CyberTestify

Automatisierter Sicherheits-Scan-Bericht

rest.vulnweb.com · Reconnaissance-Paket

CyberTestify-Sicherheitsscan — Abgeschlossen

Bericht-Nr.: **CT-ÖRNEK-F82E**

Verifizierungscode: **5A53-5E8E**

Dieser Bericht ist kein formeller Penetrationstest / keine Zertifizierung.

Inhaltsverzeichnis

ZIEL	PAKET	BERICHT-NR.
rest.vulnweb.com	Reconnaissance-Paket	CT-ÖRNEK-F82E

1. Managementzusammenfassung

GESAMTBEWERTUNG — Hohes Risiko

Dieses Ziel antwortet nicht über HTTPS; die Kommunikation wird unverschlüsselt übertragen (vorrangig sollte auf HTTPS umgestellt werden). Das höchste Risiko wurde im Bereich CMS- & bekannte-CVE-Prüfung (Kein bekannter CMS-Fingerabdruck festgestellt · 21 bekannte CVE in der Server-/Software-Version) festgestellt; eine vorrangige Behebung wird empfohlen. Nachfolgend wird jeder Bereich einzeln berichtet.

⚡ Management-Entscheidung

Es wurden **2** 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** — 3 Bereiche untersucht; das höchste Risiko liegt im Bereich **CMS- & bekannte-CVE-Prüfung** (Kein bekannter CMS-Fingerabdruck festgestellt · 21 bekannte CVE in der Server-/Software-Version).
- ⚠ **HTTPS wird nicht unterstützt:** Das Ziel hat nicht über HTTPS (443) geantwortet; die Erkundung wurde über http:// durchgeführt. Unverschlüsselte Kommunikation ist für sich genommen ein ernster Befund (siehe unten).

Verbesserungsprioritäten

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

🛠 **Schnelle Erfolge (Serverkonfiguration, ~1-2 Tage):** Fehlendes HSTS.

2. Befunde

2.1 Schwachstellenverteilung

In diesem Scan wurden insgesamt **2** 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	Veraltete Komponente (CVE)	Offen	Hoch

Was wurde geprüft, und was kam heraus?

✓ GEPRÜFT — KEIN INDIKATOR GEFUNDEN

- **Subdomain-Takeover-Prüfung** 0 Subdomain-Einträge geprüft (mit der Site verbundene Subadressen — Liste mit Namen: Abschnitt „Subdomain-Takeover-Prüfung"); kein Übernahme-Indikator gefunden (kein verwaister Eintrag, den Dritte übernehmen könnten)
- **API- & Swagger-Erkundung** 12 Pfade geprüft (12 fest + 0 Sitemap-Kandidaten, aus 2 Seiten — VOLLSTÄNDIGE Pfadliste: Abschnitt „API- & Swagger-Erkundung"); kein öffentlich zugängliches API-Schema gefunden (kein von außen lesbares API-Dokument)
- **CMS-/Framework-CVE-Abgleich** 6 passive Signalkategorien untersucht (Header, Meta-Generator, HTML-Spuren, Versionsdateien, bekannte CMS-Pfade, Bibliothekshinweise — vollständige Liste: Abschnitt „CMS & bekannte CVE"); kein bekannter CMS-/Framework-Fingerabdruck festgestellt
- **Sitemap- + robots.txt-Pfaderkundung** 0 aus Sitemap + robots.txt-Disallow abgeleitete Pfade geprüft (nur Vorhanden/Nicht-Vorhanden — Kandidatenliste: Abschnitt „API- & Swagger-Erkundung"); kein sensibler/administrativer Endpunkt gefunden

⚠ INDIKATOR GEFUNDEN — DETAILS IN DEN ABSCHNITTEN UNTEN

- **Server-/Software-Banner → Bekannte CVE** 21 bekannte CVE in Banner-Version (PHP 7.1.26)

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: Das Ziel antwortet nicht über HTTPS; der gesamte Verkehr wird unverschlüsselt (Klartext) übertragen — mitlesbar/veränderbar. 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 · Veraltete Komponente (CVE)

Status: Offen

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

Wie es erkannt wurde: Details im Abschnitt „CMS- & bekannte-CVE-Prüfung" unten.

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

3. Kontrollübersicht & Methodik

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
HTTPS wird nicht unterstützt (unverschlüsselte Kommunikation)	Hoch	Das Ziel antwortet nicht über HTTPS; der gesamte Verkehr wird unverschlüsselt (Klartext) übertragen — mitlesbar/veränderbar. Lösung: gültiges TLS-Zertifikat + HTTP→HTTPS-Umleitung + HSTS.
CMS- & bekannte-CVE-Prüfung — Kein bekannter CMS-Fingerabdruck festgestellt · 21 bekannte CVE in der Server-/Software-Version	Hoch	Details im Abschnitt „CMS- & bekannte-CVE-Prüfung" unten.

UMFANG UND METHODIK

Dieser Bericht wurde in drei Erkundungsbereichen mit **passiven** (nicht ausnutzenden) Techniken automatisch aus von außen beobachtbaren Daten erzeugt:

- **Subdomain-Takeover:** Subdomains werden aus Certificate-Transparency-Logs (crt.sh, ersatzweise certSpotter) erhoben; jede wird per Cloudflare DoH einer DNS-/CNAME-Auflösung unterzogen und mit einer Signaturdatenbank bekannter „dangling" (verlassener Cloud-Dienste) verglichen.
- **API- & Swagger-Erkundung:** Eine feste Liste gängiger API-Dokumentationspfade wird per GET geprüft; gefundene OpenAPI/Swagger-Schemata werden geparst und sensible/nicht authentifizierte Endpunkte markiert (die Endpunkte werden nicht aufgerufen).
- **CMS & bekannte CVE:** Über HTTP-Header, `<meta generator>` und HTML-Muster werden CMS- und Versions-Fingerabdruck ermittelt; ist der Generator verborgen, wird über die EXISTENZ bekannter CMS-Pfade (nur GET/Existenz — kein Login-Versuch) verifiziert. Die erkannte Version wird durch Abfrage der NVD (NIST National Vulnerability Database) mit bekannten CVEs abgeglichen, die die Version ausdrücklich abdecken; ist die Version nicht lesbar, erfolgt keine CVE-Zuordnung (keine erfundenen CVEs).

> Alle Daten wurden von außen und ohne Schaden am Ziel erhoben. Authentifizierungspflichtige Bereiche, das interne Netzwerk und aktive Ausnutzung liegen außerhalb des Umfangs dieses Pakets.

Subdomain-Takeover-Prüfung

Gesamtrisikostufe: Niedrig — In den Certificate-Transparency-Einträgen wurde keine Subdomain gesehen

Aus den Certificate-Transparency-Einträgen (crt.sh / certSpotter) wurden **0** eindeutige Subdomains inventarisiert; davon wurde bei **0** der CNAME-Eintrag aufgelöst und auf Übernahme (Subdomain-Takeover) untersucht.

In den aufgelösten CNAME-Einträgen wurde keine **übernehmbare (dangling)** Subdomain festgestellt, die auf eine verlassene Cloud-Ressource zeigt. Das zeigt, dass Ihre von außen sichtbare Subdomain-Fläche derzeit **schmal und kontrolliert** wirkt.

> Umfang: Nur passive Quellen (Certificate-Transparency-Logs + beobachtbares DNS). Es wurde kein Subdomain-Brute-Force / aktiver Scan durchgeführt.

API- & Swagger-Erkundung

Gesamtrisikostufe: Niedrig — Keine öffentlich zugängliche API-/Swagger-Dokumentation gefunden

Die folgenden **12** gängigen API-Dokumentations-/Erkundungspfade wurden per GET geprüft. Kein Endpunkt wurde aufgerufen/ausgenutzt (passive Erkundung).

Geprüfte Pfade (vollständige Liste)

Pfad	HTTP	Status
/openapi.json	404	Nicht gefunden
/swagger.json	404	Nicht gefunden
/v2/api-docs	400	Nicht gefunden
/v3/api-docs	400	Nicht gefunden
/api-docs	404	Nicht gefunden
/api/docs	400	Nicht gefunden
/api/v1/docs	400	Nicht gefunden
/swagger-ui.html	404	Nicht gefunden
/swagger/index.html	400	Nicht gefunden
/redoc	404	Nicht gefunden
/.well-known/openapi.json	400	Nicht gefunden
/graphql	404	Nicht gefunden

Keiner der geprüften Pfade lieferte ein öffentlich zugängliches API-Schema/-Interface. Das Fehlen öffentlicher API-Dokumentation bedeutet, dass Angreifer Ihre API-Fläche von außen nicht leicht **kartieren** können — das ist im Hinblick auf die externe Angriffsfläche ein positives Zeichen.

> Umfang: Nur öffentlich zugängliche Dokumentationspfade wurden per GET geprüft; kein Endpunkt wurde aufgerufen/ausgenutzt (passive Erkundung).

CMS- & bekannte-CVE-Prüfung

Gesamtrisikostufe: Hoch — Kein bekannter CMS-Fingerabdruck festgestellt · 21 bekannte CVE in der Server-/Software-Version

Untersuchte Fingerabdruck-Quellen

Zur CMS-/Framework- und Versionserkennung wurden in der Startseiten-Antwort die folgenden **6 passiven Signalkategorien** betrachtet (die Angabe „6+ passive CMS-/Technologiesignale" in der Managementzusammenfassung meint genau diese Liste; das „+" steht für die mehreren Untersignale je Kategorie):

- HTTP-Antwort-Header (`Server` , `X-Powered-By` , `X-Generator` , `X-Drupal-Cache` , `X-Magento-Cache-Debug`)
- `<meta name="generator">` -Tag
- HTML-Pfad-/Musterspuren (`/wp-content/` , `/wp-includes/` , `Drupal.settings` , `/sites/all/` , `option=com_` , `/media/jui/` , `typo3conf` , `Magento_`)
- Gängige Versionsdateien (WordPress `/readme.html` , Drupal `/CHANGELOG.txt`)
- EXISTENZ bekannter CMS-Pfade (`/wp-login.php` , `/wp-json/` , `/administrator/` , `/user/login` , `/typo3/` — nur Vorhanden/Nicht-Vorhanden-Prüfung; KEIN Login-/Passwortversuch)
- Bibliotheks-/Plugin-Hinweise (WooCommerce, jQuery-Version)

Keines dieser Signale stimmte mit einem bekannten CMS/Framework überein. Das deutet auf eine eigenentwickelte Anwendung oder eine Installation hin, die CMS-Spuren bewusst verbirgt; beides erschwert die automatische CMS-/CVE-Zuordnung von außen.

Server-/Software-Banner-Version — Bekannte CVE (NVD)

Die aus dem Banner ermittelte(n) Version(en) wurden mit der NVD abgeglichen (KEINE Ausnutzung/Verifikation — nur der Indikator „gibt es bekannte CVEs für diese Version"):

Software/Version	NVD-Ergebnis	Beispiel-CVE
Apache httpd 2.4.25	nicht abfragbar (NICHT sauber)	—
PHP 7.1.26	⚠ **21 bekannte CVE**	CVE-2017-8923

> **Hinweis:** Die folgende CVE-Liste enthält bekannte Schwachstellen, die anhand der erkannten Version **automatisch über die NVD (NIST National Vulnerability Database) zugeordnet** wurden; ob sie für Ihre Version ausnutzbar sind, ist **nicht verifiziert**, und ein Teil kann von Plugins/Themes stammen. Für eine gesicherte Einschätzung werden ein Update + eine gezielte Verifikation empfohlen.

> Umfang: Passiver Fingerabdruck + CVE-Zuordnung über die NVD. Keine CVE wurde **ausgenutzt/verifiziert**.

POSITIVE ZUSICHERUNG — GEPRÜFTE ERKUNDUNGSMETHODEN

Die Erkundung fällt bei den meisten gesunden Zielen sauber aus; dieser Abschnitt macht auch das Ergebnis „nichts gefunden" TRANSPARENT — er zeigt, was TATSÄCHLICH geprüft wurde. Die unten genannten **2 Seiten** sind die Startseite plus die aus Sitemap und robots.txt abgeleiteten Seiten (keine Seite außerhalb Ihrer eigenen Website wurde abgerufen):

Erkundungsbereich	Ergebnis
Subdomain-Takeover-Prüfung	✓ **0 Subdomain-Einträge geprüft (mit der Site verbundene Subadressen — Liste mit Namen: Abschnitt „Subdomain-Takeover-Prüfung"); kein Übernahme-Indikator gefunden (kein verwaister Eintrag, den Dritte übernehmen könnten)**
API- & Swagger-Erkundung	✓ **12 Pfade geprüft (12 fest + 0 Sitemap-Kandidaten, aus 2 Seiten — VOLLSTÄNDIGE Pfadliste: Abschnitt „API- & Swagger-Erkundung"); kein öffentlich zugängliches API-Schema gefunden (kein von außen lesbares API-Dokument)**
CMS-/Framework-CVE-Abgleich	✓ **6 passive Signalkategorien untersucht (Header, Meta-Generator, HTML-Spuren, Versionsdateien, bekannte CMS-Pfade, Bibliothekshinweise — vollständige Liste: Abschnitt „CMS & bekannte CVE"); kein bekannter CMS-/Framework-Fingerabdruck festgestellt**
Server-/Software-Banner → Bekannte CVE	⚠ **21 bekannte CVE in Banner-Version (PHP 7.1.26)**
Sitemap- + robots.txt-Pfaderkundung	✓ **0 aus Sitemap + robots.txt-Disallow abgeleitete Pfade geprüft (nur Vorhanden/Nicht-Vorhanden — Kandidatenliste: Abschnitt „API- & Swagger-Erkundung"); kein sensibler/administrativer Endpunkt gefunden**

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

Was dieses Paket bewertet — und was NICHT

> **BEWERTET (passive Erkundung — nur GET, externe Quellen):** Subdomain-Inventar + Übernahme (dangling CNAME), öffentlich zugängliches API-/Swagger-/OpenAPI-Dokument, CMS-/Framework- und Server-/Software-Banner-Fingerabdruck (Apache/nginx/PHP) + bekannte CVE-Übereinstimmung (NVD), EXISTENZ-Feststellung aus Sitemap + robots.txt-Disallow

abgeleiteter API-/administrativ wirkender Pfade — über die oben definierten 2 Seiten (Startseite + Sitemap-/robots.txt-Ableitungen).

BEWERTET NICHT: aktive Injektions-/IDOR-/XSS-Verifikation und Berechtigungsprüfung entdeckter Endpunkte (Umfang **Aktive Verifikation / Umfassender Pentest**), HTTP-Sicherheits-Header-/CORS-/Cookie-/CSP-Details (Umfang **Basis-Scan / Externe Angriffsfläche**), DSGVO/PCI/ISO-Rahmenzuordnung (Umfang **Compliance**). Die Aussage „kein Indikator gefunden" in einem Bereich **BEWEIST NICHT**, dass Sie sicher sind — sie zeigt nur, dass mit den geprüften passiven Methoden kein Indikator auftrat.

BEST PRACTICES / EMPFOHLENE NÄCHSTE SCHRITTE

Unabhängig vom Ergebnis dieser Prüfung empfohlene dauerhafte Praktiken, um Ihre Angriffsfläche schmal zu halten:

- **Bereinigen Sie ungenutzte CNAME-Einträge regelmäßig** — Einträge, die auf verlassene Cloud-Ressourcen zeigen, bergen Subdomain-Takeover-Risiko; entfernen Sie den DNS-Eintrag, bevor Sie die Cloud-Ressource löschen.
- **Falls Sie eine API-Dokumentation (Swagger/OpenAPI) haben,** halten Sie sie nur für authentifizierten Zugriff offen; veröffentlichen Sie sie in der Produktion nicht öffentlich.
- **Halten Sie Ihre CMS-, Plugin- und Theme-Versionen** per Auto-Update oder regelmäßiger Verfolgung aktuell; bleiben Sie gegen bekannte CVEs gepatcht.
- **Erkennen Sie neue/unerwartete Subdomain-Zertifikate frühzeitig** mit Certificate-Transparency-(CT-)Log-Überwachungstools (crt.sh, certSpotter usw.).
- **Reduzieren Sie die Versions-/Technologieoffenlegung** — verraten Sie über Header/Tags wie `Server`, `X-Powered-By`, `<meta generator>` keine unnötigen Versionsinformationen.
- **Dokumentieren Sie Ihr Subdomain-Inventar** — zu wissen, welche Subdomain zu welchem Dienst/Team gehört, hilft Ihnen, brachliegende Einträge schnell zu erkennen.

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 Behebungsempfehlungen für alle in Ihrer Erkundungsprüfung festgestellten Mängel.

Subdomain-Takeover — proaktiver Überwachungsleitfaden

Derzeit wurde kein übernehmbares Subdomain festgestellt. Um diesen Zustand **dauerhaft** zu sichern, richten Sie ein Certificate-Transparency-(CT-)Überwachungssystem ein, das neue/unerwartete Subdomain-Zertifikate frühzeitig erkennt:

1) Kostenlose E-Mail-/Webhook-Benachrichtigung mit certSpotter

Fügen Sie Ihre Domain (z. B. `nomorelink.com`, inklusive Subdomains) auf `sslmate.com/certspotter` zur Überwachung hinzu; bei Ausstellung eines neuen Zertifikats erhalten Sie eine E-Mail-/Webhook-Benachrichtigung (ein neues Subdomain-Zertifikat kann ein Eintrag sein, den Sie nicht erstellt haben).

2) Ein einfacher Cronjob, der crt.sh regelmäßig abfragt (auf Ihrem eigenen Server)

Ein Beispielskript, das täglich die Subdomain-Liste abruft, mit der vorherigen vergleicht und bei einer neu hinzugekommenen Subdomain benachrichtigt:

```
#!/usr/bin/env bash
# /etc/cron.daily/ct-watch  (chmod +x)
DOMAIN="ornek.com"
STATE="/var/lib/ct-watch/$DOMAIN.txt"
mkdir -p "$(dirname "$STATE")"; touch "$STATE"
curl -s "https://crt.sh/?q=%25.$DOMAIN&output=json" \
  | jq -r ".[].name_value" | tr "[:upper:]" "[:lower:]" | sed "s/^\*\.//" \
  | sort -u > /tmp/ct-now.txt
NEW=$(comm -13 "$STATE" /tmp/ct-now.txt)
if [ -n "$NEW" ]; then
  echo "$NEW" | mail -s "[CT] Yeni alt domain: $DOMAIN" siz@ornek.com
  cp /tmp/ct-now.txt "$STATE"
fi
```

3) Führen Sie ein Subdomain-Inventar — dokumentieren Sie, welche Subdomain zu welchem Dienst/Team gehört; entfernen Sie brachliegende (ungenutzte) CNAME-Einträge aus dem DNS, bevor Sie die Cloud-Ressource löschen (die Reihenfolge der Entfernung ist wichtig).

API- & Swagger-Erkundung — proaktiver Härtungsleitfaden

Es wurde keine öffentlich zugängliche API-Dokumentation gefunden. Um dies dauerhaft zu sichern, stellen Sie Schema-Endpunkte wie Swagger/OpenAPI/ReDoc in der Produktion hinter eine Authentifizierung oder IP-Beschränkung. Wenden Sie das zu Ihrer Plattform passende Beispiel an:

Nginx — IP-Allowlist + Basic-Auth für `/swagger*`, `/api-docs*`, `/openapi.json`

```
location ~* ^/(swagger|api-docs|v2/api-docs|v3/api-docs|openapi\.json|redoc) {
    allow 203.0.113.0/24;   # ofis/VPN IP bloğunuz
    deny all;               # geri kalan herkese kapalı
    auth_basic "Restricted";
    auth_basic_user_file /etc/nginx/.htpasswd;  # htpasswd ile oluşturun
    # ... mevcut proxy_pass/try_files yönergeleriniz ...
}
```

Apache (.htaccess)

```
<LocationMatch "^/(swagger|api-docs|openapi\.json|redoc)">
    AuthType Basic
```

```
    AuthName "Restricted"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
    Require ip 203.0.113.0/24
</LocationMatch>
```

Caddy

```
@apidocs path /swagger* /api-docs* /openapi.json /redoc*
basic_auth @apidocs {
    admin $2a$14$...   # caddy hash-password ile üretin
}
```

Zusätzliche Empfehlungen: Fügen Sie Ihrem OpenAPI-Schema eine globale `security`-Definition hinzu; wenn Sie GraphQL verwenden, deaktivieren Sie Introspection in der Produktion (`introspection: false`).

CMS & bekannte CVE — proaktiver Aktualitätsleitfaden

Es wurde kein bekanntes CMS erkannt (möglicherweise eine eigene/verschleierte Anwendung). Automatisieren Sie das Aktuellbleiben und die Verfolgung bekannter Schwachstellen:

1) Wenn Sie WordPress verwenden — automatische Minor- + Sicherheitsupdates

In `wp-config.php` :

```
define( 'WP_AUTO_UPDATE_CORE', 'minor' );  // Sicherheits-/Minor-Versionen automatisch
```

Für automatische Plugin-/Theme-Updates (WP-CLI):

```
wp plugin auto-updates enable --all
wp theme auto-updates enable --all
```

2) Fügen Sie Ihrer CI/CD einen kostenlosen Abhängigkeits-Scan (SCA) hinzu

- **Dependabot** (GitHub, kostenlos): Fügen Sie dem Repository `.github/dependabot.yml` hinzu:

```
version: 2
updates:
  - package-ecosystem: "composer"   # für WordPress/PHP; npm/pip/… werden ebenfalls unterstützt
    directory: "/"
    schedule: { interval: "weekly" }
```

- **npm audit** (Node-Projekte): Fügen Sie Ihrem CI-Schritt `npm audit --audit-level=high` hinzu; bei hohen/kritischen Schwachstellen soll der Build fehlschlagen.

3) Reduzieren Sie die Versionsoffenlegung — entfernen/deaktivieren Sie versionsverratende Stellen wie `<meta generator>` , `X-Powered-By` , `/readme.html` , `/CHANGELOG.txt` ; so wird die automatische CVE-Zuordnung für Angreifer erschwert.

5. Anhang — Glossar

XSS	Cross-Site Scripting — das Einschleusen schädlicher Skripte in Seiten.
IDOR	Insecure Direct Object Reference — Zugriff auf fremde Datensätze durch ID-Manipulation.
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.