HINWEIS

Ein etabliertes, international tätiges Softwareunternehmen entwickelt und betreibt seit über einem Jahrzehnt eigene webbasierte Softwarelösungen im Bereich Dateikonvertierung und Dateiverarbeitung. Das Unternehmen fokussiert sich auf Software-as-a-Service (SaaS) und betreibt eine Reihe bekannter Online-Tools, die von Millionen Nutzern weltweit verwendet werden.

Die Plattformen ermöglichen unter anderem die Bearbeitung, Umwandlung und Komprimierung von Dateien verschiedenster Formate. Die Anwendungen sind vollständig webbasiert, mehrsprachig verfügbar und auf hohe Nutzerzahlen ausgelegt – mit einer skalierbaren Infrastruktur, die täglich Millionen von Zugriffen verarbeitet.

https://github.com/drdecoded

Beobachtungen zur Frontend-Performance

Auffälligkeit #1

Kategorie: Sicherheit / Informationsleakage - Offenlegung von Server-Technologien

Schweregrad: mittel

Potentielles Risiko:

Potentielles Risiko: Offenlegung von serverseitigen Technologien ermöglicht gezieltere Angriffsversuche (Reconnaissance)

Beschreibung:

Beim Aufruf der Website via curl geben die HTTP-Response-Header mehrere interne Details preis:

Quelle:

```
C:\Users\      >curl -i https://       com
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Mon, 24 Mar 2025 07:40:02 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-Powered-By: PHP/8.2.28
Vary: Accept-Encoding,Cookie
X-Redirect-By: WordPress
Location: https://            com/
X-Cache-Status: MISS
X-Powered-By: PleskLin
```

Diese Informationen verraten den eingesetzten Webserver, die verwendete PHP-Version, das CMS (WordPress) sowie das Hosting-Panel (Plesk). Ein Angreifer könnte diese Informationen gezielt nutzen, um bekannte Schwachstellen in exakt diesen Komponenten zu identifizieren und auszunutzen.

Lösungsansatz:

Die Offenlegung dieser Header sollte unterbunden oder reduziert werden. Mögliche Maßnahmen:

```
# Nginx Konfigurations-Datei

server_tokens off;

## php.ini

expose_php = Off

## Entfernung entsprechender Meta-Tags in Wordpress u.a.

remove_action('wp_head', 'wp_generator');
```

Auffälligkeit #2

Kategorie: Sicherheit / Zugriffskontrolle

Schweregrad: kritisch

Potentielles Risiko:

Unautorisierter Zugriff auf das WordPress-Admin-Panel / Angriffsfläche für Brute-Force- und Login-Angriffe

Beschreibung:

Das Verzeichnis /wp-admin/ ist öffentlich über das Internet erreichbar. Selbst wenn eine Login-Authentifizierung vorhanden ist, stellt der öffentliche Zugang ein erhöhtes Sicherheitsrisiko dar – insbesondere durch automatisierte Angriffe (z. B. Credential Stuffing oder Brute Force) auf die Login-Maske.

Quelle:

https://***.com/wp-admin u. /admin

Lösungsansatz:

Beschränken Sie den Zugriff auf das /wp-admin/-Verzeichnis per Webserver-Konfiguration auf bestimmte IP-Adressen. Beispiele:

```
# Session-Verwaltung
location ~ ^/wp-admin {
   allow 123.123.123.123;  # Erlaubte IP-Adresse
   deny all;
}
```

Auffälligkeit #3

Kategorie: Keine CORS-Richtlinie

Schweregrad: mittel

Potentielles Risiko:

Die Seite erlaubt keine Cross-Origin-Requests. Das ist grundsätzlich sicher, kann aber in komplexeren Architekturen zu Problemen führen. Zudem zeigt es, dass CORS nicht aktiv verwaltet wird – ein Versäumnis, das später zu Fehlkonfigurationen führen kann.

Beschreibung:

CORS-Header wie Access-Control-Allow-Origin fehlen vollständig. Zwar ist das sicherer als eine offene Konfiguration, es deutet jedoch darauf hin, dass CORS nicht bewusst kontrolliert wird – ein Risikofaktor bei der aktuellen Infrastruktur oder späteren API-Integrationen und auch bestehenden API-Requests und Handling.

Quelle:

Content-Encoding:	gzip
Content-Length:	14586
Content-Type:	text/html; charset=UTF-8
Date:	Mon, 24 Mar 2025 07:35:42 GMT
Link:	<https://[redacted]com/wp-json/>; rel="https://api[redacted]/", <https://[redacted].com/wp-json/wp/v2/pages/3700>; rel="alternate"; title="JSON"; type="application/json", <https://[redacted].com/>; rel=shortlink
Server:	nginx
Vary:	Accept-Encoding,Cookie
X-Cache-Status:	BYPASS
X-Powered-By:	PHP/8.2.28
X-Powered-By:	PleskLin

Lösungsansatz:

CORS bewusst setzen oder durchsetzen:

```
# Nginx Konfigurations-Datei
add_header Access-Control-Allow-Origin "https://example.com" always;
add_header Access-Control-Allow-Methods "GET, POST" always;
add_header Access-Control-Allow-Headers "Content-Type" always;
```

Auffälligkeit #4

Kategorie: Kein CSP (Content-Security-Policy) Header

Schweregrad: kritisch

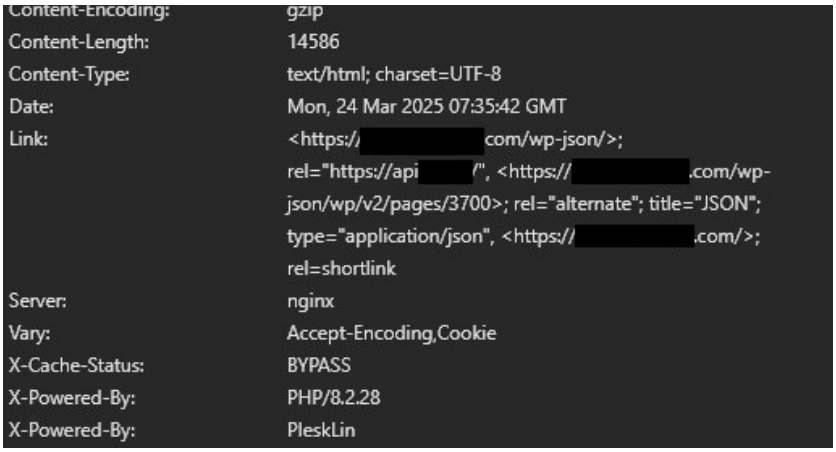
Potentielles Risiko:

Ohne CSP besteht ein erhöhtes Risiko für XSS-Angriffe, da der Browser keine Einschränkungen für eingebettete Skripte, Styles oder Drittinhalte hat.

Beschreibung:

Eine Content-Security-Policy (CSP) ist ein effektives Mittel gegen viele Angriffsarten. Ihre Abwesenheit ist ein Hinweis auf fehlende moderne Sicherheitsmaßnahmen.

Quelle:



Content-Encoding:	gzip
Content-Length:	14586
Content-Type:	text/html; charset=UTF-8
Date:	Mon, 24 Mar 2025 07:35:42 GMT
Link:	<https:// [redacted] com/wp-json/>; rel="https://api [redacted] /", <https:// [redacted] .com/wp-json/wp/v2/pages/3700>; rel="alternate"; title="JSON"; type="application/json", <https:// [redacted] .com/>; rel=shortlink
Server:	nginx
Vary:	Accept-Encoding,Cookie
X-Cache-Status:	BYPASS
X-Powered-By:	PHP/8.2.28
X-Powered-By:	PleskLin

Lösungsansatz:

Eine strikte CSP einführen, z. B.:

```
# Nginx Konfigurations-Datei
add_header Content-Security-Policy "
  default-src 'self';
  script-src 'self' 'nonce-abc123' 'sha256-xyz='
https://cdn.example.com;
  style-src 'self' 'nonce-xyz';
  img-src 'self' data:;
  connect-src 'self' https://api.example.com;
  font-src 'self' https://fonts.gstatic.com;
  object-src 'none';
  frame-ancestors 'none';
  base-uri 'self';
  form-action 'self';
" always;
```

Auffälligkeit #5

Kategorie: Client-Security

Schweregrad: kritisch

Potentielles Risiko:

Cross-Site Scripting (XSS), Code Injection
OWASP Top 10 A03:2021 – Injection

Das eingebettete Inline-Skript wird beim Seitenaufruf direkt im HTML ausgeführt und lädt zusätzlich externen JavaScript-Code. In der aktuellen Konfiguration existiert keine Content-Security-Policy (CSP) mit Nonce oder Hash, was bedeutet, dass potenziell beliebiger Inline-JavaScript-Code ausgeführt werden kann, wenn er erfolgreich injiziert wird (z. B. durch Reflected-XSS).

Beschreibung:

Die Nutzung eines ungeschützten Inline-Skripts stellt eine potentielle Angriffsfläche für XSS-Angriffe dar. Ohne eine restriktive CSP (z. B. script-src 'self' 'nonce-xyz') oder ein Hash-basiertes Whitelisting (sha256-...) ist der Browser gezwungen, alle Inline-Skripte zuzulassen. In Kombination mit dynamisch nachgeladenem Code aus Drittquellen (z. B. Google) wird der Angriffsvektor zusätzlich erweitert. Dies widerspricht modernen Sicherheitsstandards (z. B. OWASP A03:2021 – Injection).

Beispiel Quelle:

```
<script type="text/javascript">
    var _paq = window._paq = window._paq || [];
    /* tracker methods like "setCustomDimension" should be called before "trackPageView" */
    _paq.push(['trackPageView']);
    _paq.push(['enableLinkTracking']);
    (function() {
      var u="[redacted]";
      _paq.push(['setTrackerUrl', u+'matomo.php']);
      _paq.push(['setSiteId', '6']);
      var d=document, g=d.createElement('script'), s=d.getElementsByTagName('script')[0];
      g.type='text/javascript'; g.async=true; g.src=u+'matomo.js'; s.parentNode.insertBefore(g,s);
    })();
</script>
<!-- End Matomo Code -->
<style id=[redacted]> == $0
[redacted]
</style>
```

Lösungsansatz:

Das Skript in eine separate .js-Datei auslagern, die von der eigenen Domain geladen wird oder CSP mit nonce- oder sha256-Mechanismus implementieren, um die Ausführung gezielt erlaubter Inline-Skripte abzusichern.

```
# Beispiel
<script nonce="abc123">...</script>
```

Auffälligkeit #6

Kategorie: Sicherheit / Informationsleakage

Schweregrad: kritisch

Potentielles Risiko:

Angreifer können gezielt bekannte Sicherheitslücken in öffentlich erkennbaren Plugins ausnutzen

Beschreibung:

Durch einfache Mechanismen (z. B. Zugriff auf /wp-content/plugins/, öffentlich einsehbare HTML-Kommentare oder JavaScript- und CSS-Dateien mit Pluginnamen) ist erkennbar, dass folgende WordPress-Plugins im Einsatz sind:

Yoast SEO
Matomo (Analytics)
WP Super Cache

Beispiel Quelle:

```
<!-- Dynamic page generated in 0.308 seconds. -->
<!-- Cached page generated by WP-Super-Cache on 2025-03-22 17:01:42 -->
<!-- super cache -->
```

```
<!-- This site is optimized with the Yoast SEO plugin v24.7 - ht
```

Lösungsansatz:

Vermeidung sprechender Pfade und Kommentare: Keine Pluginnamen oder -pfade in HTML-Kommentaren oder Skriptpfaden offenlegen.

Plugins aktuell halten: Regelmäßige Updates sind essenziell. Optional können sicherheitskritische Plugins durch eigene, schlanke Alternativen ersetzt werden.

Sicherheitsplugins verwenden: Tools wie iThemes Security oder WP Hide & Security Enhancer helfen, WordPress-Komponenten vor neugierigen Scannern zu verstecken.

```
## Nginx Konfigurations-Datei

location ~* ^/wp-content/plugins/ {
   deny all;
}
```

Auffälligkeit #7

Kategorie: Performance / Caching

Schweregrad: mittel

Potentielles Risiko:

Hohe Serverlast, langsame Ladezeiten, ineffiziente Ressourcennutzung trotz aktivem Caching

Beschreibung:

Trotz Einsatz des Plugins WP Super Cache wird bei jeder Seitenanfrage im HTTP-Header X-Cache-Status der Wert MISS oder BYPASS zurückgegeben. Dies deutet darauf hin, dass entweder kein Caching stattfindet oder das Caching-System aufgrund fehlerhafter Konfiguration umgangen wird.

Beispiel Quelle:

```
C:\Users\      >curl -i https://        com
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Mon, 24 Mar 2025 07:40:02 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-Powered-By: PHP/8.2.28
Vary: Accept-Encoding,Cookie
X-Redirect-By: WordPress
Location: https://          com/
X-Cache-Status: MISS
X-Powered-By: PleskLin
```

▼ Response Headers

Content-Encoding:	gzip
Content-Length:	14586
Content-Type:	text/html; charset=UTF-8
Date:	Mon, 24 Mar 2025 07:35:42 GMT
Link:	<https:// com/wp-json/>; rel="https://api /", <https:// .com/wp-json/wp/v2/pages/3700>; rel="alternate"; title="JSON"; type="application/json", <https:// .com/>; rel=shortlink
Server:	nginx
Vary:	Accept-Encoding,Cookie
X-Cache-Status:	BYPASS
X-Powered-By:	PHP/8.2.28
X-Powered-By:	PleskLin

Plugin-Konfiguration prüfen:
In den Einstellungen von WP Super Cache sollte überprüft werden, ob Caching aktiviert ist, der richtige Modus verwendet wird (z. B. „Mod_Rewrite“ statt „PHP“), und ob keine Ausnahmen gesetzt sind (z. B. für eingeloggte Nutzer oder bestimmte URLs).

Positive Beobachtungen

Dateien werden per gzip Encoding kompimiert.
Formulare sind mit einer Nonce versehen die als CSFR-Token gedeutet werden kann.
SSL Cipher und all. Konfiguration ist gut. Entsprechende Tests sind positiv, veraltete Technologien erlauben keinen Key-Exchange.

Protocols

TLS 1.3	Yes
TLS 1.2	Yes*
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

(*) Experimental: Server negotiated using No-SNI

Cipher Suites

# TLS 1.3 (suites in server-preferred order)	
TLS_AES_256_GCM_SHA384 (0x1302) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_AES_128_GCM_SHA256 (0x1301) ECDH x25519 (eq. 3072 bits RSA) FS	128

# TLS 1.2 (suites in server-preferred order)	
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) DH 2048 bits FS	128
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) DH 2048 bits FS	256

Handshake Simulation

Android 4.4.2	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 5.0.0	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 6.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Android 8.0	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Android 8.1	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Android 9.0	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
BingPreview Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 69 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Chrome 70 / Win 10	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Chrome 80 / Win 10 R	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 47 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Firefox 73 / Win 10 R	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Googlebot Feb 2018	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
IE 11 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 2048 FS
IE 11 / Win 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 2048 FS
IE 11 / Win Phone 8.1 R	Server sent fatal alert: handshake_failure		
IE 11 / Win Phone 8.1 Update R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 2048 FS
IE 11 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Edge 16 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Edge 18 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Java 8u161	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Java 11.0.3	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH secp256r1 FS
Java 12.0.1	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH secp256r1 FS
OpenSSL 1.0.1l R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.0.2s R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.1.0k R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS

Positive Beobachtungen

OpenSSL 1.0.2s R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.1.0k R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
OpenSSL 1.1.1c R	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Safari 6 / iOS 6.0.1	Server sent fatal alert: handshake_failure		
Safari 7 / iOS 7.1 R	Server sent fatal alert: handshake_failure		
Safari 7 / OS X 10.9 R	Server sent fatal alert: handshake_failure		
Safari 8 / iOS 8.4 R	Server sent fatal alert: handshake_failure		
Safari 8 / OS X 10.10 R	Server sent fatal alert: handshake_failure		
Safari 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 10 / iOS 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 10 / OS X 10.12 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 12.1.2 / MacOS 10.14.6 Beta R	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Safari 12.1.1 / iOS 12.3.1 R	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Apple ATS 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Yahoo Slurp Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
YandexBot Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS