CyberTestify

Automatisierter Sicherheits-Scan-Bericht

testasp.vulnweb.com · Basis-Scan

CyberTestify-Sicherheitsscan — Abgeschlossen

Bericht-Nr.: **CT-ÖRNEK-0D8B**

Verifizierungscode: **F1DE-4625**

Dieser Bericht ist kein formeller Penetrationstest / keine Zertifizierung.

Inhaltsverzeichnis

ZIEL
testasp.vulnweb.com

PAKET
Basis-Scan

BERICHT-NR.
CT-ÖRNEK-0D8B

1. Managementzusammenfassung

GESAMTBEWERTUNG — Hohes Risiko

Ein dringendes Problem, das Besuchern eine direkte Sicherheitswarnung anzeigen kann (z. B. Zertifikatsgültigkeit/Hostname), wurde festgestellt; es sollte vorrangig behoben werden.

⚡ Management-Entscheidung

Es wurden **1** 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.

- **Allgemeine Risikostufe: Hoch** — die Website unterstützt kein HTTPS; die Kommunikation wird unverschlüsselt (im Klartext) übertragen — sie kann abgehört/verändert werden. Vorrangig sollte auf HTTPS umgestellt werden.
- ⚠ Dieses Ziel hat über HTTPS (443) nicht geantwortet; die Prüfung wurde **über http://** durchgeführt. Das Fehlen von HTTPS ist für sich genommen ein Befund (siehe unten).
- 6/6 wichtige Sicherheits-Header fehlen: Content-Security-Policy, X-Frame-Options, X-Content-Type-Options, Strict-Transport-Security, Referrer-Policy, Permissions-Policy.

Umfang und Grenzen: Dieses Paket ist eine **passive, GET-basierte** externe Beobachtung; es wurde kein aktiver Angriff und keine aktive Sonde versucht. Die Aussage „kein Befund" in einem Bereich **BEWEIST NICHT**, dass er sicher ist, da kein aktiver Test durchgeführt wurde — sie zeigt lediglich, dass die von außen beobachtete Konfiguration sauber ist.

Verbesserungsprioritäten

⚡ **Am dringendsten:** HTTPS wird nicht unterstützt (unverschlüsselte Kommunikation).

🛠 **Schnelle Erfolge (Serverkonfiguration, ~1-2 Tage):** Fehlende Content-Security-Policy; Fehlendes X-Frame-Options (Clickjacking); Fehlendes X-Content-Type-Options; Fehlendes HSTS; Fehlende Referrer-Policy.

🗓 **Mittelfristig / Prozess (manuelle Verifizierung oder Code/Architektur):** Fehlende Permissions-Policy.

2. Befunde

2.1 Schwachstellenverteilung

Kritisch	Hoch	Mittel	Niedrig

In diesem Scan wurden insgesamt **7** Befund-Indikatoren festgestellt; die Verteilung nach Schweregrad ist unten dargestellt.

2.2 Master-Befundtabelle

ID	Titel	Status	Schweregrad
CT-1	HTTPS wird nicht unterstützt (unverschlüsselte Kommunikation)	Offen	Hoch
CT-2	Fehlende Content-Security-Policy	Offen	Mittel
CT-3	Fehlendes X-Frame-Options (Clickjacking)	Offen	Mittel
CT-4	Fehlendes X-Content-Type-Options	Offen	Mittel
CT-5	Fehlendes HSTS	Offen	Mittel
CT-6	Fehlende Referrer-Policy	Offen	Mittel
CT-7	Fehlende Permissions-Policy	Offen	Mittel

3 Kontrollbereiche wurden ausgeführt — 1 ohne Befund, 2 mit Befund

✓ GEPRÜFT — KEIN PROBLEM GEFUNDEN

- **Server-/Software-Versionssignatur (auf 8 Seiten)** Kein Problem gefunden (keine bekannte veraltete/EOL-Versionssignatur erkannt)

⚠ BEFUND VORHANDEN — DETAILS IN DEN ABSCHNITTEN UNTEN

- **HTTP-Sicherheits-Header (auf 8 Seiten)** Befund vorhanden (6/6 empfohlene Header fehlen — oben ausführlich)
- **TLS / Zertifikat** Befund vorhanden (HTTPS antwortete nicht — unverschlüsselte Kommunikation)

2.3 Detaillierte Befunde

[MITTEL] CT-2 · 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 8 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-3 · 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 8 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-4 · 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: Der Browser kann den Inhaltstyp erraten (MIME-Sniffing); hochgeladene Dateien könnten wie Skripte ausgeführt werden. Fehlt auf ALLEN 8 geprüften einzigartigen 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-5 · Fehlendes HSTS

Status: Offen

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

Wie es erkannt wurde: Die HTTPS-Pflicht wird dem Browser nicht mitgeteilt; bei Erstanfragen besteht SSL-Stripping-/MITM-Risiko. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.

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

[MITTEL] CT-6 · Fehlende Referrer-Policy

Status: Offen

Beschreibung: Keine Referrer-Policy; die vollständige URL kann an externe Links durchsickern.

Wie es erkannt wurde: An externe Links wird die vollständige URL (Referer) gesendet; Sitzungs-/Datenschutzinformationen können abfließen. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.

Geschäftliche Auswirkung: Adress-/Session-Informationen können an externe Links durchsickern; geringfügige Informationspreisgabe.

EMPFOHLENE BEHEBUNG

Fügen Sie `Referrer-Policy: strict-origin-when-cross-origin` hinzu.

Referenz: CWE-200 · OWASP A05:2021 Security Misconfiguration

[MITTEL] CT-7 · Fehlende Permissions-Policy

Status: Offen

Beschreibung: Kein Permissions-Policy-Header; der Browser-Feature-Zugriff ist unbeschränkt.

Wie es erkannt wurde: Sensible APIs wie Kamera/Mikrofon/Standort sind nicht eingeschränkt; Drittinhalte könnten sie missbrauchen. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.

Geschäftliche Auswirkung: Fehlende Permissions-Policy; Browser-Funktionen (Kamera/Mikrofon/Geolokalisierung) sind nicht eingeschränkt.

EMPFOHLENE BEHEBUNG

Fügen Sie eine `Permissions-Policy` hinzu, die ungenutzte Funktionen deaktiviert (z. B. `geolocation=(), camera=(), microphone=()`).

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

3. Kontrollübersicht & Methodik

HTTP-SICHERHEITS-HEADER

Header	Status	Beschreibung
Strict-Transport-Security	Fehlt	Dem Browser wird die HTTPS-Pflicht nicht mitgeteilt; bei ersten Anfragen besteht das Risiko von SSL-Stripping/MITM.
Content-Security-Policy	Fehlt	Der Browser kann nicht einschränken, welche Ressourcen geladen werden; es gibt keine grundlegende Verteidigung gegen XSS und Content-Injektion.
X-Frame-Options	Fehlt	Die Seite kann in ein iframe einer anderen Website eingebettet werden; der Nutzer kann per Clickjacking getäuscht werden.
X-Content-Type-Options	Fehlt	Der Browser kann den Inhaltstyp erraten (MIME-Sniffing); hochgeladene Dateien könnten wie Skripte ausgeführt werden.
Referrer-Policy	Fehlt	An externe Links wird die vollständige URL (Referer) gesendet; Sitzungs-/Datenschutzinformationen können abfließen.
Permissions-Policy	Fehlt	Sensible APIs wie Kamera/Mikrofon/Standort sind nicht eingeschränkt; Drittanbieter-Inhalte könnten sie missbrauchen.
X-XSS-Protection	Fehlt	Der XSS-Filter älterer Browser ist nicht gesetzt (in modernen Browsern nicht kritisch; der eigentliche Schutz ist die CSP).

Header	Status	Beschreibung
Content-Type	Vorhanden	text/html

TLS-ZERTIFIKATSSTATUS

⚠ Dieses Ziel hat **über HTTPS (443) nicht geantwortet**; es wurde kein gültiges TLS-Zertifikat gefunden. Die Website ist nur über **unverschlüsseltes HTTP** erreichbar (siehe Festgestellte Risiken → „HTTPS wird nicht unterstützt"). Die folgenden Header-Prüfungen wurden über http:// durchgeführt.

SERVER-/TECHNOLOGIESIGNATUR

- Server: Microsoft-IIS/8.5
- X-Powered-By: ASP.NET

FESTGESTELLTE RISIKEN

Befund	Schweregrad	Beschreibung
HTTPS wird nicht unterstützt (unverschlüsselte Kommunikation)	Hoch	Die Website antwortet nicht auf HTTPS; der gesamte ein- und ausgehende Datenverkehr wird unverschlüsselt (im Klartext) übertragen — ein Angreifer im selben Netzwerk kann den Verkehr abhören, Sitzungen/Passwörter stehlen oder Inhalte verändern. Die Prüfung wurde über http:// durchgeführt.
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 8 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 8 geprüften einzigartigen Seiten.
Sicherheits-Header fehlt: X-Content-Type-Options	Mittel	Der Browser kann den Inhaltstyp erraten (MIME-Sniffing); hochgeladene Dateien könnten wie Skripte ausgeführt werden. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.
Sicherheits-Header fehlt: Strict-Transport-Security	Mittel	Die HTTPS-Pflicht wird dem Browser nicht mitgeteilt; bei Erstanfragen besteht SSL-Stripping-/MITM-Risiko. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.
Sicherheits-Header fehlt: Referrer-Policy	Mittel	An externe Links wird die vollständige URL (Referer) gesendet; Sitzungs-/Datenschutzinformationen können abfließen. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.
Sicherheits-Header fehlt: Permissions-Policy	Mittel	Sensible APIs wie Kamera/Mikrofon/Standort sind nicht eingeschränkt; Drittinhalte könnten sie missbrauchen. Fehlt auf ALLEN 8 geprüften einzigartigen Seiten.

POSITIVE ZUSICHERUNG — GEPRÜFTE BEREICHE

Einschließlich der Bereiche ohne Befund wurden die Prüfungen des Einfachen Scans auf **8 eindeutigen Seiten** einschließlich der Startseite tatsächlich ausgeführt. Die folgende Tabelle zeigt transparent auch die Ergebnisse „kein Problem gefunden":

Prüfbereich	Ergebnis
HTTP-Sicherheits-Header (auf 8 Seiten)	**⚠ Befund vorhanden (6/6 empfohlene Header fehlen — oben ausführlich)**

Prüfbereich	Ergebnis
TLS / Zertifikat	**⚠ Befund vorhanden (HTTPS antwortete nicht — unverschlüsselte Kommunikation)**
Server-/Software-Versionssignatur (auf 8 Seiten)	**✓ Kein Problem gefunden (keine bekannte veraltete/EOL-Versionssignatur erkannt)**

Drei-Zustands-Unterscheidung (Ehrlichkeit): ✓ *Kein Problem gefunden* = Prüfung lief, sauberes Ergebnis · ⚠ *Befund vorhanden* = oben ausführlich · ⚠ *Nicht überprüfbar* = keine Daten erhoben (bedeutet NICHT sicher).

Was dieses Paket prüft und was NICHT

PRÜFT (passiv — nur Seitenabruf per GET, es wird keine Sonde/kein Payload gesendet): HTTP-Sicherheits-Header, TLS-/Zertifikatsstatus (Gültigkeit · Hostname · TLS-Version), Server-Software-Versionssignatur und Erkennung bekannter veralteter/EOL-Versionen — auf den 8 entdeckten Seiten.

PRÜFT NICHT: CORS-Richtlinie, Details der Cookie-Flags, Content-Security-Policy-Analyse, DNS-/E-Mail-Einträge (SPF/DKIM/DMARC) und Suche nach offen zugänglichen sensiblen Dateien sind im Paket **Externe Angriffsfläche**; die Zuordnung zu DSGVO/PCI/ISO-Rahmenwerken im Paket **Compliance**; Subdomain-/API-/CVE-Discovery im Paket **Reconnaissance**; die aktive Schwachstellenverifikation (SQLi/XSS/IDOR-Sonde) in den Paketen **Aktive Verifikation** und **Umfassender Pentest**. Dieser Bericht beruht auf passiver Beobachtung; „kein Befund" **BEWEIST NICHT**, dass es sicher ist, da kein aktiver Angriff versucht wurde.

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

Die folgenden Empfehlungen dienen dazu, die auf der Startseite von testasp.vulnweb.com festgestellten fehlenden Sicherheits-Header zu beheben. Für jeden Header folgt zunächst eine kurze Erläuterung, anschließend ein Nginx-Beispiel; ganz am Ende finden Sie fertige Blöcke für andere Plattformen als Nginx (Firebase, Vercel, Next.js, Apache). Kopieren Sie den für Ihren Server passenden Block.

1. Content-Security-Policy (CSP) hinzufügen

Dies ist die wirksamste browserseitige Verteidigung gegen XSS und Content-Injektion. Beginnen Sie mit einer für Ihre 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 ein iframe einer anderen Website eingebettet und so für Clickjacking anfällig wird:

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

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

3. X-Content-Type-Options hinzufügen

Verhindert, dass der Browser MIME-Type-Sniffing betreibt und Inhalte falsch interpretiert (und das dadurch entstehende 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

Verringert den Informationsabfluss im `Referer`-Header, der an externe Links gesendet wird:

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

6. Permissions-Policy hinzufügen

Schränkt Browser-APIs (Kamera, Mikrofon, Standort usw.) ein; verhindert den unerwünschten Zugriff durch Drittanbieter-iframes:

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

7. X-XSS-Protection (Tiefenverteidigung)

Ein historischer Header für ältere Browser; der eigentliche Schutz liegt in der CSP. Sie können ihn optional hinzufügen:

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

Alles zusammenfassende Konfiguration — Nginx

Fügen Sie dies Ihrem Server-Block (server { … }) hinzu, prüfen Sie es mit `nginx -t` und laden Sie anschließend mit `systemctl reload nginx` neu:

```
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

Wenn Ihr Server nicht Nginx ist, können Sie dieselben Header mit dem passenden 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>
```

Überprüfen Sie nach den Änderungen mit den Entwicklertools des Browsers (Registerkarte „Network") oder mit `curl -I https://testasp.vulnweb.com`, ob die Header der Antwort hinzugefügt wurden.

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.
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.