

Sicherheitsanforderung

OWASP Mobile Application Security Verification Standard (Android / iOS)

Deutsche Telekom Gruppe

Version	1.1
Datum	01.07.2023
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 4.1.1	Dokumententyp Sicherheitsanforderung
Version 1.1	Stand 01.07.2023	Status Freigegeben
Fachlicher Ansprechpartner Telekom Security psa.telekom.de	Gültigkeitsdauer 01.07.2023 - 30.06.2028	Freigegeben von Stefan Pütz, Leiter SEC-T-TST

Zusammenfassung

Android / iOS Anforderungen nach dem OWASP [Mobile Application Security Verification Standard](#) und OWASP [Mobile Application Security Testing Guide](#). Dieses Dokument enthält Anforderungen für Softwarearchitekten und -entwickler, die mobile Anwendungen sicher entwickeln möchten. Es dient als Industriestandard, um die Sicherheit mobiler Anwendungen zu überprüfen. Es klärt auch die Rolle von Softwareschutzmechanismen bei der mobilen Sicherheit und gibt Anforderungen zur Überprüfung ihrer Wirksamkeit. Das Dokument enthält auch spezifische Empfehlungen für das Sicherheitsniveau, das für verschiedene Anwendungsfälle empfohlen wird.

[MASVS Release 1.5.0](#), [MASTG Release 1.5.0](#)

Copyright © 2023 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.	Enterprise Appstore	5
2.2.	Standards	5
3.	Android / iOS Entwicklung	6
3.1.	Architecture, Design und Threat Modeling Anforderungen	6
3.2.	Data Storage Anforderungen	11
3.3.	Kryptographie Anforderungen	20
3.4.	Authentication und Session Management Anforderungen	24
3.5.	Network Communication Anforderungen	31
3.6.	Platform Interaction Anforderungen	35
3.7.	Code Quality und Build Setting Anforderungen	42
3.8.	Resilience Anforderungen	48

1. Einleitung

Dieses Dokument wurde auf Basis der Vorgaben aus den im Konzern gültigen Sicherheitsrichtlinien erstellt. Die Sicherheitsanforderung dient u.a. als Grundlage zur Freigabe im PSA-Prozess. Sie dient darüber hinaus als Umsetzungsempfehlung in 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.

2. Allgemein

2.1. Enterprise Appstore

Jenseits der in diesem Dokument beschriebenen Anforderungen, ist zu berücksichtigen, dass für die Distribution von Applikationen, die Daten der Schutzklasse ab „Intern“ verarbeiten und die Telekom intern verwendet werden, der Enterprise Appstore zu nutzen ist. Ausserdem muss bei Eigenentwicklungen solcher Apps das SDK für Container verwendet werden.

2.2. Standards

Dieses Anforderungsdokument ist in Form und Inhalt identisch mit den Anforderungen aus dem OWASP Mobile App-Sec Verification Standard ([MASVS](#)) und dem OWASP Mobile Security Testing Guide ([MSTG](#)). Damit orientiert sich das Dokument an einem internationalen Standard, um einerseits eine Vergleichbarkeit an Ergebnissen auch außerhalb der DTAG sicherzustellen und andererseits auch automatische, statische Tests mit frei verfügbaren Tools (wie z.B. mit [MobSF](#) - dem Mobile Security Framework) ermöglichen zu können.

3. Android / iOS Entwicklung

3.1. Architecture, Design und Threat Modeling Anforderungen

Req 1 Alle Komponenten der mobilen App sind identifiziert und für den Betrieb der App erforderlich.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Android und iOS bieten zur Entwicklung von Applikationen den Zugriff auf viele Standard Bibliotheken und APIs die direkt in den SDKs der Anbieter integriert sind. Während der Entwicklung einer Applikation sollte unbedingt sichergestellt werden, dass nur die Komponenten verlinkt werden, die auch für die Ausführung der Applikation benötigt werden, um die Angriffsfläche so gering wie möglich zu halten. So benötigt eine Taschenlampe App typischerweise keinen Zugriff auf Kamera Funktionen des Endgeräts.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-for-insecure-configuration-of-instant-apps-mstg-arch-1-mstg-arch-7>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-1/1.1

Req 2 Sicherheitsfunktionen sind niemals nur auf Client-Seite implementiert, sondern immer auch im entsprechenden entfernten API-Endpunkt.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die einseitige Abbildung oder Kontrolle von Sicherheitsfunktionen können Angreifer ausnutzen, um durch gezielte Manipulation von API Aufrufen weiterreichende Zugriffsrechte auf dem Backend zu erhalten.

Umsetzungsbeispiel: **Android:**

<https://mobile-security.gitbook.io/mobile-security-testing-guide/general-mobile-app-testing-guide/0x04h-testing-code-quality#injection-flaws-mstg-arch-2-and-mstg-platform-2>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-2/1.1

Req 3	Es existiert eine Architekturübersicht über die mobile App und alle verbundenen API-Endpunkte und Security wurde in der Gesamtarchitektur berücksichtigt.
-------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die mit der App verbundenen API Endpunkte sind in der Architektur dokumentiert und sicherheitstechnisch beschrieben. Dies kann mit Tools wie bspw. Swagger erreicht werden, um eine schnelle und detaillierte Übersicht sicherzustellen und damit Fehlern wie z.B. nicht authentisierten Zugriffen vorzubeugen und bereits in der Design Phase entsprechend zu begegnen.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-3/1.1

Req 4	Alle sensiblen Daten im Kontext der mobilen App wurden klar identifiziert.
-------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die von der App zu verarbeitenden und zu übertragenden Daten sind eindeutig nach den Schutzklassen des Konzerns in "öffentlich", "intern", "vertraulich" oder "TBS" (Top Business Secret) klassifiziert. Dies ist notwendig, um sicherzustellen, dass anhand der Requirements zu jedem Zeitpunkt die Daten der Schutzklasse entsprechend geschützt 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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-4/1.1

Req 5	Für jede Komponente der App ist der angebotene fachliche Funktionsumfang und/oder Sicherheitsfunktionen/-mechanismen klar definiert.
-------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die von der App verwendeten Komponenten werden fachlich für die Ausführung der App benötigt und die Sicherheitsfunktionen sind eindeutig beschrieben und festgelegt. Zu jedem Zeitpunkt ist somit sichergestellt, dass die

sicherheitstechnischen Abhängigkeiten bereits im Entwicklungsprozess berücksichtigt 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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-5/1.1

Req 6	Für die mobile App und die genutzten API-Endpunkte wurde eine Bedrohungsanalyse durchgeführt und potenzielle Bedrohungen und Gegenmaßnahmen identifiziert.
-------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Für die genutzten API Endpunkte wird bereits während der Entwicklungsphase geprüft, dass die Sicherheitsfunktionen wie geplant implementiert wurden. Für den Fall einer etwaigen, ungewollten Exponierung sind Gegenmassnahmen verfügbar, um auf die Bedrohung entsprechend zu reagieren.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-6/1.1

Req 7	Alle Sicherheitsfunktionen wurden in Form von zentralen Komponenten implementiert.
-------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Es gibt App Typen, wie z.B. Google Play Instant Apps, die ohne Installation ausführbar sind. Dabei verzichten diese Apps auf einen Großteil verfügbarer APIs und Permissions (u.a. wegen Größenbeschränkungen) was mit entsprechend geringeren Möglichkeiten zur sicherheitstechnischen Absicherung zur Laufzeit erreicht wird.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-for-insecure-configuration-of-instant-apps-mstg-arch-1-mstg-arch-7>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-7/1.1

Req 8	Eine dedizierte Richtlinie zum Management von kryptografischen Schlüsseln (falls in der App genutzt) beschreibt den sicheren Umgang mit Schlüsseln über den gesamten Lebenszyklus, idealerweise basierend auf Standards wie NIST SP 800-57.
-------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: In allen Fällen, die das Schlüsselmaterial betreffen, ist es notwendig, dass das Schlüssel Management ausschließlich auf Basis gängiger Grundsätze erfolgt. Hier kann es je nach Land und Unternehmen unterschiedliche Anforderungen geben, die zu berücksichtigen sind.

Idealerweise wird Schlüsselmaterial direkt auf dem Endgerät erzeugt und auf die Hinterlegung von privatem Schlüsselmaterial außerhalb des Endgeräts verzichtet. Es kann aber Fälle geben, in denen es nicht möglich ist das Schlüsselmaterial auf dem Endgerät zu erzeugen oder es gibt gesetzliche Verpflichtungen Schlüsselmaterial zentral zu hinterlegen. In beiden Fällen muss gewährleistet sein, dass das Schlüsselmaterial adäquat geschützt übermittelt und gespeichert wird, um Missbrauch durch Dritte zu verhindern.

Umsetzungsbeispiel: **Android:**

<https://mobile-security.gitbook.io/mobile-security-testing-guide/general-mobile-app-testing-guide/0x04g-testing-cryptography#cryptographic-policy-mstg-arch-8>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-8/1.1

Req 9	Es gibt einen Mechanismus in der mobilen App, um App-Aktualisierungen zu erzwingen.
-------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Apple Appstore und Google Playstore bieten zwar automatische Update Mechanismen an, diese können aber durch den Anwender verzögert oder deaktiviert werden. Um sicherzustellen, dass insbes. kritische Sicherheitsupdates schnellstmöglich ausgerollt werden müssen Apps über entsprechend eigene Mechanismen verfügen.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#testing-logs-for-sensitive-data-mstg-storage-3>

<https://developer.android.com/guide/playcore/in-app-updates>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#checking-logs-for-sensitive-data-mstg-storage-3>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-9/1.1

Req 10	Security wird in allen Teilen des Softwareentwicklungszyklus berücksichtigt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Sicherheitsanforderungen werden in allen Phasen der Entwicklung berücksichtigt, um kostspielige Verzögerungen oder kritische Sicherheitsprobleme aufgrund einer verspäteten Einführung der Sicherheitsanforderungen zu vermeiden.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-10/1.1

Req 11	Eine "Responsible Disclosure Policy" ist vorhanden und wird umgesetzt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Für Applikationen, die in den Appstores veröffentlicht werden, sollte sichergestellt sein, dass ein direkter Informationskanal für Responsible Disclosures existiert. Von Anwendern, Unternehmen oder Sicherheitsforschern entdeckte Sicherheitslücken können somit direkt gemeldet und im Austausch gelöst werden, um (kritische) Sicherheitslücken schnell und unkompliziert lösen zu 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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-11/1.1

Req 12	Die App sollte Datenschutzgesetze und Regulierungen befolgen.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Entwicklung der Applikation berücksichtigt die Anforderungen des Datenschutzes in jeder Phase der Entwicklung und wendet Sicherheitsfunktionen wie z.B. Verschlüsselung an, um den Anforderungen technisch zu entsprechen.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-12/1.1

3.2. Data Storage Anforderungen

Req 13	Die App verwendet die vom jeweiligen Betriebssystem angebotenen sicheren Speichermechanismen, um sensible Daten zu speichern, wie personenbezogene Daten, Anmeldedaten oder kryptografische Schlüssel.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Daten der Schutzklassen "intern" und "vertraulich" verwenden die von Android und iOS bereitgestellten Funktionen für Datenverschlüsselung und Key Management. So wird sichergestellt, dass keine Eigenentwicklungen zum Einsatz kommen die vermeintlich unsicherer sind und entsprechende Angriffsflächen bieten.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05d-Testing-Data-Storage.md#testing-local-storage-for-sensitive-data-mstg-storage-1-and-mstg-storage-2>

iOS:

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06d-Testing-Data-Storage.md#testing-local-data-storage-mstg-storage-1-and-mstg-storage-2>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-13/1.1

Req 14	Es werden keine sensiblen Daten außerhalb des App-Containers oder außerhalb des vom jeweiligen Betriebssystem angebotenen sicheren Speichermechanismus abgelegt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Speicherorte außerhalb des App-Containers können durch die App in den seltensten Fällen beeinflusst werden und sind häufig auch durch andere Applikationen im Zugriff, um Daten wie z.B. Bilder auszutauschen, so sind die Ordner für Fotos bspw. durch alle Apps lese- und schreibberechtigt. Auf Android Geräten kann es vorkommen, dass der Speicherort ein Wechselmedium darstellt und nicht von den Verschlüsselungsoptionen des Betriebssystems nachhaltig erfaßt wird.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#testing-local-storage-for-sensitive-data-mstg-storage-1-and-mstg-storage-2>

<https://developer.android.com/guide/playcore/in-app-updates>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#testing-local-data-storage-mstg-storage-1-and-mstg-storage-2>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-14/1.1

Req 15	Es werden keine sensiblen Daten in die Logfiles der App geschrieben.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Daten der Schutzklasse "vertraulich" oder personenbezogene Daten, wie z.B. Passwörter, Private Keys, etc. dürfen nicht in den Logs enthalten sein. Logs können typischerweise mit Dritten geteilt werden, bspw. für notwendige Analysezwecke, und es gilt auszuschließen, dass Mitarbeiter oder Dritte Zugriff auf Informationen des Anwenders erhalten.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#testing-logs-for-sensitive-data-mstg-storage-3>

<https://developer.android.com/guide/playcore/in-app-updates>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#checking-logs-for-sensitive-data-mstg-storage-3>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-15/1.1

Req 16	Es werden keine sensiblen Daten mit Dritten geteilt - es sei denn dies wurde in der App-Architektur definiert und ist zur Erfüllung des Zwecks der App erforderlich.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die App teilt keine Daten mit Dritten, es sei denn die App Architektur sieht das entsprechend vor und alle beteiligten Dienste wurden im Rahmen des PSA Verfahrens beschrieben, freigegeben und / oder entsprechende Schutzmaßnahmen zur Speicherung, Übertragung und ggf. Weiterverarbeitung der Daten sichergestellt.

In Apps eingebaute Funktionen, mit denen es möglich ist Daten mit Dritten zu teilen, was typischerweise der Einwilligung des Anwenders und aktiven Nutzung eines anderen Dienstes vorausgeht, soll sichergestellt werden, dass die Schnittstellen sicher bedient werden und von Diensten wie z.B. Crashlytics, Embrace oder Sentry erfasste Daten weder unaufgefordert mit Dritten geteilt noch die Daten ohne Schutzmaßnahmen ausgeleitet werden.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#determining-whether-sensitive-data-is-shared-with-third-parties-mstg-storage-4>

<https://developer.android.com/guide/playcore/in-app-updates>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#determining-whether-sensitive-data-is-shared-with-third-parties-mstg-storage-4>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-16/1.1

Req 17	Der Tastatur-Cache ist für alle sensiblen Texteingaben deaktiviert.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Felder, in denen schutzbedürftige Informationen eingegeben werden, müssen Keyboard Caching deaktivieren. Mit dieser Maßnahme wird verhindert, dass schutzbedürftige Daten potentiell unsicher abgelegt und weiterverarbeitet werden. Schutzbedürftige Eingabefelder sind z.B. persönliche Informationen (z.B. Geburtstag), Bankdaten (z.B. Konto No., Kreditkartendaten) oder Passwörter.

Umsetzungsbeispiel: **Android:**

```
<EditText  
android:id="@+id/KeyboardCache"  
android:inputType="textNoSuggestions" />
```

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#determining-whether-the-keyboard-cache-is-disabled-for-text-input-fields-mstg-storage-5>

iOS:

```
textObject.autocorrectionType = UITextAutocorrectionTypeNo;  
textObject.secureTextEntry = YES;
```

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#finding-sensitive-data-in-the-keyboard-cache-mstg-storage-5>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-17/1.1

Req 18	Es werden keine sensiblen Daten über Interprozesskommunikation zur Verfügung gestellt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Android bietet mit Provider, Intents und Permissions ein sehr offenes Konzept für die Inter Process Communication (IPC), das über "Signature" auf Austausch von Daten nur durch den gleichen Entwickler eingeschränkt werden kann während iOS mittels NSMachPort wesentlich eingeschränktere Möglichkeiten zur Verfügung stellt. Soll auf Interaktion durch den Anwender verzichtet werden, also kein Share Sheet oder ähnliches zum Einsatz kommen, dann kann auf die IPC Möglichkeiten zurückgegriffen werden.

Da der Austausch von Daten typischerweise ohne Interaktion mit dem Anwender stattfindet, sollte die IPC ausschließlich mit Apps des gleichen Entwicklers erfolgen und keine sensiblen Daten der App oder des Anwenders exponieren, um sicherzustellen, dass Apps Dritter definitiv keinen Zugriff auf die Daten erreichen können.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#determining-whether-sensitive-stored-data-has-been-exposed-via-ipc-mechanisms-mstg-storage-6>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#determining-whether-sensitive-data-is-exposed-via-ipc-mechanisms-mstg-storage-6>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-18/1.1

Req 19 Es werden keine sensiblen Daten wie Passwörter oder Pins über die Benutzeroberfläche exponiert.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Eingabe von sensiblen Daten wie z.B. Passwörtern erfolgt verschleiert, um zu verhindern, dass Dritte durch Shoulder Surfing Einsicht in die Daten bekommen. Im besten Fall wird auf die Nutzung von Passwörtern verzichtet und vorrangig alternative Anmeldungsmethoden zu verwenden wie z.B. FIDO2, 3rd Party Identity Provider (Apple, Google, Verimi, etc.) oder ähnlich sichere Passwordless Methoden.

Umsetzungsbeispiel: Android:

```
android:inputType="textPassword"
```

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#checking-for-sensitive-data-disclosure-through-the-user-interface-mstg-storage-7>

iOS:

```
sensitiveTextField.isSecureTextEntry = true
```

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#checking-for-sensitive-data-disclosed-through-the-user-interface-mstg-storage-7>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-19/1.1

Req 20 Es ist sichergestellt, dass betriebssystemgesteuerte Backups keine sensiblen App-Daten enthalten.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Android und iOS bieten für die Geräte unterschiedlich weitreichende Möglichkeiten um die Daten eines Geräts in der Cloud, auf Wechselmedien oder einem anderen Rechner abzuspeichern. In allen Fällen kann dabei jedoch sichergestellt werden, dass die Daten entweder erst gar nicht außerhalb des Geräts abgelegt oder gespeichert werden können.

Die Anforderung soll dabei nicht verhindern, dass eine App Daten außerhalb des Geräts lesen und schreiben darf, sondern soll sicherstellen, dass nicht ungewollt bspw. Private Keys, vertrauliche Konfigurationen, etc. ohne zusätzlichen Schutz in das Backup aufgenommen werden. Sensible und vertrauliche Elemente sollten ausschließlich von den sicheren Speicherungs- und Ablagemöglichkeiten der Betriebssysteme Gebrauch machen (Android Key Attestation, Apple Keychain) während sensible und vertrauliche Dateien mit den Möglichkeiten des Betriebssystems zusätzlich verschlüsselt werden.

Umsetzungsbeispiel: **Android**:

`android:allowBackup="false"`

<https://developer.android.com/training/articles/security-key-attestation>

<https://source.android.com/security/encryption/file-based>

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#testing-backups-for-sensitive-data-mstg-storage-8>

iOS:

`let kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly: CFString`

<https://developer.apple.com/documentation/security/ksecattraccessiblewhenpasscodesetthisdeviceonly>

https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/encrypting_your_app_s_files

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#testing-backups-for-sensitive-data-mstg-storage-8>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-20/1.1

Req 21	Die App entfernt sensible Daten aus der aktuellen Ansicht, wenn der Hintergrundmodus aktiviert wird.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Wird die App in den Hintergrundmodus versetzt wird u.a. auch ein Vorschaubild (Snapshot) erzeugt, um die Auswahl durch den Anwender für den Vordergrundmodus zu verschönern und zu vereinfachen. Dabei speichern die Betriebssysteme das Vorschaubild in einem Ordner auf dem Endgerät wobei auch sensible Daten enthalten sein können die ohne zusätzlichen Schutz bspw. wieder in Backups enthalten sein können.

Umsetzungsbeispiel: **Android:**

https://developer.android.com/reference/android/view/Display#FLAG_SECURE

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#finding-sensitive-information-in-auto-generated-screenshots-mstg-storage-9>

iOS:

`private var backgroundImage: UIImageView?`

```
func applicationDidEnterBackground(_ application: UIApplication) {
    let myBanner = UIImageView(image: #imageLiteral(resourceName:
"overlayImage"))
    myBanner.frame = UIScreen.main.bounds
    backgroundImage = myBanner
    window?.addSubview(myBanner)
}
```

```
func applicationWillEnterForeground(_ application: UIApplication) {
```

```
        backgroundImage?.removeFromSuperview()  
    }
```

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#testing-auto-generated-screenshots-for-sensitive-information-mstg-storage-9>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-21/1.1

Req 22	Die App hält sensible Daten nur solange wie nötig im Speicher und betroffene Speicherbereiche werden nach Nutzung explizit gelöscht.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Trotz aller Anstrengungen durch Apple, Google und App Entwickler können auch durch regelmäßige Updates Programmierfehler nicht vermieden werden. Dabei kommen immer neuere Methoden zum Einsatz, um auf Daten der Betriebssysteme oder anderer installierter Apps anzugreifen. Je geringer die Angriffsfläche gehalten wird, also je weniger Daten auf dem Endgerät vorgehalten werden, desto unwahrscheinlicher ist es, dass Angreifer Zugriff auf größere Datenmengen erhalten.

Idealerweise werden Daten dabei nur im RAM gecached und nach deren Verarbeitung die entsprechenden Speicherbereiche wieder gelöscht. Je nach Applikation muss ggf. auch mit Abstufungen gearbeitet werden, d.h. bspw. auch Caching oder permanente Speicherung im Filesystem (z.B. nach Eingaben und gleichzeitigem Verlust der Netzwerkverbindung, Netzwechsel zw. WiFi <=> Mobilfunk, etc.), wobei Methoden der sicheren und verschlüsselten Speicherung zum Einsatz kommen sollten.

Umsetzungsbeispiel: Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05d-Testing-Data-Storage.md#testing-memory-for-sensitive-data-mstg-storage-10>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06d-Testing-Data-Storage.md#testing-memory-for-sensitive-data-mstg-storage-10>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-22/1.1

Req 23	Die App erzwingt ein Minimum an Geräteschutz-Richtlinien wie das Definieren eines Gerätepassworts.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Außerhalb vom Unternehmen kann nicht garantiert werden, dass Endgeräte über einen PIN- oder Passwortschutz verfügen. Ohne einen entsprechenden Schutz können die in einer App abgelegten Daten durch OS Security Features nicht angemessen geschützt werden, da entweder die entsprechenden APIs und / oder Funktionen bspw. zur verschlüsselten Speicherung nicht sicher zur Verfügung stehen. Zwar könnte die App durch ein individuelles Passwort geschützt werden, aber die Implementierung eigener Schutz- und Verschlüsselungsmaßnahmen ist viel zu aufwändig und aus sicherheitstechnischer Sicht auf gar keinen Fall zu empfehlen.

Aus dem Grund bieten Apple und Google Möglichkeiten zu prüfen, ob ein entsprechender Geräteschutz gesetzt ist. Ist das nicht der Fall und die App ist aus unterschiedlichen Gründen darauf angewiesen, kann eine entsprechende Reaktion erfolgen und der Anwender aufgefordert werden entsprechende Schutzmechanismen zu aktivieren.

Umsetzungsbeispiel: Android:

<https://developer.android.com/reference/android/provider/Settings.Secure.html>

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05f-Testing-Local-Authentication.md#testing-confirm-credentials-mstg-auth-1-and-mstg-storage-11>

iOS:

<https://developer.apple.com/documentation/localauthentication/lapolicy/deviceownerauthentication>

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06f-Testing-Local-Authentication.md#testing-local-authentication-mstg-auth-8-and-mstg-storage-11>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-23/1.1

Req 24	Die App klärt den Nutzer über die Art und Weise der verarbeiteten personenbezogenen Daten auf und gibt dem Nutzer Security-Best-Practice-Empfehlungen zum Umgang mit der App.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Der Nutzer kann darüber entscheiden, ob und wie er die App im Rahmen der Erklärung nutzen möchte.

Umsetzungsbeispiel: **Allgemein**:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04i-Testing-User-Privacy-Protection.md#testing-user-education-mstg-storage-12>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-24/1.1

Req 25	Sensible Daten sollten nicht lokal auf dem mobilen Gerät gespeichert werden. Daten sollten stattdessen direkt vom API-Endpunkt abgerufen und nur im Arbeitsspeicher vorgehalten werden.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Trotz aller Anstrengungen durch Apple, Google und App Entwickler können auch durch regelmäßige Updates Programmierfehler nicht vermieden werden. Dabei kommen immer neuere Methoden zum Einsatz, um auf Daten der Betriebssysteme oder anderer installierter Apps zuzugreifen. Je geringer die Angriffsfläche gehalten wird, also je weniger Daten auf dem Endgerät vorgehalten werden, desto unwahrscheinlicher ist es, dass Angreifer Zugriff auf größere Datenmengen erhalten.

Idealerweise werden Daten dabei nur im RAM gecached und nach deren Verarbeitung die entsprechenden Speicherbereiche wieder gelöscht. Je nach Applikation muss ggf. auch mit Abstufungen gearbeitet werden, d.h. bspw. auch Caching oder permanente Speicherung im Filesystem (z.B. nach Eingaben und gleichzeitigem Verlust der Netzwerkverbindung, Netzwechsel zw. WiFi <=> Mobilfunk, etc.), wobei Methoden der sicheren und verschlüsselten Speicherung zum Einsatz kommen sollten.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-25/1.1

Req 26	Falls doch sensible Daten lokal vorgehalten werden müssen, sollten diese mit einem Schlüssel verschlüsselt werden, der im vom jeweiligen Betriebssystem angebotenen sicheren Speichermechanismus abgelegt ist. Der Zugriff auf den Schlüssel sollte Authentifizierung erfordern.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Sensible und vertrauliche Elemente (das sind typischerweise kleine Datenmengen wie z.B. Zugangsdaten, Keys, Tokens, etc.) sollten ausschließlich von den sicheren Speicherungs- und Ablagemöglichkeiten der Betriebssysteme Gebrauch machen (Android Key Attestation, Apple Keychain), während sensible und vertrauliche Dateien (das sind typischerweise größere Datenmengen wie z.B. Datenbanken) mit den Möglichkeiten des Betriebssystems zusätzlich verschlüsselt werden. Um sicherzustellen, dass der Zugriff nur durch einen berechtigten Anwender erfolgt wird bei Zugriff eine Authentisierung verlangt.

Umsetzungsbeispiel: **Android:**

<https://developer.android.com/training/articles/security-key-attestation>
<https://source.android.com/security/encryption/file-based>

iOS:

let kSecAttrAccessibleWhenUnlockedThisDeviceOnly: CFString

<https://developer.apple.com/documentation/security/ksecattraccessiblewhenunlockedthisdeviceonly>
https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/encrypting_your_app_s_files

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-26/1.1

Req 27	Der lokale Speicher von der App sollte gelöscht werden, nach Erreichen einer exzessiven Anzahl von fehlgeschlagenen Login-Versuchen.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Um sicherzustellen, dass ein Angreifer unter Verwendung von Brute-force Methoden keinen Zugriff auf die sensiblen oder vertraulichen Daten der Applikation erhält, sollte nach spätestens drei Anmeldeversuchen der Zeitraum zwischen zwei Anmeldeversuchen vergrößert werden (Tarpit). Nach dem siebten, fehlgeschlagenen Anmeldeversuch sollte der Anwender darauf hingewiesen werden, dass die App nach weiteren drei Versuchen in den Auslieferungszustand zurückgesetzt wird und dabei alle lokalen Daten gelöscht werden.

Die Option sollte direkt im Einrichtungsdialog (z.B. beim Setzen der App PIN) angesprochen werden, dem User kann ein Opt-Out angeboten werden, wenn er gleichzeitig über die Risiken aufgeklärt wird.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-27/1.1

3.3. Kryptographie Anforderungen

Req 28	Verschlüsselung in der App basiert nicht nur auf symmetrischer Kryptographie mit hart-codierten Schlüsseln.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die App stellt für die symmetrische Verschlüsselung immer einen Hardware- und Anwender Bezug her, um sicherzustellen, dass die verschlüsselten Daten generell nicht ohne den Hardware und Anwender bezogenen Schlüssel entschlüsselt werden können. In die App eingebautes Schlüsselmaterial (hardcoded keys, Passwörter) ist für Entwicklungen in keinem Fall gestattet, da Angreifer spätestens nach der Veröffentlichung in den Appstores Rückschlüsse auf dessen Verwendung ziehen könnten.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04g-Testing-Cryptography.md#common-configuration-issues-mstg-crypto-1-mstg-crypto-2-and-mstg-crypto-3>

Android:

<https://developer.android.com/training/articles/security-key-attestation>
<https://source.android.com/security/encryption/file-based>
<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05e-Testing-Cryptography.md#testing-symmetric-cryptography-mstg-crypto-1>

iOS:

<https://developer.apple.com/documentation/security/ksecattraccessiblewhenunlockedthisdeviceonly>
https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/encrypting_your_app_s_files
<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06e-Testing-Cryptography.md#testing-key-management-mstg-crypto-1-and-mstg-crypto-5>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-28/1.1

Req 29	Die App nutzt bewährte Implementierungen zur Umsetzung kryptografischer Primitive.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Entwicklung, Implementierung und Pflege sicherer Crypto Funktionen und Bibliotheken ist nicht nur komplex und zeitaufwändig, sondern sollte vor allem einem breiten Fachpublikum zur Bewertung vorliegen, um rechtzeitig Schwachstellen zu identifizieren. Aus diesem Grund sollten für die App Entwicklung ausschließlich die von Apple und Google bereitgestellten Funktionen und Bibliotheken verwendet werden. Diese werden seit Jahren weiterentwickelt und immer wieder von Sicherheitsforschern auf Schwachstellen überprüft und getestet.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04g-Testing-Cryptography.md#common-configuration-issues-mstg-crypto-1-mstg-crypto-2-and-mstg-crypto-3>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05e-Testing-Cryptography.md#testing-the-configuration-of-cryptographic-standard-algorithms-mstg-crypto-2-mstg-crypto-3-and-mstg-crypto-4>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06e-Testing-Cryptography.md#verifying-the-configuration-of-cryptographic-standard-algorithms-mstg-crypto-2-and-mstg-crypto-3>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-29/1.1

Req 30	Die App nutzt kryptografische Primitive passend zum spezifischen Anwendungsfall, konfiguriert nach Best-Practice Vorgaben dem Stand der Technik entsprechend.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Je nach Anwendungsfall kommen die dafür passenden kryptografischen Methoden, Funktionen, Bibliotheken, Chiffren und Algorithmen unter Verwendung zeitgemäßer Security Best Practices zum Einsatz. Nach dem Salting eines Passworts wird z.B. SHA-2 384 verwendet, um den Hash zu erstellen während MD5 und SHA-1 als unsicher gelten und nicht verwendet werden dürfen.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04g-Testing-Cryptography.md#common-configuration-issues-mstg-crypto-1-mstg-crypto-2-and-mstg-crypto-3>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05e-Testing-Cryptography.md#testing-the-configuration-of-cryptographic-standard-algorithms-mstg-crypto-2-mstg-crypto-3-and-mstg-crypto-4>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06e-Testing-Cryptography.md#verifying-the-configuration-of-cryptographic-standard-algorithms-mstg-crypto-2-and-mstg-crypto-3>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-30/1.1

Req 31	Die App nutzt keine kryptografischen Protokolle oder Algorithmen die allgemein als veraltet und unsicher gelten.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Veraltete oder als unsicher geltende Methoden, Funktionen, Bibliotheken, Chiffren und Algorithmen dürfen nicht verwendet werden, da Angreifer die bereits veröffentlichten Lücken ausnutzen, um Zugriff auf die vermeintlich sicher verschlüsselten Daten oder Konten zu erhalten. Nach dem Salting eines Passworts wird z.B. SHA-2 384 verwendet, um den Hash zu erstellen während MD5 und SHA-1 als unsicher gelten und nicht verwendet werden dürfen.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04g-Testing-Cryptography.md#identifying-insecure-andor-deprecated-cryptographic-algorithms-mstg-crypto-4>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05e-Testing-Cryptography.md#testing-the-configuration-of-cryptographic-standard-algorithms-mstg-crypto-2-mstg-crypto-3-and-mstg-crypto-4>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-31/1.1

Req 32	Die App verwendet einen kryptografischen Schlüssel für genau einen Zweck und nicht für mehrere Zwecke.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Wiederverwendung kryptografischer Schlüssel ist genauso wie die Wiederverwendung von Passwörtern zu werten. Nach erfolgreicher Kompromittierung sind alle mit dem Schlüssel gesicherten Daten exponiert. Bei entsprechender Zweckbindung des Schlüsselmaterials wird das Risiko der Kompromittierung aller Daten entsprechend reduziert.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05e-Testing-Cryptography.md#testing-the-purposes-of-keys-mstg-crypto-5>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06e-Testing-Cryptography.md#testing-key-management-mstg-crypto-1-and-mstg-crypto-5>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-32/1.1

Req 33	Alle Zufallswerte werden über einen ausreichend sicheren kryptografischen Zufallszahlengenerator erzeugt.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Für die Erzeugung kryptografisch sicherer Pseudozufallszahlen müssen die durch Android und iOS zur Verfügung gestellten Funktionen verwendet werden. Eigene Implementierungen oder die Erzeugung einfacher Zufallszahlen ist kryptografisch unsicher und deshalb nicht erlaubt.

Umsetzungsbeispiel: **Android:**

```
SecureRandom random = new SecureRandom();  
byte bytes = new byte[20];  
random.nextBytes(bytes);
```

<https://developer.android.com/reference/java/security/SecureRandom.html>

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05e-Testing-Cryptography.md#testing-random-number-generation-mstg-crypto-6>

iOS:

```
func SecRandomCopyBytes(  
_ rnd: SecRandomRef?,  
_ count: Int,  
_ bytes: UnsafeMutableRawPointer  
) -> Int32
```

<https://developer.apple.com/documentation/security/1399291-secrandomcopybytes>

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06e-Testing-Cryptography.md#testing-random-number-generation-mstg-crypto-6>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-33/1.1

3.4. Authentication und Session Management Anforderungen

Req 34	Falls die App Nutzern Zugriff auf entfernte Service APIs bietet, wird am API-Endpunkt eine Authentifizierung z.B. mit Nutzernamen/Passwort durchgeführt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Um zu verhindern, dass Daten ohne Berechtigung ausgelesen werden, wird jeder Remote Endpunkt authentisiert und autorisiert, eine anonyme Nutzung ist nur dann gestattet, wenn der Dienst für diesen Zweck ausgelegt ist und bei Zugriff auf Daten der Schutzklasse ab "intern" eine entsprechende Authentisierung und Autorisierung durchführt.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#verifying-that-appropriate-authentication-is-in-place-mstg-arch-2-and-mstg-auth-1>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05f-Testing-Local-Authentication.md#testing-confirm-credentials-mstg-auth-1-and-mstg-storage-11>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-34/1.1

Req 35	Kommt Session-Management am API-Endpunkt zum Einsatz, so werden zufällig generierte Session-IDs erzeugt, um Client-Anfragen zu authentifizieren und keine Nutzer-Anmeldedaten versandt.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Um zu verhindern, dass User Daten durch Dritte kompromittiert werden können, werden Methoden verwendet, mit denen das Risiko einer Kompromittierung entsprechend reduziert wird. Dabei wird sich u.a. zufällig generierter Session IDs bedient.

Umsetzungsbeispiel: **Allgemein:**

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#session-attacks-detection

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-stateful-session-management-mstg-auth-2>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-35/1.1

Req 36 Kommt statuslose Token-basierte Authentifizierung zum Einsatz, so werden die Token am Server mit einem sicheren Algorithmus signiert.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Um sicherzustellen, dass erstellte Tokens nicht durch Dritte manipuliert werden, werden die Token bei Ausstellung kryptografisch sicher signiert und stets durch den Server verifiziert.

Umsetzungsbeispiel: **Allgemein:**

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#session-attacks-detection

https://cheatsheetseries.owasp.org/cheatsheets/JSON_Web_Token_for_Java_Cheat_Sheet.html

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-stateless-token-based-authentication-mstg-auth-3>

Android:

<https://developers.google.com/identity/protocols/oauth2/native-app?hl=en>

iOS:

https://developer.apple.com/documentation/sign_in_with_apple/sign_in_with_apple_rest_api/authenticating_users_with_sign_in_with_apple

https://developer.apple.com/documentation/sign_in_with_apple/sign_in_with_apple_js/incorporating_sign_in_with_apple_into_other_platforms

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-36/1.1

Req 37 Der API-Endpunkt beendet die existierende Nutzersitzung, sobald sich der Nutzer abmeldet.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Viele mobile Anwendungen melden die Benutzer nicht automatisch ab. Dafür kann es verschiedene Gründe geben, z. B. weil es für die Kunden unbequem ist oder aufgrund von Entscheidungen, die bei der Implementierung der zustandslosen Authentifizierung getroffen wurden. Die Anwendung sollte dennoch über eine Abmeldefunktion verfügen, und diese sollte nach bewährten Verfahren implementiert werden, wobei alle lokal gespeicherten Token oder Sitzungskennungen vernichtet werden. Wenn Sitzungsinformationen auf dem Server gespeichert sind, sollten sie ebenfalls durch Senden einer Abmeldeanforderung an diesen Server zerstört werden. Im Falle einer Hochrisikoanwendung sollten die Token ungültig gemacht werden. Das Nichtentfernen von Token oder Sitzungskennungen kann dazu führen, dass Unbefugte Zugriff auf die Anwendung erhalten, falls die Token durchsickern. Es ist zu beachten, dass auch andere sensible Informationen entfernt werden sollten, da alle nicht ordnungsgemäß gelöschten Informationen zu einem späteren Zeitpunkt durchsickern können, z. B. bei einer Gerätesicherung.

Umsetzungsbeispiel: **Allgemein:**

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-user-logout-mstg-auth-4>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-37/1.1

Req 38	Es existiert eine Passwort-Richtlinie, die am entfernten API-Endpunkt erzwungen wird.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Passwortstärke ist ein wichtiges Kriterium, wenn Passwörter für die Authentifizierung verwendet werden. Die Kennwortrichtlinie definiert die Anforderungen, an die sich die Endbenutzer halten sollten. In einer Kennwortrichtlinie werden in der Regel Kennwortlänge, Kennwortkomplexität und Kennworttopologien festgelegt. Eine "starke" Kennwortrichtlinie macht das manuelle oder automatische Knacken von Kennwörtern schwierig oder unmöglich.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-best-practices-for-passwords-mstg-auth-5-and-mstg-auth-6>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-38/1.1

Req 39	Der entfernte API-Endpunkt implementiert einen Mechanismus, um sich gegen eine exzessive Anzahl von Login-Versuchen zu schützen.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Das Remote Backend implementiert Schutzmaßnahmen gegen Passwort Bruteforcing. Dabei wird darauf geachtet, dass berechtigte User im Rahmen eines Denial of Service Angriffs nicht ausgesperrt werden. Zu diesem Zweck kann bspw. von Tarpitting (verzögerte Anmeldeversuche), temporären IP- Adress-Sperren oder ähnlichen Mitteln Gebrauch gemacht werden.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-best-practices-for-passwords-mstg-auth-5-and-mstg-auth-6>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-39/1.1

Req 40	Nach einer definierten Inaktivitätsdauer werden Nutzersitzungen am entfernten API-Endpunkt beendet und Zugriffs-Tokens werden nach Ablauf der Gültigkeitsdauer abgelehnt.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Minimierung der Lebensdauer von Sitzungskennungen und Token verringert die Wahrscheinlichkeit eines erfolgreichen Account-Hijackings.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-session-timeout-mstg-auth-7>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-40/1.1

Req 41	Biometrische Authentifizierung basiert auf dem Betriebssystem-basierten Entsperren des Keystores (Android)/der Keychain (iOS) und nicht auf einer z.B. event-basierten API die einfach "true" oder "false" zurück liefert.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die APIs zur Auswertung, ob eine biometrische Authentisierung durch den Anwender erfolgreich war, liefern typischerweise nur "wahr" oder "falsch" zurück. Sollte die App nur den booleschen Wert überprüfen, könnten Angreifer diese Situation ausnutzen, um unberechtigten Zugang zur Applikation zu erhalten. Richtig ist, bei Einrichtung Schlüssel im Keystore (Android) bzw. der Keychain (iOS) abzulegen und diese nach erfolg-

reicher Authentisierung zur Anmeldung an der App bzw. zur Entsperrung der Inhalte oder der Verschlüsselungsschlüssel auszulesen und zu verwenden. Nur so kann sichergestellt werden, dass tatsächlich ein erfolgreicher (biometrischer) Entsperrvorgang stattgefunden hat.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05f-Testing-Local-Authentication.md#testing-biometric-authentication-mstg-auth-8>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06f-Testing-Local-Authentication.md#testing-local-authentication-mstg-auth-8-and-mstg-storage-11>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-41/1.1

Req 42	Es gibt einen 2. Authentifizierungsfaktor und er wird am entfernten API-Endpunkt konsistent erzwungen.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Das entfernte System verlangt eine Mehrfaktor Authentisierung, um zu verhindern, dass selbst wenn Passwörter kompromittiert wurden ein Zugang zum System mit entwendeten Nutzerdaten nicht ohne weiteres möglich ist.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-two-factor-authentication-and-step-up-authentication-mstg-auth-9-and-mstg-auth-10>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-42/1.1

Req 43	Sensible Transaktionen erfordern eine zusätzliche Authentifizierung (durch einen weiteren Authentifizierungsfaktor).
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Der Zugriff auf sensible, personenbezogene, Daten der Schutzklasse ab "vertraulich" oder je nach App kritischen Anteilen (z.B. Löschung des Accounts, Ändern von Abrechnungsdaten, etc.) wird durch eine interaktive Re-Authentisierung des Anwenders abgesichert, um zu verhindern, dass Dritte im Fall einer Kompromittierung Zugriff auf entsprechende Inhalte bekommen.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-two-factor-authentication-and-step-up-authentication-mstg-auth-9-and-mstg-auth-10>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-43/1.1

Req 44	Die App informiert den Nutzer über alle Anmelde-Vorgänge am Nutzerkonto. Nutzer können eine Liste aller mit dem Konto verbundenen Geräte sowie kontextbezogene Informationen (IP Adresse, Lokation usw.) sehen und ausgewählte Geräte blockieren.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich" Sensible und personenbezogene Daten Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Mehrfache Logins, insbesondere von unterschiedlichen Endgeräten, werden heutzutage von den Anwendern erwartet. Häufig ist es allerdings so, dass es Anwendern selten bewußt ist wie viele (langlebige) Sitzungen sich mit der Zeit von den unterschiedlichsten Endgeräten und Browsern ansammeln. Die App soll den Anwender aktiv dabei unterstützen den sicheren Überblick über alle aktive Logins zu behalten und insbes. bei neuen Logins z.B. durch Push Notifications und eMails zu informieren. Um bei den Anwendern jedoch keine Sicherheits-Müdigkeit durch zu viele Mitteilungen hervorzurufen kann die Applikation auch aktiv Lernen und weiß im Laufe der Zeit zwischen normalen Logins des Anwenders und denen eines möglichen Angreifers zu unterscheiden. Z.B. loggt sich der Anwender typischerweise nicht zw. 23:00 und 07:00 Uhr morgens ein und ansonsten nur innerhalb der EU warnt die App den Anwender bei Login Versuchen von außerhalb der EU oder seiner typischen Nutzungszeit. Innerhalb der App kann der Anwender sich alle aktiven Logins und deren Herkunft (IP, Endgerät, etc.) anzeigen lassen und aktiv einzelne oder alle Sessions (außer der gerade aktiven) beenden. Diese Maßnahmen stellen sicher, dass der Anwender jederzeit über die offenen Sitzungen informiert ist und unbekannte entsprechend sofort eliminieren und Gegenmaßnahmen einleiten kann, so z.B. Änderung des Passworts usw.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md#testing-login-activity-and-device-blocking-mstg-auth-11>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

Req 45 Authentifizierung wird am API-Endpunkt definiert und überprüft.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Authentisierung und Authorisierung finden am entfernten System statt, da letztlich dort die zur Identität passenden Konten, Informationen und Daten des Anwenders zusammenlaufen und vorgehalten bzw. hergestellt 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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-45/1.1

3.5. Network Communication Anforderungen

Req 46 Daten werden zur Netzwerkkommunikation innerhalb der App durchgängig mit TLS verschlüsselt.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Eine der Hauptfunktionen mobiler Apps ist das Senden/Empfangen von Daten über nicht vertrauenswürdige Netzwerke wie das Internet. Wenn die Daten bei der Übertragung nicht ordnungsgemäß geschützt sind, kann ein Angreifer mit Zugang zu einem beliebigen Teil der Netzwerkinfrastruktur (z. B. einem Wi-Fi-Zugangspunkt) die Daten abfangen, lesen oder verändern.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04f-Testing-Network-Communication.md#verifying-data-encryption-on-the-network-mstg-network-1>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05g-Testing-Network-Communication.md#testing-data-encryption-on-the-network-mstg-network-1>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06g-Testing-Network-Communication.md#testing-data-encryption-on-the-network-mstg-network-1>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-46/1.1

Req 47	Die TLS-Einstellungen entsprechen aktuellen Best Practices, oder erfüllen die Best Practices so gut wie möglich, falls das mobile Betriebssystem die Empfehlung nicht unterstützt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Verbindungen sind bei Verwendung von TLS bzw. HTTPS nicht automatisch sicher oder vertrauensvoll. Dies wird erst in Kombination mit entsprechend sicheren Cipher Suites erreicht. Die Cipher Suites müssen dabei PFS (Perfect Forward Secrecy) mit ECDHE (Elliptic Curve Diffie-Hellmann Ephemeral) Schlüsselaustausch zusammen mit AES256 unterstützen.

Für Entwicklungen sind TLS 1.3 (Cipher Suites werden vorgegeben) und TLS 1.2 mit folgenden Cipher Suites zulässig:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TLS 1.3 ist bereits seit Android 10 und iOS 12.2 standardmäßig aktiviert, so dass in der Fläche keine Inkompatibilitäten mehr zu erwarten sind. TLS 1.2 sollte deshalb nur noch dann eingesetzt werden, wenn Inkompatibilitäten im Rahmen der Entwicklung oder bei Feldtests auftreten und für die Produktion nicht ausgeschlossen werden können.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04f-Testing-Network-Communication.md#verifying-the-tls-settings-mstg-network-2>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05g-Testing-Network-Communication.md#testing-the-tls-settings-mstg-network-2>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06g-Testing-Network-Communication.md#testing-the-tls-settings-mstg-network-2>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-47/1.1

Req 48	Die App überprüft das X.509-Zertifikat des Servers beim TLS-Handshake. Die App akzeptiert nur Zertifikate, die von einer vertrauenswürdigen Zertifizierungsstelle signiert wurden.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Verwendung von TLS zur Übertragung sensibler Daten über das Netz ist für die Sicherheit unerlässlich. Die Verschlüsselung der Kommunikation zwischen einer mobilen Anwendung und ihrer Backend-API ist jedoch nicht trivial. Weniger sichere Lösungen (z. B. solche, die jedes beliebige Zertifikat akzeptieren) erleichtern den Entwicklungsprozess, und manchmal schaffen es diese schwachen Lösungen in die Produktionsversion, wodurch Benutzer potenziell Man-in-the-Middle-Angriffen ausgesetzt sind.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05g-Testing-Network-Communication.md#testing-endpoint-identity-verification-mstg-network-3>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06g-Testing-Network-Communication.md#testing-endpoint-identity-verification-mstg-network-3>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-48/1.1

Req 49	Die App nutzt entweder ihren eigenen Zertifikatspeicher oder nutzt Public Key-/ Zertifikats-Pinning und akzeptiert deshalb keine Verbindungen zu Endpunkten die andere Zertifikate/Schlüssel präsentieren - selbst wenn die Zertifikate von einer vertrauenswürdigen Zertifizierungsstelle signiert sind.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Beim Zertifikats-Pinning wird der Backend-Server mit einem bestimmten X.509-Zertifikat oder öffentlichen Schlüssel verknüpft, anstatt jedes von einer vertrauenswürdigen Zertifizierungsstelle signierte Zertifikat zu akzeptieren. Nach dem Speichern ("Pinning") des Serverzertifikats oder des öffentlichen Schlüssels verbindet sich die mobile Anwendung nur noch mit dem bekannten Server. Der Entzug des Vertrauens in externe Zertifizierungsstellen verringert die Angriffsfläche (schließlich gibt es viele Fälle von Zertifizierungsstellen, die kompromittiert oder dazu verleitet wurden, Zertifikate für Betrüger auszustellen).

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05g-Testing-Network-Communication.md#testing-custom-certificate-stores-and-certificate-pinning-mstg-network-4>
<https://developer.android.com/training/articles/security-config#CertificatePinning>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06g-Testing-Network-Communication.md#testing-custom-certificate-stores-and-certificate-pinning-mstg-network-4>
<https://developer.apple.com/news/?id=g9ejcf8y>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-49/1.1

Req 50	Die App nutzt keine unsicheren Kommunikationskanäle wie E-Mail oder SMS für kritische Operationen wie Konten-Registrierung oder Konten-Reaktivierung.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Bei E-Mail und SMS handelt es sich grundsätzlich um keine sicheren Kommunikationskanäle und sie sollten dementsprechend auch nicht für Apps, die Daten der Schutzklasse "vertraulich", sensible Daten oder strikt personenbezogene Daten verarbeiten, standardmäßig eingesetzt werden. Zu solchen Apps zählen bspw. Banking Apps, Kollaborationstools (mit Zugriffsmöglichkeiten auf viele vertrauliche Daten), Health Apps usw.

Entsprechenden Alternativen sollte Vorrang eingeräumt werden. Dazu zählen u.a. Hardware basierte Token, Push Notifications (Zustimmung / Ablehnung des Vorgangs), Google Sign-In, Sign In With Apple, QR-Code, TOTP (Time-based Token wie z.B. Google Authenticator oder Microsoft Authenticator), Aktivierungsbrief, etc.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04f-Testing-Network-Communication.md#making-sure-that-critical-operations-use-secure-communication-channels-mstg-network-5>

Android:

<https://developers.google.com/identity/sign-in/android/start-integrating>

iOS:

https://developer.apple.com/documentation/authenticationservices/implementing_user_authentication_with_sign_in_with_apple

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-50/1.1

Req 51 Die App nutzt aktuelle Bibliotheken zum Aufbau von sicheren Kommunikationsverbindungen.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Werden veraltete Bibliotheken für die App Entwicklung verwendet besteht die Gefahr, dass Fehler oder bekanntgewordene Sicherheitslücken mit der Applikation ausgeliefert werden und in Folge die Applikation für Angriffe anfällig ist. Entwickler müssen deshalb sicherstellen, dass eingesetzte Bibliotheken dem aktuellen Stand entsprechen und rechtzeitig auf aktualisierte Versionen umstellen. Insbesondere auf die Einbindung von Bibliotheken Dritter sollte verzichtet werden, wenn es um die Kernkomponenten wie bspw. sichere Netzwerkverbindungen geht.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05g-Testing-Network-Communication.md#testing-the-security-provider-mstg-network-6>

<https://developer.android.com/training/basics/network-ops/connecting>

<https://developer.android.com/training/articles/security-ssl>

iOS:

<https://developer.apple.com/documentation/network>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-51/1.1

3.6. Platform Interaction Anforderungen

Req 52 Die App fordert vom Nutzer nur die unbedingt erforderlichen App-Berechtigungen.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Apps können mehr Zugriffsrechte eingeräumt werden, als sie für die Ausführung benötigen. Eine möglicherweise vorhandene Schwachstelle könnte dann dazu ausgenutzt werden die App oder das Endgerät zu manipulieren. Ein QR-Code Reader benötigt bspw. Zugriff auf die Kamera, während die App genutzt wird, aber keinesfalls Zugriff auf das Mikrofon. Zudem gelten sowohl für Android als auch für iOS Richtlinien wie Berechtigungen beim User anzufragen sind.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-app-permissions-mstg-platform-1>

<https://developer.android.com/guide/topics/permissions/overview.html>

<https://developer.android.com/training/permissions/requesting>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#testing-app-permissions-mstg-platform-1>
https://developer.apple.com/documentation/uikit#apple_ref/doc/uid/TP40007072-CH3-SW7
<https://developer.apple.com/design/human-interface-guidelines/patterns/accessing-private-data/>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-52/1.1

Req 53	Alle Eingaben von externen Quellen und dem Nutzer werden validiert und falls erforderlich bereinigt. Dazu gehören Daten aus der GUI, IPC Mechanismen wie Intents oder spezifische URL-Schemas und Netzwerk-Daten.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Insbesondere Apps, die WebView (Android) oder UIWebView (iOS) einsetzen, sind häufig anfälliger für XSS (Cross-Site Scripting) Verwundbarkeiten bzw. Angriffe. Ähnlich verhält es sich mit SQL / XML Injection basierten Angriffen. In allen Fällen werden Schwachstellen ausgenutzt, die auf mangelnde Eingabeprüfung und -bereinigung zurückzuführen sind. Darum ist es erforderlich jede auf dem Client als auch dem Backend getätigte Eingabe zu validieren und ggf. zu bereinigen, um zu verhindern, dass Schwachstellen ausgenutzt werden, um die Applikation oder die gespeicherten Daten zu kompromittieren.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04h-Testing-Code-Quality.md#injection-flaws-mstg-arch-2-and-mstg-platform-2>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-for-injection-flaws-mstg-platform-2>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-53/1.1

Req 54	Die App bietet keine sensible Funktionalität über App-eigene URL-Schemas an - wenn doch, ist der Mechanismus angemessen geschützt.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: URL-Schemata stellen einen potenziellen Angriffsvektor auf Apps dar. Alle URL-Parameter sollten deshalb validiert und nicht konforme URLs / URIs verworfen werden. Die verfügbaren Aktionen sollte zudem auf solche eingeschränkt werden, die keine Gefahr für die Daten des Benutzers darstellen. Andere Anwendungen sollten keinen Zugriff auf sensible Inhalte oder Daten der Schutzklasse ab "intern" erhalten. Bei Durchführung von Tests sollte unbedingt darauf geachtet werden, dass die URL-Verarbeitung falsch formatierte URLs / URIs erkennt und angemessen (z.B. Verwerfen) reagiert.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-deep-links-mstg-platform-3>
<https://developer.android.com/training/app-links>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#testing-custom-url-schemes-mstg-platform-3>
<https://developer.apple.com/documentation/xcode/defining-a-custom-url-scheme-for-your-app>
https://developer.apple.com/documentation/uikit#//apple_ref/doc/uid/TP40007072-CH6-SW1

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-54/1.1

Req 55	Die App bietet keine sensible Funktionalität über Interprozesskommunikation (IPC) an - wenn doch, ist der Mechanismus angemessen geschützt.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die IPC-Systemfunktionalität, die von mobilen Anwendungsplattformen angeboten wird, sollte genutzt werden, da sie viel ausgereifter ist als traditionelle Techniken. Die Verwendung von IPC-Mechanismen ohne Berücksichtigung von Sicherheitsfunktionen kann dazu führen, dass die Anwendung sensible Daten preisgibt oder den Zugriff darauf erleichtert, daher muss die gesamte IPC entsprechend geschützt werden.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-for-sensitive-functionality-exposure-through-ipc-mstg-platform-4>
<https://developer.android.com/guide/topics/manifest/activity-element#exported>
<https://developer.android.com/guide/topics/manifest/activity-element#prmsn>
<https://developer.android.com/guide/topics/manifest/permission-element#plevel>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#testing->

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-55/1.1

Req 56	JavaScript ist in WebViews deaktiviert, wenn es nicht unbedingt erforderlich ist.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: JavaScript kann über reflektiertes, gespeichertes oder DOM-basiertes Cross-Site Scripting (XSS) in Webanwendungen eingeschleust werden. Mobile Anwendungen werden in einer Sandbox-Umgebung ausgeführt und weisen diese Schwachstelle nicht auf, wenn sie nativ implementiert sind. Dennoch können WebViews Teil einer nativen Anwendung sein, um die Anzeige von Webseiten zu ermöglichen. Wenn die WebView-Implementierung nicht ausreichend gesichert ist und die Verwendung von JavaScript zulässt, kann JavaScript verwendet werden, um die Anwendung anzugreifen und Zugriff auf ihre Daten zu erhalten.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-javascript-execution-in-webviews-mstg-platform-5>
<https://developer.android.com/reference/android/webkit/WebSettings#setJavaScriptEnabled%28boolean%29>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#testing-ios-webviews-mstg-platform-5>
<https://developer.apple.com/documentation/webkit/wkwebview>
<https://developer.apple.com/documentation/safariservices/sfsafariviewcontroller>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-56/1.1

Req 57	WebViews sind so konfiguriert, dass nur die minimal nötigen Protokoll-Handler erlaubt sind (Idealerweise nur HTTPS). Potenziell gefährliche Handler wie file, tel und app-id sind deaktiviert.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Wenn ein WebView Inhalte aus dem Datenverzeichnis der Anwendung lädt, sollten die Benutzer nicht in der Lage sein, den Dateinamen oder den Pfad zu ändern, aus dem die Datei geladen wird, und sie sollten die geladene Datei nicht bearbeiten können.

Umsetzungsbeispiel: **Allgemein:**

- Create a list that defines local and remote web pages and URL schemes that are allowed to be loaded.
- Create checksums of the local HTML/JavaScript files and check them while the app is starting up.

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-webview-protocol-handlers-mstg-platform-6>
[https://developer.android.com/reference/android/webkit/WebSettings#setAllowContentAccess\(boolean\)](https://developer.android.com/reference/android/webkit/WebSettings#setAllowContentAccess(boolean))
[https://developer.android.com/reference/android/webkit/WebSettings#setAllowFileAccess\(boolean\)](https://developer.android.com/reference/android/webkit/WebSettings#setAllowFileAccess(boolean))
[https://developer.android.com/reference/android/webkit/WebSettings#setAllowFileAccessFromFileURLs\(boolean\)](https://developer.android.com/reference/android/webkit/WebSettings#setAllowFileAccessFromFileURLs(boolean))
[https://developer.android.com/reference/android/webkit/WebSettings#setAllowUniversalAccessFromFileURLs\(boolean\)](https://developer.android.com/reference/android/webkit/WebSettings#setAllowUniversalAccessFromFileURLs(boolean))

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#testing-webview-protocol-handlers-mstg-platform-6>
<https://developer.apple.com/documentation/webkit/wkwebview/1415004-loadhtmlstring>
<https://developer.apple.com/documentation/webkit/wkwebview/1415011-load>
<https://developer.apple.com/documentation/foundation/bundle/1410989-path>
<https://developer.apple.com/documentation/foundation/bundle/1411540-url>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-57/1.1

Req 58	Wenn eine WebView über Javascript Zugriff auf native Methoden einer App bekommt, ist sichergestellt, dass die WebView nur vorgegebenes Javascript aus der App rendert und kein Javascript aus externen Quellen.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Android:

Android bietet eine Möglichkeit für JavaScript, das in einer WebView ausgeführt wird, native Funktionen einer Android-App (mit `@JavascriptInterface` annotiert) aufzurufen und zu verwenden, indem die Methode `addJavaScriptInterface` verwendet wird. Dies ist bekannt als eine WebView JavaScript Brücke oder native Brücke.

Die Verwendung von `addJavaScriptInterface` gewährt explizit den Zugriff auf das registrierte JavaScript Interface Objekt für alle Seiten, die innerhalb dieser WebView geladen werden. Dies bedeutet, dass, wenn der Benutzer außerhalb der Anwendung oder Domäne navigiert, alle anderen externen Seiten ebenfalls Zugriff auf diese JavaS-

cript-Schnittstellenobjekte haben, was ein potenzielles Sicherheitsrisiko darstellen könnte, wenn sensible Daten über diese Schnittstellen offengelegt werden.

iOS:

Apple hat APIs eingeführt, die die Kommunikation zwischen der JavaScript-Laufzeit im WebView und den nativen Swift- oder Objective-C-Objekten ermöglichen. Wenn diese APIs unvorsichtig verwendet werden, können wichtige Funktionen für Angreifer offengelegt werden, denen es gelingt, bösartige Skripte in die WebView zu injizieren (z. B. durch einen erfolgreichen Cross-Site-Scripting-Angriff).

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#determining-whether-java-objects-are-exposed-through-webviews-mstg-platform-7>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#determining-whether-native-methods-are-exposed-through-webviews-mstg-platform-7>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-58/1.1

Req 59	Objektserialisierung wird, falls vorhanden, über eine sichere Serialisierungs-API implementiert.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Die Serialisierung ist nicht von Natur aus sicher. Es handelt sich lediglich um ein Binärformat (oder eine Darstellung) für die lokale Speicherung von Daten in einer Datei.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-object-persistence-mstg-platform-8>

<https://developer.android.com/reference/java/io/Serializable.html>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06h-Testing-Platform-Interaction.md#testing-object-persistence-mstg-platform-8>

https://developer.apple.com/documentation/foundation/archives_and_serialization

<https://developer.apple.com/documentation/foundation/NSSecureCoding>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-59/1.1

Req 60	Die App verwendet Mechanismen, um sich vor Screen Overlay Angriffen zu schützen (nur Android).
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Bildschirmüberlagerungsangriffe treten auf, wenn es einer böartigen Anwendung gelingt, sich über eine andere Anwendung zu legen, die normal weiterarbeitet, als ob sie im Vordergrund wäre. Die böartige Anwendung kann Benutzeroberflächenelemente erstellen, die das Aussehen und die Bedienung der ursprünglichen Anwendung oder sogar die Benutzeroberfläche des Android-Systems nachahmen. Die Absicht besteht in der Regel darin, den Benutzern vorzugaukeln, dass sie weiterhin mit der legitimen Anwendung interagieren, und dann zu versuchen, ihre Privilegien zu erhöhen (z. B. durch die Gewährung einiger Berechtigungen), heimliches Phishing zu betreiben, Benutzereingaben und Tastenanschläge zu erfassen usw.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05h-Testing-Platform-Interaction.md#testing-for-overlay-attacks-mstg-platform-9>

<https://developer.android.com/reference/android/view/View#onFilterTouchEventForSecurity%28android.view.MotionEvent%29>

https://developer.android.com/reference/android/view/View#attr_android:filterTouchesWhenObscured

<https://developer.android.com/reference/android/view/View.html#setFilterTouchesWhenObscured%28boolean%29>

https://developer.android.com/reference/android/view/MotionEvent.html#FLAG_WINDOW_IS_OBSCURED

https://developer.android.com/reference/android/view/MotionEvent.html#FLAG_WINDOW_IS_PARTIALLY_OBSCURED

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-60/1.1

Req 61	Cache, Speicher und geladene Ressourcen (JavaScript usw.) eines WebViews sollten gelöscht werden, bevor der WebView geschlossen wird.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Insbesondere Daten der Schutzklasse ab "vertraulich" unterliegen einem besonderen Schutzbedarf und sollten dementsprechend gelöscht werden bevor der WebView geschlossen wird. Damit wird sichergestellt, dass weder betroffene Bereiche im RAM oder Festspeicher Artefakte enthalten die durch maliziöse Aktivitäten exponiert werden können.

Umsetzungsbeispiel: **Android:**

<https://developer.android.com/reference/android/webkit/WebView.html#clearCache%28boolean%29>

iOS:

<https://developer.apple.com/documentation/foundation/urlcache/1417802-removeallcachedresponses>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-61/1.1

Req 62	Die App sollte verhindern das 3rd Party Tastaturen verwendet werden, wenn sensible Daten eingegeben werden (nur iOS).
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "vertraulich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Es gibt jede Menge nützlicher Tastaturangebote, doch deren Einstellungen lassen sich leider nicht kontrollieren und sensible oder Daten der Schutzklasse ab "vertraulich" könnten unbeabsichtigt zwischengespeichert oder mit Dritten geteilt werden.

Umsetzungsbeispiel: **iOS:**

```
[size]func application(_ application: UIApplication, shouldAllowExtensionPointIdentifier extensionPointIdentifier: UIApplication.ExtensionPointIdentifier) -> Bool {
    switch extensionPointIdentifier {
        case UIApplication.ExtensionPointIdentifier.keyboard: return false
        default: return true
    }
}
```

<https://developer.apple.com/documentation/uikit/uiapplication/extensionpointidentifier>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-62/1.1

3.7. Code Quality und Build Setting Anforderungen

Req 63	Die App ist signiert und mit einem gültigen Zertifikat provisioniert, dessen privater Schlüssel angemessen geschützt ist.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: **Android:**

Android verlangt, dass alle APKs mit einem Zertifikat digital signiert werden, bevor sie installiert oder ausgeführt werden. Die digitale Signatur wird verwendet, um die Identität des Eigentümers bei Anwendungsaktualisierungen zu überprüfen. Dieser Prozess kann verhindern, dass eine App manipuliert oder so verändert wird, dass sie bösartigen Code enthält.

iOS:

Das Signieren des Codes Ihrer App garantiert den Nutzern, dass die App aus einer bekannten Quelle stammt und seit der letzten Signierung nicht verändert wurde. Bevor eine App Dienste integrieren, auf einem Standard Gerät installiert oder im App Store eingereicht werden kann, muss sie mit einem von Apple ausgestellten Zertifikat signiert werden.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#making-sure-that-the-app-is-properly-signed-mstg-code-1>
<https://developer.android.com/studio/publish/app-signing.html>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#making-sure-that-the-app-is-properly-signed-mstg-code-1>
<https://developer.apple.com/documentation/xcode/using-the-latest-code-signature-format>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-63/1.1

Req 64	Die App wurde im Release-Modus gebaut und mit passenden Release-Einstellungen (kein Debugging).
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Obwohl das Debugging eine nützliche Funktion bei der Entwicklung einer App ist, muss es vor der Freigabe von Apps im Apple App Store, Google Playstore oder innerhalb eines Unternehmensprogramms deaktiviert werden. Das Erstellen von Apps im Release-Modus verhindert, dass mittels Debugging-Konsolen Zugriff auf die App und die entsprechenden Daten genommen werden kann.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#testing-whether-the-app-is-debuggable-mstg-code-2>
<https://developer.android.com/guide/topics/manifest/application-element.html#debug>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#determining-whether-the-app-is-debuggable-mstg-code-2>
<https://developer.apple.com/documentation/xcode/testing-a-release-build>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-64/1.1

Req 65 Debugging Symbole wurden von nativen Binärdateien entfernt.

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Im Allgemeinen sollte kompilierter Code mit so wenig Erklärungen wie möglich bereitgestellt werden. Einige Metadaten, wie z. B. Debugging-Informationen, Zeilennummern und beschreibende Funktions- oder Methodennamen, machen den Binär- oder Bytecode für einen Reverse Engineer leichter verständlich, werden aber in einem Release-Build nicht benötigt und sollten daher weggelassen werden, ohne die Funktionalität der Anwendung zu beeinträchtigen.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#testing-for-debugging-symbols-mstg-code-3>
<https://developer.android.com/studio/build/shrink-code>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#finding-debugging-symbols-mstg-code-3>

Um die Einbeziehung von Debugsymbolen zu verhindern, setzen Sie in den Build-Einstellungen des XCode-Projekts die Option `Strip Debug Symbols During Copy` auf YES.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-65/1.1

Req 66	Debugging Code und "Entwicklerüberbleibsel" (z.B. Test Code, backdoors oder versteckte Einstellungen) wurden entfernt und die App-Logdateien enthalten keine ausführlichen Fehler oder Debug-Meldungen.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Debugging-Code, Entwicklungsartefakte oder der die Entwicklung unterstützender Code, wie z.B. die Generierung ausführlicher Fehler- oder Debug-Meldungen muß entfernt werden, um sicherzustellen, dass die Sicherheit und Stabilität der App in der Produktion gewährleistet wird. Damit können potenzielle Schwachstellen und Angriffsvektoren vermieden werden, die von Angreifern ausgenutzt werden könnten.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#testing-for-debugging-code-and-verbose-error-logging-mstg-code-4>
<https://developer.android.com/reference/android/os/StrictMode.html>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#finding-debugging-code-and-verbose-error-logging-mstg-code-4>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-66/1.1

Req 67	Bibliotheken und Frameworks von Drittanbietern, die die App nutzt, wurden auf Schwachstellen geprüft.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Anwendungen verwenden oft Bibliotheken von Drittanbietern, die die Entwicklung beschleunigen, da der Entwickler weniger Code schreiben muss, um ein Problem zu lösen. Bibliotheken von Drittanbietern können jedoch Schwachstellen, inkompatible Lizenzen oder bösartige Inhalte enthalten. Außerdem ist es schwierig, die Abhängigkeiten von Anwendungen zu verwalten, einschließlich der Überwachung von Bibliotheksversionen und der Anwendung verfügbarer Sicherheitspatches.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#checking-for-weaknesses-in-third-party-libraries-mstg-code-5>

iOS:

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-67/1.1

Req 68	Die App führt eine sichere Fehlerbehandlung durch, indem sie Exceptions abfängt und kontrolliert behandelt.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Ausnahmen treten auf, wenn eine Anwendung in einen abnormalen oder fehlerhaften Zustand gerät. Beim Testen der Ausnahmebehandlung geht es darum, sicherzustellen, dass die Anwendung eine Ausnahme behandelt und in einen sicheren Zustand übergeht, ohne sensible Informationen über die Benutzeroberfläche oder die Protokollierungsmechanismen der Anwendung preiszugeben.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#testing-exception-handling-mstg-code-6-and-mstg-code-7>
<https://developer.android.com/reference/java/lang/RuntimeException.html>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#testing-exception-handling-mstg-code-6>
<https://docs.swift.org/swift-book/LanguageGuide/ErrorHandling.html>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-68/1.1

Req 69	Treten in Sicherheitsfunktionen Fehler auf, lehnt die App-Fehlerbehandlung die Zugriffe standardmäßig ab.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Ausnahmen treten auf, wenn eine Anwendung in einen abnormalen oder fehlerhaften Zustand gerät. Beim Testen der Ausnahmebehandlung geht es darum, sicherzustellen, dass die Anwendung eine Ausnahme behandelt und in einen sicheren Zustand übergeht, ohne sensible Informationen über die Benutzeroberfläche oder die Protokollierungsmechanismen der Anwendung preiszugeben.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#testing-exception-handling-mstg-code-6-and-mstg-code-7>
<https://developer.android.com/reference/java/lang/SecurityException>

iOS:

<https://developer.apple.com/documentation/swift/error>
<https://docs.swift.org/swift-book/LanguageGuide/ErrorHandling.html>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-69/1.1

Req 70	Das Speichermanagement (Allokation und Freigabe von Speicher) erfolgt in unmanaged code auf sichere Weise.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"
Sensible und personenbezogene Daten
Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Android:

Android-Anwendungen werden häufig in einer VM ausgeführt, in der die meisten Probleme mit der Speicherkorruption beseitigt wurden. Das bedeutet aber nicht, dass es keine Speicherkorruptionsfehler gibt.

iOS:

Bei iOS-Anwendungen gibt es verschiedene Möglichkeiten, Speicherfehler zu verursachen: Zunächst gibt es Probleme mit nativem Code, verschiedene unsichere Operationen sowohl in Objective-C als auch in Swift, die um nativen Code gewrappert werden, was zu Problemen und Speicherlecks führen kann, weil Objekte, die nicht mehr verwendet und weiter vorgehalten werden.

Umsetzungsbeispiel: **Allgemein:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x04h-Testing-Code-Quality.md#memory-corruption-bugs-mstg-code-8>

Android:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#memory-corruption-bugs-mstg-code-8>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#memory-corruption-bugs-mstg-code-8>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-70/1.1

Req 71	Angebote Sicherheitsfunktionen der Entwicklungsumgebung wie Byte-Code Minimierung, Stack-Protection, PIE-Support und automatisches Reference-Counting sind aktiviert.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Schutzklasse ab "öffentlich"

Sensible und personenbezogene Daten

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Motivation: Während die neuesten Installationen der Entwicklungsumgebungen für Android und iOS bereits die entsprechenden Einstellungen standardmäßig voreingestellt haben, kann es sein, dass die Sicherheitsmechanismen bei aktualisierten Installationen noch inaktiv sind oder testweise deaktiviert wurden.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md#make-sure-that-free-security-features-are-activated-mstg-code-9>

iOS:

<https://github.com/OWASP/owasp-mastg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md#make-sure-that-free-security-features-are-activated-mstg-code-9>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-71/1.1

3.8. Resilience Anforderungen

Req 72	Die App erkennt und reagiert auf ein gerootetes Gerät oder Geräte mit Jailbreak, indem der Nutzer gewarnt oder die App beendet wird.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zu Erreichung des beschriebenen Schutzniveaus müssen die Apps sicherstellen, dass die Inhalte angemessen geschützt sind. Auf Geräten mit aktivem Root / Jailbreak ist das nicht sichergestellt und die App muss den Anwender darauf hinweisen und die Ausführung verweigern.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-root-detection-mstg-resilience-1>

iOS:

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md#jailbreak-detection-mstg-resilience-1>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-72/1.1

Req 73	Die App verhindert Debugging und/oder erkennt und reagiert auf einen Debugger. Alle verfügbaren Debugging-Protokolle müssen abgedeckt werden.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps Debugging unterbinden bzw. auf Debugging Versuche angemessen reagieren, um Reverse Engineering Versuche und damit Tests auf mögliche Schwachstellen und Angriffsmöglichkeiten entsprechend zu erschweren.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-anti-debugging-detection-mstg-resilience-2>

iOS:

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-73/1.1

Req 74	Die App erkennt und reagiert auf Manipulationen an ausführbaren Dateien und kritischen Daten innerhalb der App-eigenen Sandbox.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps Manipulationen erkennen und angemessen reagieren, um sicherzustellen, dass eingeschleuster Code nicht zur Ausführung kommt. So wird verhindert, dass die App oder Daten kompromittiert werden und in Folge ein Schaden für den Anwender entsteht.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-file-integrity-checks-mstg-resilience-3>

iOS:

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md#file-integrity-checks-mstg-resilience-3-and-mstg-resilience-11>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-74/1.1

Req 75	Die App erkennt und reagiert auf typische und allgemein bekannte Reverse Engineering Tools und Frameworks auf dem Gerät.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps installierte Reverse Engineering Tools, Apps oder Frameworks erkennen und angemessen darauf reagieren, um sicherzustellen, dass durch diese Massnahmen der Analyse Aufwand erheblich erhöht und somit mögliche Angreifer abgeschreckt werden.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-reverse-engineering-tools-detection-mstg-resilience-4>

iOS:

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-reverse-engineering-tools-detection-mstg-resilience-4>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-75/1.1

Req 76	Die App erkennt und reagiert darauf, wenn sie in einem Emulator ausgeführt wird.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps eine Emulatorerkennung implementieren, um die Ausführung der Anwendung auf einem emulierten Gerät zu erschweren, was einige Tools und Techniken, die Reverse Engineers gerne verwenden, behindert. Dieser erhöhte Schwierigkeitsgrad zwingt potentielle Angreifer, die Emulatorprüfungen zu umgehen oder das physische Gerät zu verwenden, wodurch der für eine groß angelegte

Geräteanalyse erforderliche Zugang verwehrt wird.

Konnten in der Vergangenheit ausschließlich Android Geräte komfortabel emuliert werden gibt es mittlerweile Anbieter wie z.B. [Corellium](#) die entsprechende Services nicht nur für Android, sondern auch iOS Geräte inkl. Jailbreaking und Large Scale zur Verfügung stellen. Aus diesem Grund ist ein entsprechender Schutz auch in iOS Apps notwendig.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-emulator-detection-mstg-resilience-5>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-76/1.1

Req 77	Die App erkennt und reagiert auf Manipulation von Code und Daten im eigenen Arbeitsspeicherbereich.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps die Integrität ihres Speicherbereichs überprüfen, um die App gegen Speicherpatches zu schützen, die während der Laufzeit angewendet werden. Zu solchen Patches gehören unerwünschte Änderungen am Binärcode, Bytecode, Funktionszeigertabellen und wichtigen Datenstrukturen sowie in den Prozessspeicher geladener schadhafter Code.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-runtime-integrity-checks-mstg-resilience-6>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-77/1.1

Req 78	Die App realisiert mehrere Mechanismen in jeder Kategorie (Requirements von 72 bis 77). Die Resilienz steigt mit der Anzahl, Vielfalt und Originalität der genutzten Mechanismen.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps mehrere Mechanismen aus den Requirements 72 bis 77 implementieren, um so die Resilienz gegen Angreifer entsprechend und angemessen zu erhöhen.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-78/1.1

Req 79	Die Erkennungsmechanismen der App lösen verschiedene Arten von Reaktionen aus z.B. verzögerte, versteckte oder heimliche Reaktionen.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps Resilienz Mechanismen implementieren die verschiedene Reaktionen auslösen, um Angreifern immer wieder erneute Hürden zu setzen und damit entsprechend abzuschrecken.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-79/1.1

Req 80	Programmatische Abwehrmaßnahmen werden durch Obfuskierung geschützt was wiederum De-Obfuskierung mittels dynamischer Analyse verhindert.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps Obfuskierungsmaßnahmen umsetzen, um so angemessen auf Versuche zur De-Obfuskierung und damit Reverse Engineering vorzubeugen.

Umsetzungsbeispiel: **Android:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/Ox05j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-obfuscation-mstg-resilience-9>

iOS:

<https://github.com/OWASP/owasp-mstg/blob/master/Document/Ox06j-Testing-Resiliency-Against-Reverse-Engineering.md#testing-obfuscation-mstg-resilience-9>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-80/1.1

Req 81	Die App implementiert einen Mechanismus zur Gerätebindung, bei dem der Geräte-Fingerabdruck aus mehreren einzigartigen Geräteeigenschaften ermittelt wird.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

ten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps einen Fingerabdruck des Geräts implementieren der aus unterschiedlichen Eigenschaften des Geräts ermittelt wird und nicht reproduzierbar auf ein anderes Gerät dupliziert werden kann. Eine Ausführung der App oder sicherheitskritischer Bereiche kann somit auch nach einer Duplizierung entsprechend verhindert werden.

Umsetzungsbeispiel: **iOS:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md#device-binding-mstg-resilience-10>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-81/1.1

Req 82	Alle ausführbaren Dateien und Bibliotheken der App sind entweder auf Dateiebene verschlüsselt und/oder wichtige Code- und Datensegmente in ausführbaren Dateien sind verschlüsselt oder durch Packing obfuskiert. Triviale statische Analyse offenbart keinen wichtigen Code oder Daten.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps von der Verschlüsselung wichtiger Dateien und Code-Segmente Gebrauch machen, um zu verhindern, dass Dritte bereits im Rahmen einer statischen Analyse Rückschlüsse auf die Sicherheitsmechanismen ziehen können.

Umsetzungsbeispiel: **iOS:**

<https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md#file-integrity-checks-mstg-resilience-3-and-mstg-resilience-11>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-82/1.1

Req 83	Wenn das Ziel der Obfuskierung der Schutz sensibler Logik wie Algorithmen oder Berechnungen ist, so wird ein angemessener Obfuskierungsmechanismus, dem Stand der Technik entsprechend, genutzt, der resilient gegen manuelle und automatisierte De-Obfuskierungsangriffe ist. Die Wirksamkeit der Obfuskierungsmethode muss durch manuelle Tests überprüft werden. Es ist zu beachten, dass hardware-basierte Isolations-Mechanismen softwarebasierter Obfuskierung vorzuziehen sind.
--------	--

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps Obfuskierungsmethoden einsetzen, um De-Obfuskierungsversuchen angemessen zu verhindern. Dabei werden Reverse Engineering Tätigkeiten und damit Tests auf mögliche Schwachstellen und Angriffsmöglichkeiten entsprechend erschwert.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-83/1.1

Req 84	Neben einer gehärteten Kommunikation zwischen Client und API-Endpunkt sollte auch der übertragene Payload verschlüsselt werden, um das Abhören von Daten zu erschweren.
--------	---

Betrifft Apps, die folgende Daten verarbeiten:

Urheberrechts-Schutz, besonders sensible und unternehmenskritische Daten

Apps, die sensible Daten oder urheberrechtlich geschützte Inhalte verarbeiten, müssen die zu verarbeitenden Daten besonders schützen. Dazu gehören beispielsweise Banking- und Versicherungs-Apps, Gesundheits-Apps, Video-, Musik-Player und eReader sowie Messenger. Diese Apps benötigen ein höheres Sicherheitsniveau, um sie angemessen gegen Analyse und Angriffe zu schützen.

Fertige SDKs wie z.B. [Trusteer](#) oder No-Code Lösungen wie z.B. [Appdome](#) bieten dabei den Vorteil einer permanenten Weiterentwicklung während Entwickler sich auf die Kernfunktionen der App konzentrieren können.

Bestehen keine besonderen Schutzanforderungen an die App ist die Umsetzung der Anforderung nicht zwingend erforderlich!

Motivation: Zur Erreichung des beschriebenen Schutzniveaus sollten die Apps auch im Rahmen einer per TLS verschlüsselten Verbindung die übertragenen Daten (Payload) bereits vor der Übertragung verschlüsseln (Ende-Zu-Ende Verschlüsselung), um die zu übertragenen Daten selbst gegen mögliche Man-In-The-Middle Angriffe zu verteidigen. Sollte die TLS Verbindung durch Dritte kompromittiert werden, wären die zu übertragenen Daten weiterhin verschlüsselt und für den Angreifer nicht ohne weiteres einsehbar.

Umsetzungsbeispiel: **Android:**

<https://developer.android.com/guide/topics/security/cryptography>

iOS:

Dies ist ein einfaches Beispiel unter Verwendung von CloudKit. Je nach Einsatzzweck stehen iCloud Features, insbes. in Unternehmen, nicht zur Verfügung:

<https://github.com/apple/sample-cloudkit-encryption>

<https://support.apple.com/de-de/guide/security/sec3cac31735/web>

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 4.1.1-84/1.1