

Abschlussbericht

„Webaudit JuiceShop“

für
Mustermann GmbH

Übersicht	
Version	1.0
Datum	1. Juni 2022
Ansprechpartner Kunde	Herr Max Mustermann
Ansprechpartner Q_PERIOR	Herr David Wollmann

Versionshistorie

Version	Datum	Author	Beschreibung
0.1	30. Mai 2022	David Wollmann	Draft
0.2	31. Mai 2022	Dr. Michael Hanspach	Qualitätssicherung
1.0	1. Juni 2022	David Wollmann	Finale Version

Inhalt

Versionshistorie.....	1
1 Ausgangslage und Zielsetzung	3
2 Management Summary	4
2.1 Ergebnis der Überprüfung	4
2.2 Empfehlungen.....	4
3 Schwachstellenübersicht	5
4 Projektinformationen.....	7
5 Detaillierte Findings	8
5.1  Findings Risikostufe Kritisch	8
5.1.1 Zugriff auf Sensitive Daten ohne Authentisierung	8
5.1.1.1  Proof of Concept Zugriff auf Sensitive Daten ohne Authentisierung.....	9
5.1.2 Fehlende Zugriffsprüfung bei Funktionen und/oder Daten	10
5.1.2.1  Proof of Concept Zugriff auf Daten anderer Benutzer möglich	11
5.2  Findings Risikostufe Hoch.....	14
5.2.3 Verzeichnisaufistung auf dem Webserver.....	14
5.2.3.1  Proof of Concept Verzeichnisaufistung	15
5.2.4 Fehlende Möglichkeit zur Verschlüsselung der Kommunikation	16
5.2.4.1  Proof of Concept Fehlende Möglichkeit zur verschlüsselten Kommunikation.....	17
5.3  Findings Risikostufe Mittel	18
5.3.5 Informationen in Robots.txt Datei.....	18
5.3.5.1  Proof of Concept Informationen in Datei Robots.txt	19
5.4  Findings Risikostufe Niedrig	20
5.4.6 Detaillierte Fehlermeldung.....	20
5.4.6.1  Proof of Concept Detaillierte Fehlermeldung	21
Anhang 1 – Erklärung Q_SCORE und der Risikoeinstufung.....	22

1 Ausgangslage und Zielsetzung

Ausgangslage

Die Mustermann GmbH plant vor der Inbetriebnahme des Webshops eine Überprüfung auf mögliche Schwachstellen mit Hilfe eines Webanwendungs-Penetrationstests.

Die Überprüfung des Webshops erfolgt remote über das Internet. Dabei wird eine Testumgebung zur Verfügung gestellt, die technisch dem tatsächlichen Webshop gleicht.

Zielsetzung und Vorgehen

Ziel der Überprüfung ist es, die in der Anwendung enthaltenen Schwachstellen zu ermitteln und eine Empfehlung zur Behebung zu geben.

Da es sich bei der zu überprüfenden Anwendung um eine Webanwendung handelt, wird nach den Vorgaben des Open Web Application Security Projects (OWASP) vorgegangen.

Neben der reinen Webanwendung werden ebenfalls die Systeme im Rahmen dieser Überprüfung betrachtet, die für den Betrieb der Webanwendung unmittelbar erforderlich sind. In der Regel handelt es sich dabei um den Webserver direkt.

Die Prüfung erfolgte auf einem Black-Box-Ansatz. Außer der URL für die Überprüfung wurden keine weiteren Informationen von der Mustermann GmbH zur Verfügung gestellt.

Im Rahmen der Überprüfung wurden sowohl automatisierte Tests mit Hilfe von entsprechender Software durchgeführt, als auch manuelle Überprüfungen. Dieser Ansatz hilft, die hohe Anzahl an technischen Angriffsmöglichkeiten durchzuspielen, während manuell Angriffe wie beispielsweise logische Fehler gesucht werden, die nicht oder nur sehr schwer über automatisierte Scans erkannt werden können. Weiterhin werden jegliche Schwachstellen, die über eine Software gefunden worden sind, manuell verifiziert, um auszuschließen, dass es sich um ein sogenanntes False-Positive-Ergebnis handelt.

2 Management Summary

Die Management Summary soll einen schnellen Überblick über die Ergebnisse des durchgeführten Penetrationstests bieten. Dabei werden die Schwachstellen und deren Auswirkungen erklärt, jedoch wird auf technische Beschreibungen verzichtet. Diese Details befinden sich in Kapitel 5 dieses Berichtes. Eine kurze Übersicht über die Schwachstellen und die Bewertung können Sie Kapitel 3 entnehmen.

2.1 Ergebnis der Überprüfung

Die überprüfte Webanwendung entspricht nicht den Empfehlungen für sichere Webanwendungen.

Im Rahmen der Überprüfung konnten einige als kritisch anzusehende Schwachstellen identifiziert werden, die den Zugriff auf sensitive Daten ohne vorherige Anmeldung an der Anwendung zulassen. Zwei weitere in der Anwendung identifizierte Schwachstellen begünstigen diesen Angriffsvektor.

Weiterhin ist es durch eine andere Schwachstelle möglich, auf personenbezogene Daten anderer Benutzer zuzugreifen, sofern man über einen gültigen Account für die Anwendung verfügt.

Neben diesen kritischsten Findings sind weitere Schwachstellen in der Anwendung enthalten, die von der Kritikalität niedriger einzustufen sind. Diese Schwachstellen werden an dieser Stelle nicht aufgelistet, da sie beispielsweise nicht den unbefugten Zugriff auf Daten erlauben. Dennoch sollten diese Schwachstellen möglichst zeitnah adressiert und behoben werden.

2.2 Empfehlungen

Q_PERIOR empfiehlt, alle Schwachstellen, die mit der Risikostufe  *kritisch* und  *hoch* klassifiziert wurden, so schnell wie möglich zu beheben. Im Falle der überprüften Anwendung bedeutet dies konkret:

- Das Verschieben aller Dateien und Verzeichnisse an einen Ort, an dem nicht direkt über einen Browser zugegriffen werden kann.
- Beim Zugriff auf Daten und Funktionen wird jedes Mal eine Überprüfung durchgeführt, ob der zugreifende Benutzer entsprechende Zugriffsrechte besitzt.
- Die Konfiguration des Webservers wird so angepasst, dass Verzeichnisinhalte nicht mehr aufgelistet werden können.

Alle weiteren Empfehlungen für die Risikostufen  *mittel* und  *niedrig*, die in Kapitel 5, für die Findings aufgeführt worden sind, sollten nachfolgend auch so zeitnah wie möglich umgesetzt werden.

Generell empfiehlt Q_PERIOR, Anwendungen und Systeme in regelmäßigen Abständen auf die Sicherheit hin zu überprüfen.

David Wollmann

Offensive Security Service Team

3 Schwachstellenübersicht

In diesem Kapitel erhalten Sie eine tabellarische Übersicht über die gefundenen Schwachstellen. Die Details zu den Schwachstellen, sowie die empfohlenen Maßnahmen entnehmen Sie bitte Kapitel 5. Eine Beschreibung zum Q_SCORE und der allgemeinen Risikoeinstufung befinden sich in Anhang 1.

Kritikalität	Name	Beschreibung	Empfehlung	Q_SCORE	CVSSv3.1 Score	OWASP Kategorie
Kritisch	Zugriff auf Sensitive Daten ohne Authentisierung	Es ist möglich, auf sensitive Daten ohne vorherige Authentisierung zuzugreifen.	Stellen Sie sicher, dass für den Zugriff auf sensitive Daten eine vorherige Authentisierung erfolgen muss.	9.9	7.5	A7 – Identification and Authentication Failures
Kritisch	Fehlende Zugriffsprüfung bei Funktionen und/oder Daten	Die Anwendung prüft nicht, ob ein Anwender die Berechtigung besitzt, auf gewisse Daten innerhalb der Anwendung zuzugreifen.	Stellen Sie beim Aufruf jeglicher Funktionen bzw. beim Zugriff auf Daten sicher, dass der angemeldete Benutzer dazu berechtigt ist.	9.9	6.4	A1 – Broken Access Control
Hoch	Verzeichnisauflistung auf dem Webserver	Der Webserver erlaubt die Auflistung der Inhalte von Verzeichnissen.	Deaktivieren Sie das Directory Listing auf für den Webserver.	8.5	5.3	A5 – Security Misconfiguration
Hoch	Fehlende Möglichkeit zur Verschlüsselung der Kommunikation	Der Server bietet keine Möglichkeit für eine verschlüsselte Kommunikation	Konfigurieren Sie den Server so, dass eine verschlüsselte Kommunikation möglich ist.	8.5	6.4	A2 – Cryptographic Failures
Mittel	Informationen in Robots.txt Datei	Die Datei Robots.txt liefert Hinweise für weitere Angriffspunkte.	Versuchen Sie die Informationen in der entsprechenden Datei so gering wie möglich zu halten.	5.3	5.3	A5 – Security Misconfiguration
Niedrig	Detaillierte Fehlermeldung	Die Anwendung gibt detaillierte Informationen in	Geben Sie nur generische Fehlermeldungen	3.7	5.3	

		Fehlermeldungen preis.	aus. Fügen Sie Details zu Fehlern in Logfiles hinzu, auf die ein Anwender keinen Zugriff besitzt.			
--	--	------------------------	---	--	--	--

4 Projektinformationen

Ansprechpartner Kunde : Max Mustermann
E-Mail-Adresse : max.mustermann@mustergmbh.de
Telefonnummer : +49 555 123456

Ansprechpartner Q_PERIOR : David Wollmann
E-Mail-Adresse : david.wollmann@q-perior.com
Telefonnummer : +49 152 52192760

Durchführungszeitraum:

Das Audit wurde vom 4. bis 8. April 2022 durchgeführt.

Einstiegspunkt:

Die folgende URL wurde als Einstiegspunkt für die Überprüfung vom Kunden angegeben:

- <http://juiceshop:3000>

Zugangsdaten:

Da es sich bei dem Audit um einen Black-Box-Ansatz gehandelt hat, wurden keine Zugangsdaten für die Anwendung mitgeteilt.

Anmerkungen zum Beispielbericht:

Bei dem OWASP Juice Shop, der als Grundlage für diesen Bericht gedient hat, sind zahlreiche Schwachstellen enthalten. In dem Bericht wurden bewusst nicht alle Schwachstellen der Webanwendung aufgeführt, sondern nur exemplarisch einige herausgesucht. Bei der Überprüfung einer Webanwendung werden selbstverständlich alle Schwachstellen gesucht und aufgeführt.

5 Detaillierte Findings

5.1 Findings Risikostufe Kritisch

5.1.1 Zugriff auf Sensitive Daten ohne Authentisierung

Beschreibung der Schwachstelle:

In der Webanwendung ist ein Zugriff auf sensitive Daten möglich, ohne dass dafür eine vorherige Anmeldung erforderlich ist. Der Zugriff kann erfolgen, sofern die URL bzw. ID für die Daten bekannt sind oder zufällig erraten werden.

Potentieller Schaden:

Ein Angreifer kann problemlos auf sensitive Daten zugreifen. Diese Daten befinden sich komplett ungeschützt im Internet und können von jedem angesehen und heruntergeladen werden.

Empfohlene Maßnahme:

Stellen Sie sicher, dass der Zugriff auf sensitive Daten einer Berechtigungsprüfung unterliegt. Dateien sollten in der Regel nicht direkt in einem öffentlich zugreifbaren Verzeichnis auf dem Webserver liegen. Besser wäre hier die Speicherung entweder in einer Datenbank oder in einem Verzeichnis außerhalb des Webroot-Verzeichnisses auf dem Webserver.

OWASP 2021 Kategorie:

A7 – Identification and Authentication Failures

5.1.1.1 Proof of Concept Zugriff auf Sensitive Daten ohne Authentisierung

Im betroffenen Verzeichnis wurde eine Vielzahl an Dateien und ein Verzeichnis gefunden, auf die ohne Authentisierung zugegriffen werden konnte:

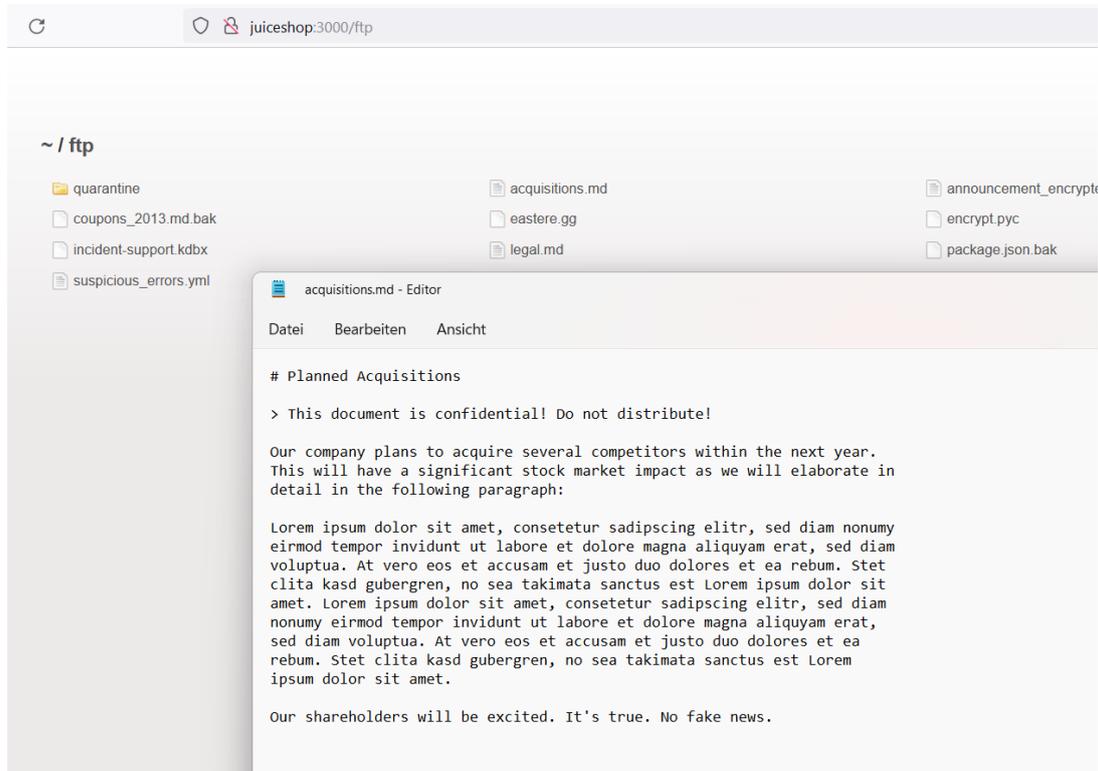


Abbildung 1: Beispielhafter Zugriff auf eine Datei aus dem Verzeichnis

Betroffene Komponente:

Das folgende Verzeichnis inklusive aller darin befindlichen Daten und Unterverzeichnisse ist betroffen:

- <http://juiceshop:3000/ftp>

Bewertungskriterien:

CVSS 3.1 Vektor und Score: [AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N](#) (7.5)

Q-Scoring: Aus Sicht des Offensive Security Service Teams wird im CVSS Score nicht korrekt die Kritikalität widerspiegelt, dass diese Schwachstelle den freien Zugriff auf sensible Unternehmensdaten ermöglicht. Daher erfolgt eine Einstufung abweichend vom eigentlichen CVSS Score. (Erhöhung auf 9.9)

Zuständigkeit für die Behebung:

Diese Schwachstelle sollte durch die Anwendungsentwickler behoben werden.

5.1.2 Fehlende Zugriffsprüfung bei Funktionen und/oder Daten

Beschreibung der Schwachstelle:

In der Anwendung ist ein Rechte- und Rollenkonzept für die Steuerung von Zugriffsrechten einzelner Benutzer integriert. Durch die Manipulation von Anfragen an die Anwendung ist es möglich, auf Funktionen und/oder Daten zuzugreifen, auf die der angemeldete Benutzer normalerweise keine Zugriffsrechte besitzt. Ein Beispiel für solch eine Funktion wäre die Möglichkeit, auf ein administratives Backend zuzugreifen oder auf die Daten anderer Benutzer.

Potentieller Schaden:

Angreifer können zumindest lesend auf Funktionen oder Daten zugreifen, auf die sie keine Berechtigung besitzen. In dem jeweiligen Proof of Concept werden die Fälle genauer beschrieben, und ob auch eine Manipulation von Funktionen bzw. Daten anderer Benutzer durch einen Angreifer möglich wäre.

Empfohlene Maßnahme:

Es wird empfohlen, dass die Webanwendung bei jedem Zugriff auf Funktionen und Datensätze prüft, ob der zugreifende Benutzer ausreichende Zugriffsrechte besitzt.

OWASP 2021 Kategorie:

A1 – Broken Access Control

5.1.2.1 Proof of Concept Zugriff auf Daten anderer Benutzer möglich

Bei der überprüften Webanwendung handelt es sich um einen Webshop des Kunden. Eine der enthaltenen Funktionen stellt die Nutzung eines Warenkorbs dar. Dabei wird der Warenkorb nach Benutzern getrennt und der Warenkorb soll nur von dem jeweiligen Benutzer einsehbar und zu verändern sein:

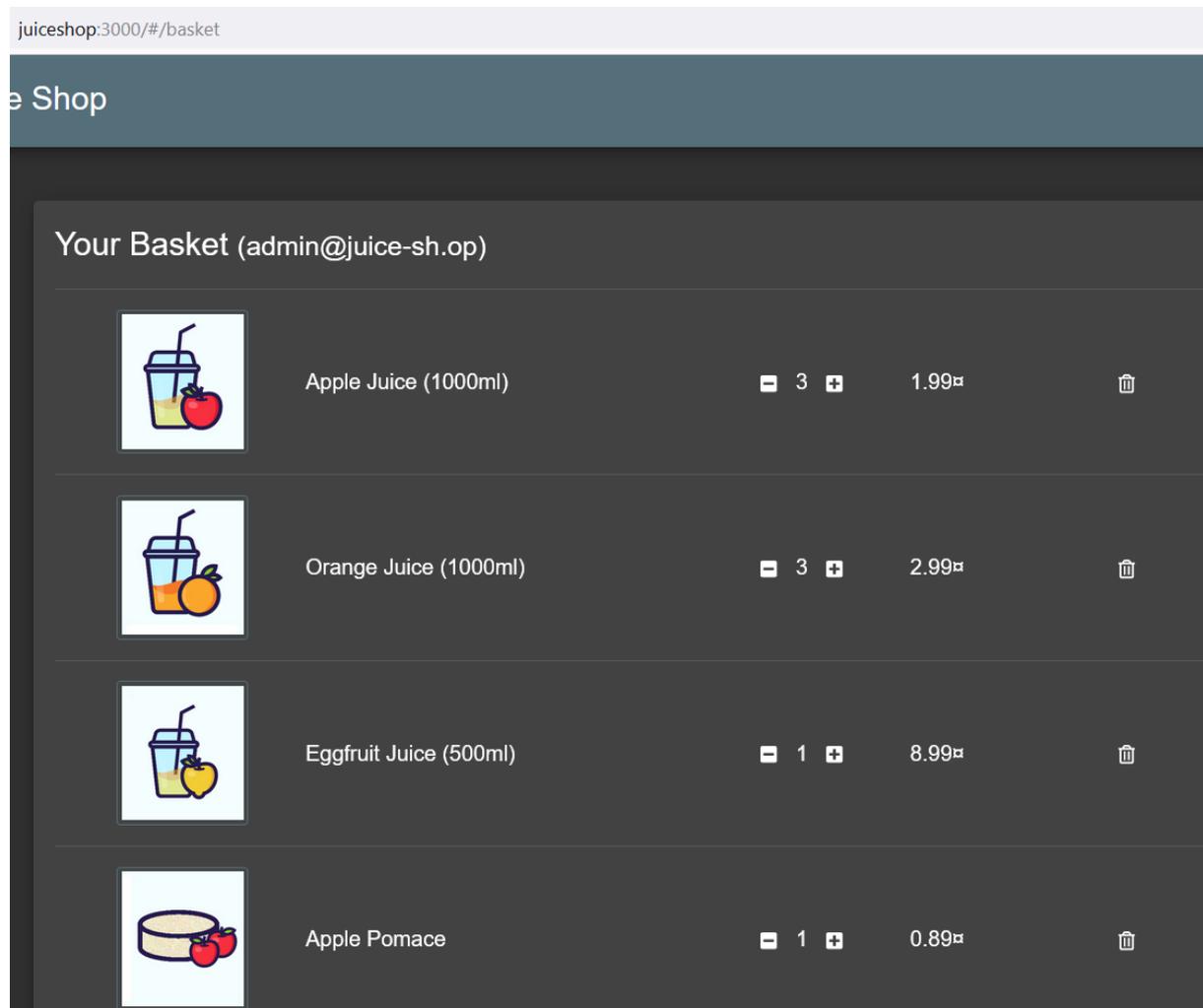


Abbildung 2: Der Warenkorb des angemeldeten Benutzers.

Der Zugriff auf den Warenkorb wird über eine ID im Session Storage gehandhabt (Name des Keys *bid*):

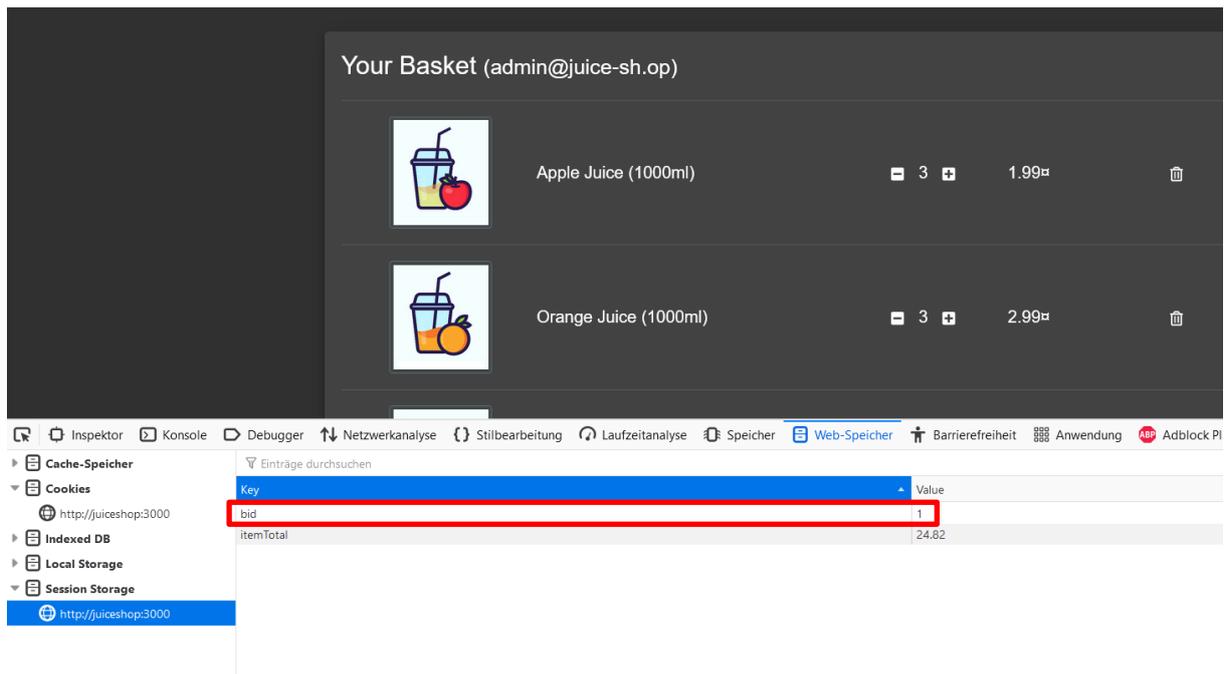


Abbildung 3: Der Originalwert für den Key *bid* für den angemeldeten Benutzer ist auf 1 gesetzt.

Dieser Wert ist im Browser des Users hinterlegt und kann entsprechend auch dort verändert werden:

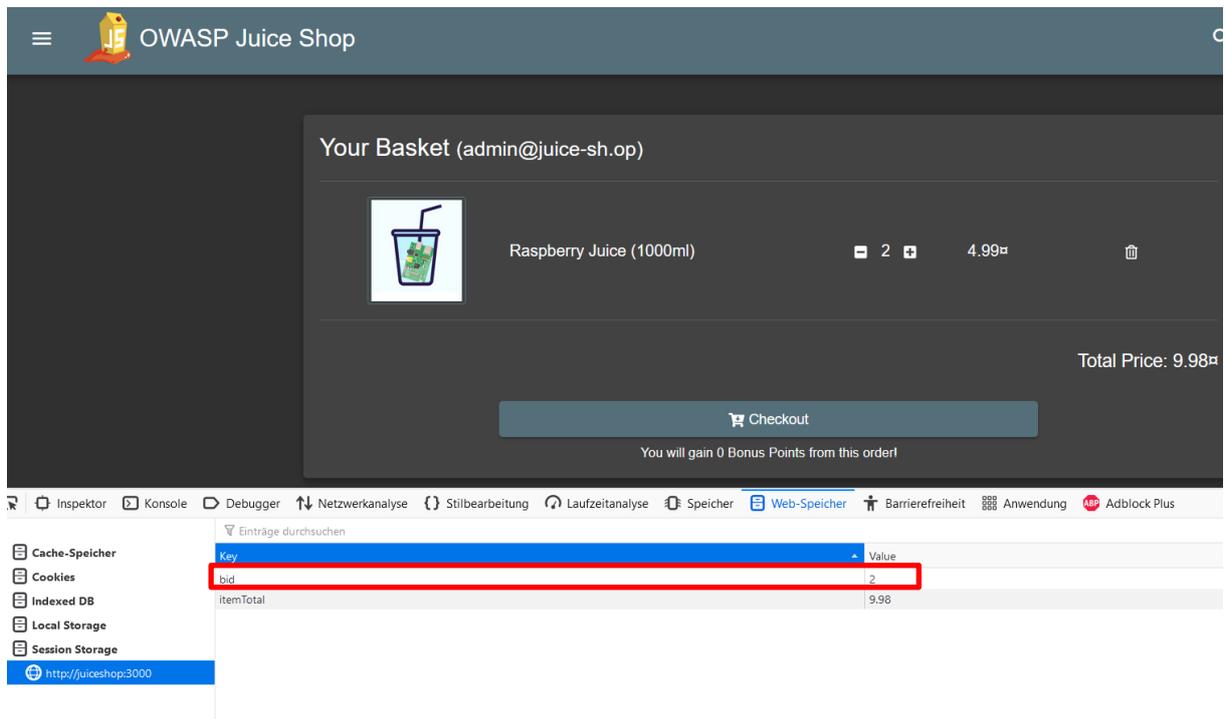


Abbildung 4: Der Wert für den Key *bid* wurde auf 2 geändert.

Nachdem der Wert abgeändert wurde, wird beim erneuten Laden der Webseite der Warenkorb eines anderen Benutzers angezeigt:

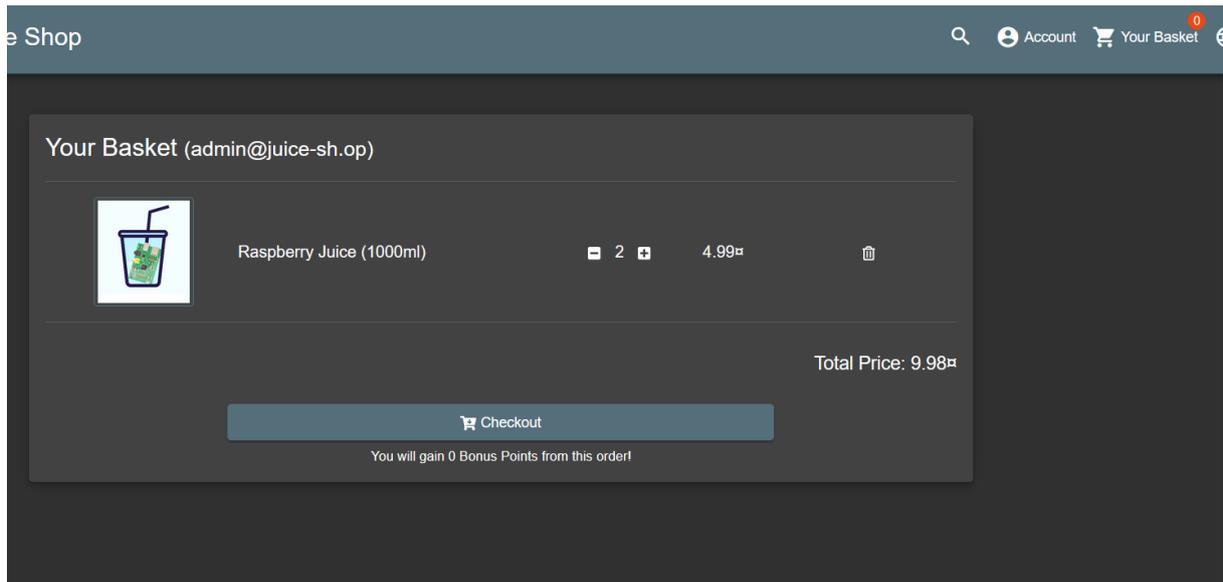


Abbildung 5: Der Warenkorb eines anderen Benutzers.

Neben der Anzeige des Warenkorbes war auch das Ändern des Warenkorbes möglich.

Betroffene Komponente:

Der Warenkorb der Webanwendung ist betroffen. Insbesondere der Key *bid* der im Session Storage verwendet wird.

Bewertungskriterien:

CVSS 3.1 Vektor und Score: [AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N](#) (6.4)

Q-Scoring: Aus Sicht des Offensive Security Service Teams wird im CVSS Score nicht korrekt die Kritikalität widerspiegelt, dass diese Schwachstelle den Zugriff auf personenbezogene Daten von anderen Nutzern des Shops ermöglicht. Daher erfolgt eine Einstufung abweichend vom eigentlichen CVSS Score. (Erhöhung auf 9.9)

Zuständigkeit für die Behebung:

Diese Schwachstelle sollte durch die Anwendungsentwickler behoben werden.

5.2 Findings Risikostufe Hoch

5.2.3 Verzeichnisaufistung auf dem Webserver

Beschreibung der Schwachstelle:

Es wurde mindestens ein Vorkommen auf dem Webserver gefunden, bei dem der Inhalt eines Verzeichnisses aufgelistet wird. Bei dieser Auflistung werden alle im Verzeichnis befindlichen Dateien und Unterverzeichnisse aufgeführt.

Potentieller Schaden:

Ein Angreifer kann Zugriff auf Informationen erhalten, auf die er normalerweise keinen Zugriff gehabt hätte.

Empfohlene Maßnahme:

Deaktivieren Sie die Möglichkeit für die Verzeichnisaufistung (Directory Listing) für alle Verzeichnisse auf dem Webserver.

OWASP 2021 Kategorie:

A5 – Security Misconfiguration

Referenz:

- <https://www.simplified.guide/apache/disable-directory-listing>

5.2.3.1 Proof of Concept Verzeichnisauflistung

Das Verzeichnis `/ftp` lässt eine Auflistung der Inhalte zu:

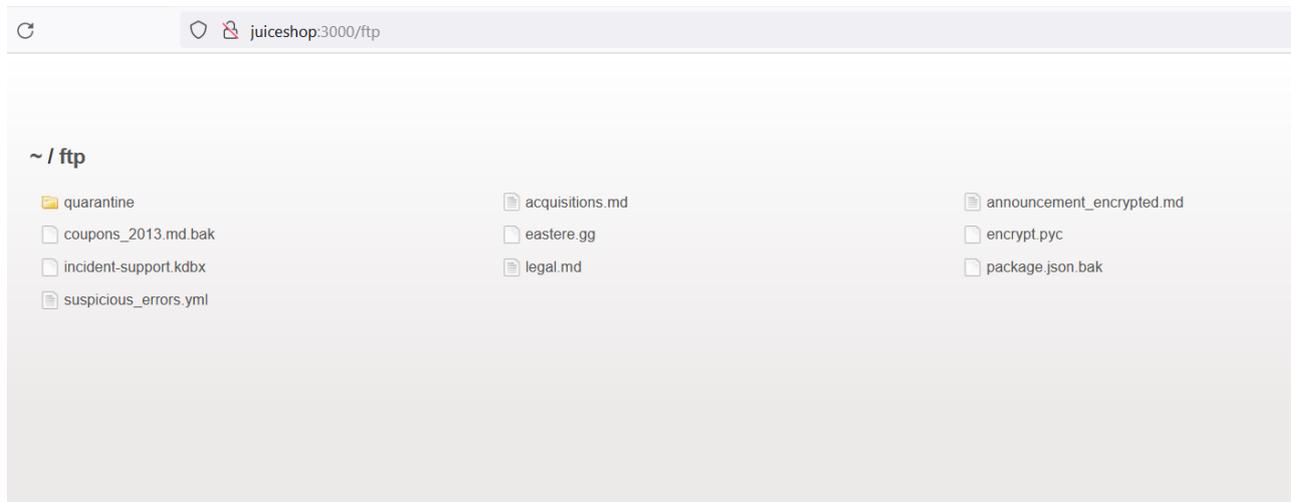


Abbildung 6: Auflistung der Inhalte des Verzeichnisses

Dabei ist zu beachten, dass diese Schwachstelle die Ausnutzung der Schwachstelle aus Kapitel 5.1.1 erleichtert.

Betroffene Komponente:

- <http://juiceshop:3000/ftp>

Bewertungskriterien:

CVSS 3.1 Vektor und Score: [AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N](#) (5.3)

Q-Scoring: Aufgrund der Tatsache, dass es sich in dem Verzeichnis um sensitive Informationen handelt und die Ausnutzung der Schwachstelle in Kapitel 5.1.1 vereinfacht, wurde die Schwachstelle abweichend vom initialen CVSS Score eingestuft (Erhöhung auf 8.5).

Abgrenzung zu anderen Schwachstellen:

Bitte beachten Sie, dass es sich bei dieser Schwachstelle um eine separate Schwachstelle zu der Schwachstelle aus Kapitel 5.1.1 handelt. Die Behebung der hier beschriebenen Schwachstelle würde nicht die Schwachstelle aus Kapitel 5.1.1 beheben, sondern nur das Auffinden der Dateien erschweren.

Zuständigkeit für die Behebung:

Diese Schwachstelle sollte durch die Administratoren des Webservers behoben werden.

5.2.4 Fehlende Möglichkeit zur Verschlüsselung der Kommunikation

Beschreibung der Schwachstelle:

Der Webserver bietet nicht die Möglichkeit, über einen verschlüsselten Kanal zu kommunizieren. Jegliche Form der Kommunikation zwischen Client und Server erfolgt somit im Klartext.

Potentieller Schaden:

Mit Hilfe eines Man-in-the-Middle (MitM) Angriffs kann ein Angreifer sehr einfach den Datenstrom zwischen einem Client und Server mitlesen und auch manipulieren.

Empfohlene Maßnahme:

Ermöglichen Sie die Kommunikation über einen verschlüsselten Kanal (HTTPS). Im Idealfall sollte eine Kommunikation nur über einen verschlüsselten Kanal möglich sein.

OWASP 2021 Kategorie:

A2 – Cryptographic Failures

5.2.4.1 Proof of Concept Fehlende Möglichkeit zur verschlüsselten Kommunikation

Während der Überprüfung konnte weder durch einen Portscan, noch durch eine Weiterleitung oder sonstige technische Möglichkeiten das Vorhandensein einer Möglichkeit zur verschlüsselten Kommunikation identifiziert werden.

Betroffene Komponente:

Der gesamte Webserver ist betroffen.

Bewertungskriterien:

CVSS 3.1 Vektor und Score: AV:A/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:N (6.4)

Q-Scoring: Das Ausnutzen dieser Schwachstelle erfordert, dass ein Angreifer sich in eine Man-in-the-Middle Position zwischen einem Opfer und dem Server bringen kann. Dies wird im CVSS Vektor für diese Schwachstelle reflektiert. Aus Sicht des Q_PERIOR Offensive Security Service Teams trägt der Score jedoch nicht der Tatsache Rechnung, dass es sich bei der Anwendung um einen Webshop handelt und dort sensitive und personenbezogene Daten verarbeitet werden. Daher wurde eine Einstufung abweichend vom CVSS Score vorgenommen (Erhöhung auf 8.5).

Zuständigkeit für die Behebung:

Diese Schwachstelle sollte durch die Administratoren des Webserver behoben werden. Es sollte jedoch auch eine Abstimmung mit den Anwendungsentwicklern erfolgen, um zu verhindern, dass Links nachfolgend auch auf die richtige Adresse (https) verweisen.

5.3 Findings Risikostufe Mittel

5.3.5 Informationen in Robots.txt Datei

Beschreibung der Schwachstelle:

Auf dem Webserver befindet sich die Datei Robots.txt, die genutzt wird, um den Zugriff von Suchmaschinen-Crawlern zu steuern. In dieser Datei werden oft Verzeichnis- oder Dateinamen festgelegt, die von der automatisierten Indizierung durch Suchmaschinen ausgeschlossen werden sollen.

Potentieller Schaden:

Ein Angreifer kann durch die Inhalte der Robots.txt Hinweise auf die Existenz von Verzeichnissen und Dateien hingewiesen werden, die er sonst nicht oder nur sehr schwer hätte finden könnten.

Empfohlene Maßnahme:

Sofern Sie die Datei Robots.txt nutzen möchten, wird empfohlen, die Namen für Dateien und Verzeichnisse so gut es geht zu kürzen und Wildcards zu nutzen.

Beispielsweise könnte der Eintrag:

```
Disallow: /admindirectory
```

Auf

```
Disallow: /a*
```

geändert werden.

Diese Anpassung ist nicht immer so einfach möglich und wird auch nicht komplett verhindern, dass ein Angreifer ein entsprechendes Verzeichnis oder Datei finden kann. Dennoch erschwert diese Maßnahme das Auffinden solcher Dateien und Verzeichnisse.

OWASP 2021 Kategorie:

A5 – Security Misconfiguration

5.3.5.1 Proof of Concept Informationen in Datei Robots.txt

Die Datei Robots.txt wurde auf dem Server gefunden und konnte aufgerufen werden. Darin war der Eintrag für ein Verzeichnis enthalten:

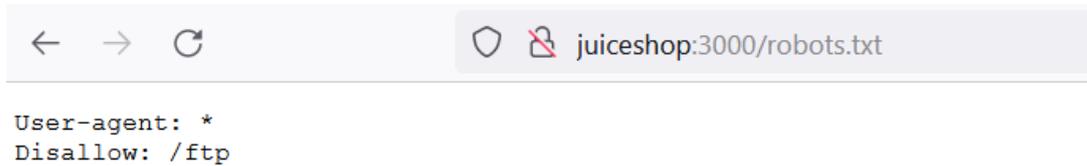


Abbildung 7: Die Datei Robots.txt auf dem Server

Dabei ist zu beachten, dass diese Schwachstelle die Ausnutzung der Schwachstelle aus Kapitel 5.1.1 erleichtert.

Betroffene Komponente:

- <http://juiceshop:3000/robots.txt>

Bewertungskriterien:

CVSS 3.1 Vektor und Score: [AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N](#) (5.3)

Q-Scoring: An dieser Stelle wird keine Abweichung zu dem CVSS Score gesehen.

Zuständigkeit für die Behebung:

Diese Schwachstelle sollte durch die Anwendungsentwickler behoben werden.

5.4 Findings Risikostufe Niedrig

5.4.6 Detaillierte Fehlermeldung

Beschreibung der Schwachstelle:

Bei der Eingabe von unerwarteten Parametern wird eine detaillierte Fehlermeldung vom Server ausgegeben.

Potentieller Schaden:

Unter Umständen könnte ein Angreifer Informationen erhalten, die andere Angriffe erleichtern. Beispielsweise könnten verwendete Softwarekomponenten und deren Version, die Verzeichnisstruktur auf dem Server oder Teile des Quellcodes in einer Fehlermeldung angezeigt werden.

Empfohlene Maßnahme:

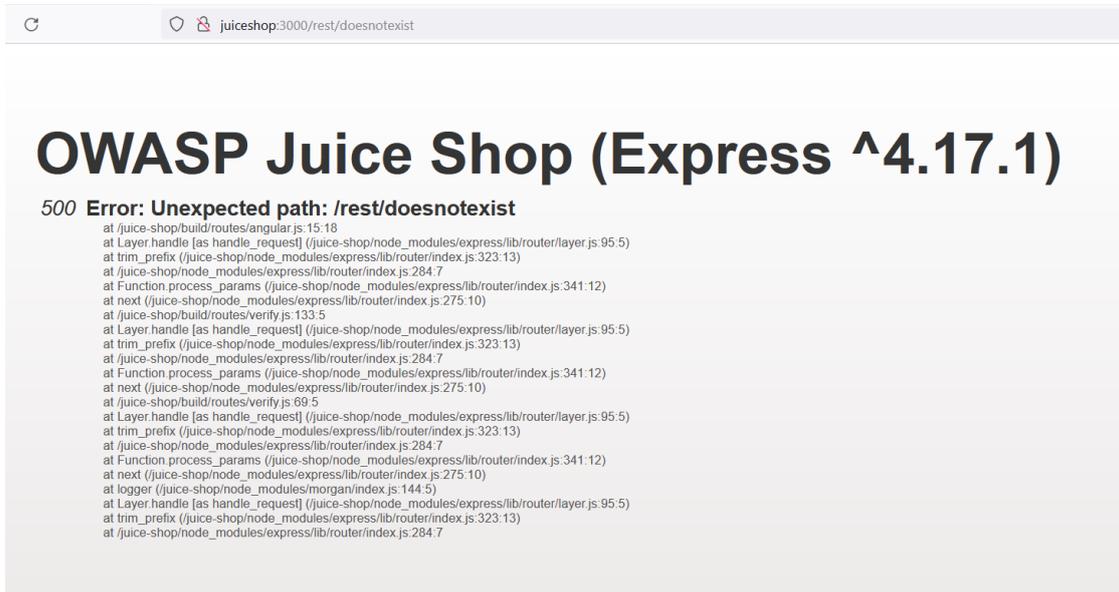
Stellen Sie sicher, dass in einer Fehlermeldung nur eine generische Fehlermeldung gegebenfalls mit einer Referenz angezeigt wird. Die Details zu dem Fehler sollten in ein Logfile geschrieben werden, auf die Anwender keinen Zugriff besitzen. So kann sichergestellt werden, dass Fehler analysiert werden können, aber Informationen nicht potentiellen Angreifern in die Hände fallen.

OWASP 2021 Kategorie:

A5 – Security Misconfiguration

5.4.6.1 Proof of Concept Detaillierte Fehlermeldung

Sofern im Unterverzeichnis `/rest` ein nicht existierendes Verzeichnis angegeben wird, so antwortet der Server mit einer detaillierten Fehlermeldung.



```
juiceshop:3000/rest/doesnotexist

OWASP Juice Shop (Express ^4.17.1)

500 Error: Unexpected path: /rest/doesnotexist
at /juice-shop/build/routes/angular.js:15:18
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/build/routes/verify.js:133:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/build/routes/verify.js:69:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at logger (/juice-shop/node_modules/morgan/index.js:144:5)
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
```

Abbildung 8: Eine detaillierte Fehlermeldung des Anwendungsservers

Für das Beispiel wurde die URL <http://juiceshop:3000/rest/doesnotexist> aufgerufen.

Betroffene Komponente:

Jegliche Fehlermeldungen mit dem Responsecode 500, die vom Anwendungsserver zurückgeliefert werden, sind betroffen.

Bewertungskriterien:

CVSS 3.1 Vektor und Score: [AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N](#) (5.3)

Q-Scoring: Die Informationen, die durch diese Schwachstelle preisgegeben werden, begünstigen keine weiteren Angriffe. Daher ist diese Schwachstelle aus Sicht des Q_PERIOR Offensive Security Service Teams als ein niedrigeres Risiko eingestuft (Reduzierung auf 3.7).

Zuständigkeit für die Behebung:

Diese Schwachstelle sollte durch die Anwendungsentwickler behoben werden.

Anhang 1 – Erklärung Q_SCORE und der Risikoeinstufung

Als Grundlage zur Bewertung von Schwachstellen nutzt Q_PERIOR das Common Vulnerability Scoring System (CVSS) in Version 3.1. Allerdings sind die Einstufungsmöglichkeiten für Schwachstellen in vielen Fällen aus Sicht von Q_PERIOR nicht ausreichend bzw. nicht granular genug.

Daher wird neben dem klassischen CVSS Wert für eine Schwachstelle auch der Q_SCORE angegeben. Der Q_SCORE nimmt den CVSS Wert als Grundlage und modifiziert diesen um weitere Faktoren zu berücksichtigen. Bei diesen Faktoren handelt es sich um:

- Die Möglichkeit, dass eine Schwachstelle andere Schwachstellen bedingt oder deren Ausnutzung erleichtert
- Den Gesamtkontext der Anwendung und die Kritikalität der Daten in der Anwendung bzw. auf die im Rahmen einer Schwachstelle zugegriffen werden kann.
- Die Betrachtung, ob ein Exploit für die Schwachstelle bekannt und veröffentlicht ist, sofern dies zutreffend ist.
- Sofern zutreffend, welche Ausbreitungsmöglichkeiten eine solche Schwachstelle erlauben würde (z.B. Zugriffe auf andere Systeme).

Dies führt dazu, dass Schwachstellen abweichend vom ursprünglichen CVSS Score eingestuft und bewertet werden kann.

Bei einer Abweichung wird eine Begründung, sowie der von Q_PERIOR ermittelte Score angegeben.

Zu beachten ist, dass diese Bewertung auf einer Einschätzung von Q_PERIOR beruht, die nach bestem Wissen und Gewissen auf Basis der ihnen bekannten Informationen getroffen wurde. Die Einschätzung kann durch Kunden aufgrund von internen Kenntnissen noch einmal abweichend bewertet werden.

Für die Unterteilung in die Risikoeinstufung wurde die folgende auf den CVSS Vorgaben basierende Tabelle genutzt:

Risikoeinstufung	Q_SCORE-Wert
 Information	0.0
 Niedrig	0.1 – 3.9
 Mittel	4.0 – 6.9
 Hoch	7.0 – 8.9
 Kritisch	9.0 – 10.0

Die Risikoeinstufungen lassen sich dabei wie folgt beschreiben:

Risikoeinstufung Information:

Bei der Risikoeinstufung handelt es sich um keine mittelbare oder unmittelbare Gefahr für die Anwendung, die Anwender oder die Daten. Ein Finding aus dieser Kategorie ist im Laufe der Tests aufgefallen und der Auditor empfand es zumindest als so relevant, es im Bericht an einer Stelle zu erwähnen.

Risikoeinstufung Niedrig:

Bei einer Schwachstelle in dieser Kategorie handelt es sich um ein sicherheitsrelevantes Finding. Allerdings bietet eine Schwachstelle in dieser Kategorie alleine nur ein sehr geringes Risiko für die Gesamtsicherheit der Anwendung dar und ist keine unmittelbare Gefahr für die Anwendung oder die Benutzer. Dennoch sollte dieses Finding langfristig adressiert und behoben werden, um die Angriffsfläche auf weiter zu reduzieren.

 **Risikoeinstufung Mittel:**

Ein Finding dieser Risikoeinstufung stellen ein deutliches Risiko für die Anwendung bzw. deren Nutzer dar. Zwar erlauben Schwachstellen in dieser Kategorie nicht, auf Daten anderer Benutzer oder Komponenten der Anwendung zuzugreifen, dennoch erhöhen sie die Angriffsfläche in einem solchen Maß, dass sie mindestens mittelfristig behoben werden sollten.

 **Risikoeinstufung Hoch:**

Durch eine Schwachstelle dieser Kategorie besteht ein signifikantes Risiko für die Anwendung bzw. deren Anwender. Ein Angreifer kann mit Hilfe solch einer Schwachstelle zumindest einen Teil der Anwendung unter seine Kontrolle bringen oder auf, als sensitiv einzustufende Daten zugreifen oder diese manipulieren. Daher sollte das Beheben einer solchen Schwachstelle so schnell wie möglich erfolgen.

 **Risikoeinstufung Kritisch:**

Eine Schwachstelle aus dieser Kategorie handelt es sich um eine Schwachstelle, die sehr leicht auszunutzen ist, eine schwerwiegende Auswirkung auf die Anwendung bzw. Anwender hat und/oder den Zugriff auf besonders sensitive oder personenbezogene Daten ermöglicht. Solch eine Schwachstelle sollte unverzüglich behoben werden.