A¹ Digital International GmbH

Security-Assessment

Sicherheitsprüfung der Web-Applikationen

Empfänger:

Example Company GmbH
Musterstraße 7
A-1010 Wien

Klassifizierung: Vertraulich
Datum: 30. Jänner 2019
Version: **1.0**

Kontakt A1 Digital International GmbH:

Max Mustermann
max.mustermann@a1.digital
+43 664 66 32123
Department Security

Lassallestraße 9, A-1020 Wien

1 Änderungsverzeichnis

Datum	Version	Autor	Beschreibung
23.01.2019	0.1	Max Mustermann	Initiale Erstellung
30.01.2019	0.9	Christina Köhler	Review
30.01.2019	1.0	Max Mustermann	Freigabe

Tabelle 1: Änderungsverzeichnis

Inhalt

2 Management Summary

Im Folgenden werden die Ergebnisse des Sicherheitstests kurz zusammengefasst. Ausführliche Beschreibungen der einzelnen Sachverhalte mit Referenzen zu weiteren Ressourcen sowie empfohlenen Gegenmaßnahmen sind in Kapitel 5 zu finden.

2.1 Ergebnisse

Die Login-Funktion des **Shop-Systems** war anfällig für **SQL-Injection Angriffe,** welche einfach zu identifizieren war. Die Schwachstelle ermöglichte **Zugriff auf die Datenbank**– es konnten die Daten von mehr als **1000 Shop-Benutzern** inklusive **Passwort-Hashes** abgezogen werden. Von ca. 300 Benutzern konnte aufgrund von schwachen Passwörtern das **Klartextpasswort** ausfindig gemacht werden.

Die Subdomain https://takeover.example.com hatte einen CNAME zu **someotherdomain.trafficmanager.net** gesetzt. Zum Zeitpunkt der Prüfung war **someotherdomain.trafficmanager.net** nicht vergeben und könnte somit über AZURE registriert werden. Das bedeutet, dass die Subdomain **takeover.example.com** unter die Kontrolle von Angreifern gelangen könnte und beispielsweise für Phishing-Angriffe verwendet werden kann.

Im Zuge der Prüfung wurde eine **Reflected Cross-Site-Scripting (XSS)** Schwachstelle auf www.example.com identifiziert, welche einfach ausnutzbar war. Angreifer könnten durch erfolgreiche Ausnutzung dieser Schwachstelle eventuell **Session-Informationen stehlen** und somit den Account des Opfers übernehmen.

Auf dem **Shop-System** kam eine **veraltete Tomcat-Version** (v8.0.41) zum Einsatz. Diese Version beinhaltet bekannte Schwachstellen, welche beispielsweise den Zugriff auf private Daten eines Nutzers erlauben können.

Zum Zeitpunkt der Prüfung verfügte die E-Mail-Ändern-Funktion der Example-Applikation über keinen **Cross-Site Request Forgery** Schutz. Dies hat zur Folge, dass Angreifer potentiell die Möglichkeit haben, angemeldete Opfer zu Aktionen in deren Konto zu bewegen. Beispielsweise könnten Angreifer die E-Mail-Adresse des Opfers ändern um anschließend das Passwort zurückzusetzen.

Weitere Schwachstellen umfassen **nicht optimale SSL-Einstellungen** auf den geprüften Systemen sowie **nicht gesetzte Sicherheitsheader**. Weitere Details können dem Kapitel 5 entnommen werden.

2.2 Empfohlene nächste Schritte

- Es sollte eine tiefgehende Analyse der Shop-Applikation durchgeführt werden (inklusive Source-Code Review), da mit der SQL-Injection eine kritische Schwachstelle identifiziert werden konnte und nicht auszuschließen ist, dass ähnliche Schwachstellen an anderen Stellen in der Applikation weiterhin existieren.
- Nach größeren Änderungen in der Infrastruktur, bei Berechtigungskonzepten, oder in gewissen Applikationen, sollte ein Regressionstest durchgeführt werden um zu prüfen, ob etwaige Schwachstellen auch tatsächlich behoben wurden.

2.3 Schwachstellenübersicht

In der folgenden Tabelle wird eine Übersicht der identifizierten Sachverhalte geliefert sowie eine Einschätzung der A1 Digital International GmbH, wie komplex die Umsetzung von Gegenmaßnahmen sein könnte. Abbildung 1 stellt die identifizierten Schwachstellen diagrammisch dar.

Schwachstelle	Risiko (laut CVSS3)	Gegenmaßnahmen	Kommentar
SQL-Injection	Critical (9.8)	Mittel	
Subdomain Takeover	Medium (6.5)	Leicht	
Reflected Cross-Site-Scripting	Medium (6.1)	Mittel	
Veraltete Tomcat-Installation	Medium (5.9)	Leicht	
Fehlende Security Header	Medium (5.4)	Mittel	
Cross-Site Request Forgery	Medium (5.0)	Mittel	
Schwache SSL Einstellungen	Medium (4.4)	Leicht	

Tabelle 2: Schwachstellenübersicht

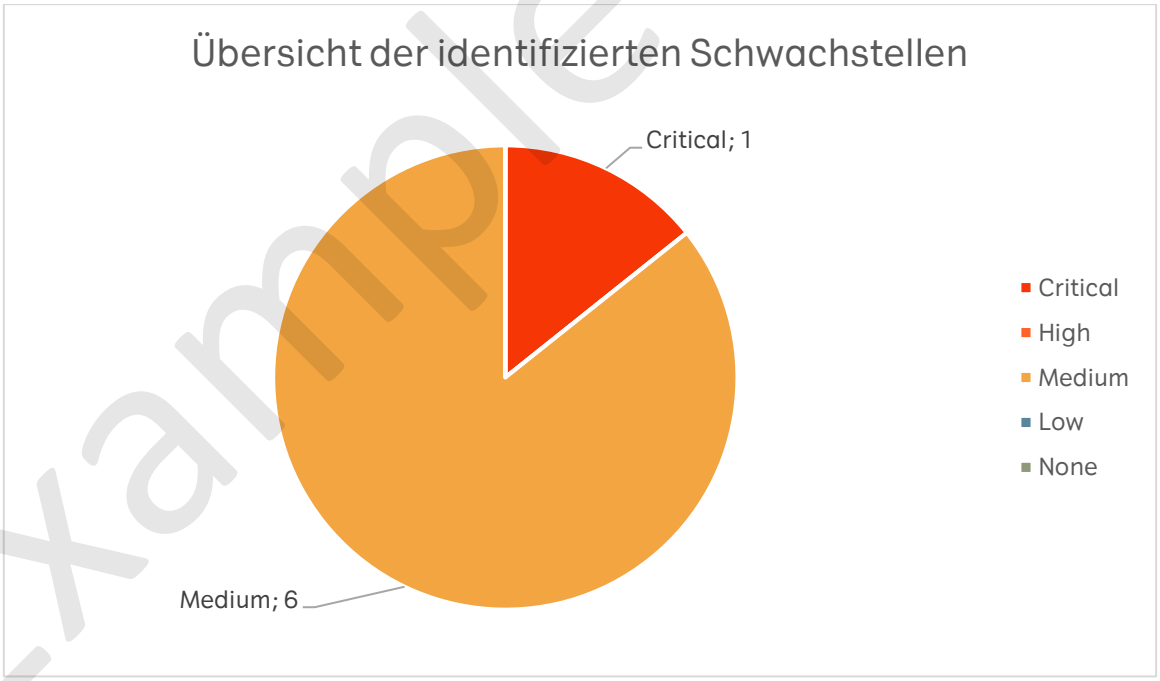


Abbildung 1: Übersicht der identifizierten Schwachstellen

2.3.1 Schwachstellen-Kategorisierung

Es wurde eine grobe Kategorisierung der identifizierten Schwachstellen durchgeführt, um einen Überblick geben zu können, in welchen Gebieten die meisten sicherheitsrelevanten Feststellungen identifiziert werden konnten. Es handelt sich dabei um die folgenden Schwachstellen-Kategorien:

- **Configuration Issue:** Fehler in der Konfiguration von Software- oder Hardware-Komponenten.
  - Sollten in dieser Kategorie gehäuft Schwachstellen identifiziert worden sein, könnten Schulungen für System-Administratoren zur sicheren Konfiguration der betreuten Komponenten Abhilfe schaffen.

- **Outdated Software:** Veraltete Software-Komponenten mit bekannten sicherheitsrelevanten Problemen.
  - Sollte gehäuft veraltete Software ein Problem darstellen, wird die Etablierung eines kontinuierlichen Update- und Patchmanagement- Prozesses empfohlen, um sicherheitskritische Updates zeitnah einspielen zu können.

- **Input Validation/Output Encoding:** Fehlende Validierung von Benutzereingaben bzw. fehlende korrekte Enkodierung von Ausgaben der Software.
  - Häufige Fehler in dieser Kategorie lassen auf fehlende Schulungen im Secure Coding-Bereich schließen. Regelmäßige Secure Coding-Schulungen für Software-Entwickler könnten die Sicherheit und die Software-Qualität erhöhen.

- **Other:** Nicht eindeutig zuordenbar zu den obigen drei Kategorien.

Folgende Tabelle zeigt die Schwachstellenkategorisierung der identifizierten Findings:

Schwachstelle	Kategorie
SQL-Injection	Input Validation/Output Encoding
Subdomain Takeover	Configuration Issue
Reflected Cross-Site-Scripting	Input Validation/Output Encoding
Veraltete Tomcat-Installation	Outdated Software
Fehlende Security Header	Configuration Issue
Cross-Site Request Forgery	Other
Schwache SSL Einstellungen	Configuration Issue

Tabelle 3: Schwachstellen-Kategorisierung

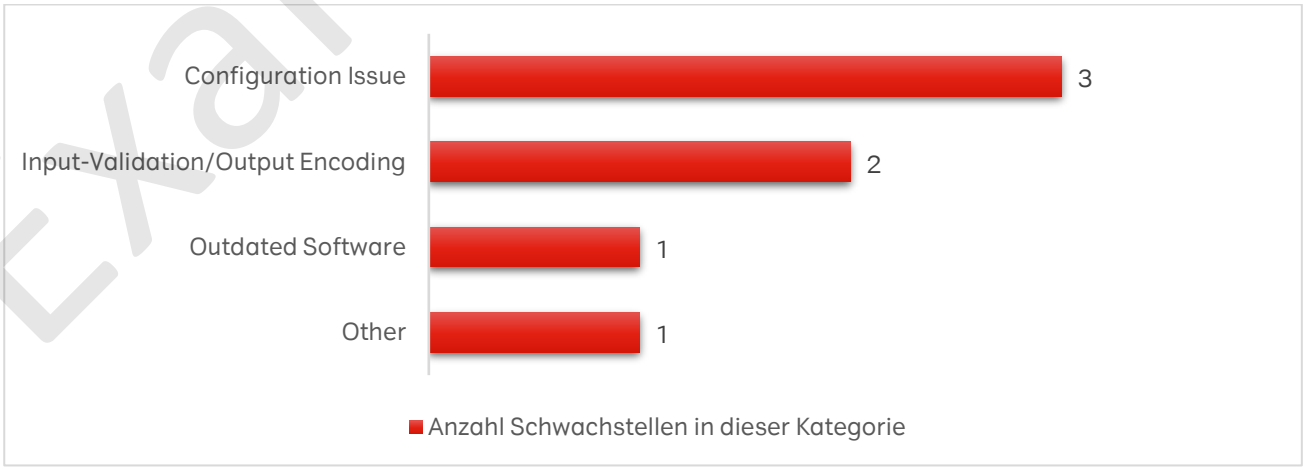


Abbildung 2: Schwachstellen-Kategorisierung

2.4 Disclaimer

Der Aufwand für diese Prüfung wurde nach einem Timebox-Ansatz geschätzt was bedeutet, dass etwaige Schwachstellen nur innerhalb des vereinbarten Zeitfensters identifiziert wurden. Das Ziel dabei war es, möglichst alle sicherheitsrelevanten Schwachstellen der zu prüfenden Systeme aufzudecken und zu dokumentieren – jedoch besteht keinerlei Anspruch auf Vollständigkeit der im Bericht aufgeführten Sachverhalte.
Diese Sicherheitsprüfung stellt eine Momentaufnahme zum Zeitpunkt der Prüfung dar, was bedeutet, dass sich keine zukünftigen IT-Sicherheits-Risiken davon ableiten lassen.

3 Scope

Example Company GmbH beauftrage die A1 Digital International GmbH mit der Durchführung einer Sicherheitsprüfung der unten angeführten Web-Applikationen.

Die Prüfung fand zwischen **23.01.2019** und **30.01.2019** statt.
Eine detailliertere Beschreibung zur Vorgehensweise finden Sie in Kapitel 4.

3.1 Geprüfte Systeme

Die folgenden Systeme wurden im externen Assessment berücksichtigt.

IP	Hostname	Kommentar
1.1.1.1	www.example.com	Hauptseite
1.1.1.1	shop.example.com	Onlineshop

Tabelle 4: Geprüfte Systeme

3.2 Verwendende Benutzerkonten

Seitens des Kunden wurden **keine Benutzerkonten zur Verfügung gestellt**. Im Zuge der Prüfung war es möglich, selbst Nutzer im Webshop zu registrieren, welche dann für die Sicherheitsprüfungen verwendet wurden.

4 Vorgehensweise

Um ein möglichst breites Spektrum an möglichen Schwachstellenkategorien abdecken zu können, wurde der Test in Anlehnung an den Open Web Application Security Project (OWASP) Testing Guide Version 4 durchgeführt (siehe Kapitel 6.6). Ziel war die Identifikation möglichst aller sicherheitsrelevanten Schwachstellen, welche zum Zeitpunkt der Prüfung auf den getesteten Systemen vorhanden waren.

Im Vorfeld des Tests wurden einige Kriterien bestimmt, welche eine Klassifizierung des durchgeführten Penetrationtests erlauben. Folgende Abbildung orientiert sich an der Studie „Durchführungskonzept für Penetrationtests“[1] des BSI und soll den Ansatz für die Durchführung dieser Prüfung wiederspiegeln.

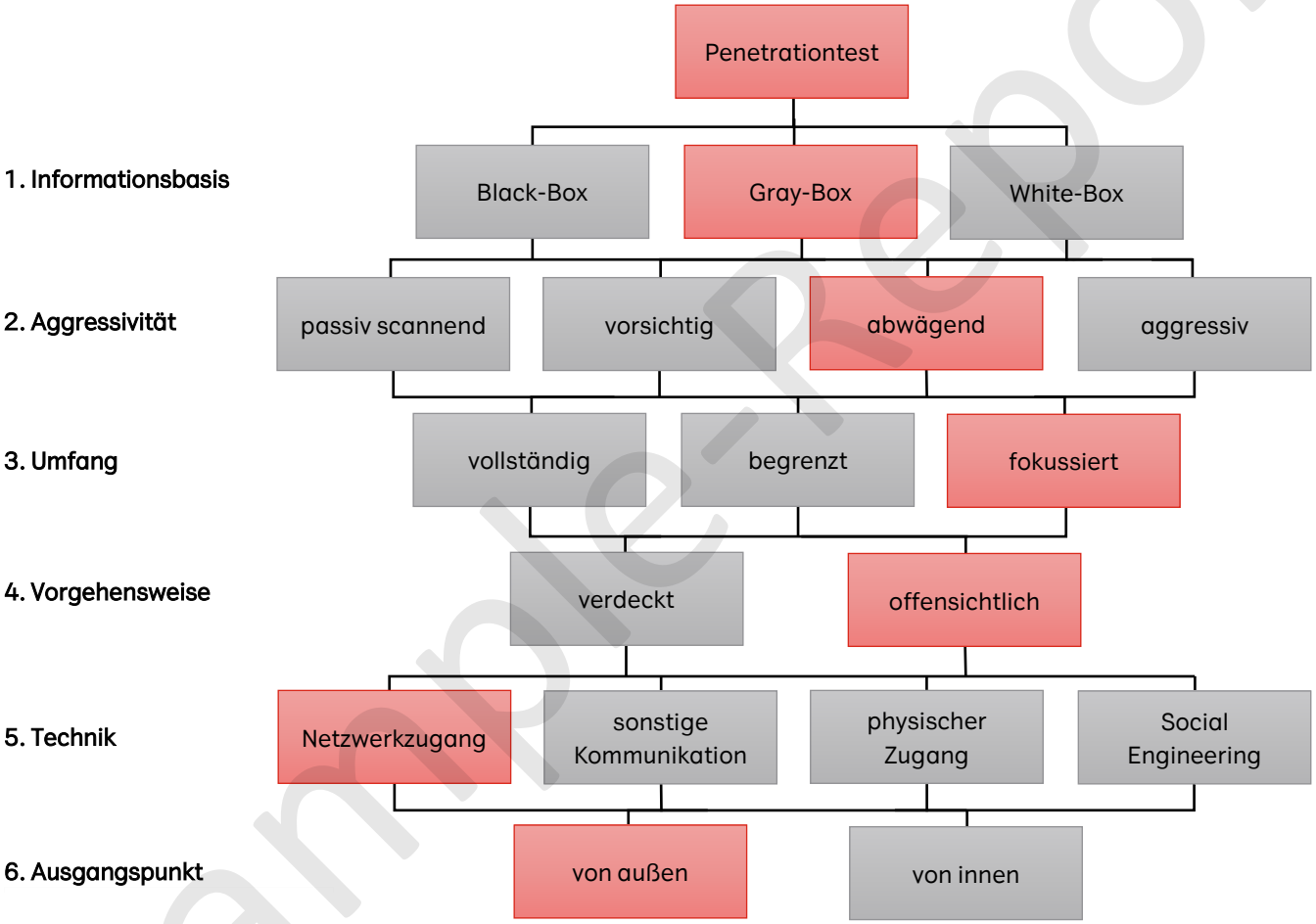


Abbildung 3: Durchführungskonzept für Penetrationtests

4.1 Risikobewertung nach CVSSv3

Das Common Vulnerability Scoring System (CVSS) bietet die Möglichkeit, die grundlegenden Charakteristiken einer Schwachstelle abzubilden bzw. zu bewerten. Das Ergebnis ist ein numerischer Wert, welcher sich zwischen **0.0 und 10.0** bewegen kann, wobei **10.0** als der höchste und somit als der kritischste definiert ist. Eine detaillierte Beschreibung der CVSS-Metriken ist in Kapitel 6.2 zu finden. Um das Risiko in Worten ausdrücken zu können, werden fünf verschiedene Wertebereiche definiert, welche in Kapitel 6.3 beschrieben sind. Ein Risiko kann danach mit „**None**“, „**Low**“, „**Medium**“, „**High**“ und „**Critical**“ bewertet werden.

[1] https://www.bsi.bund.de/DE/Publikationen/Studien/Pentest/index_htm.html

5 Identifizierte Schwachstellen

Im Folgenden werden die im Zuge der Prüfung identifizierten Schwachstellen beschrieben und mit einem Risiko bewertet. Diese Risikoeinschätzung erfolgt nach dem CVSSv3-Standard und wird vom Tester nach bestem Wissen und Gewissen vorgenommen. Die Bewertungen des Risikos können also von den Bewertungen des Auftraggebers abweichen, da der Tester in den meisten Fällen nicht über genügend Hintergrundwissen verfügt, um eine konkrete Einschätzung des Business-Risikos vornehmen zu können.
Jede Schwachstellenbeschreibung enthält empfohlene Gegenmaßnahmen sowie Referenzen auf externe Ressourcen für nähere Details.

5.1 SQL-Injection

CVSSv3 Score	**Critical (9.8)**
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Betroffene Systeme

- https://shop.example.com (1.1.1.1:443 (TCP))

Beschreibung

Die Login-Funktion war zum Zeitpunkt der Prüfung anfällig für SQL-Injection Angriffe. Dies ermöglichte es, Zugriff auf die Datenbank zu erlangen – es konnten die Daten von mehr als 1000 Shop-Benutzern inklusive Passwort-Hashes abgezogen werden. Von ca. 300 Benutzern konnte das Klartextpasswort ausfindig gemacht werden.

Empfehlung

- Um SQL-Injection Angriffe zu unterbinden sollen folgende Maßnahmen getroffen werden:
  - Einsatz von Prepared-Statements und Stored-Procedures, um das Einschleusen von SQL-Kommandos durch Angreifer zu unterbinden.
  - Whitelisting von erlaubten Eingabewerten.
  - Escaping von jeglichen Benutzereingaben.
- Es sollte evaluiert werden, ob die beschriebe Schwachstelle bereits in der Vergangenheit erfolgreich ausgenutzt wurde. Wenn ja, sollte evaluiert werden, welche Daten durch diesen Angriff abhandengekommen sein könnten. Gegebenenfalls empfehlen wir die Kontaktaufnahme mit der zuständigen Datenschutzbehörde.
- Es sollte eine tiefgehende Prüfung aller Applikationen durchgeführt werden, um sicherzustellen, dass derartige Angriffe auf keinen anderen Systemen möglich sind.
- Es sollten regelmäßige Secure Coding-Schulungen abgehalten werden, um die Entwickler über etwaige Sicherheitsrisiken zu informieren.

Technische Beschreibung

Die Login-Funktion war zum Zeitpunkt der Prüfung anfällig für SQL-Injection Angriffe.
Der anfällige Parameter dabei war:

- **Username**

Folgende Abbildung zeigt den Request an die Applikation, welcher einen SQL-Fehler hervorruft:

```
POST /cgi-bin/badstore.cgi?action=login HTTP/1.1
Host: store.example.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101
Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://store.example.com/cgi-bin/badstore.cgi?action=loginregister
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Connection: close
Upgrade-Insecure-Requests: 1

username=%27&passwd=aaa
```

Abbildung 4: Request an die Applikation, welcher einen SQL-Fehler hervorruft

Folgende Abbildung zeigt den SQL-Syntax Fehler in der Applikation:

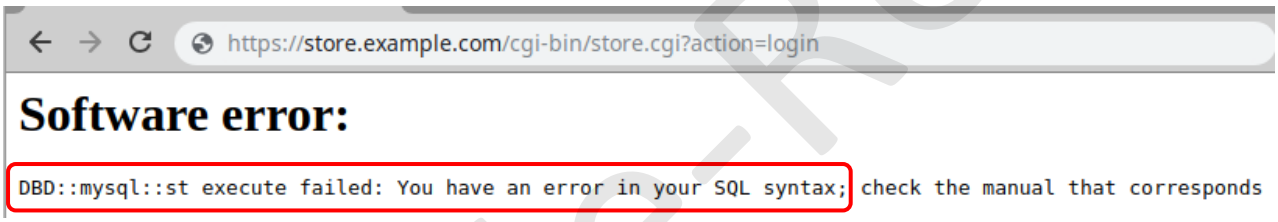


Abbildung 5: SQL-Injection in store.example.com

Im Zuge der Prüfung konnte die Schwachstelle erfolgreich und ohne Authentifizierung ausgenutzt werden. Die Daten von mehr als 1000 Store-Benutzern konnten ausgelesen werden, wobei auf Klarnamen, Adressdaten und gehashte Passwörter zugegriffen werden konnte.

Im Folgenden ist der Zugriff auf die Datenbank zu sehen:

```
Database: storedb
Table: userdb
[5 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| email    | varchar(40)  |
| fullname | varchar(50)  |
| address  | varchar(50)  |
| passwd   | varchar(32)  |
| pwdhint  | varchar(8)   |
| role     | char(1)      |
+----------+--------------+
```

Wie im Folgenden gezeigt wird, befinden sich über 1000 Benutzerdaten in der Datenbank:

```
Database: storedb
+--------+---------+
| Table  | Entries |
+--------+---------+
| userdb | 1241    |
+--------+---------+
```

Die Passwörter von mehr als 300 Benutzern konnten aufgrund von schwachen Passwörtern zurückgerechnet werden.

Weiterführende Informationen / Referenzen

- https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)
- https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
- https://www.owasp.org/index.php/SQL_Injection

5.2 Subdomain Takeover

CVSSv3 Score	Medium (6.5)
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

Betroffene Systeme

- https://takeover.example.com

Beschreibung

Die Subdomain `takeover.example.com` hatte zum Zeitpunkt der Prüfung einen CNAME zu `someotherdomain.trafficmanager.net` gesetzt. Zum Zeitpunkt der Prüfung war `someotherdomain.trafficmanager.net` nicht vergeben und könnte somit über AZURE registriert werden. Das bedeutet, dass die Subdomain `takeover.example.com` unter die Kontrolle von Angreifern gelangen könnte und beispielsweise für Phishing-Angriffe verwendet werden kann.

Empfehlung

- Alle nicht benutzten CNAME Einträge sollten entfernt werden.
- Falls dies nicht möglich ist, sollte der CNAME wieder übernommen werden, um mögliche Fremdübernahmen zu verhindern.
- Weiters sollte ein Lifecycle-Management Prozess für Domains erstellt oder adaptiert werden.

Technische Beschreibung

Abbildung 6: Subdomain Takeover von https://takeover.example.com

Im Folgenden ist der konfigurierte CNAME-Record zu sehen:

```
# nslookup takeover.example.com 8.8.8.8
Server:     8.8.8.8
Address:    8.8.8.8#53

Non-authoritative answer:
takeover.example.com    canonical name = someotherdomain.trafficmanager.net.
Name:   someotherdomain.trafficmanager.net
Address: 54.192.96.244
```

In weiterer Folge konnte `someotherdomain.trafficmanager.net` über AZURE registriert werden wodurch die volle Kontrolle über die Subdomain `takeover.example.com` möglich war.

Weiterführende Informationen / Referenzen

- https://blog.sweepatic.com/subdomain-takeover-principles/
- https://0xpatrik.com/subdomain-takeover-basics/

5.3 Reflected Cross-Site-Scripting

CVSSv3 Score	Medium (6.1)
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

Betroffene Systeme

- www.example.com (1.1.1.1:443 (TCP))

Beschreibung

Im Zuge der Prüfung wurde eine Reflected Cross-Site-Scripting (XSS) Schwachstelle identifiziert, welche praktisch einfach ausnutzbar war. Angreifer könnten durch erfolgreiche Ausnutzung dieser Schwachstelle eventuell Session-Informationen stehlen und somit den Account des Opfers übernehmen.

Empfehlung

- Es sollte sichergestellt werden, dass Benutzereingaben validiert und bei einer Ausgabe im Quelltext enkodiert werden (Input Validation, Output Encoding).
- Beim Output Encoding ist besonders auf folgende Zeichen zu achten:

Zeichen	Enkodiertes Zeichen
&	&amp;
<	&lt;
>	&gt;
“	&quot;
‘	&#x27;
/	&#x2F;

- Bevor Benutzerinput direkt in JavaScript ausgegeben wird, muss der Benutzerinput escaped werden, um das Einschleusen von bösartigem JavaScript Code zu verhindern.

Technische Beschreibung

Im Zuge der Prüfung wurde in der **Example-Applikation** eine Reflected Cross-Site-Scripting Schwachstelle identifiziert. Folgende Parameter waren zum Zeitpunkt der Prüfung anfällig:

- **parameter1**

Folgende URL dient als Proof of Concept:

- https://www.example.com/app?parameter1=”><script>alert(document.domain)</script>

Folgender Screenshot zeigt die Ausführung von JavaScript Code im Kontext der betroffenen Webseite.

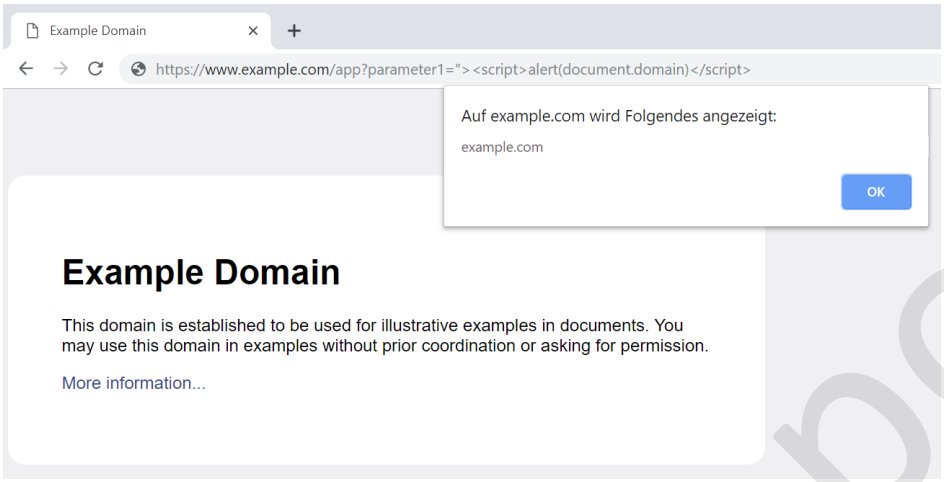


Abbildung 7: Reflected XSS auf example.com

Angreifer könnten dies potentiell nutzen, um Session-Informationen des betroffenen Nutzers auszulesen und in weiterer Folge den Account zu übernehmen.

Weiterführende Informationen / Referenzen

- https://www.owasp.org/index.php/Testing_for_Reflected_Cross_site_scripting_(OTG-INPVAL-001)
- https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet
- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

5.4 Veraltete Tomcat-Installation

CVSSv3 Score	Medium (5.9)
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L/E:U/RL:O/RC:U

Betroffene Systeme:

- https://shop.example.com (1.1.1.1:443 (TCP))

Beschreibung

Zum Zeitpunkt der Prüfung kam auf dem Shop-System eine veraltete Tomcat-Version (v8.0.41) zum Einsatz. Diese Version beinhaltet bekannte Schwachstellen, welche beispielsweise den Zugriff auf private Daten eines Nutzers erlauben können.

Empfehlung

- Es wird empfohlen, die veraltete Software auf einen aktuellen Stand zu bringen.
- Es sollte ein kontinuierlicher Updateprozess etabliert werden, welcher garantiert, dass sicherheitskritische Updates zeitnah eingespielt werden können.
- Sollten Updates nicht möglich sein, sollten die betroffenen Systeme isoliert und abgeschottet werden, um den Zugriff zu erschweren bzw. zu verhindern.
- Es sollten generische Fehlermeldungen an Endbenutzer ausgeliefert werden, welche keine Rückschlüsse auf eingesetzte Soft- oder Hardware zulassen.

Technische Beschreibung

Wurde auf https://shop.example.com im Webroot eine nicht existierende Seite aufgerufen, offenbarte die Applikation die eingesetzte Tomcat-Version, wie in folgender Abbildung zu sehen ist:

Abbildung 8: Eingesetzte Tomcat Version wird offenbart

Die Version 8.0.41 beinhaltet mehrere bekannte Schwachstellen, welche von Angreifern eventuell missbraucht werden können, um weitere Angriffe vereinfacht durchzuführen oder auf private Daten von angemeldeten Benutzern zugreifen zu können.
Weitere Informationen zu den bekannten Schwachstellen in dieser Tomcat-Version sind in den Referenzen zu finden.

Weiterführende Informationen / Referenzen

- https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-887/version_id-212327/Apache-Tomcat-8.0.41.html
- https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities
- https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration

5.5 Fehlende Security-Header

CVSSv3 Score	Medium (5.4)
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:N

Betroffene Systeme:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Beschreibung

Im Zuge der Prüfung wurde festgestellt, dass die betroffenen Web-Applikationen nicht durchgehend gängige Sicherheits-Header implementiert hatten. Das Setzen von Sicherheits-Headern kann die allgemeine Sicherheit einer Web-Applikation erhöhen und es Angreifern erschweren, beispielsweise Cross-Site-Scripting Angriffe durchzuführen.

Empfehlung

- Es sollte evaluiert werden, ob die empfohlenen Sicherheits-Header durchgehend gesetzt werden können.
- Insbesondere die Content-Security-Policy (CSP) kann einen guten Schutz gegen Cross-Site-Scripting Angriffe bieten. Jedoch kann die Konfiguration der CSP komplex sein und unter Umständen Fehler in der Web-Applikation hervorrufen. Daher wird empfohlen, initial die CSP mit „Report-Only“ zu testen.

Technische Beschreibung

Im Folgenden werden die Sicherheits-Header, welche zum Zeitpunkt der Prüfung nicht (durchgehend) gesetzt waren, im Detail beschrieben:

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **Content-Security Policy Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Die Content-Security Policy wird benutzt, um beispielsweise Cross-Site-Scripting und Framing-Angriffe über Zugriffs-Policies einzuschränken, zu reporten und zu verhindern. Die Quelleneinschränkung sollte nur direkt kontrollierte Adressen erlauben, von den „unsafe“ Optionen wird dringendst abgeraten. Vorerst sollte mit „Report-Only“ und „self“ als Origin gestartet werden um zu sehen welche externen Requests benötigt werden. Nach genauerer Spezifikation der Origins wird empfohlen, die CSP blockierend durchzusetzen.

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **X-Frame-Options Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Der X-Frame-Options Header spezifiziert, ob die Seite in einer anderen Seite als „frame“, „iframe“ oder „embed“ eingebunden werden darf. Damit werden sogenannte „Clickjacking“-Attacken verhindert, in denen Nutzer dazu gebracht werden können, auf Elemente zu klicken, die hinter anderen Elementen auf fremden Webseiten versteckt sind. Es wird empfohlen, zumindest die Einbettung auf externe und wenn möglich auch die auf Seiten unter derselben Domain zu unterbinden.

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **X-XSS-Protection Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Der X-XSS-Protection Header instruiert den Browser, mögliche Cross-Site-Scripting-Angriffe zu erkennen und zu blockieren. Es wird die Option „X-XSS-Protection: 1;“ bzw. „X-XSS-Protection: 1; mode=block“ empfohlen.

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **X-Content-Type-Options Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Der X-Content-Type-Options Header kann mit der Option „nosniff“ die automatische Erkennung und Korrektur von MIME-Typen für JavaScript und CSS-Dateien im Browser deaktivieren und damit Schwachstellen blockieren, bei denen aus anderen Dateien vermeintliches JavaScript geladen werden kann. Es wird geraten, den Header mit „nosniff“ zu aktivieren.

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **Referrer-Policy Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Der Referrer-Policy Header gibt an, welche „Referer“ [sic] zu welchen Ziel-Seiten gesendet werden sollen. Damit kann verhindert werden, dass genaue Informationen über die Herkunft von Usern an externe Seiten oder Seiten ohne Verschlüsselung weitergegeben werden. Diese Informationen könnten problematisch sein, wenn in ihnen Session Tokens, Namen, IDs und andere sensible Daten enthalten sind. Es wird empfohlen, den Header mit „strict-origin-when-cross-origin“ zu senden, um ein Tracking der Nutzer zu minimieren. Damit wird auf der selben Origin der gesamte URL, zwischen zwei Origins mit gleicher Sicherheit (HTTPS->HTTPS) nur der Origin und zwischen zwei Origins mit unterschiedlicher Sicherheit (HTTPS->HTTP) nichts gesendet.

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **Feature-Policy Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Der Feature-Policy Header kontrolliert, welche JavaScript Quellen welche Browser Features von der Seite benutzen dürfen. Darunter fallen die Benutzung der Kamera, des Mikrofons, Geolocation und Payment Requests, welche standardmäßig deaktiviert sein sollten. Es wird geraten, so viel wie möglich zu deaktivieren, da es damit für Angreifer schwieriger wird, Daten über Nutzer zu sammeln.

Folgende Applikationen hatten zum Zeitpunkt der Prüfung den **HTTP-Strict-Transport-Security (HSTS) Header** nicht (durchgehend) gesetzt:

- https://www.example.com (1.1.1.1:443 (TCP))
- https://shop.example.com (1.1.1.1:443 (TCP))

Der HSTS-Header schützt vor Aushebelung der Verschlüsselung indem er für eine Domain und optional dessen Subdomains für einen gewissen Zeitraum festlegt, dass diese nur verschlüsselt aufgerufen werden dürfen. Dafür ist der Strict-Transport-Security Header mit einer Zeit zu senden, für die diese Restriktion gelten soll. Es wird geraten, diesen Header mit einer Zeit von einem Jahr zu setzen.

Weiterführende Informationen / Referenzen

- https://securityheaders.com/
- https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet
- https://www.owasp.org/index.php/Security_Headers
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security

5.6 Cross-Site Request Forgery (CSRF)

CVSSv3 Score	Medium (5.0)
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N/IR:H

Betroffene Systeme

- https://shop.example.com (1.1.1.1:443 (TCP))

Beschreibung

Zum Zeitpunkt der Prüfung verfügte die E-Mail-Ändern-Funktion der Example-Applikation über keinen Cross-Site Request Forgery-Schutz. Dies hat zur Folge, dass Angreifer potentiell die Möglichkeit haben, angemeldete Opfer zu Aktionen in deren Konto zu bewegen. Beispielsweise könnten Angreifer die E-Mail-Adresse des Opfers ändern um anschließend das Passwort zurückzusetzen.

Empfehlung

- Es sollte ein CSRF-Schutz implementiert werden. Dies wird üblicherweise über sogenannte CSRF-Tokens gelöst, welche einer Session zugeordnet sind und bei jedem schreibenden Request als Body- oder Header-Wert mitgesendet und serverseitig validiert werden. Da ein Angreifer den Wert des CSRF-Tokens nicht kennt, hat dieser nicht mehr die Möglichkeit, diesen Angriff durchzuführen.
- Das Ändern der E-Mail-Adresse sollte nach Möglichkeit nur nach vorheriger Eingabe des aktuellen Passwortes möglich sein.
    - Sollte dies nicht möglich sein, sollte nach der Änderung der E-Mail-Adresse eine Benachrichtigungsmail mit der Möglichkeit, die Änderung abzubrechen, an die alte E-Mail-Adresse gesendet werden. Die neue E-Mail-Adresse sollte erst nach einer gewissen Zeitspanne (z. B. 24 Stunden) akzeptiert werden.

Technische Beschreibung

Die E-Mail-Ändern-Funktion der Example-Applikation schien zum Zeitpunkt der Prüfung keinen CSRF Schutz implementiert zu haben.
Angreifer könnten daher bei einem erfolgreichen Angriff die E-Mail-Adresse des Benutzers ändern und im Anschluss das Passwort des Accounts zurücksetzen.

Folgendes Formular demonstriert den Angriff:

```
<form action="https://shop.example.com/account/edit_profile" method="post" name="main">
<input type="hidden" name="user_id" value="1234">
<input type="hidden" name="email" value="attacker@evil.com">
<input type="hidden" name="btn_save" value="Save">
</form><script>document.main.submit();</script>
```

Weiterführende Informationen / Referenzen

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet

5.7 Schwache SSL-Einstellungen

CVSSv3 Score	Medium (4.4)
CVSSv3 Vektor String	CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N/CR:H

Betroffene Systeme

- www.example.com (1.1.1.1:443 (TCP))

Beschreibung

Im Zuge der Prüfung stellte sich heraus, dass `www.example.com` zum Zeitpunkt der Prüfung keine sicheren SSL-Einstellungen konfiguriert hatte. Angreifer mit Zugriff auf den Netzwerkverkehr hätten potentiell die Möglichkeit, die Verschlüsslung zu brechen und somit Zugriff auf sensible Daten wie Nutzernamen und Passwörter zu erlangen.

Empfehlung

- Die als unsicher geltenden Ciphersuites sollten deaktiviert werden, da es für diese bekannte Schwachstellen und Angriffe gibt (siehe unten).
- Nach Möglichkeit solle nur noch TLS 1.2 und TLS 1.3 zum Einsatz kommen.
- Anleitungen zur sicheren SSL/TLS Konfiguration können dem Paper „Applied Crypto Hardening“ auf bettercrypto.org entnommen werden (siehe Referenzen).

Technische Beschreibung

Zum Zeitpunkt der Prüfung konnten auf den betroffenen Systemen einige unsichere SSL-Einstellungen identifiziert werden. Die Prüfung der SSL-Einstellungen wurde mithilfe von Qualis SSL Labs[2] am 29.01.2019 durchgeführt. Die Prüfung ergab folgende Ergebnisse:

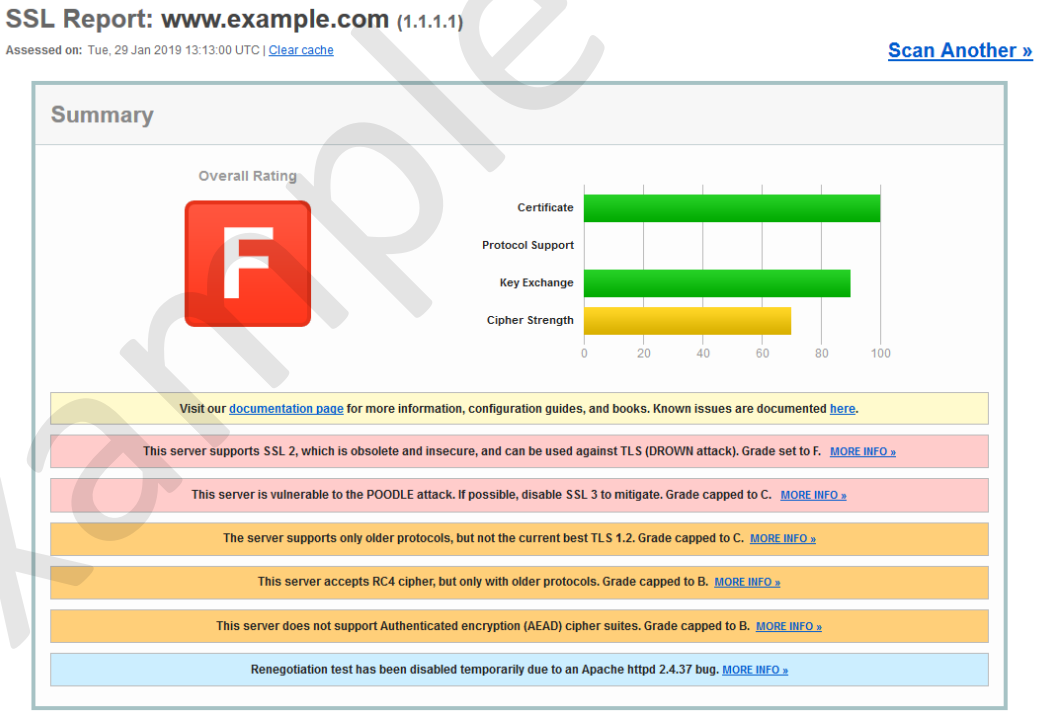


Abbildung 9: www.example.com Ergebnisse SSL Test

Einsatz von SSLv2 und SSLv3: Das betroffene System unterstützte zum Zeitpunkt der Prüfung SSLv2 und SSLv3, wobei es sich um veraltete Verschlüsselungsprotokolle handelt, welche in produktiven Umgebungen nicht mehr

[2] https://www.ssllabs.com

zum Einsatz kommen sollten. Angreifer mit Zugriff auf den Netzwerkverkehr könnten versuchen, die schwache Verschlüsselung zu brechen und somit Zugriff auf sensible Klartextdaten erhalten.

Einsatz weiterer schwacher Verschlüsselungsmethoden: Zum Zeitpunkt der Prüfung unterstützte das betroffene System neben SSLv2 und v3 auch noch TLSv1.0 und RC4 Cipher. RC4 ist als unsicher zu betrachten und sollte deaktiviert werden. TLSv1.0 gilt ebenfalls als unsicheres Protokoll – seit 30. Juni 2018 ist der Einsatz von TLSv1.0 beispielsweise nicht mehr PCI DSS Compliant, da es bekannte Angriffe gegen TLSv1.0 gibt.

Kein Einsatz von TLSv1.2 oder TLSv1.3: Das betroffene System unterstützte zum Zeitpunkt der Prüfung kein TLSv1.2 oder TLSv1.3. Diese modernen Protokolle sollten nach Möglichkeit unterstützt werden, um eine sichere Verbindung für die Nutzer zu gewährleisten.

Keine Unterstützung von sogenannten AEAD Cipher Suiten: AEAD (Authenticated Encryption with Associated Data) Ciphers sind als sicher zu betrachtende Cipher-Suiten. Beispielsweise setzt TLSv1.3 nur noch auf AEAD Cipher Suiten.

Weiterführende Informationen / Referenzen

- https://bettercrypto.org/static/applied-crypto-hardening.pdf
- https://blog.qualys.com/ssllabs/2014/10/15/ssl-3-is-dead-killed-by-the-poodle-attack

6 Anhang

6.1 Ansprechpersonen

A1 Digital International GmbH

Name	Rolle	Telefon	E-Mail
Max Mustermann	Durchführung der Sicherheitsprüfung	+43 664 66 32123	max.mustermann@a1.digital
Christina Köhler	Review des Berichts	+43 664 66 32124	christina.koehler@a1.digital

Tabelle 5: Ansprechpersonen A1 Digital International GmbH

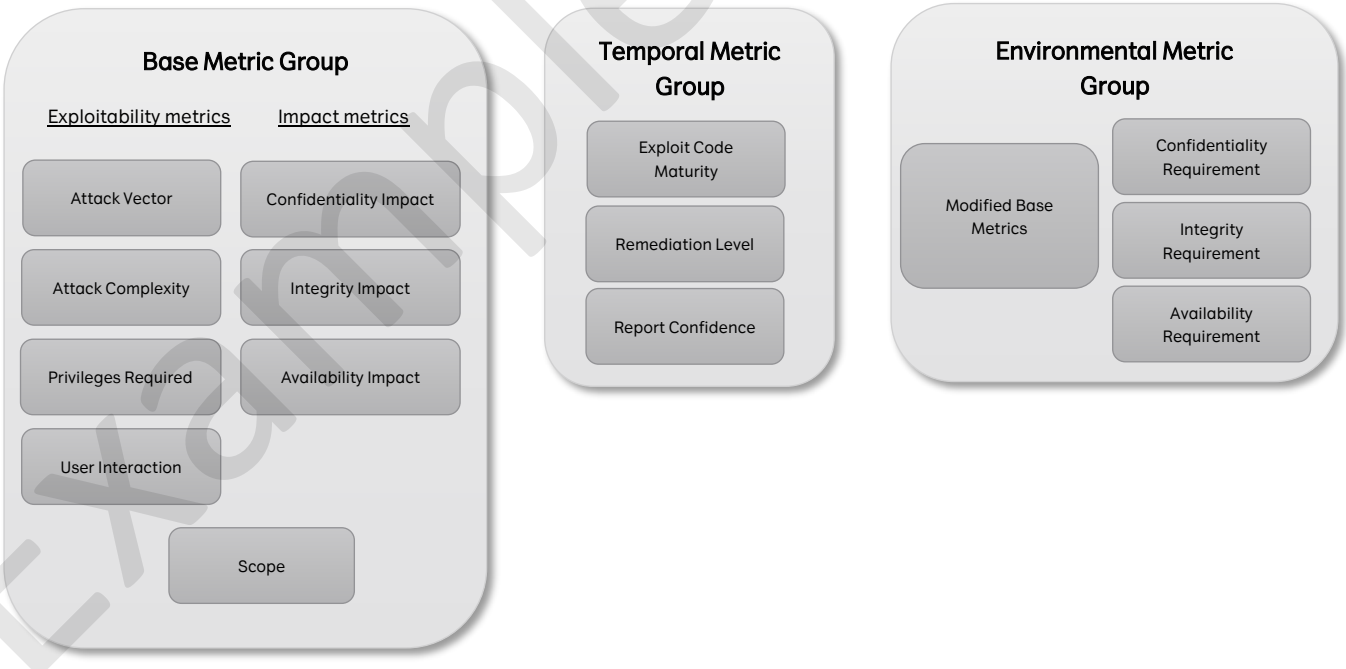
Example Company GmbH

Name	Rolle	Telefon	E-Mail
Ulrich Bohm	Auftraggeber	+43 1 154 14572	ulrich.bohm@example.com
Tim Gaertner	Technischer Kontakt	+43 1 154 14577	tim.gaertner@example.com

Tabelle 6: Ansprechpersonen Example Company GmbH

6.2 CVSS v3.0 Metriken

CVSS setzt sich aus drei Metrik Gruppen zusammen: **Base**, **Temporal** und **Environmental** wie in der unteren Abbildung dargestellt ist:



Die **Base Metric Group** drückt das grundlegende Risiko einer Schwachstelle aus und bewertet die anfällige Komponente. Ohne eine Base Metric bildet sich kein gültiger CVSS-Wert. Die Base Metric wird wiederum in **Exploitability Metrics** und **Impact Metrics** unterteilt.
Die **Exploitability Metric** spiegelt die Leichtigkeit und benötigte Voraussetzungen für eine erfolgreiche Ausnutzung der Schwachstelle wider.

Die **Impact Metric** spiegelt auf der anderen Seite die direkte Folge der erfolgreichen Ausnutzung der Schwachstelle wider – ist die Vertraulichkeit, die Integrität oder die Verfügbarkeit der betroffenen Daten/des betroffenen Systems gefährdet?

Base Metric Group

Metric	Mögliche Werte
Attack Vector (V) – Angriffsvektor	Network (N), Adjacent (A), Local (L), Physical (P)
Attack Complexity (AC) – Angriffskomplexität	Low (L), High (H)
Privileges Required (PR) – Benötigte Privilegien	None (N), Low (L), High (H)
User Interaction (UI) – Benötigte Benutzerinteraktion	None (N), Required (R)
Scope (S) – Betroffener Bereich	Changed (C), Unchanged (U)
Confidentiality Impact (C) – Vertraulichkeitsverlust	None (N), Low (L), High (H)
Integrity Impact (I) – Integritätsverlust	None (N), Low (L), High (H)
Availability Impact (A) – Verfügbarkeitsverlust	None (N), Low (L), High (H)

Tabelle 7: Übersicht Base Metric Group

Die **Temporal Metric Group** drückt die Charakteristik einer Schwachstelle aus, welche sich im Laufe der Zeit ändern kann. Beispielsweise kann nach einiger Zeit ein offizieller Patch veröffentlicht werden, wodurch der Temporal Score sinken würde.

Temporal Metric Group

Metric	Mögliche Werte
Exploit Code Maturity (E) – Reifegrad des vorhandenen Exploit Codes	Not Defined (X), High (H), Functional (F), Proof of Concept (P), Unproven (U)
Remediation Level (RL) – Vorhandene Gegenmaßnahmen	Not Defined (X), Unavailable (U), Workaround (W), Temporal Fix (T), Official Fix (O)
Report Confidence (RC) – Misst die Zuverlässigkeit der vorhandenen Informationen bezüglich der Schwachstelle	Not Defined (X), Confirmed (C), Reasonable (R), Unknown (U)

Tabelle 8: Übersicht Temporal Metric Group

Die **Environmental Metric Group** ist speziell auf das Benutzer-Environment ausgelegt. Diese Metric erlaubt die Anpassung des Scores hinsichtlich der Wichtigkeit eines betroffenen Systems für den Nutzer/Kunden. Die Anpassung erfolgt anhand der Anforderungen an Vertraulichkeit, Integrität und Verfügbarkeit.

Environmental Metric Group

Metric	Mögliche Werte
Confidentiality Requirement (CR)– Vertraulichkeitsanforderung	Network (N), Adjacent (A), Local (L), Physical (P)
Integrity Requirement (IR) – Integritätsanforderung	Low (L), High (H)
Availability Requirement (AR) - Verfügbarkeitsanforderung	None (N), Low (L), High (H)

Tabelle 9: Übersicht Environmental Metric Group

Zusätzlich lassen sich die Basismetriken als modifizierter Wert (Modified Base Metric) angeben.
Dies kann dazu verwendet werden, um Situationen zu beschreiben, welche den Base Score erhöhen.
Beispielsweise könnte eine Komponente standardmäßig mehrere Faktoren zur Authentifizierung erfordern (PR: High) um bestimmte Ressourcen zu erreichen, wobei in der getesteten Umgebung keine Authentifizierung erforderlich war (PR: None).

Metric	Mögliche Werte
Modified Attack Vector (MAV)	Dieselben Werte wie die zugehörigen Basismetriken + Not Defined (N).
Modified Attack Complexity (MAC)	
Modified Privileges Required (MPR)	
Modified User Interaction (MUI)	
Modified Scope (MS)	
Modified Confidentiality (MC)	
Modified Integrity (MI)	
Modified Availability (MA)	

Tabelle 10: Übersicht Modified Base Metric

Detaillierte Informationen zu den Base, Temporal und Environmental Metriken und deren Werte sind auf der *first.org* Webseite zu finden.[3]

6.3 Textuelle Darstellung von CVSS v3.0 Scores

In den meisten Fällen ist es hilfreich, eine textuelle Darstellung der numerischen CVSS Scores zu haben. Jede einzelne Metrik (Base, Temporal und Environmental) können mittels folgender Tabelle in textuelle Form gebracht werden.[4] [5]

Severity	CVSS v3 Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Tabelle 11: Textuelle Darstellung von CVSS v3.0 Scores

[3] https://www.first.org/cvss/specification-document
[4] https://nvd.nist.gov/vuln-metrics/cvss
[5] https://www.first.org/cvss/specification-document#5-Qualitative-Severity-Rating-Scale

6.4 Tabellenverzeichnis

6.5 Abbildungsverzeichnis

6.6 OWASP Testing Guide Version 4.0

Information Gathering
Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)
Fingerprint Web Server (OTG-INFO-002)
Review Webserver Metafiles for Information Leakage (OTG-INFO-003)
Enumerate Applications on Webserver (OTG-INFO-004)
Review Webpage Comments and Metadata for Information Leakage (OTG-INFO-005)
Identify application entry points (OTG-INFO-006)
Map execution paths through application (OTG-INFO-007)
Fingerprint Web Application Framework (OTG-INFO-008)
Fingerprint Web Application (OTG-INFO-009)
Map Application Architecture (OTG-INFO-010)

Configuration and Deployment Management Testing
Test Network/Infrastructure Configuration (OTG-CONFIG-001)
Test Application Platform Configuration (OTG-CONFIG-002)
Test File Extensions Handling for Sensitive Information (OTG-CONFIG-003)
Review Old, Backup and Unreferenced Files for Sensitive Information (OTG-CONFIG-004)
Enumerate Infrastructure and Application Admin Interfaces (OTG-CONFIG-005)
Test HTTP Methods (OTG-CONFIG-006)
Test HTTP Strict Transport Security (OTG-CONFIG-007)
Test RIA cross domain policy (OTG-CONFIG-008)
Test File Permission (OTG-CONFIG-009)

Identity Management Testing
Test Role Definitions (OTG-IDENT-001)
Test User Registration Process (OTG-IDENT-002)
Test Account Provisioning Process (OTG-IDENT-003)
Testing for Account Enumeration and Guessable User Account (OTG-IDENT-004)
Testing for Weak or unenforced username policy (OTG-IDENT-005)

Authentication Testing
Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)
Testing for default credentials (OTG-AUTHN-002)
Testing for Weak lock out mechanism (OTG-AUTHN-003)
Testing for bypassing authentication schema (OTG-AUTHN-004)
Test remember password functionality (OTG-AUTHN-005)
Testing for Browser cache weakness (OTG-AUTHN-006)
Testing for Weak password policy (OTG-AUTHN-007)
Testing for Weak security question/answer (OTG-AUTHN-008)
Testing for weak password change or reset functionalities (OTG-AUTHN-009)
Testing for Weaker authentication in alternative channel (OTG-AUTHN-010)

Authorization Testing
Testing Directory traversal/file include (OTG-AUTHZ-001)
Testing for bypassing authorization schema (OTG-AUTHZ-002)
Testing for Privilege Escalation (OTG-AUTHZ-003)
Testing for Insecure Direct Object References (OTG-AUTHZ-004)

Session Management Testing
Testing for Bypassing Session Management Schema (OTG-SESS-001)
Testing for Cookies attributes (OTG-SESS-002)
Testing for Session Fixation (OTG-SESS-003)
Testing for Exposed Session Variables (OTG-SESS-004)
Testing for Cross Site Request Forgery (CSRF) (OTG-SESS-005)
Testing for logout functionality (OTG-SESS-006)
Test Session Timeout (OTG-SESS-007)
Testing for Session puzzling (OTG-SESS-008)

Input Validation Testing
Testing for Reflected Cross Site Scripting (OTG-INPVAL-001)
Testing for Stored Cross Site Scripting (OTG-INPVAL-002)
Testing for HTTP Verb Tampering (OTG-INPVAL-003)
Testing for HTTP Parameter pollution (OTG-INPVAL-004)
Testing for SQL Injection (OTG-INPVAL-005)
Oracle Testing
MySQL Testing
SQL Server Testing
Testing PostgreSQL (from OWASP BSP)
MS Access Testing
Testing for NoSQL injection
Testing for LDAP Injection (OTG-INPVAL-006)
Testing for ORM Injection (OTG-INPVAL-007)
Testing for XML Injection (OTG-INPVAL-008)
Testing for SSI Injection (OTG-INPVAL-009)
Testing for XPath Injection (OTG-INPVAL-010)
IMAP/SMTP Injection (OTG-INPVAL-011)
Testing for Code Injection (OTG-INPVAL-012)
Testing for Local File Inclusion
Testing for Remote File Inclusion
Testing for Command Injection (OTG-INPVAL-013)
Testing for Buffer overflow (OTG-INPVAL-014)
Testing for Heap overflow
Testing for Stack overflow
Testing for Format string
Testing for incubated vulnerabilities (OTG-INPVAL-015)
Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016)
Testing for HTTP Incoming Requests (OTG-INPVAL-017)

Testing for Error Handling
Analysis of Error Codes (OTG-ERR-001)
Analysis of Stack Traces (OTG-ERR-002)

Testing for weak Cryptography
Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)
Testing for Padding Oracle (OTG-CRYPST-002)
Testing for Sensitive information sent via unencrypted channels (OTG-CRYPST-003)

Testing for Weak Encryption (OTG-CRYPST-004)
Business Logic Testing
Test Business Logic Data Validation (OTG-BUSLOGIC-001)
Test Ability to Forge Requests (OTG-BUSLOGIC-002)
Test Integrity Checks (OTG-BUSLOGIC-003)
Test for Process Timing (OTG-BUSLOGIC-004)
Test Number of Times a Function Can be Used Limits (OTG-BUSLOGIC-005)
Testing for the Circumvention of Work Flows (OTG-BUSLOGIC-006)
Test Defenses Against Application Mis-use (OTG-BUSLOGIC-007)
Test Upload of Unexpected File Types (OTG-BUSLOGIC-008)
Test Upload of Malicious Files (OTG-BUSLOGIC-009)
Client Side Testing
Testing for DOM based Cross Site Scripting (OTG-CLIENT-001)
Testing for JavaScript Execution (OTG-CLIENT-002)
Testing for HTML Injection (OTG-CLIENT-003)
Testing for Client Side URL Redirect (OTG-CLIENT-004)
Testing for CSS Injection (OTG-CLIENT-005)
Testing for Client Side Resource Manipulation (OTG-CLIENT-006)
Test Cross Origin Resource Sharing (OTG-CLIENT-007)
Testing for Cross Site Flashing (OTG-CLIENT-008)
Testing for Clickjacking (OTG-CLIENT-009)
Testing WebSockets (OTG-CLIENT-010)
Test Web Messaging (OTG-CLIENT-011)
Test Local Storage (OTG-CLIENT-012)

7 Impressum

A1 Digital International GmbH

Unternehmensgegenstand: Machine-to-Machine-Kommunikation Dienstleistungen, IT-Lösungen, Geräte und andere damit zusammenhängende Produkte und Dienstleistungen
UID-Nummer: ATU 66624566

Vertretungsberechtigte Personen:
Dr. Elisabetta Castiglioni (CEO)
Roland Haidner (CFO)

FB-Nummer: 366000k
Firmenbuchgericht: HG Wien
Firmensitz: Wien
Anschrift: Lassallestraße 9, A-1020 Wien
Kontaktdaten: Telefon: (+43) 5 06640; E-Mail: info@a1.digital
Kammerzugehörigkeit: Wirtschaftskammer Wien
Anwendbare Rechtsvorschriften: Telekommunikationsgesetz: www.ris.bka.gv.at
Aufsichtsbehörde/Gewerbebehörde: Österreichische Regulierungsbehörde für Rundfunk und Telekommunikation (RTR GmbH)