CyberTestify

Automatisierter Sicherheits-Scan-Bericht

rest.vulnweb.com · Paket Externe Angriffsfläche & Konfiguration

CyberTestify-Sicherheitsscan — Abgeschlossen

Bericht-Nr.: **CT-ÖRNEK-7CE5**

Verifizierungscode: **73DD-A368**

Dieser Bericht ist kein formeller Penetrationstest / keine Zertifizierung.

Inhaltsverzeichnis

ZIEL	PAKET	BERICHT-NR.
rest.vulnweb.com	Paket Externe Angriffsfläche & Konfiguration	CT-ÖRNEK-7CE5

1. Managementzusammenfassung

GESAMTBEWERTUNG — Hohes Risiko

Dieses Ziel antwortet nicht über HTTPS; die Kommunikation läuft unverschlüsselt (Klartext) — vorrangig sollte mit einem gültigen TLS-Zertifikat auf HTTPS umgestellt werden. Die übrigen Bereiche wurden über http:// geprüft. Das höchste Risiko wurde im Bereich SSL/TLS-Konfigurationsaudit (auf Zertifikats- und/oder Protokollebene wurde ein dringend zu behebendes Problem festgestellt.) festgestellt; eine vorrangige Behebung wird empfohlen. Nachfolgend wird jeder Bereich einzeln berichtet.

⚡ Management-Entscheidung

Es wurden **3** Indikator(en) mit hohem/kritischem Schweregrad festgestellt, die vorrangig zu behandeln sind; ein dringender Behebungsplan wird empfohlen. Mittlere/geringe Punkte können durch geplante Verbesserung behoben werden.

- **Gesamtrisikostufe: Hoch** — 5 Bereiche wurden geprüft; das höchste Risiko liegt im Bereich **SSL/TLS-Konfigurationsaudit** (auf Zertifikats- und/oder Protokollebene wurde ein dringend zu behebendes Problem festgestellt.).
- ⚠ **HTTPS wird nicht unterstützt:** Das Ziel hat nicht über HTTPS (443) geantwortet; die Kommunikation läuft unverschlüsselt (Klartext). Die Prüfung wurde über http:// durchgeführt. Das ist für sich genommen ein ernster Befund (siehe unten).

Verbesserungsprioritäten

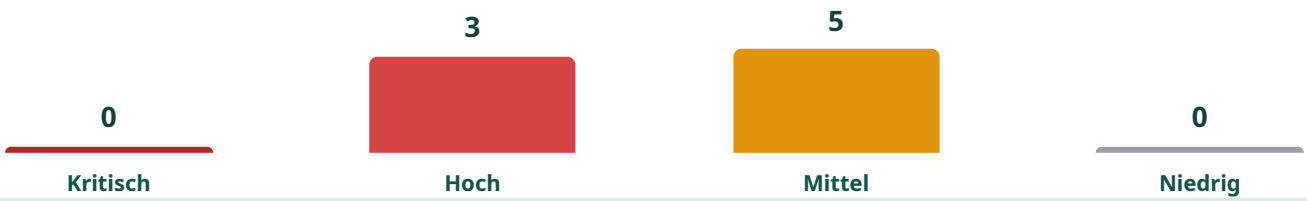
⚡ **Am dringendsten:** Veraltete Komponente (CVE).

🛠 **Schnelle Erfolge (Serverkonfiguration, ~1-2 Tage):** Fehlendes HSTS; Offengelegte sensible Datei — /db.sql; Fehlende Content-Security-Policy; Fehlendes X-Frame-Options (Clickjacking); Fehlendes X-Content-Type-Options; Unzureichende DMARC-Richtlinie (fehlend oder schwach).

🗓 **Mittelfristig / Prozess (manuelle Verifizierung oder Code/Architektur):** Nicht aktuelle Softwareversion wird offengelegt (Apache/2.4.25 — sehr alter Patchstand).

2. Befunde

2.1 Schwachstellenverteilung



In diesem Scan wurden insgesamt **8** Befund-Indikatoren festgestellt; die Verteilung nach Schweregrad ist unten dargestellt.

2.2 Master-Befundtabelle

ID	Titel	Status	Schweregrad
CT-1	Fehlendes HSTS	Offen	Hoch
CT-2	Offengelegte sensible Datei — /db.sql	Offen	Hoch
CT-3	Veraltete Komponente (CVE)	Offen	Hoch
CT-4	Nicht aktuelle Softwareversion wird offengelegt (Apache/2.4.25 — sehr alter Patchstand)	Offen	Mittel
CT-5	Fehlende Content-Security-Policy	Offen	Mittel
CT-6	Fehlendes X-Frame-Options (Clickjacking)	Offen	Mittel
CT-7	Fehlendes X-Content-Type-Options	Offen	Mittel
CT-8	Unzureichende DMARC-Richtlinie (fehlend oder schwach)	Offen	Mittel

8 Kontrollbereiche wurden ausgeführt — 4 ohne Befund, 4 mit Befund

✓ GEPRÜFT — KEIN PROBLEM GEFUNDEN

- **CORS- & Cookie-Sicherheit** Kein Problem gefunden
- **Verzeichnisauflistung (autoindex / „Index of /") · CWE-548** 6 Verzeichnisse geprüft, keine Auflistung
- **Ausführlicher Fehler / Serverpfad-Offenlegung · CWE-209** Kein Indikator gefunden
- **autocomplete-Richtlinie im Passwortfeld · CWE-522** Angemessen / kein Passwortfeld beobachtet

⚠ BEFUND VORHANDEN — DETAILS IN DEN ABSCHNITTEN UNTEN

- **SSL/TLS-Konfigurationsaudit** Befund vorhanden (Hoch — oben im Detail)
- **Sicherheits-Header & Informationslecks** Befund vorhanden (Hoch — oben im Detail)
- **DNS- & E-Mail-Sicherheit** Befund vorhanden (Mittel — oben im Detail)
- **CSP-Analyse (Content Security Policy)** Befund vorhanden (Mittel — oben im Detail)

2.3 Detaillierte Befunde

[HOCH] CT-1 · Fehlendes HSTS

Status: Offen

Beschreibung: Kein HSTS; der Browser wird nicht auf HTTPS gezwungen.

Wie es erkannt wurde: Die Website antwortet nicht über HTTPS; der gesamte Verkehr wird unverschlüsselt (Klartext) übertragen — mitlesbar/veränderbar, Sitzungen/Passwörter können gestohlen werden. Lösung: gültiges TLS-Zertifikat + HTTP→HTTPS-Umleitung + HSTS.

Geschäftliche Auswirkung: Ohne erzwungenes HTTPS kann der Verkehr per Man-in-the-Middle abgehört/umgeleitet werden.

EMPFOHLENE BEHEBUNG

Fügen Sie `Strict-Transport-Security: max-age=31536000; includeSubDomains` hinzu (wenn vollständig HTTPS).

Referenz: CWE-319 · OWASP A05:2021 Security Misconfiguration

[HOCH] CT-2 · Offengelegte sensible Datei — /db.sql

Status: Offen

Beschreibung: Eine sensible Datei (.git/.env/Backup) ist offengelegt.

Wie es erkannt wurde: Inhalt verifiziert; Risiko eines Konfigurations-/Quellcode-Lecks. Der Zugriff muss umgehend gesperrt werden.

Geschäftliche Auswirkung: Sensible Dateien (Quellcode, Backups, Secrets) sind offengelegt; Risiko der Preisgabe von Zugangsdaten/Geheimnissen.

EMPFOHLENE BEHEBUNG

Blockieren Sie diese Pfade; verschieben Sie Quell-/Backup-/Secret-Dateien aus dem Web-Root.

Referenz: CWE-538 · OWASP A05:2021 Security Misconfiguration

[HOCH] CT-3 · Veraltete Komponente (CVE)

Status: Offen

Beschreibung: Fingerabdruck einer veralteten Komponente/eines CMS und mögliche bekannte CVE.

Wie es erkannt wurde: Die PHP-7.x-Serie wird offiziell nicht mehr unterstützt (Sicherheitsupdates für 7.x endeten Ende 2022). Zahlreiche bekannte Sicherheitslücken bleiben ungepatcht. CWE-1104 · OWASP A06:2021 (Veraltete/Verwundbare Komponenten). Lösung: Auf eine aktuelle, unterstützte PHP-Version (8.2+) aktualisieren; die Versionssignatur verbergen (expose_php=Off).

Geschäftliche Auswirkung: Eine veraltete Komponente ist bekannten CVEs ausgesetzt; Risiko gezielter Angriffe.

EMPFOHLENE BEHEBUNG

Halten Sie Komponenten/Plugins/Themes aktuell; ergänzen Sie Auto-Update + Dependency-Scanning.

Referenz: CWE-1104 · OWASP A06:2021 Vulnerable & Outdated Components

[MITTEL] CT-5 · Fehlende Content-Security-Policy

Status: Offen

Beschreibung: Keine Content-Security-Policy; keine browserseitige XSS-Minderung.

Wie es erkannt wurde: Der Browser kann nicht einschränken, welche Ressourcen geladen werden; keine grundlegende Verteidigung gegen XSS und Content-Injection. Fehlt auf ALLEN 2 geprüften einzigartigen Seiten.

Geschäftliche Auswirkung: Keine browserseitige Minderung gegen Content-Injection/XSS; injizierte Skripte können ausgeführt werden.

EMPFOHLENE BEHEBUNG

Definieren Sie eine strikte CSP (`default-src 'self'`) und setzen Sie Drittanbieter-Quellen auf eine Allowlist.

Referenz: CWE-693 · OWASP A05:2021 Security Misconfiguration

[MITTEL] CT-6 · Fehlendes X-Frame-Options (Clickjacking)

Status: Offen

Beschreibung: Die Seite kann in einen iframe einer anderen Website eingebettet werden (kein X-Frame-Options / CSP frame-ancestors).

Wie es erkannt wurde: Die Seite kann in das iframe einer fremden Website eingebettet werden; Nutzer können per Clickjacking getäuscht werden. Fehlt auf ALLEN 2 geprüften einzigartigen Seiten.

Geschäftliche Auswirkung: Die Seite kann unsichtbar in einen Frame eingebettet werden, um Nutzer zu unbeabsichtigten Aktionen zu verleiten (Clickjacking); schwächt Konto- und Transaktionssicherheit.

EMPFOHLENE BEHEBUNG

Fügen Sie `X-Frame-Options: SAMEORIGIN` hinzu oder setzen Sie CSP `frame-ancestors 'self'`.

Referenz: CWE-1021 · OWASP A05:2021 Security Misconfiguration

[MITTEL] CT-7 · Fehlendes X-Content-Type-Options

Status: Offen

Beschreibung: Kein X-Content-Type-Options; der Browser kann den Inhaltstyp erraten (MIME-Sniffing).

Wie es erkannt wurde: Die Verteidigungstiefe ist schwach. (fehlt auf 2/2 Seiten)

Geschäftliche Auswirkung: Der Browser kann Inhaltstypen erraten und schädliche Inhalte ausführen, was die XSS-/Injection-Angriffsfläche vergrößert.

EMPFOHLENE BEHEBUNG

Fügen Sie `X-Content-Type-Options: nosniff` hinzu.

Referenz: CWE-693 · OWASP A05:2021 Security Misconfiguration

[MITTEL] CT-8 · Unzureichende DMARC-Richtlinie (fehlend oder schwach)

Status: Offen

Beschreibung: Kein DMARC; SPF/DKIM werden nicht durchgesetzt.

Wie es erkannt wurde: Die SPF/DKIM-Ergebnisse werden nicht durchgesetzt.

Geschäftliche Auswirkung: Ohne DMARC-Richtlinie werden gefälschte E-Mails nicht blockiert; Phishing- und Reputationsrisiko.

EMPFOHLENE BEHEBUNG

Veröffentlichen Sie `_dmarc` `v=DMARC1; p=quarantine` (mit Monitoring beginnen, dann auf reject erhöhen).

Referenz: CWE-290 · OWASP A07:2021 Identification & Authentication Failures

3. Kontrollübersicht & Methodik

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
HTTPS wird nicht unterstützt (unverschlüsselte Kommunikation)	Hoch	Die Website antwortet nicht über HTTPS; der gesamte Verkehr wird unverschlüsselt (Klartext) übertragen — mitlesbar/veränderbar, Sitzungen/Passwörter können gestohlen werden. Lösung: gültiges TLS-Zertifikat + HTTP→HTTPS-Umleitung + HSTS.

SSL/TLS-Konfigurationsaudit

TLS-ZERTIFIKATSSTATUS

⚠ Dieses Ziel **hat nicht über HTTPS (443) geantwortet**; es wurde kein gültiges TLS-Zertifikat gefunden. Die Website ist nur über **unverschlüsseltes HTTP** erreichbar.

TLS-PROTOKOLL & CIPHER

- **Aktives Protokoll:** nicht ermittelbar
- **Cipher:** nicht ermittelbar
- **Unterstützung veralteter/schwacher Versionen:** Nicht beobachtet (nur TLS 1.2+ gesehen)

HSTS (HTTP Strict Transport Security)

- **Status:** Fehlt — Dem Browser wird die HTTPS-Pflicht nicht mitgeteilt; bei Erstanfragen besteht das Risiko von SSL-Stripping-/Downgrade-Angriffen.

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
HTTPS wird nicht unterstützt (unverschlüsselte Kommunikation)	Hoch	Die Website antwortet nicht über HTTPS; der gesamte Datenverkehr wird unverschlüsselt (Klartext) übertragen. Ein Angreifer im selben Netzwerk kann mithören, Sitzungen/Passwörter stehlen oder Inhalte verändern. Lösung: gültiges TLS-Zertifikat + HTTP→HTTPS-Umleitung + HSTS.
HSTS fehlt	Mittel	Die HTTPS-Pflicht wird dem Browser nicht mitgeteilt; anfällig für Downgrade-Angriffe.

Sicherheits-Header & Informationslecks

HTTP-SICHERHEITS-HEADER

Header	Status	Beschreibung
Strict-Transport-Security	Fehlt	Die HTTPS-Pflicht wird nicht mitgeteilt; SSL-Stripping-Risiko.
Content-Security-Policy	Fehlt	Keine Browser-Verteidigung gegen XSS/Injection.
X-Frame-Options	Fehlt	Anfällig für Clickjacking; kann in ein iframe eingebettet werden.
X-Content-Type-Options	Fehlt	MIME-Sniffing möglich.
Referrer-Policy	Fehlt	Referrer-Informationen können an externe Quellen abfließen.
Permissions-Policy	Fehlt	Sensible Browser-APIs sind nicht eingeschränkt.
X-XSS-Protection	Fehlt	Der Legacy-XSS-Filter älterer Browser ist nicht gesetzt (in modernen Browsern unkritisch).

INFORMATIONSLECKS / OFFENLIEGENDE DATEIEN

Häufige sensible Pfade wurden mit einem einzigen GET geprüft (Inhalt verifiziert — allein HTTP 200 gilt nicht als Nachweis):

Pfad	Status	Hinweis
`/.git/config`	Geschlossen	HTTP 400 / leerer Body — nicht erreichbar
`/.env`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar

Pfad	Status	Hinweis
`/.git/HEAD`	Geschlossen	HTTP 400 / leerer Body — nicht erreichbar
`/backup.zip`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/.DS_Store`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/wp-config.php.bak`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/ftp`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/backup`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/backups`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/uploads`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/files`	Geschlossen	HTTP 400 / leerer Body — nicht erreichbar
`/admin`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/.svn/entries`	Geschlossen	HTTP 400 / leerer Body — nicht erreichbar
`/.htaccess`	Geschlossen	HTTP 403 / leerer Body — nicht erreichbar
`/config.php.bak`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/db.sql`	⚠ **OFFEN**	erwartetes Dateiformat bestätigt (kein Catch-all)
`/dump.sql`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/backup.tar.gz`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/backup.tar`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/www.zip`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/site.zip`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/backup.old`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/backup.backup`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/index.php.bak`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/index.php~`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/.env.bak`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/.env.old`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar
`/database.sql`	Geschlossen	HTTP 404 / leerer Body — nicht erreichbar

ZUSÄTZLICHE INFORMATIONSLECK-BEOBACHTUNGEN

Prüfung	Ergebnis
Verzeichnisauflistung (autoindex / „Index of /") · CWE-548	✓ **6 Verzeichnisse geprüft, keine Auflistung**

Prüfung	Ergebnis
Ausführlicher Fehler / Serverpfad-Offenlegung · CWE-209	✓ **Kein Indikator gefunden**
autocomplete-Richtlinie im Passwortfeld · CWE-522	✓ **Angemessen / kein Passwortfeld beobachtet**

> Alles GET-only/passive Beobachtung — Inhalte werden NICHT abgerufen/angezeigt; ein geleakter Pfad wird REDIGIERT. „Kein Indikator gefunden" BEWEIST NICHT, dass es sicher ist; es zeigt nur, dass mit den passiven Methoden kein Indikator auftrat.

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
Sensible Datei erreichbar (`/db.sql`)	Hoch	Inhalt verifiziert; Risiko eines Konfigurations-/Quellcode-Lecks. Der Zugriff muss umgehend gesperrt werden.
Veraltete/nicht mehr unterstützte Softwareversion wird offengelegt (PHP/7.1.26 — EOL)	Hoch	Die PHP-7.x-Serie wird offiziell nicht mehr unterstützt (Sicherheitsupdates für 7.x endeten Ende 2022). Zahlreiche bekannte Sicherheitslücken bleiben ungepatcht. CWE-1104 · OWASP A06:2021 (Veraltete/Verwundbare Komponenten). Lösung: Auf eine aktuelle, unterstützte PHP-Version (8.2+) aktualisieren; die Versionssignatur verbergen (expose_php=Off).
Nicht aktuelle Softwareversion wird offengelegt (Apache/2.4.25 — sehr alter Patchstand)	Mittel	Die Apache-2.4-Serie wird zwar unterstützt, Apache/2.4.25 ist jedoch ein sehr alter Patchstand; zwischenzeitliche Sicherheitspatches scheinen nicht eingespielt zu sein. CWE-1104 · OWASP A06:2021 (Veraltete/Verwundbare Komponenten). Lösung: Auf den aktuellen Patch der 2.4-Serie aktualisieren; die Versionssignatur verbergen (ServerTokens Prod).
Kritischer Sicherheits-Header fehlt: Content-Security-Policy	Mittel	Der Browser kann nicht einschränken, welche Ressourcen geladen werden; keine grundlegende Verteidigung gegen XSS und Content-Injection. Fehlt auf ALLEN 2 geprüften einzigartigen Seiten.
Kritischer Sicherheits-Header fehlt: X-Frame-Options	Mittel	Die Seite kann in das iframe einer fremden Website eingebettet werden; Nutzer können per Clickjacking getäuscht werden. Fehlt auf ALLEN 2 geprüften einzigartigen Seiten.
Weitere Header fehlen (Strict-Transport-Security, X-Content-Type-Options, Referrer-Policy, Permissions-Policy, X-XSS-Protection)	Mittel	Die Verteidigungstiefe ist schwach. (fehlt auf 2/2 Seiten)

DNS- & E-Mail-Sicherheit

SPF (Sender-Richtlinie) — geprüfte Domain: `vulnweb.com`

- **Status:** Vorhanden — `v=spf1 ~all`
- **Härte:** `~all` — weich (softfail; akzeptabel, nicht ideal).

DMARC (Authentifizierungsrichtlinie) — geprüfte Domain: `vulnweb.com`

- **Status:** Fehlt — Es wurde kein DMARC-Eintrag gefunden. Es erfolgt keine Durchsetzung anhand der SPF/DKIM-Ergebnisse; der Schutz vor Spoofing ist schwach.

DKIM (Signatur)

- **Status:** Nicht erkannt — Bei gängigen Selektoren (default/google/selector1…) wurde kein DKIM-Eintrag gefunden. Möglicherweise verwenden Sie einen anderen Selektor; dies bedeutet nicht sicher „kein Eintrag".

DNSSEC

- **Status:** Passiv/fehlt — DNS-Antworten sind nicht signiert; erhöhtes Risiko für DNS-Spoofing/Cache-Poisoning.

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
DMARC fehlt	Mittel	Die SPF/DKIM-Ergebnisse werden nicht durchgesetzt.
DKIM nicht erkannt	Hinweis	Bei gängigen Selektoren nicht gefunden (möglicherweise anderer Selektor).
DNSSEC passiv	Hinweis	Die DNS-Antworten sind nicht signiert.

CORS- & Cookie-Sicherheit

CORS-KONFIGURATION (auf 2/2 Seiten getestet)

- **Test-Origin:** `https://cybertestify-cors-probe.example` — an jede Seite wurde ein harmloser Origin-Header gesendet und die Antwort ausgewertet.
- **Offenste beobachtete Richtlinie** (`/`): Access-Control-Allow-Origin: wird nicht gesendet (geschlossen — sichere Voreinstellung); Allow-Credentials: wird nicht gesendet.

COOKIE-FLAGS (alle auf 2 Seiten beobachteten Cookies)

- Auf keiner der 2 geprüften Seiten wurde ein Set-Cookie beobachtet.

FESTGESTELLTE RISIKEN

- Bei der CORS- und Cookie-Konfiguration wurde kein deutliches Risiko festgestellt.

CSP-Analyse (Content Security Policy)

CSP-STATUS

- **Status:** Fehlt — Es wird kein Content-Security-Policy-Header gesendet.

CSP-DIREKTIVENANALYSE

- Da keine durchgesetzte CSP vorhanden ist, konnte keine Direktivenanalyse durchgeführt werden.

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
CSP vollständig fehlend	Mittel	Keine Verteidigung auf Browser-Ebene gegen XSS und Content-Injection. Auf ALLEN 2 geprüften Seiten fehlt der CSP-Header.

POSITIVE ZUSICHERUNG — GEPRÜFTE BEREICHE

Auch die Bereiche ohne Befund eingeschlossen, wurden die Kontrollen der externen Angriffsfläche tatsächlich auf **2 einzigartigen Seiten** einschließlich der Startseite ausgeführt. Die folgende Tabelle zeigt auch die „kein Problem gefunden"-Ergebnisse transparent:

Kontrollbereich	Ergebnis
SSL/TLS-Konfigurationsaudit	⚠ **Befund vorhanden (Hoch — oben im Detail)**
Sicherheits-Header & Informationslecks	⚠ **Befund vorhanden (Hoch — oben im Detail)**
DNS- & E-Mail-Sicherheit	⚠ **Befund vorhanden (Mittel — oben im Detail)**
CORS- & Cookie-Sicherheit	✓ **Kein Problem gefunden**
CSP-Analyse (Content Security Policy)	⚠ **Befund vorhanden (Mittel — oben im Detail)**
Verzeichnisauflistung (autoindex / „Index of /") · CWE-548	✓ **6 Verzeichnisse geprüft, keine Auflistung**
Ausführlicher Fehler / Serverpfad-Offenlegung · CWE-209	✓ **Kein Indikator gefunden**
autocomplete-Richtlinie im Passwortfeld · CWE-522	✓ **Angemessen / kein Passwortfeld beobachtet**

> **Drei-Zustands-Unterscheidung (Ehrlichkeit):** ✓ *Kein Problem gefunden* = Kontrolle lief, Ergebnis sauber · ⚠ *Befund vorhanden* = oben im Detail · ⚠ *Nicht prüfbar* = keine Daten erhebbar (bedeutet NICHT sicher).

Was dieses Paket prüft — und was NICHT

> **PRÜFT (passiv — nur Seitenabruf per GET + harmlose Origin-/DNS-Abfrage):** TLS/Zertifikat, HTTP-Sicherheits-Header, CORS-Richtlinie, Cookie-Flags (Secure/HttpOnly/SameSite), Content-Security-Policy, DNS-/E-Mail-Einträge (SPF/DKIM/DMARC/DNSSEC), offenliegende sensible Dateien (inkl. gängiger Backup-Muster), Verzeichnisauflistung (autoindex), ausführliche-Fehler-/Serverpfad-Offenlegung (redigiert), autocomplete-Richtlinie im Passwortfeld, veraltete/nicht unterstützte Softwareversionen — auf den 2 entdeckten Seiten.

> **PRÜFT NICHT:** Aktive Schwachstellenverifikation (Payload-/Probe-Versuche wie SQLi/XSS/IDOR), authentifizierte Ablauftests, Missbrauch der Geschäftslogik. Diese gehören zum Umfang der Pakete **Aktive Verifikation** und **Umfassender Pentest**. Dieser Bericht beruht auf passiver Beobachtung; die Aussage „kein Befund" in einem Bereich **BEWEIST NICHT**, dass er sicher ist, da kein aktiver Exploit versucht wurde — sie zeigt lediglich, dass die von außen beobachtete Konfiguration sauber ist.

Methodik

Phase	Beschreibung
Erkundung	Die externe Oberfläche, Seiten und (bei SPAs) echte Endpunkte/Parameter aus dem JS-Bundle werden extrahiert.
Automatische Erkennung	Auf der ermittelten Oberfläche werden deterministische, sichere (read-only) Indikatoren gesucht.
Manuell-deterministische Verifizierung	Befunde werden mit code-basierten Regeln verifiziert; „nachweisen, nicht ausnutzen“.
Autorisierung & Session	Im authentifizierten Paket werden Cookie/Session/Autorisierung und die Post-Login-Oberfläche geprüft.
Konfiguration	Header-, TLS-, E-Mail-/DNS- und Preisgabe-Konfigurationen werden beobachtet.

Risikobewertungskriterien

Schweregrad	Definition
Kritisch	Direkt/leicht ausnutzbarer Indikator mit erheblicher Daten-/Zugriffsauswirkung.
Hoch	Hervorstechender, vorrangig zu behebender starker Indikator.
Mittel	Erfordert Aufmerksamkeit; kontextabhängige Verifizierung empfohlen.
Niedrig	Härtungs-/Reifegrad-Chance; niedrige Priorität.

Der Schweregrad ist nur ein Bandlabel (kein numerischer CVSS-Score); alle Befunde sind als „Indikator, Verifizierung erforderlich" gerahmt.

---

4. KI-Lösungsempfehlungen

Dieser Abschnitt enthält bereichsweise Korrekturvorschläge für alle in Ihrer Prüfung der externen Angriffsfläche festgestellten Lücken. Kopieren Sie die für Ihren Server passenden Beispiele (Nginx/Firebase/Apache/DNS).

SSL/TLS-Konfigurationsaudit

Die folgenden Empfehlungen stärken die TLS-/Verschlüsselungskonfiguration für rest.vulnweb.com.

1. HSTS-Header ergänzen

Weist den Browser an, die Website nur über HTTPS aufzurufen (nur anwenden, wenn Ihre Website vollständig auf HTTPS läuft):

Nginx:

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

Firebase Hosting — `firebase.json`:

```
{
  "hosting": {
    "headers": [
      {
        "source": "**",
        "headers": [
          { "key": "Strict-Transport-Security", "value": "max-age=31536000; includeSubDomains" }
        ]
      }
    ]
  }
}
```

Apache — `.htaccess`:

```
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

Sicherheits-Header & Informationslecks

Die folgenden Empfehlungen dienen der Behebung der auf der Startseite von rest.vulnweb.com festgestellten fehlenden Sicherheits-Header. Zu jedem Header folgt zuerst eine kurze Erläuterung, dann ein Nginx-Beispiel; am Ende finden Sie fertige Blöcke für andere Plattformen (Firebase, Vercel, Next.js, Apache). Kopieren Sie den zu Ihrem Server passenden Block.

1. Content-Security-Policy (CSP) hinzufügen

Die wirksamste Browser-Verteidigung gegen XSS und Content-Injection. Beginnen Sie mit einer zur Website passenden Basisrichtlinie und verschärfen Sie sie mit der Zeit:

```
add_header Content-Security-Policy "default-src 'self'; img-src 'self' data:; script-src 'self'; style-src '
```

Wenn Sie Drittanbieter-Skripte verwenden (GTM, Analytics, Pixel), fügen Sie die entsprechenden Domains zu `script-src` / `connect-src` hinzu.

2. X-Frame-Options hinzufügen

Verhindert, dass Ihre Seite in das iframe einer fremden Website eingebettet und für Clickjacking missbraucht wird:

```
add_header X-Frame-Options "SAMEORIGIN" always;
```

Als moderne Alternative bietet CSP `frame-ancestors 'self'` denselben Schutz.

3. X-Content-Type-Options hinzufügen

Verhindert, dass der Browser per MIME-Type-Sniffing Inhalte falsch interpretiert (und das damit verbundene XSS-Risiko):

```
add_header X-Content-Type-Options "nosniff" always;
```

4. Strict-Transport-Security (HSTS) hinzufügen

Erzwingt HTTPS und erschwert SSL-Stripping-/MITM-Angriffe. (Nur anwenden, wenn Ihre Website vollständig über HTTPS läuft.)

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

5. Referrer-Policy hinzufügen

Reduziert den Informationsabfluss über den `Referer`-Header bei externen Links:

```
add_header Referrer-Policy "strict-origin-when-cross-origin" always;
```

6. Permissions-Policy hinzufügen

Schränkt Browser-APIs (Kamera, Mikrofon, Standort usw.) ein; verhindert unerwünschte Zugriffe durch Drittanbieter-iframes:

```
add_header Permissions-Policy "geolocation=(), microphone=(), camera=()" always;
```

7. X-XSS-Protection (Defense-in-Depth)

Ein historischer Header für ältere Browser; der eigentliche Schutz liegt in der CSP. Kann optional ergänzt werden:

```
add_header X-XSS-Protection "1; mode=block" always;
```

Kombinierte Konfiguration — Nginx

In Ihren Server-Block (server { ... }) einfügen, mit `nginx -t` prüfen und anschließend mit `systemctl reload nginx` neu laden:

```
add_header Content-Security-Policy "default-src 'self'; frame-ancestors 'self'" always;
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff" always;
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

```
add_header Referrer-Policy "strict-origin-when-cross-origin" always;
add_header Permissions-Policy "geolocation=(), microphone=(), camera=()" always;
```

Fertige Konfiguration für andere Plattformen

Falls Ihr Server nicht Nginx ist, können Sie dieselben Header mit einem der folgenden fertigen Blöcke hinzufügen.

Firebase Hosting — `firebase.json`:

```
{
  "hosting": {
    "headers": [
      {
        "source": "**",
        "headers": [
          { "key": "Content-Security-Policy", "value": "default-src 'self'; frame-ancestors 'self'" },
          { "key": "X-Frame-Options", "value": "SAMEORIGIN" },
          { "key": "X-Content-Type-Options", "value": "nosniff" },
          { "key": "Strict-Transport-Security", "value": "max-age=31536000; includeSubDomains" },
          { "key": "Referrer-Policy", "value": "strict-origin-when-cross-origin" },
          { "key": "Permissions-Policy", "value": "geolocation=(), microphone=(), camera=()" }
        ]
      }
    ]
  }
}
```

Vercel — `vercel.json`:

```
{
  "headers": [
    {
      "source": "/(.*)",
      "headers": [
          { "key": "Content-Security-Policy", "value": "default-src 'self'; frame-ancestors 'self'" },
          { "key": "X-Frame-Options", "value": "SAMEORIGIN" },
          { "key": "X-Content-Type-Options", "value": "nosniff" },
          { "key": "Strict-Transport-Security", "value": "max-age=31536000; includeSubDomains" },
          { "key": "Referrer-Policy", "value": "strict-origin-when-cross-origin" },
          { "key": "Permissions-Policy", "value": "geolocation=(), microphone=(), camera=()" }
      ]
    }
  ]
}
```

Next.js — `next.config.js`:

```
module.exports = {
  async headers() {
    return [
      {
        source: '/(.*)',
        headers: [
          { key: 'Content-Security-Policy', value: 'default-src 'self'; frame-ancestors 'self'' },
          { key: 'X-Frame-Options', value: 'SAMEORIGIN' },
          { key: 'X-Content-Type-Options', value: 'nosniff' },
          { key: 'Strict-Transport-Security', value: 'max-age=31536000; includeSubDomains' },
```

```
          { key: 'Referrer-Policy', value: 'strict-origin-when-cross-origin' },
          { key: 'Permissions-Policy', value: 'geolocation=(), microphone=(), camera=()' }
        ],
      },
    ];
  },
};
```

Apache — `.htaccess` (mod_headers):

```
Header always set Content-Security-Policy "default-src 'self'; frame-ancestors 'self'"
Header always set X-Frame-Options "SAMEORIGIN"
Header always set X-Content-Type-Options "nosniff"
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
Header always set Referrer-Policy "strict-origin-when-cross-origin"
Header always set Permissions-Policy "geolocation=(), microphone=(), camera=()"
```

IIS / ASP.NET — `web.config`:

```
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
      <add name="Content-Security-Policy" value="default-src 'self'; frame-ancestors 'self'" />
      <add name="X-Frame-Options" value="SAMEORIGIN" />
      <add name="X-Content-Type-Options" value="nosniff" />
      <add name="Strict-Transport-Security" value="max-age=31536000; includeSubDomains" />
      <add name="Referrer-Policy" value="strict-origin-when-cross-origin" />
      <add name="Permissions-Policy" value="geolocation=(), microphone=(), camera=()" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

Prüfen Sie nach den Änderungen mit den Browser-Entwicklertools (Tab „Network") oder mit `curl -I https://rest.vulnweb.com`, dass die Header in der Antwort enthalten sind.

Zugriff auf offenliegende Dateien sperren

Festgestellte Pfade: `/db.sql`. Sperren Sie den Zugriff auf Serverebene:

Nginx:

```
location ~ /\.(git|env|ht) { deny all; return 404; }
location ~* \.(bak|old|zip|sql)$ { deny all; return 404; }
```

Apache — `.htaccess`:

```
RedirectMatch 404 /\.git
RedirectMatch 404 /\.env
<FilesMatch "\.(bak|old|zip|sql)$">
  Require all denied
</FilesMatch>
```

Am besten legen Sie diese Dateien zudem gar nicht erst in das Web-Root (public).

DNS- & E-Mail-Sicherheit

Die folgenden Empfehlungen stärken die E-Mail-Authentifizierung der Domain vulnweb.com. Fügen Sie die Einträge über die TXT-Oberfläche Ihres DNS-Anbieters (Cloudflare/GoDaddy/...) hinzu.

DMARC-Eintrag hinzufügen/verschärfen

Überwachen Sie zunächst mit `p=none`; sobald Sie die Berichte geprüft und Fehlalarme ausgeschlossen haben, wechseln Sie zu `quarantine` → `reject`:

```
_dmarc.vulnweb.com.  TXT  "v=DMARC1; p=quarantine; rua=mailto:dmarc@vulnweb.com; fo=1"
```

DKIM-Signierung aktivieren

Erzeugen Sie DKIM über das Panel Ihres E-Mail-Anbieters (Google Workspace/Microsoft 365/Versanddienst) und fügen Sie den bereitgestellten TXT-Eintrag unter `selektor._domainkey` hinzu:

```
google._domainkey.vulnweb.com.  TXT  "v=DKIM1; k=rsa; p=<vom-Anbieter-bereitgestellter-Schlüssel>"
```

DNSSEC aktivieren

Aktivieren Sie **DNSSEC** im Panel Ihres DNS-Anbieters; der Anbieter erzeugt den DS-Eintrag, den Sie bei Ihrem Domain-Registrar eintragen. Es signiert die DNS-Antworten und erschwert Cache-Poisoning.

CORS- & Cookie-Sicherheit

Die folgenden Empfehlungen stärken die CORS- und Cookie-Sicherheit für rest.vulnweb.com.

Ihre CORS- und Cookie-Konfiguration ist nahe an sicheren Voreinstellungen. Behalten Sie beim Hinzufügen neuer Endpunkte das Prinzip der Origin-Allowlist und der Cookie-Flags (Secure/HttpOnly/SameSite) bei.

CSP-Analyse (Content Security Policy)

Die folgenden Empfehlungen stärken die Content-Security-Policy für rest.vulnweb.com. Testen Sie die CSP zunächst mit dem Header `Content-Security-Policy-Report-Only` und wechseln Sie erst dann zum durchgesetzten (enforce) Header, wenn Sie sicher sind, dass die Website nicht beeinträchtigt wird.

Basis-CSP (Einstieg)

Erweitern Sie sie, indem Sie Ihre eigenen Drittanbieter-Domains (Analytics, CDN, Schriftarten) zu `script-src` / `connect-src` / `img-src` hinzufügen:

Nginx:

```
add_header Content-Security-Policy "default-src 'self'; img-src 'self' data:; script-src 'self'; style-src '
```

Firebase Hosting — `firebase.json`:

```
{
  "hosting": {
    "headers": [
      {
        "source": "**",
        "headers": [
          { "key": "Content-Security-Policy", "value": "default-src 'self'; img-src 'self' data:; script-src
        ]
      }
    ]
```

```
  }
}
```

Apache — `.htaccess`:

```
Header always set Content-Security-Policy "default-src 'self'; img-src 'self' data:; script-src 'self'; styl
```

Test mit Report-Only

```
Content-Security-Policy-Report-Only: default-src 'self'; report-uri /csp-report
```

Nachdem Sie die Berichte überwacht und Fehlalarme beseitigt haben, veröffentlichen Sie den Header als `Content-Security-Policy`.

5. Anhang — Glossar

SQL Injection	Eine Injection-Schwachstelle, bei der Nutzereingaben in eine Datenbankabfrage gelangen.
XSS	Cross-Site Scripting — das Einschleusen schädlicher Skripte in Seiten.
IDOR	Insecure Direct Object Reference — Zugriff auf fremde Datensätze durch ID-Manipulation.
CWE	Common Weakness Enumeration — Klassifizierung von Schwachstellentypen.
OWASP	Open Web Application Security Project; bekannt für die Top 10 und Testleitfäden.
TLS	Transport Layer Security (Grundlage von HTTPS).
HSTS	Sicherheitsheader, der den Browser zwingt, nur HTTPS zu verwenden.
CSP	Content Security Policy — Header zur Minderung von XSS/Injection.
CORS	Cross-Origin Resource Sharing; eine lockere Konfiguration ist riskant.
Clickjacking	Klicks werden durch unsichtbares Framing getäuscht.