicherheitsanforderungen zu erfüllen, die im Sicherheitskonzept und der Risikoanalyse festgelegt wurden.
- Vor Abnahme der Anwendung muss diese überprüft werden (z. B. durch Quellcode-Überprüfung, peer reviews) um sicherzustellen, dass keine versteckten Sicherheitslücken (z. B. Hintertüren) vorhanden sind. Das Ergebnis ist zu dokumentieren.
- Existieren hohe Schutzanforderungen für die durch die Anwendung zu verarbeitenden Daten (Vertraulichkeit = vertraulich, geheim; Integrität = hoch oder sehr hoch; Verfügbarkeit = hoch oder sehr hoch), muss vor Inbetriebnahme ein Penetrationstest der Anwendung durchgeführt werden. Kritische Schwachstellen müssen beseitigt oder abgeschwächt werden.
- Unabhängige Überprüfungen sollten hinsichtlich Schadsoftware (wie Computerviren, Würmer, Time Bombs, Backdoors, Trojaner, etc.) durchgeführt werden.
- Lizenzvereinbarungen, Eigentum des Codes und geistiges Eigentum

[13] Siehe Anhang A.1.3

[14] Eine Richtlinie findet sich in OWASP (https://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex)

II. Verantwortlichkeiten

II.I Kapitel 1: Mindestanforderungen für Software

Diese Richtlinie muss von allen Anwendungsentwicklern und den zuständigen Stellen für Softwareerwerb beachtet werden.

Abweichungen von dieser Richtlinie, die zu einem geringeren Sicherheitsniveau führen, sind zeitlich begrenzt möglich nach Absprache mit der IT-Sicherheit.

II.II Kapitel 2: Sichere Entwicklung von Anwendungen

Diese Richtlinie muss von allen Anwendungsentwicklern beachtet werden.

Abweichungen von dieser Richtlinie, die zu einem geringeren Sicherheitsniveau führen, sind zeitlich begrenzt möglich nach Absprache mit der IT-Sicherheit.

Anhang

A. Allgemeines

A.1 Mitgeltende Dokumente

A.1.1 Informationssicherheit Regelung Nr. 03.01.09 Ausnahmeprozess

A.1.2 Informationssicherheitshandlungsleitlinien für Systembetreiber und Administratoren

A.1.3 Informationssicherheitshandlungsleitlinien Systementwickler

A.1.4 Informationssicherheit Regelung Nr. 03.01.02 Kryptographie

A.1.5 Informationssicherheit Regelung Nr. 03.01.05 Authentifizierung und IAM

A.1.6 Informationssicherheit Regelung Nr. 03.01.04 Sicherheitsprotokollierung und -monitoring

A.1.7 Security Guidelines – https://group-wiki.wob.vw.vwg/wikis/display/SSCA2017/Security+Guidelines

A.1.8 Book of Standards (BoS)

A.2 Anlagen

A.2.1 Anlage 1 Feedback-Formular

Das Feedbackformular für Änderungsvorschläge zu Regelungen kann von der MyNet-Webseite Gesellschaften → Audi Brussels → Organisation → Finanz (B/F) → IT (B/FP) → IT-Sicherheit → Regelwerk heruntergeladen werden.

Das Feedback-Formular ist folgendermaßen auszufüllen:

Die Spalten 2 bis 6 müssen für jede vorgeschlagene Änderung ausgefüllt werden. Änderungen, für die nicht alle Spalten 2 bis 6 ausgefüllt sind, werden automatisch ohne weitere Prüfung abgelehnt.

Spalte 3: Folgende Kommentararten sind möglich: a=allgemein, f=fachlich, r=redaktionell

Spalte 4: Bitte bisherigen Text, Tabelle oder Bild einfügen

Spalte 5: Bitte vollständig umformulierten Text, geänderte Tabelle oder geändertes Bild einfügen

Spalte 6: Bitte begründen Sie die gewünschte Änderung möglichst ausführlich.

Bitte senden Sie das ausgefüllte Formular an: : it-security.audibx@audi.de

A.3 Quellen und Referenzen

- Open Web Application Security Project „Development Guide“ http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- Open Web Application Security Project „Development Guide“, Abschnitt „Data Validation“ http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- Open Web Application Security Project „Development Guide“, Abschnitt „Session Management“ und „Authentication“ http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- Open Web Application Security Project „Development Guide“, Abschnitt „Authorization“ http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- Open Web Application Security Project „Development Guide“, Abschnitt „Error Handling, Auditing and Logging“ http://www.owasp.org/index.php/Category:OWASP_Guide_Project

A.4 Abkürzungen und Definitionen

Begriff	Definition
OWASP	Open Web Application Security Project

A.5 Gültigkeit

Diese Regelung ist mit der Veröffentlichung sofort gültig.

Bereits bestehende Ausnahmeregelungen sind spätestens bei der nächsten Änderung (z.B. Verlängerung, Änderung der Auflagen) an diese Regelung anzupassen.

Nächstes Überprüfungsdatum: 06.08.2021

Bitte verwenden Sie das angegebene Formular für Änderungsanträge[15].

A.6 Dokumentenhistorie

Version	Name	Org. Einheit	Datum	Kommentar
1.0	Andreas Walter	B/FP	07.08.2019	Veröffentlicht

[15] Siehe Anhang A.2.1 Anlage 1 Feedback-Formular

B. Spezifische Ausprägungen

B.1 Kapitel 1: Mindestanforderungen für Software

B.1.1 IT-Sicherheit

B.2 Kapitel 2: Sichere Entwicklung von Anwendungen

B.2.1 CERT AUDI BRUSSELS und CERT AUDI AG