Kundeninformation und Handlungsanleitung

Sicherheitslücke in JTL-Shop / CVE-2026-54390

Stand: 06.07.2026 **Zielgruppe:** Betreiber von JTL-Shop 5, Administratoren, Hosting-Verantwortliche, Datenschutzbeauftragte, IT-Dienstleister

---

1. Kernaussage

In JTL-Shop wurde eine kritische Sicherheitslücke im Bereich der E-Mail-Verarbeitung festgestellt. Die Schwachstelle betrifft nach JTL-Kundeninformation alle JTL-Shop-5-Versionen ab 5.0.0, wobei die öffentliche CVE-Beschreibung die technische Verwundbarkeit insbesondere für JTL-Shop 5.2.0 bis 5.7.1 beschreibt. Die Lücke ist als **CVE-2026-54390** veröffentlicht und wird als kritisch eingestuft. Über die Schwachstelle konnten Angreifer sensible serverseitige Werte wie Datenbankzugangsdaten und Verschlüsselungsschlüssel auslesen; ab Shop 5.4.0 ist zusätzlich das Schreiben einer Webshell und damit Remote Code Execution möglich. (NVD)

JTL hat hierfür gepatchte Versionen bereitgestellt: **JTL-Shop 5.5.4**, **5.6.2** und **5.7.2**. Diese Releases enthalten laut JTL-Forum jeweils einen kritischen Security Fix. (JTL-Supportforum)

Wichtig: Ein eingespielter Hotfix schützt vor weiterer Ausnutzung der bekannten Lücke. Er beweist aber nicht automatisch, dass der Shop vorher nicht bereits kompromittiert wurde. Deshalb muss zusätzlich geprüft werden, ob Schadcode, Webshells, neue PHP-Dateien, unbekannte Admin-Konten oder auffällige Logeinträge vorhanden sind.

---

2. Was ist passiert?

JTL hat gemeinsam mit dem Sansec Threat Research Team eine Server-Side Template Injection, kurz **SSTI**, in JTL-Shop identifiziert. Die Schwachstelle befindet sich im Zusammenhang mit dem Rendering des E-Mail-Betreffs über Smarty. Sansec beschreibt die Lücke als kritisch, weil ein nicht angemeldeter Angreifer Datenbankzugangsdaten auslesen und in bestimmten Versionen Code auf dem Server ausführen kann. (Sansec)

Eine SSTI entsteht grundsätzlich, wenn Benutzereingaben nicht als normale Daten behandelt, sondern in eine Template-Engine eingebettet und serverseitig ausgewertet werden. PortSwigger beschreibt diese Angriffsklasse als besonders gefährlich, weil sie je nach Template-Engine bis zur vollständigen Kontrolle über den Backend-Server führen kann. (PortSwigger)

Nach den späteren JTL-Informationen wurde die Lücke inzwischen aktiv ausgenutzt. Dabei wurden Backdoor-Dateien in den Webroot geschrieben. Besonders relevant ist die beobachtete Datei:

```
/site.php
```

Zusätzlich wurden in einem konkreten Analysefall folgende Dateien festgestellt:

```
/s_off.php
```

```
/site_off.php
/site.php
/includes/src/Mail/Renderer/SmartyRenderer.php
```

Die Datei `s_off.php` wurde in der vorliegenden Analyse als Datenexfiltrations-Trojaner bewertet. Sie konnte unter anderem Konfigurationsdaten, Datenbankzugänge, den `BLOWFISH_KEY`, SMTP-/FTP-/SFTP-Zugangsdaten, OAuth-Tokens und SSH-Schlüssel suchen beziehungsweise exfiltrieren. Die Datei `site_off.php` wurde als PHP-Webshell/File-Manager bewertet, mit Möglichkeiten zum Durchsuchen, Hochladen, Bearbeiten und Löschen von Dateien sowie zur Ausführung von PHP- und Systembefehlen.

---

3. Was hat JTL unternommen?

JTL hat die Schwachstelle laut Kundeninformation gemeinsam mit externen Sicherheitsforschern verifiziert, eingegrenzt und Patches bereitgestellt.

Öffentlich verfügbare JTL-Releaseinformationen bestätigen:

Shop-Zweig	Gepatchte Version	Bewertung
JTL-Shop 5.5	5.5.4	enthält kritischen Security Fix
JTL-Shop 5.6	5.6.2	enthält kritischen Security Fix
JTL-Shop 5.7	5.7.2	enthält kritischen Security Fix

Die öffentlichen JTL-Releaseposts nennen ausdrücklich, dass es sich um eine SSTI im E-Mail-Betreff handelt, über die Dritte an Blowfish-Key oder Datenbankpasswort gelangen können; ab Shop 5.4.0 ist zusätzlich RCE möglich. (JTL-Supportforum)

Für JTL-Hosting-Kunden gilt laut JTL-Kommunikation: JTL hat die Absicherung bereits vorgenommen. Im JTL-Guide wird außerdem klargestellt, dass JTL-Hosting-Kunden Shop-Updates nicht selbst per Dateiupload durchführen sollen, sondern Updates über das JTL-Kundencenter beziehungsweise den Hosting-Prozess anstoßen. (JTL-Guide)

---

4. Prüfen, ob der eigene Shop den Hotfix bereits hat

4.1 Shop-Version im Backend prüfen

Melden Sie sich im JTL-Shop-Backend an:

```
https://ihre-domain.de/admin
```

Prüfen Sie im Dashboard das Informations-Widget beziehungsweise den Bereich mit der Shop-Version.

Bewertung:

Angezeigte Version	Bewertung
5.7.2 oder höher im 5.7-Zweig	Sicherheitsfix enthalten
5.6.2 oder höher im 5.6-Zweig	Sicherheitsfix enthalten
5.5.4 oder höher im 5.5-Zweig	Sicherheitsfix enthalten

Angezeigte Version	Bewertung
5.7.1, 5.6.1, 5.5.3 oder niedriger	nicht ausreichend, sofern kein manueller Patch eingespielt wurde
5.0.0 bis 5.7.0 mit manuellem Patch	Version kann gleich bleiben; Datei muss geprüft werden

Die öffentliche CVE-/NVD-Beschreibung nennt 5.5.4, 5.6.2 und 5.7.2 als nicht betroffene beziehungsweise bereinigte Versionen. (NVD)

4.2 Dateistruktur im JTL-Backend prüfen

Nach dem Update sollte im Backend geprüft werden:

`Administration → Fehlerbehebung → Diagnose → Dateistruktur`

JTL weist in der Update-Dokumentation darauf hin, dass nach aufgespielten Patches die Dateistruktur geprüft werden kann und dass verwaiste Dateien sicherheitsrelevant sein können. (JTL-Guide)

Wichtig bei manuellem Patch älterer Versionen:

Wenn nur die Datei

`/includes/src/Mail/Renderer/SmartyRenderer.php`

manuell ersetzt wurde, kann JTL-Shop diese Datei als „modifiziert“ anzeigen. Das ist laut JTL-Kundeninformation für diesen konkreten manuellen Patch erwartbar. Werden jedoch weitere unbekannte Dateien als verändert oder verwaist angezeigt, muss dies technisch geprüft werden.

4.3 Datei nicht nur nach Datum bewerten

Ein Änderungsdatum alleine ist kein sicherer Nachweis. Angreifer können Zeitstempel manipulieren. NSA/ASD empfehlen bei Webshell-Verdacht den Vergleich mit einer bekannten sauberen Version der Anwendung, also einen „Known-Good“-Vergleich.

Praktisch bedeutet das:

```
cd /pfad/zum/shop
stat includes/src/Mail/Renderer/SmartyRenderer.php
```

Das Datum ist nur ein Hinweis. Sicherer ist der Vergleich mit der Datei aus dem offiziellen JTL-Patchpaket oder dem offiziellen JTL-Komplettpaket.

---

5. Hotfix oder Update einspielen

5.1 Vorbereitungen

Vor jedem Eingriff:

1. Vollständiges Dateisystem-Backup des Shops erstellen.
2. Datenbank-Backup erstellen.
3. Datei `includes/config.JTL-Shop.ini.php` gesondert sichern.
4. Wartungsmodus aktivieren.

5. JTL-Wawi-Abgleich und JTL-Worker für diesen Shop stoppen.
6. ObjectCache deaktivieren, falls genutzt.
7. Prüfen, ob Template und Plugins kompatibel sind.

Der JTL-Guide beschreibt für Build-/Patch-Updates unter anderem Wartungsmodus, Stopp des Wawi-Abgleichs, Deaktivierung des ObjectCache, Upload der neuen Dateien und anschließende Datenbankaktualisierung im Backend. (JTL-Guide)

5.2 Richtigen Update-Weg wählen

Aktuelle Version	Maßnahme
5.7.1	Patch auf 5.7.2
5.7.0	Patch auf 5.7.2 oder Updatepfad laut JTL
5.6.1	Patch auf 5.6.2
5.6.0	Patch auf 5.6.2
5.5.3	Patch auf 5.5.4
5.5.0 bis 5.5.2	Patch auf 5.5.4
5.0.0 bis 5.4.x	Update auf unterstützte gepatchte Version oder manueller Patch laut JTL-Kundeninformation
JTL-Hosting	nicht selbst per FTP patchen; JTL-/Kundencenter-Prozess nutzen

Patches und Komplettpakete dürfen ausschließlich aus offiziellen JTL-Quellen verwendet werden. Drittquellen sind bei Sicherheitsupdates nicht akzeptabel.

5.3 Dateien übertragen

Bei selbst gehosteten Shops:

1. Patchpaket aus dem JTL-Releaseforum laden.
2. Lokal entpacken.
3. Per SFTP/FTP im **Binärmodus** hochladen.
4. Vorhandene Dateien überschreiben.
5. Übertragungsfehler im FTP-Client prüfen.
6. Backend aufrufen.
7. Falls Datenbankupdates angeboten werden: Sicherung erstellen und Datenbankupdate starten.
8. Danach Dateistruktur prüfen.
9. Verwaiste Dateien prüfen und nur nach Bewertung löschen.
10. Cache leeren und ObjectCache ggf. wieder aktivieren.
11. Wartungsmodus erst nach Funktionstest deaktivieren.

JTL nennt als Nacharbeiten ausdrücklich die Prüfung des Frontends, Registrierung, Kaufprozess, Dateistruktur, Templates, Plugins, E-Mail-Vorlagen, Cache und Logbücher. (JTL-Guide)

---

6. Prüfen, ob der Shop bereits kompromittiert wurde

Ein gepatchter Shop kann vor dem Patch bereits betroffen gewesen sein. Deshalb ist eine Kompromittierungsprüfung erforderlich.

6.1 Access-Logs ab 17.06.2026 prüfen

JTL nennt als Prüfzeitraum in der Folgemeldung **seit dem 17.06.2026**. Prüfen Sie insbesondere Zugriffe auf:

```
/site.php
/s_off.php
/site_off.php
```

Beispiele für Apache/nginx unter Debian/Ubuntu:

```
zgrep -Ei "site\.php|s_off\.php|site_off\.php" /var/log/apache2/access.log*
/var/log/nginx/access.log*
```

Auffällig sind:

- viele Aufrufe in kurzer Zeit
- HTTP-Status `200`
- unterschiedliche Response-Größen
- POST-Anfragen auf unbekannte PHP-Dateien
- Zugriffe von ungewöhnlichen IP-Adressen
- Aufrufe zu Uhrzeiten ohne normale Shopaktivität

Ein einzelner negativer Logfund ist keine Entwarnung, weil Logs rotiert, gelöscht oder unvollständig sein können.

6.2 Neue oder veränderte PHP-Dateien suchen

Pfad anpassen:

```
cd /var/www/shop5
find . -type f -name "*.php" -newermt "2026-06-17" -ls
```

Gezielt nach bekannten verdächtigen Dateinamen suchen:

```
find /var/www/shop5 -type f \( -name "site.php" -o -name "s_off.php" -o -name
"site_off.php" \) -ls
```

Zusätzlich verdächtige PHP-Dateien im Webroot prüfen:

```
find /var/www/shop5 -maxdepth 1 -type f -name "*.php" -ls
```

Wichtig: Nicht jede neue PHP-Datei ist automatisch Schadcode. Updates, Plugins und Wartungsarbeiten erzeugen ebenfalls neue oder geänderte Dateien. Jede unbekannte Datei im Webroot ist jedoch kritisch zu bewerten.

6.3 YARA-Scan ausführen

Wenn die von JTL bereitgestellten YARA-Regeln vorliegen, können bekannte Schadcode-Muster gesucht werden:

```
yara -r jtl-shop-CVE-2026-54390.yar /var/www/shop5/
```

Falls YARA fehlt:

```
apt update
apt install yara
```

Ein negativer YARA-Scan ist keine vollständige Entwarnung. Webshells sind laut NSA/ASD schwer zuverlässig zu erkennen, da sie häufig verschlüsselt, kodiert, obfuskiert oder leicht verändert werden. Deshalb sollten Signaturprüfung, Dateivergleich, Logprüfung und manuelle Bewertung kombiniert werden.

6.4 Verdächtige PHP-Funktionen suchen

Diese Prüfung erzeugt False Positives, ist aber für eine erste Triage hilfreich:

```
grep -RInE "eval\s*\(|shell_exec\s*\(|passthru\s*\(|system\s*\(|popen\s*\(|
proc_open\s*\(|base64_decode\s*\(" /var/www/shop5 --include="*.php"
```

Treffer in Cache-, Vendor-, Plugin- oder legitimen Systemdateien können normal sein. Treffer in unbekannten Dateien im Webroot sind kritisch.

6.5 Admin-Benutzer prüfen

Im JTL-Shop-Backend prüfen:

```
Administration / Benutzerverwaltung / Admin-Benutzer
```

Zu prüfen:

- unbekannte Admin-Konten
- kürzlich angelegte Admin-Konten
- geänderte E-Mail-Adressen
- unerklärliche Rechteerweiterungen
- alte nicht mehr benötigte Benutzer

Alle Admin-Passwörter sollten nach einem Verdachtsfall geändert werden.

6.6 Datenbank-Erreichbarkeit prüfen

Bei älteren Versionen beziehungsweise bei möglichem Auslesen der Datenbankzugangsdaten muss geprüft werden, ob MySQL/MariaDB von außen erreichbar ist.

Auf dem Server:

```
ss -lntp | grep -E ":3306|:3307"
```

Von einem externen System:

```
nc -vz shop-domain.de 3306
```

oder:

```
nmap -Pn -p 3306 shop-domain.de
```

Wenn die Datenbank extern erreichbar ist, muss sie sofort abgesichert werden. MySQL/MariaDB sollte grundsätzlich nicht öffentlich erreichbar sein, sondern nur lokal oder aus explizit freigegebenen internen Netzen.

6.7 Cronjobs und ausgehende Verbindungen prüfen

Webserver-Benutzer prüfen, je nach System meist `www-data`, `apache`, `nginx` oder ein Hosting-Benutzer:

```
crontab -l -u www-data
```

Ausgehende Verbindungen prüfen:

```
ss -tp | grep -E "www-data|apache|nginx"
```

Unbekannte Verbindungen, Reverse-Shell-Verhalten oder wiederkehrende externe HTTP-Verbindungen sind kritisch.

6.8 Shop-Datenbank und Konfiguration prüfen

Prüfen Sie insbesondere:

- unbekannte Admin-Konten
- geänderte Zahlungsarten
- geänderte PayPal-/Payment-/API-Zugangsdaten
- unbekannte Plugins
- veränderte Plugin-Konfigurationen
- veränderte E-Mail-Templates
- unbekannte Weiterleitungen
- unbekannte JavaScript-Einbindungen
- unbekannte Einträge in CMS-/Footer-/Header-Inhalten

Bei E-Commerce-Angriffen ist das Ablegen von Webshells nur ein Teil des Problems. Vergleichbare Shop-Angriffe zeigen regelmäßig, dass Angreifer nach Serverzugriff auch Checkout-Code, JavaScript, Zahlungsprozesse oder Kundendaten ins Visier nehmen. Sansec dokumentiert seit Jahren solche E-Commerce-Angriffe und nennt unter anderem Fälle mit massenhafter Kompromittierung von Magento-Shops. (Sansec)

---

7. Was tun, wenn ein Befall festgestellt wird?

7.1 Sofortmaßnahmen

Wenn `site.php`, `s_off.php`, `site_off.php`, eine Webshell, unbekannte PHP-Dateien oder verdächtige Logzugriffe gefunden werden:

1. Shop sofort offline nehmen.
2. Serverzugang auf bekannte Administrator-IP-Adressen beschränken.
3. Beweise sichern, bevor Dateien gelöscht werden:

    - verdächtige Dateien kopieren
    - Access-Logs sichern
    - Error-Logs sichern
    - SSH-Logs/Auth-Logs sichern
    - Datenbankdump sichern
    - Zeitpunkt der Feststellung dokumentieren
4. Sicherheitspatch einspielen.
5. Verdächtige Dateien aus dem produktiven Webroot entfernen oder isolieren.
6. `SmartyRenderer.php` aus offizieller sauberer Quelle ersetzen.
7. Alle Zugangsdaten rotieren.
8. Datenschutzbeauftragten und Geschäftsleitung informieren.
9. Meldepflicht prüfen.

Polizeiliche Empfehlungen weisen darauf hin, vorhandenes Datenmaterial als Beweismittel möglichst unverändert zu lassen beziehungsweise sauber zu sichern. (Polizei-Beratung)

7.2 Zugangsdaten rotieren

Folgende Zugangsdaten müssen als kompromittiert betrachtet werden, wenn ein Befall festgestellt wurde oder Datenexfiltration nicht ausgeschlossen werden kann:

- JTL-Shop-Admin-Passwörter
- Datenbankpasswort
- `BLOWFISH_KEY`
- FTP-/SFTP-Zugänge
- SSH-Keys
- Hosting-Panel-Zugänge
- SMTP-Zugangsdaten
- OAuth-Tokens
- API-Keys
- Payment-/PSP-/PayPal-/Marktplatz-Zugangsdaten
- JTL-Extension-Store-Anbindung
- sonstige in der Shop-Konfiguration gespeicherte Zugangsdaten

Passwörter ändern allein reicht nicht aus, solange Webshells, Backdoors oder manipulierte Dateien noch vorhanden sind.

7.3 Blowfish-Key rotieren

Laut JTL-Ticketkontext gibt es keinen normalen Shop-internen Standardweg zur Rotation des Blowfish-Keys. In dem vorliegenden Supportkontext wurde ein Skript `rotate_key.php` genannt, das per CLI aus dem Shop-Root ausgeführt werden kann.

Vorgehen nur mit vertrauenswürdiger Skriptquelle:

```
php rotate_key.php --dry-run
```

Nur wenn der Dry-Run fehlerfrei läuft:

```
php rotate_key.php
```

Vorher zwingend:

- Wartungsmodus aktivieren
- Datenbank-Backup erstellen
- Datei `includes/config.JTL-Shop.ini.php` sichern
- Dateisystem sichern

Nachher:

- Extension Store neu anbinden
- Skript sofort löschen
- prüfen, dass das Skript nicht öffentlich per Browser aufrufbar war
- Shop-Funktion testen

Wenn der Shop nachweislich kompromittiert wurde, ist die Rotation des Blowfish-Keys nur eine Teilmaßnahme. Sie ersetzt keine vollständige Bereinigung oder Neuinstallation.

7.4 Beste technische Lösung bei bestätigter Kompromittierung

Die sauberste Lösung ist:

1. neues Betriebssystem / neues Hosting sauber aufsetzen
2. frische JTL-Shop-Dateien aus offizieller Quelle installieren
3. Datenbank nur nach Prüfung migrieren
4. Uploads/Bilder/Medien nur nach Schadcodeprüfung übernehmen
5. Templates und Plugins aus vertrauenswürdigen Quellen neu installieren
6. alle Passwörter und Keys neu setzen
7. Zahlungsarten und Admin-Konten prüfen
8. Logs und Monitoring aktivieren

Diese Vorgehensweise ist aufwendiger, aber bei Webshell-Befall technisch sauberer als das bloße Löschen einzelner Dateien. Webshells dienen Angreifern häufig als persistenter Zugang und können leicht verändert werden, was die vollständige Erkennung erschwert.

---

8. Behördliche Meldung und externe Stellen

8.1 Datenschutzaufsichtsbehörde

Wenn personenbezogene Daten betroffen sein können, muss der Verantwortliche eine Datenschutzverletzung prüfen und gegebenenfalls an die zuständige Datenschutzaufsichtsbehörde melden.

Nach Art. 33 DSGVO muss eine Verletzung des Schutzes personenbezogener Daten unverzüglich und möglichst binnen 72 Stunden nach Bekanntwerden an die zuständige Aufsichtsbehörde gemeldet werden, sofern kein nur geringes Risiko für die Rechte und Freiheiten natürlicher Personen vorliegt. Die Meldung muss unter anderem Art der Verletzung, Kategorien und ungefähre Zahl betroffener Personen und Datensätze, Kontakt des Datenschutzbeauftragten, wahrscheinliche Folgen und ergriffene Maßnahmen enthalten. (Datenschutz-Grundverordnung (DSGVO))

Zuständig ist in der Regel die Datenschutzaufsichtsbehörde des Bundeslandes, in dem das Unternehmen seinen Sitz hat. Die Datenschutzkonferenz stellt eine Übersicht der Datenschutzaufsichtsbehörden von Bund und Ländern bereit. (Datenschutzkonferenz)

Für Unternehmen mit Sitz in Berlin ist die **Berliner Beauftragte für Datenschutz und Informationsfreiheit** zuständig. Dort steht ein Onlineformular für Datenpannen zur Verfügung; die Berliner Behörde weist ebenfalls auf die 72-Stunden-Frist und eine mögliche Information betroffener Personen hin. (datenschutz-berlin.de)

8.2 Betroffene Kunden informieren

Wenn der Vorfall voraussichtlich ein hohes Risiko für die persönlichen Rechte und Freiheiten natürlicher Personen zur Folge hat, müssen betroffene Personen nach Art. 34 DSGVO unverzüglich informiert werden. Die Information muss klar und einfach beschreiben, was passiert ist, welche Folgen wahrscheinlich sind und welche Maßnahmen ergriffen wurden beziehungsweise empfohlen werden. (Datenschutz-Grundverordnung (DSGVO))

Beispiele, bei denen eine Kundeninformation ernsthaft zu prüfen ist:

- bestätigter Abfluss von Kundendaten
- Zugriff auf Bestellungen, Adressen, Telefonnummern oder E-Mail-Adressen
- Zugriff auf Zahlungsdaten oder Zahlungsanbieter-Konfigurationen
- Zugriff auf Zugangsdaten, Tokens oder verschlüsselte Geheimnisse
- unbekannte Manipulationen an Zahlungsarten oder Checkout
- nicht ausschließbare Exfiltration bei vorhandener Webshell

8.3 Polizei / Zentrale Ansprechstelle Cybercrime

Bei einem tatsächlichen Angriff sollte Strafanzeige erstattet werden. Die polizeiliche Kriminalprävention weist darauf hin, dass Cybercrime bei jeder Polizeidienststelle angezeigt werden kann und vorhandene Beweismittel gesichert werden sollten. (Polizei-Beratung)

Für Unternehmen gibt es in den Bundesländern die **Zentralen Ansprechstellen Cybercrime**, kurz ZAC. Für Berliner Unternehmen ist die ZAC beim LKA Berlin zuständig; die Polizei Berlin nennt diese Stelle ausdrücklich als Ansprechpartner für Cyberangriffe und Sicherheitsvorfälle gegen Unternehmen, Behörden, Verbände und Vereine. (Berlin)

Für Berlin:

```
Landeskriminalamt Berlin
LKA 72 – Zentrale Ansprechstelle Cybercrime
Telefon: (030) 4664 972972
```

Bei akuter Gefahr gilt weiterhin der Notruf 110. Für Unternehmen außerhalb Berlins ist die ZAC des jeweiligen Bundeslandes zuständig.

8.4 BSI / CERT-Bund

Eine Meldung an das BSI kann freiwillig erfolgen. Für bestimmte Betreiber, etwa KRITIS, regulierte oder künftig NIS2-relevante Einrichtungen, können zusätzliche gesetzliche Meldepflichten bestehen. Das BSI stellt hierfür Meldewege für IT-Sicherheitsvorfälle bereit. (BSI)

Für normale Onlinehändler ohne besondere Regulierung besteht nicht automatisch eine BSI-Meldepflicht. Dennoch kann eine freiwillige Meldung sinnvoll sein, insbesondere wenn mehrere Shops betroffen sind, ein aktiver Angriff läuft oder neue Indicators of Compromise vorliegen.

8.5 Weitere Stellen

Zusätzlich sollten je nach Befund informiert werden:

- Hostinganbieter
- JTL beziehungsweise JTL-Servicepartner
- Datenschutzbeauftragter
- Rechtsberatung
- Cyberversicherung, falls vorhanden
- Zahlungsdienstleister / PSP / Acquirer, wenn Zahlungsdaten oder Zahlungsprozesse betroffen sein könnten
- betroffene Auftragsverarbeiter oder Verantwortliche, falls der Shop für Dritte betrieben wird

---

9. Dokumentation des Vorfalls

Führen Sie eine zentrale Vorfallsdokumentation. Mindestens erfassen:

```
Datum/Uhrzeit der ersten Kenntnis
betroffene Domain
Shop-Version
Hostingart
gefundenes Schadbild
gefundene Dateien
Logauszüge
betroffene Systeme
betroffene Datenarten
getroffene Sofortmaßnahmen
rotierte Zugangsdaten
informierte Stellen
Entscheidung zur DSGVO-Meldung
Entscheidung zur Kundeninformation
forensische Sicherungen
verantwortliche Personen
```

Nach Art. 33 Abs. 5 DSGVO müssen Verletzungen des Schutzes personenbezogener Daten einschließlich Fakten, Auswirkungen und Abhilfemaßnahmen dokumentiert werden. (Datenschutz-Grundverordnung (DSGVO))

---

10. Praktische Entscheidungslogik

Fall A: Shop ist JTL-Hosting und keine Auffälligkeiten

Maßnahmen:

1. Backend-Version prüfen.
2. JTL-Hinweise im Backend prüfen.
3. Dateistruktur prüfen.

4. Admin-Konten prüfen.
5. Logs auf `/site.php` prüfen, soweit verfügbar.
6. Keine eigenen Dateiuploads im Hosting durchführen.
7. Bei Unsicherheit JTL-Support oder Servicepartner beauftragen.

Fall B: Selbst gehosteter Shop, noch nicht gepatcht

Maßnahmen:

1. Sofort Wartungsfenster planen.
2. Backup erstellen.
3. Patch oder Update aus offizieller JTL-Quelle einspielen.
4. Danach Kompromittierungsprüfung durchführen.
5. Admin-Passwörter ändern.
6. Logs seit 17.06.2026 prüfen.

Fall C: Verdächtige Datei oder verdächtiger Zugriff gefunden

Maßnahmen:

1. Shop offline nehmen.
2. Beweise sichern.
3. Webroot und Logs sichern.
4. Datenschutzbeauftragten informieren.
5. Patch einspielen.
6. Webshells entfernen oder System neu aufsetzen.
7. Alle Credentials rotieren.
8. Datenbank und Admin-Konten prüfen.
9. DSGVO-Meldung prüfen.
10. Polizei/ZAC kontaktieren.

Fall D: Bestätigter Befall mit Webshell oder Datenexfiltration

Maßnahmen:

1. Incident als Sicherheitsvorfall behandeln.
2. Produktivsystem nicht weiter vertrauen.
3. Neuinstallation von Betriebssystem und Shop bevorzugen.
4. Datenmigration nur nach Prüfung.
5. Alle Schlüssel und Passwörter neu setzen.
6. Datenschutzaufsichtsbehörde innerhalb der Frist informieren.
7. Betroffeneninformation prüfen.
8. Strafanzeige/ZAC.
9. Forensik-Dienstleister hinzuziehen.

---

11. Was nicht ausreicht

Nicht ausreichend ist:

- nur das Admin-Passwort zu ändern
- nur `site.php` zu löschen
- nur den Patch einzuspielen, ohne Logs zu prüfen
- sich auf Dateidatum oder Dateigröße zu verlassen
- verdächtige PHP-Dateien im Browser aufzurufen
- Backups ohne Prüfung zurückzuspielen
- die Datenbankzugänge unverändert zu lassen
- SSH-Keys weiterzuverwenden
- den Blowfish-Key unverändert zu lassen, wenn Exfiltration möglich war
- Datenschutzprüfung aufzuschieben

---

12. Einordnung vergleichbarer Vorfälle

Die JTL-Sicherheitslücke passt in ein bekanntes Muster bei E-Commerce-Systemen: Kritische Shop-Schwachstellen werden nach Veröffentlichung sehr schnell automatisiert ausgenutzt, um Webshells, Backdoors oder Skimmer zu platzieren. Sansec beobachtete beispielsweise bei einer Magento-/Adobe-Commerce-Angriffswelle 471 kompromittierte Shops innerhalb einer Stunde. (Sansec)

Bereits beim sogenannten Cardbleed-Vorfall wurden laut Sansec 2.806 Magento-1-Shops infiziert; dabei wurde schädlicher Checkout-Code eingesetzt, um Zahlungsinformationen von Kunden abzugreifen. (Sansec)

Daraus folgt für JTL-Shop-Betreiber: Ein Patch ist zwingend, aber nicht ausreichend. Bei aktiv ausgenutzten Shop-Lücken muss immer zusätzlich geprüft werden, ob bereits Dateien geschrieben, Zugangsdaten entwendet oder Shop-/Checkout-Daten manipuliert wurden.

---

13. Zusammenfassung für Kunden

Prüfen Sie sofort Ihre JTL-Shop-Version. Sicher sind die von JTL bereitgestellten Security-Fix-Versionen **5.5.4**, **5.6.2** und **5.7.2** beziehungsweise ein korrekt eingespielter manueller Patch für ältere Versionen. Selbst gehostete Shops müssen aktiv aktualisiert oder manuell gepatcht werden.

Prüfen Sie danach zwingend, ob Ihr Shop bereits kompromittiert wurde. Relevante Hinweise sind unbekannte Dateien wie `site.php`, `s_off.php` oder `site_off.php`, auffällige Access-Logs seit dem 17.06.2026, unbekannte Admin-Konten, veränderte Zahlungsdaten, verdächtige Cronjobs oder ausgehende Verbindungen des Webserver-Benutzers.

Bei Befall: Shop offline nehmen, Beweise sichern, Schadcode entfernen oder System neu aufsetzen, alle Zugangsdaten rotieren, Datenschutzbeauftragten einbinden, Meldepflicht prüfen und bei Cybercrime-Verdacht Polizei/ZAC kontaktieren.

Haftungsausschluss:

Dieses Dokument wurde nach bestem Wissen auf Grundlage der zum Zeitpunkt der Erstellung vorliegenden Informationen erstellt. Die CIN GmbH übernimmt keine Haftung für die

Vollständigkeit, Aktualität oder rechtliche Verbindlichkeit der Inhalte. Die Umsetzung der beschriebenen Maßnahmen erfolgt in eigener Verantwortung des Shopbetreibers. Eine rechtliche Prüfung, insbesondere zu Datenschutz- und Meldepflichten, sollte durch den Datenschutzbeauftragten oder eine entsprechend qualifizierte Rechtsberatung erfolgen.