Assessment Report

AdServeX – Portal zur Verwaltung und Platzierung von Online-Werbung

Vorbereitet für
Mustermann GmbH

Datum: 3. Februar 2025
Version: 1.0
Status: Final

Vorbereitet von
Wavestone AG

Autor: Gabriel Duschl
E-Mail: gabriel.duschl@wavestone.eu

Wavestone AG
Leopoldstraße 28A
80802 München

Versionsverlauf

Version	Status	Datum	Autor
0.8	Draft	29. Januar 2025	Gabriel Duschl
0.9	QA	31. Januar 2025	Alexander Strassheim
1.0	Final	3. Februar 2025	Gabriel Duschl

Vertraulichkeitshinweis

Dieses Dokument konzentriert sich ausschließlich auf die während der Bewertung identifizierten technischen Sicherheitsaspekte. Dieser Bericht ist streng vertraulich. Er enthält technische Details zu Sicherheitslücken, die bei Zugriff durch unbefugte Personen ausgenutzt werden könnten. Der Zugang zu diesem Bericht muss daher eingeschränkt werden. Er darf ohne ausdrückliche schriftliche Genehmigung des Auftraggebers weder weitergegeben, veröffentlicht noch verteilt werden.

Inhaltsverzeichnis

1 Management Summary

Die Ergebnisse in diesem Bericht basieren auf der Analyse, die vom 20. Januar bis 31. Januar 2025 durchgeführt wurden. Dieser Bericht gibt einen Hinweis auf die Sicherheitslage der betrachteten Systeme am letzten Tag des Testzeitraums.

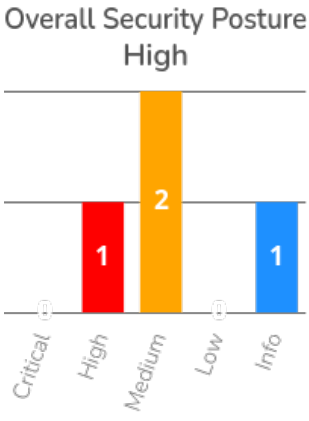
Assessment Scope

Dieser Abschnitt beschreibt den Umfang der Bewertung und spezifiziert die Ziele und Schlüsselaspekte des Penetrationstests.

Target	Description
example.com	Im Rahmen des Penetrationstests wurde die Example WebApp, eine öffentlich zugängliche Webanwendung, auf Sicherheitslücken überprüft. Die Anwendung bietet Authentifizierungsfunktionen, Benutzerverwaltung und API-Schnittstellen. Ziel des Tests war es, potenzielle Schwachstellen in der Webarchitektur, der Zugriffskontrolle und dem Datenhandling zu identifizieren und zu bewerten.

Schlüsselerkenntnisse und Maßnahmen

Im Verlauf dieses Penetrationstests wurden 1 High und 2 Medium Schwachstellen identifiziert. Diese Schwachstellen könnten zu einer Beeinträchtigung der Vertraulichkeit, Integrität und Verfügbarkeit der in der Anwendung gespeicherten Daten führen. 1 Info Fund(e) mit einem CVSS-Score von **0.0** wurden identifiziert, die keine unmittelbare Bedrohung darstellen. Die empfohlenen Maßnahmen sollten dennoch umgesetzt werden, um die allgemeine Sicherheitslage der Anwendung weiter zu verbessern.



Die Abbildung auf der linken Seite zeigt die Sicherheitslage der Anwendung. Basierend auf der höchsten identifizierten Risikoeinstufung wird das Gefährdungsniveau zum Zeitpunkt des Tests als High eingestuft.

Für weitere Details und eine detailliertere technische Beschreibung aller identifizierten Erkenntnisse und unserer Empfehlungen siehe Abschnitt "Identifizerte Schwachstellen" in diesem Bericht.

Eine Zusammenfassung der wichtigsten identifizierten Erkenntnisse sowie empfohlene Maßnahmen zur Minderung oder Reduzierung des höchsten identifizierten Risikos ist unten aufgeführt:

Persistent Cross-Site Scripting (PXSS): Cross-Site-Scripting
Empfehlung: Validierung von Benutzereingaben

2 Observation Summary

Die folgende Tabelle gibt einen Überblick über unsere Beobachtungen aus der Sicherheitsbewertung. Detaillierte Beschreibungen jeder Beobachtung finden Sie im Abschnitt 5 Identifizerte Schwachstellen dieses Dokuments.

ID	Name	CVSS	Severity
H1	Persistent Cross-Site Scripting (PXSS)	7.2	High
M1	HTTP Strict Transport Security (HSTS)	5.4	Medium
M2	TLSv1.0 - BEAST	5.3	Medium
I1	Port Scan	0.0	Info

3 Projekt-Dashboard

Der detaillierte Umfang des durchgeführten Tests vom 20. Januar bis 31. Januar 2025 stellt sich wie folgt dar:

Target	Approach	Environment
example.com	Gray-Box	Staging

Die folgenden Details bieten zusätzlichen Kontext und Überlegungen, die in den Umfang des Penetrationstests einbezogen wurden und alle relevanten Komponenten und Aspekte des Projekts abdecken:

- https://example.com
- https://api.example.com

Projektplan

Das Projekt wurde gemäß dem folgenden Zeitplan durchgeführt. Der Aufwand wird in Personentagen (PT) angegeben:

Datum oder Zeitraum	Beschreibung	Aufwand
20. Januar bis 24. Januar 2025	Web Application	5 PT
27. Januar bis 29. Januar 2025	Web Service	3 PT
30. Januar bis 31. Januar 2025	Dokumentation	2 PT

Kontaktdaten

Die folgenden Kontaktdaten wurden bereitgestellt:

Rolle	Name	E-Mail
Manager	Max Musterman	max.musterman@company.com
Pentester	Gabriel Duschl	gabriel.duschl@wavestone.eu

Testing User Accounts

Die folgenden Testbenutzer und zugewiesenen Rollen wurden im Rahmen der Bewertung bereitgestellt oder erstellt. Bitte stellen Sie sicher, dass diese Konten nach dem Testen deaktiviert oder entfernt werden:

User	Description
test-user	Test Benutzer für Webapplikation
test-admin	Administrator Benutzer für Webapplikation

Testing Tools

Die folgenden Tools wurden verwendet:

Tool Name	Short Description
Burp Suite Professional	SSL Proxy
Nessus Professional	Vulnerability Scanner
Nmap Port Scanner	Network port scanner
Nikto	Check server configuration
dirsearch	Check pathes
ssh-audit	Analyse SSH configuration
sslscan	Analyse SSL configuration
testssl.sh	Analyse SSL configuration

Schwachstellenbewertung und -management

CVSS: Das Common Vulnerability Scoring System (CVSS) wird verwendet, um die Schwere von entdeckten Sicherheitslücken zu bestimmen. Dieser Bericht liefert speziell den CVSS-Basiswert für jede entdeckte Schwachstelle, zusammen mit dem zugehörigen CVSS-Vektor-String. Der numerische Basiswert wird gemäß der CVSS-Spezifikation auch in eine qualitative Darstellung (wie niedrig, mittel, hoch und kritisch) übersetzt. Weitere Informationen zu CVSS finden Sie in der Dokumentation unter https://www.first.org/cvss/

Wave Score: Wavestone verwendet das Common Vulnerability Scoring System (CVSS) als Grundlage für die Schwachstellenbewertung, erweitert es jedoch mit dem proprietären Wave Score, um eine präzisere und kontextbewusstere Risikobewertung zu ermöglichen. Während der CVSS-Basiswert standardisierte Bewertungen liefert, erweitert der Wave Score diese Methodik, indem er zusätzliche Faktoren berücksichtigt. Dazu gehören das Potenzial einer Schwachstelle, andere Sicherheitsprobleme zu ermöglichen oder zu verschärfen, die Kritikalität der betroffenen Anwendung und der Daten, die sie verarbeitet oder offenlegt, sowie der breitere operative Einfluss auf die Systeme oder die Sicherheitslage der Organisation. Durch die Berücksichtigung dieser Dimensionen geht der Wave Score über numerische Bewertungen hinaus und liefert eine maßgeschneiderte Bewertung, die eine fundierte Grundlage für die Priorisierung und Behandlung von Schwachstellen im Einklang mit den einzigartigen Risiken und Anforderungen einer Organisation bietet.

4 Best Practices für eine sichere Konfiguration

Fehlkonfigurationen gehören zu den häufigsten Angriffsvektoren in der Cybersicherheit. Selbst gut gestaltete Systeme können durch unsachgemäße Einstellungen oder fehlende Sicherheitsmaßnahmen verwundbar werden.

Dieses Kapitel gibt einen Überblick über wesentliche Sicherheitskonfigurationen.

4.1 TLS-Konfiguration

Während des Sicherheitstests wurde die Konfiguration hinsichtlich unterstützter Protokollversionen und anderer sicherheitsrelevanter Einstellungen überprüft. In den folgenden Tabellen zeigt das Symbol ❌, dass ein kryptografisches Problem oder eine Fehlkonfiguration vorliegt. Das Symbol ✔ bedeutet, dass keine Maßnahmen erforderlich sind.

	SSLv2	SSLv3	TLSv1.0	TLSv1.1	TLSv1.2	TLSv1.3	OCSP-Stapling	PFS	Certificate Trust	Expiry date
example.com:443	✔	✔	❌	✔	✔	✔	✔	✔	✔	2025-12-31

Während des Sicherheitstests wurde die TLS-Konfiguration auf gängige kryptografische Probleme überprüft.

	Crime	Breach	Poodle	Sweet32	Freak	Drown	Logjam	Beast	Padding Oracle	RC4
example.com:443	✔	✔	✔	✔	✔	✔	✔	❌	✔	✔

4.2 Sicherheitsrelevante HTTP-Header

Im Rahmen des Sicherheitstests wurde die Anwendung auf fehlende sicherheitsrelevante HTTP-Header überprüft. Basierend auf dem OWASP Secure Headers Project[1] sollten mehrere HTTP-Header hinzugefügt werden. In der folgenden Tabelle zeigt das Symbol ❌, dass der sicherheitsrelevante HTTP-Header fehlt oder verbessert werden sollte. Das Symbol ✔ zeigt, dass keine Aktion erforderlich ist.

HTTP Header	example.com	Comment
Strict-Transport-Security	❌	-
X-Frame-Options	❌	-
X-Content-Type-Options	✔	-
Content-Security-Policy	✔	-
X-Permitted-Cross-Domain-Policies	✔	-
Referrer-Policy	✔	-
Clear-Site-Data	✔	-
Cross-Origin-Embedder-Policy	✔	-
Cross-Origin-Opener-Policy	✔	-
Cross-Origin-Resource-Policy	✔	-
Cache-Control	✔	-

Laut OWASP sind die folgenden HTTP-Header veraltet:

HTTP header	Comment
Feature-Policy	Der HTTP-Header Feature-Policy ist veraltet, da er durch den HTTP-Header Permissions-Policy ersetzt wird. Da die Spezifikation der Permissions-Policy noch in Entwicklung ist, wird empfohlen, den HTTP-Header Feature-Policy weiterhin zu verwenden, um die Nutzung von Browserfunktionen zu beschränken, bis der HTTP-Header Permissions-Policy fertiggestellt und von allen aktuellen Browsern unterstützt wird.
Public Key Pinning Extension for HTTP	Dieser HTTP-Header wird derzeit von keinem gängigen Browser unterstützt. Zusätzlich besteht die Möglichkeit, dass durch eine falsche Konfiguration des Headers, Deep-Packet-Inspection oder durch das Austauschen des TLS-Zertifikats Nutzer ausgesperrt werden können. .

[1] https://owasp.org/www-project-secure-headers/

HTTP header	Comment
X-XSS-Protection	Der HTTP-Header `X-XSS-Protection` sollte von modernen Browsern als veraltet betrachtet werden. Zudem kann seine Verwendung zusätzliche Sicherheitsprobleme auf der Client-Seite verursachen. Laut OWASP wird empfohlen, den Header auf `X-XSS-Protection: 0` zu setzen, um den XSS-Filter zu deaktivieren. Stattdessen sollte der HTTP-Header `Content-Security-Policy` verwendet werden, um die Ausführung von JavaScript zu steuern.
Expect-CT	Dieser Header ist seit Juni 2021 veraltet. Seit Mai 2018 wird erwartet, dass neue Zertifikate standardmäßig SCTs unterstützen. Zertifikate, die vor März 2018 ausgestellt wurden, durften eine Laufzeit von bis zu 39 Monaten haben und sind im Juni 2021 abgelaufen.

Die folgende Ausgabe zeigt die beantworteten HTTP-Header für Anfragen an example.com:

```
HTTP/2 200
server: nginx
date: Tue, 04 Feb 2025 10:35:08 GMT
content-type: text/html; charset=UTF-8
x-hacker: If you're reading this, you should visit wpvip.com/careers and apply to
join the fun, mention this header.
x-powered-by: WordPress VIP <https://wpvip.com>
host-header: a9130478a60e5f9135f765b23f26593b
x-frame-options: SAMEORIGIN
content-security-policy: frame-ancestors 'self'
x-redirect-by: WordPress
location: https://example.com
x-rq: hhn1 111 254 443
cache-control: max-age=3600
x-cache: MISS
```

5 Identifizerte Schwachstellen

Die folgenden Seiten dieses Berichts enthalten detaillierte Beschreibungen der identifizierten Beobachtungen sowie Empfehlungen zur Behebung.

5.1 Persistent Cross-Site Scripting (PXSS)

Target:	example.com/registration/	**Base Score:**	7.2
CVSS Vector:	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:N	**Severity:**	High
Wave Details:	Es gibt keine Abweichung zum CVSS Score.	**Wave Score:**	7.2

Beschreibung

Während der Tests konnten wir eine gespeicherte XSS-Schwachstelle in der Webanwendung identifizieren. Aufgrund einer fehlerhaften Validierung und Kodierung von Daten konnten wir bösartige Skripte in die Webanwendung einschleusen und dauerhaft speichern.

Cross-Site-Scripting (XSS) ist eine weit verbreitete Sicherheitslücke im Web, bei der aufgrund einer unzureichenden Validierung oder Kodierung von Daten bösartige Skripte in Webanwendungen eingeschleust werden können. Bei XSS-Angriffen betten Angreifer JavaScript-Code in den von der anfälligen Webanwendung gelieferten Inhalt ein.

Das Ziel bei gespeicherten XSS-Angriffen ist es, Skriptcode auf Seiten zu platzieren, die von anderen Benutzern besucht werden. Ein einfacher Besuch der betroffenen Unterseite reicht aus, damit der Skriptcode im Webbrowser des Opfers ausgeführt wird.

Bei einem Angriff werden bösartige Skripte vom Angreifer in die Webanwendung eingeschleust und in nachfolgenden HTTP-Antworten der Anwendung gespeichert und eingebunden. Das bösartige Skript wird schließlich im Webbrowser des Opfers ausgeführt und kann möglicherweise auf Cookies, Sitzungs-Tokens oder andere sensible Informationen zugreifen.

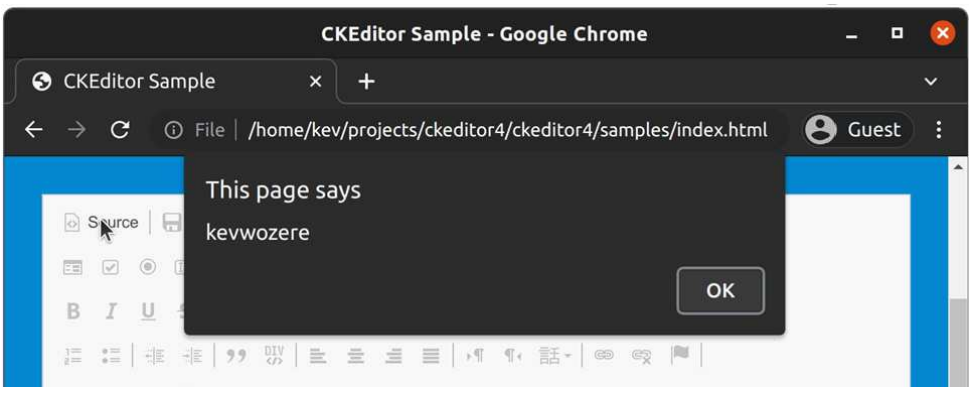


Abbildung 1 - Ausführen von JavaScript Code im Webbrowser

Impact

Ist der Angriff erfolgreich, erlangt der Angreifer die Kontrolle über Funktionen und Daten der Webanwendung im Kontext des Opfers. Wenn der betroffene Benutzer über privilegierten Zugriff verfügt, kann ein Angreifer unter Umständen die vollständige Kontrolle über die Webanwendung erlangen.

Empfehlung

- Stellen Sie sicher, dass alle verarbeiteten Daten so konsequent wie möglich gefiltert werden. Die Filterung und Validierung sollte auf der Grundlage erwarteter und gültiger Eingaben erfolgen.
- Daten sollten kodiert werden, bevor die Webanwendung sie in HTTP-Antworten einfügt. Die Kodierung sollte kontextabhängig erfolgen, d. h. je nachdem, wo die Webanwendung Daten in das HTML-Dokument einfügt, muss die entsprechende Kodierungssyntax berücksichtigt werden.
- Die HTTP-Header `Content-Type` (z.B. `text/plain`) und `X-Content-Type-Options: nosniff` können für HTTP-Antworten gesetzt werden, die kein HTML und JavaScript enthalten.
- Wir empfehlen, zusätzlich eine Content Security Policy (CSP) zu verwenden, um zu kontrollieren, welche clientseitigen Skripte erlaubt sind und welche verboten sind.
- Detaillierte Informationen und Hilfe zur Verhinderung von XSS finden Sie unter: [2]

Steps to Reproduce

Innerhalb der Webanwendung war es möglich, Persistent Cross-Site Scripting (PXSS) im Anmeldeformular einzufügen, wie in Abbildung 1 gezeigt.

Der folgende Payload wurde zu diesem Zweck verwendet:

```
<iframe// src=javaSCRIPT:alert('pXSS')>
```

Wenn ein Benutzer nun den Bereich besucht, wird das JavaScript ausgeführt.

Es war möglich, mehrere Stellen innerhalb der Webanwendung zu finden, die anfällig für Persistent Cross-Site Scripting-Angriffe sind.

[2] OWASP Cross Site Scripting Prevention Cheat Sheet

5.2 HTTP Strict Transport Security (HSTS)

Target:	example.com	**Base Score:**	5.4
CVSS Vector:	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N	**Severity:**	Medium
Wave Details:	Es gibt keine Abweichung zum CVSS Score.	**Wave Score:**	5.4

Beschreibung

Ein Benutzer, der die Webanwendung über den Link http://example.com (d.h. unverschlüsselt über HTTP) zugreift, wird nicht automatisch auf die verschlüsselte Version unter der URL https://example.com umgeleitet. Dies ist auf den fehlenden HTTP-Dienst an Port 80/TCP und eine durchgehend fehlende Umleitung auf Port 443/TCP zurückzuführen. Es ist nicht möglich, über einen unverschlüsselten Kanal mit der Webanwendung zu kommunizieren.

Wavestone befürwortet diese Konfiguration, da Man-in-the-Middle-Angriffe durch die Verwendung von SSL/TLS erheblich erschwert werden.

Impact

Es kann jedoch möglich sein, dass ein Angreifer in einer geeigneten Netzwerkposition eine Anfrage an die Webanwendung über Port 80/TCP abfängt und selbst gehostete Inhalte über diesen Dienst anbietet. Ein Angreifer kann dies tun, ohne dass der Benutzer den Angriff bemerkt.

Empfehlung

HTTP Strict Transport Security (HSTS) sollte implementiert werden, um solche Angriffe zu verhindern. Das bedeutet, dass der Webserver den HTTP-Header `Strict-Transport-Security` und einen vernünftigen `max-age`-Wert an den Browser senden sollte. Ein geeigneter Header könnte wie folgt aussehen:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

Wenn HSTS verwendet wird, kann der beschriebene Angriff nur gegen Benutzer durchgeführt werden, die die Webanwendung noch nie zuvor besucht haben (innerhalb des vom Server festgelegten Zeitrahmens). Die Gültigkeitsdauer der Header sollte in der Regel auf mehrere Monate oder sogar einige Jahre festgelegt werden.

Steps to Reproduce

Die Header können mit dem folgenden Befehl angezeigt werden:

```
$ curl -I -X GET https://example.com
```

5.3 TLSv1.0 - BEAST

Target:	example.com	**Base Score:**	5.3
CVSS Vector:	CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N	**Severity:**	Medium
Wave Details:	Es gibt keine Abweichung zum CVSS Score.	**Wave Score:**	5.3

Beschreibung

Die TLS-Version 1.0 wird angeboten. Diese Protokollversion ist veraltet. Der Dienst ist anfällig für BEAST (Browser Exploit Against SSL/TLS), weitere Informationen finden Sie unter (Link). BEAST basiert auf der Art und Weise, wie Initialisierungsvektoren (IVs) behandelt werden (bei Verwendung von CBC-Verschlüsselungsverfahren in Kombination mit SSLv2, SSLv3 oder TLSv1.0). Dies könnte zu erfolgreichen Chosen Plaintext Attacks (CPA) führen (Link).

Gemäß den Empfehlungen von Regierungsbehörden (z. B. NIST SP 800-52r2) sollte TLS-Version 1.0 nur verwendet werden, wenn es unbedingt erforderlich ist. Für Regierungssysteme sollte es überhaupt nicht verwendet werden.

Impact

Da CBC-Verschlüsselungsverfahren in TLSv1.0 obligatorisch sind (mit Ausnahme von Null-Verschlüsselungen), ist TLSv1.0 anfällig für BEAST- und Padding-Oracle-Angriffe, die zu einem Verlust der Integrität führen könnten.

Zudem könnte TLS in Version 1.0 durch Fehlkonfigurationen weiteren Sicherheitslücken ausgesetzt sein. Durch die Verwendung von Kompression könnte TLSv1.0 anfällig für CRIME- und/oder Breach-Angriffe werden. Wird Triple-DES verwendet, ist die Datenübertragung durch den Sweet32-Angriff betroffen [3].

Empfehlung

Um BEAST- und Padding-Oracle-Angriffe zu verhindern, wird empfohlen, TLSv1.0 zu deaktivieren. Allerdings könnte dies zu Problemen mit älteren Clients führen, die keine höheren TLS-Versionen unterstützen (jedoch unterstützt jeder aktuelle Browser TLS-Versionen höher als 1.0).

Steps to Reproduce

Die Kommunikation über TLSv1.0 kann mit **OpenSSL** initiiert werden:

```
$ openssl s_client -connect example.com:443 -tls1
```

[3] https://sweet32.info

5.4 Port Scan

Target:	example.com	**Base Score:**	0.0
CVSS Vector:	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	**Severity:**	Info
Wave Details:	Es gibt keine Abweichung zum CVSS Score.	**Wave Score:**	0

Beschreibung

Während des Tests wurde das System mit dem Portscanner Nmap auf offene Ports überprüft. Die während des Testzeitraums identifizierten offenen Ports sind in der folgenden Tabelle aufgeführt.

Host	Offene Ports	Dienste	Banner
example.com	80/tcp	http	nginx
	443/tcp	ssl/http	nginx
	8880/tcp	http	sw-cp-server

Um die Webanwendung auszuführen, sind in der Regel nur die TCP-Ports 80 und 443 erforderlich. Wie das Ergebnis des Portscans zeigt, ist ein weiterer Dienst erreichbar.

Impact

Ein offener Port stellt ein potenzielles Sicherheitsrisiko dar, da er eine Angriffsfläche für Cyberkriminelle bietet. Über offene Ports können Angreifer unbefugt auf Dienste zugreifen, Schwachstellen ausnutzen oder Malware einschleusen.

Empfehlung

Für die Ausführung der Webanwendung werden normalerweise nur die TCP-Ports 80 und 443 benötigt. Um den Angriffsvektor einzuschränken, ist es sehr ratsam, den Zugriff auf alle anderen Dienste zu deaktivieren oder einzuschränken.

Steps to Reproduce

```
$ nmap -sS -O -v -T4 --min-rate=100 --max-rate=500 -p- example.com
$ nmap -sU -v -T4 --max-retries=3 -p- --min-rate=100 --max-rate=500 example.com
```