

SÉCURITÉ DES SYSTÈMES D'INFORMATION

Enseignante: Dr KIE VICTOIRE

Enseignant-chercheur à ESATIC

CHAPITRE III: Sécurité des applications

1. Vulnérabilités des applications
2. Méthodes de test de sécurité des applications
3. Bonnes pratiques de développement d'applications sécurisées

CHAPITRE III: Sécurité des applications

1. Vulnérabilités des applications

Les vulnérabilités applicatives sont des faiblesses dans une application qu'un attaquant pourrait exploiter pour nuire à la sécurité de l'application. La vulnérabilité peut être introduite dans une application de différentes manières, notamment par des défaillances dans la conception, la mise en œuvre ou la configuration d'une application.

Elles peuvent entraîner une variété de problèmes, tels que :

- **Vol de données** : Les pirates peuvent exploiter des vulnérabilités pour voler des données sensibles, telles que des informations d'identification, des données financières ou des informations personnelles.
- **Prise de contrôle** : Les pirates peuvent prendre le contrôle d'une application vulnérable et l'utiliser pour lancer des attaques contre d'autres systèmes ou pour voler des données.

CHAPITRE III: Sécurité des applications

1. Vulnérabilités des applications

- **Interruption de service** : Les pirates peuvent rendre une application indisponible en exploitant une vulnérabilité.
- **Détérioration de la réputation** : Une violation de données ou une autre attaque due à une vulnérabilité applicative peut nuire à la réputation d'une entreprise.

❑ Types courants de vulnérabilités applicatives

Une liste des dix vulnérabilités applicatives les plus répandues et les plus dangereuses.

- **Contrôle d'accès défaillant** : Cela se produit lorsqu'une application ne contrôle pas correctement qui peut accéder à ses données et à ses fonctions.
- **Défaillances cryptographiques** : Cela se produit lorsqu'une application n'utilise pas correctement le chiffrement pour protéger les données.

CHAPITRE III: Sécurité des applications

1. Vulnérabilités des applications

□ Types courants de vulnérabilités applicatives

- **Injection** : Cela se produit lorsqu'un pirate peut insérer du code malveillant dans une application.
- **Conception incertaine** : Cela se produit lorsqu'une application est conçue de manière à faciliter l'exploitation des vulnérabilités.
- **Mauvaise configuration de la sécurité** : Cela se produit lorsqu'une application n'est pas correctement configurée pour la sécurité.
- **Composants vulnérables et obsolètes** : Cela se produit lorsqu'une application utilise des composants logiciels qui sont connus pour être vulnérables.

CHAPITRE III: Sécurité des applications

1. Vulnérabilités des applications

❑ Types courants de vulnérabilités applicatives

- **Défauts d'identification et d'authentification** : Cela se produit lorsqu'une application n'utilise pas des méthodes d'identification et d'authentification adéquates.
- **Défauts d'intégrité des logiciels et des données** : Cela se produit lorsqu'une application ne protège pas correctement ses logiciels et ses données contre la modification

❑ Prévenir les vulnérabilités applicatives

Il existe un certain nombre de choses que les développeurs et les entreprises peuvent faire pour prévenir les vulnérabilités applicatives, notamment :

- **Utiliser un processus de développement logiciel sécurisé** : Cela implique d'inclure la sécurité dans toutes les phases du processus de développement logiciel, de la conception à la mise en production.

CHAPITRE III: Sécurité des applications

1. Vulnérabilités des applications

❑ Prévenir les vulnérabilités applicatives

- **Effectuer des tests de sécurité réguliers** : Les applications doivent être testées régulièrement pour détecter les vulnérabilités.
- **Déployer des correctifs de sécurité rapidement** : Il est important d'appliquer les correctifs de sécurité dès qu'ils sont disponibles.
- **Utiliser des logiciels et des composants sécurisés** : Les applications doivent utiliser des logiciels et des composants qui sont connus pour être sécurisés.
- **Former les développeurs à la sécurité** : Les développeurs doivent être formés aux menaces de sécurité courantes et à la manière de les coder.

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

La sécurité des applications est essentielle pour protéger les données sensibles et maintenir la confiance des utilisateurs. Il existe plusieurs méthodes de test de sécurité des applications, chacune avec ses propres avantages et inconvénients. Voici un aperçu des principales méthodes :

❑ Tests de pénétration (Pen Testing)

- **Description:** Simule des attaques réelles contre une application pour identifier les vulnérabilités.
- **Techniques :**
 - Black Box : Sans aucune information préalable sur le système.
 - White Box : Avec des informations complètes sur le système.
 - Gray Box : Avec des informations partielles.
- **Avantages :** Révèle des vulnérabilités exploitables en conditions réelles.
- **Inconvénients :** Peut être coûteux et nécessite des compétences spécialisées

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Analyse de code statique (Static Code Analysis)

- **Description** : Analyse le code source sans exécuter l'application pour identifier les vulnérabilités potentielles.
- **Outils courants** : SonarQube, Fortify, Checkmarx.
- **Avantages** : Détecte les vulnérabilités dès la phase de développement.
- **Inconvénients** : Peut produire des faux positifs et nécessite des connaissances en codage.

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Analyse de code dynamique (Dynamic Code Analysis)

- **Description** : Analyse le comportement de l'application en cours d'exécution.
- **Outils courants** : OWASP ZAP, Burp Suite.
- **Avantages** : Permet de détecter des vulnérabilités qui ne sont visibles que pendant l'exécution.
- **Inconvénients** : Ne couvre pas l'ensemble du code et peut être limité par les conditions de test.

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Tests de sécurité des API (API Security Testing)

- **Description** : Évalue la sécurité des interfaces de programmation d'applications (API).
- **Outils courants** : Postman, SoapUI, OWASP ZAP.
- **Avantages** : important pour les applications basées sur des microservices et des architectures distribuées.
- **Inconvénients** : Peut nécessiter une compréhension approfondie des API et de leur utilisation.

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Fuzz Testing

- **Description** : Envoie des entrées aléatoires ou malformées à l'application pour provoquer des comportements inattendus.
- **Outils courants** : AFL (American Fuzzy Lop), Peach Fuzzer.
- **Avantages** : Peut révéler des bugs et des vulnérabilités inconnues.
- **Inconvénients** : Peut générer un grand nombre de faux positifs et nécessiter une analyse approfondie des résultats.

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Modélisation des menaces (Threat Modeling)

- **Description** : Identifie et évalue les menaces potentielles en examinant les architectures, les flux de données et les fonctionnalités de l'application.
- **Techniques courantes** : STRIDE, DREAD.
- **Avantages** : Permet de comprendre les risques dès la phase de conception.
- **Inconvénients** : Nécessite une connaissance approfondie de l'architecture de l'application et des scénarios de menace.

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Revues de code sécurisé (Secure Code Review)

- **Description** : Examen manuel ou automatisé du code source pour identifier les vulnérabilités de sécurité.
- **Avantages** : Complète les analyses statiques et dynamiques en fournissant une perspective humaine.
- **Inconvénients** : Peut être chronophage et dépend de l'expertise des réviseurs

CHAPITRE III: Sécurité des applications

2. Méthodes de test de sécurité

❑ Scan de vulnérabilités (Vulnerability Scanning)

- **Description** : Utilise des outils automatisés pour identifier les vulnérabilités connues dans l'application et son environnement.
- **Outils courants** : Nessus, OpenVAS.
- **Avantages** : Rapide et couvre un large éventail de vulnérabilités connues.
- **Inconvénients** : Peut ne pas détecter des vulnérabilités nouvelles ou spécifiques à l'application.

CHAPITRE III: Sécurité des applications

3. Bonnes pratiques de développement d'applications sécurisées

- **Intégrer la sécurité dès le départ:** Dès la phase de conception de votre application, il est nécessaire de considérer la sécurité comme un élément essentiel. Cela implique d'identifier les menaces potentielles et de mettre en place des protections adéquates.
- **Adopter une approche de défense en profondeur:** La mise en place de plusieurs couches de sécurité permet de minimiser les risques d'intrusions, même si une faille venait à être exploitée.
- **Utiliser des pratiques de codage sécurisées:** Respectez les bonnes pratiques de codage sécurisées pour éviter les vulnérabilités courantes telles que les injections SQL, les scripts intersites et les dépassements de tampon.

CHAPITRE III: Sécurité des applications

3. Bonnes pratiques de développement d'applications sécurisées

- **Gérer les dépendances tierces:** Assurez-vous que les bibliothèques et les composants tiers que vous utilisez sont sécurisés et à jour.
- **Tester et valider rigoureusement:** Effectuez des tests de sécurité complets tout au long du cycle de développement, y compris des tests d'intrusion et des analyses statiques et dynamiques du code
- **Déployer et mettre à jour en toute sécurité:** Mettez en place des processus de déploiement et de mise à jour sécurisés pour minimiser les risques d'exposition aux vulnérabilités.
- **Surveiller et répondre aux incidents:** Mettez en place des mécanismes de surveillance pour détecter les intrusions et les activités suspectes, et ayez un plan de réponse aux incidents prêt à être déployé.

CHAPITRE III: Sécurité des applications

3. Bonnes pratiques de développement d'applications sécurisées

- **Former et sensibiliser les développeurs:** Assurez-vous que vos développeurs sont formés aux bonnes pratiques de sécurité et comprennent les risques encourus.
- **Utiliser des outils de sécurité automatisés:** Intégrez des outils de sécurité automatisés dans votre processus de développement pour identifier et corriger les vulnérabilités plus efficacement.
- **Restez informé des menaces et des vulnérabilités:** Veillez à rester informé des dernières menaces et vulnérabilités de sécurité et mettez à jour vos pratiques en conséquence.

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec Nmap

- Nmap est un outil puissant qui peut être utilisé pour diverses tâches de sécurité réseau.
- Il est important d'utiliser Nmap de manière responsable et de respecter les lois en vigueur.
- N'utilisez jamais Nmap pour attaquer des réseaux sans autorisation.

Partie 1 : Installation de Nmap

Windows:

- ✓ Téléchargez la dernière version de Nmap pour Windows depuis le site officiel :
<https://nmap.org>
- ✓ Exécutez le fichier d'installation et suivez les instructions à l'écran.
- ✓ Une fois l'installation terminée, vous pouvez lancer Nmap en tapant nmap dans l'invite de commande.

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec Nmap

- Nmap est un outil puissant qui peut être utilisé pour diverses tâches de sécurité réseau.
- Il est important d'utiliser Nmap de manière responsable et de respecter les lois en vigueur.
- N'utilisez jamais Nmap pour attaquer des réseaux sans autorisation.

Partie 1 : Installation de Nmap

Linux:

- ✓ Ouvrez un terminal et tapez la commande suivante pour installer Nmap :
`sudo apt install nmap`
- ✓ Nmap est maintenant installé et peut être lancé en tapant nmap dans le terminal.

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec Nmap

Partie 2 : Obtenir l'adresse IP d'un hôte

Pour obtenir l'adresse IP d'un hôte, vous pouvez utiliser la commande ping. Par exemple, pour obtenir l'adresse IP de google.com, tapez la commande suivante :

```
ping google.com
```

La réponse de la commande vous indiquera l'adresse IP de l'hôte.

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec Nmap

Partie 3 : Analyser les résultats d'un scan Nmap

Nmap propose de nombreuses options pour scanner un réseau. Voici quelques exemples :

- ✓ Scan SYN (-sS): Ce type de scan est le plus rapide et le plus discret. Il envoie un paquet SYN à chaque port TCP de l'hôte cible. Si le port est ouvert, l'hôte répondra avec un paquet SYN/ACK.
- ✓ Scan UDP (-sU): Ce type de scan envoie un paquet UDP à chaque port UDP de l'hôte cible. Si le port est ouvert, l'hôte répondra avec un paquet UDP.
- ✓ Scan complet (-T4): Ce type de scan effectue un scan SYN, un scan UDP et un scan des versions des services..

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec Nmap

Partie 3 : Analyser les résultats d'un scan Nmap

Pour analyser les résultats d'un scan Nmap, vous pouvez utiliser les options suivantes :

- A: Cette option affiche des informations supplémentaires sur les hôtes cibles, telles que le nom d'hôte, le système d'exploitation et les services en cours d'exécution.
- v: Cette option augmente le niveau de verbosité de la sortie.
- oX: Cette option permet d'exporter les résultats du scan dans un fichier XML

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec Nmap

Exemple d'analyse d'un scan Nmap:

```
nmap -A -T4 google.com
```

Ce scan affichera les informations suivantes sur l'hôte google.com :

- Adresse IP
- Nom d'hôte
- Système d'exploitation
- Services en cours d'exécution
- Failles de sécurité potentielles

CHAPITRE III: Sécurité des applications

TD2

Scanner des ports avec ZeNmap

Zenmap est une interface graphique pour Nmap, un outil de scanner de réseau open source. Nmap est utilisé pour découvrir les hôtes et les services sur un réseau, donc Zenmap facilite l'utilisation de Nmap en offrant une interface utilisateur graphique.

Étape 1 : Installation de Zenmap

Assurez-vous d'avoir Nmap installé sur votre système. Vous pouvez télécharger Nmap à partir du site officiel (<https://nmap.org/download.html>) et suivre les instructions d'installation. Zenmap est généralement inclus dans la plupart des distributions Nmap.

CHAPITRE III: Sécurité des applications

TD2

Étape 2 : Ouvrir Zenmap

Une fois Nmap installé, vous pouvez ouvrir Zenmap. Selon votre système d'exploitation, cela peut être fait en recherchant Zenmap dans le menu des applications ou en utilisant la ligne de commande.

Étape 3 : Sélectionner une cible

Dans l'interface Zenmap, vous verrez un champ "Target" où vous devez spécifier la cible que vous souhaitez scanner. Cela peut être une adresse IP, un domaine ou même une plage d'adresses IP.

CHAPITRE III: Sécurité des applications

TD2

Étape 4 : Choisissez le profil de numérisation

Zenmap propose différents profils de numérisation, tels que "Intense Scan", "Quick Scan", "Ping Scan", etc. Choisissez le profil qui correspond le mieux à vos besoins. Chaque profil a des options de numérisation spécifiques.

Étape 5 : Configurer les options

Vous pouvez également configurer des options avancées en cliquant sur le bouton "Profile" pour ajuster les paramètres de numérisation. Cela peut inclure des options telles que la détection de l'OS, la détection de services, etc.

Étape 6 : Lancer le scan

Cliquez sur le bouton "Scan" pour lancer le processus de numérisation. Zenmap exécutera alors Nmap avec les paramètres que vous avez spécifiés.

CHAPITRE III: Sécurité des applications

TD2

Étape 7 : Interpréter les résultats Une fois le scan terminé, les résultats s'afficheront dans la fenêtre principale de Zenmap. Les informations peuvent inclure les hôtes découverts, les ports ouverts, les services en cours d'exécution, etc. Vous pouvez utiliser les onglets en haut de la fenêtre pour affiner les résultats.

- Topology**: Affiche la topologie du réseau.
- Hosts**: Présente une liste des hôtes découverts.
- Services**: Affiche les services en cours d'exécution sur les hôtes.
- Host Details**: Fournit des détails spécifiques sur un hôte sélectionné.

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

❑ Présentation de Damn Vulnerable Web Application(DVWA)

- DVWA est une application Web extrêmement vulnérable, codée en PHP et utilisant une base de données MySQL
- Elle est légère, facile à utiliser et plein de failles à exploiter
- DVWA est destiné personnes souhaitant s'entraîner ou voulant en apprendre plus sur les attaques Web, de tester des techniques d'attaques dans un environnement légal.
- L'objectif principal est d'aider les testeurs de pénétration et les professionnels de la sécurité à tester leurs compétences et leurs outils

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

❑ Les failles web disponible dans l'application DVWA

- **Attaque par force brute:** Téléchargement de fichiers,XSS (DOM)
- **Injection de commande:** CAPTCHA non sécurisé,XSS (réfléchi)
- **Attaques CSRF:** Injection SQL / Injection SQL (en aveugle),XSS (stocké)
- **Inclusion de fichier:** ID de session faibles, Contournement CSP
- Exécution de commande via shell_exec en PHP:
- Faille upload

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

❑ Exigences de DVWA

- serveur Web (XAMPP comme alternative)
- PHP
- MySQL
- Autres dépendances possibles (selon le système d'exploitation)

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

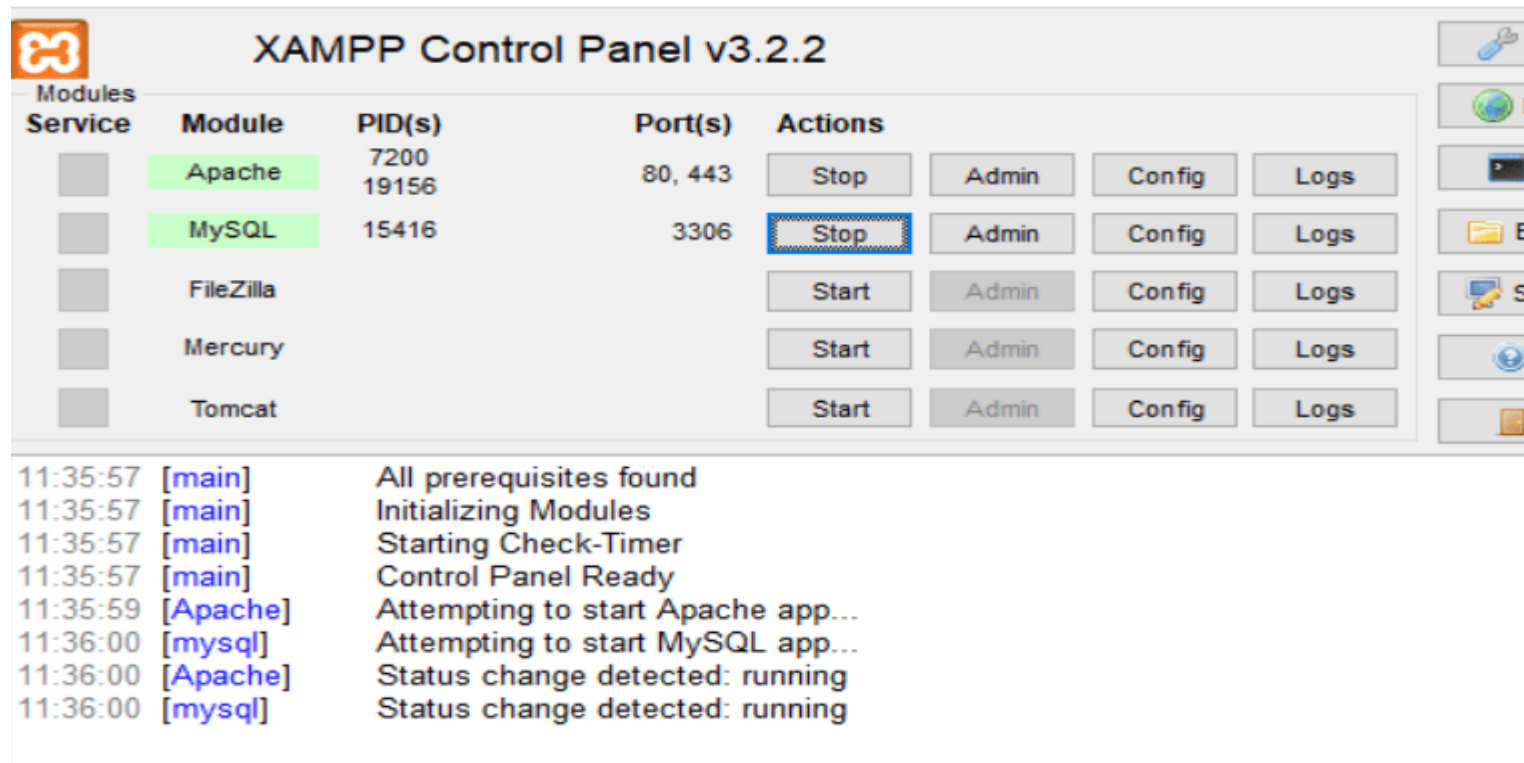
❑ Installation de DVWA

1. Tout d'abord, **Télécharger** l'application, puis placer le **dossier Dvwa** dans votre serveur web WAMP ou XAM ou encore Esayphp.
2. téléchargez et installez un serveur Wamp ou xampp pour notre exemple nous utiliseront xampp.
3. Une fois installé, ouvrez le panneau de contrôle xampp et démarrez les services MySQL et Apache.

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

❑ Installation de DVWA



3. Vous devez alors extraire le contenu du paquet DVWA téléchargé précédemment dans le dossier : C:\xampp\htdocs

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

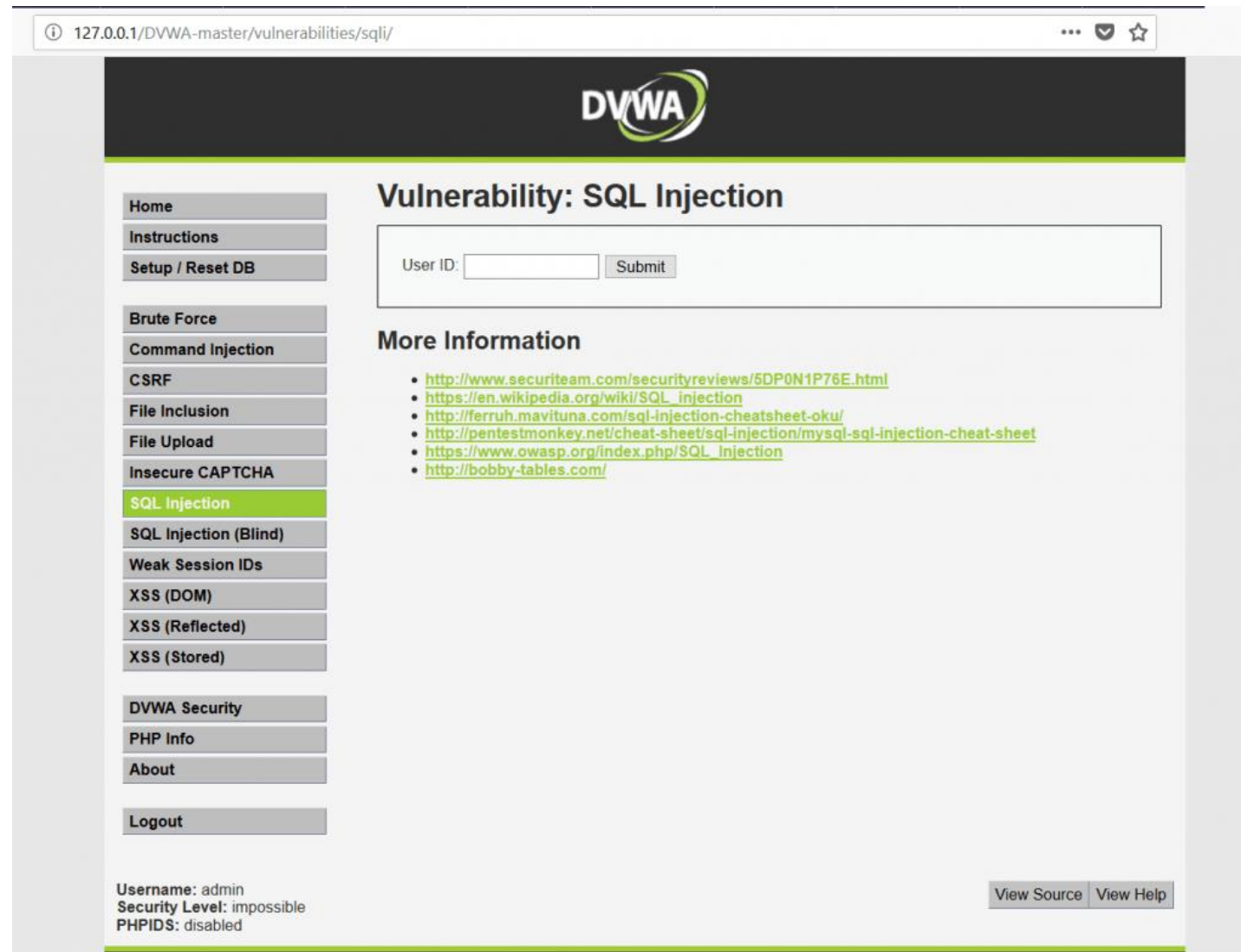
❑ Installation de DVWA

4. Puis, renommez le fichier « **config.inc.php.dist** » qui se trouve dans le dossier « \ **DVWA-master\config** » par « **config.inc.php** ». Dans ce même fichier, supprimez le mot de passe
5. Enfin, allez à l'adresse :**127.0.0.1/DVWA-master** puis créez une nouvelle base de données pour faire vos tests SQL.

CHAPITRE III: Sécurité des applications

3.TP: Détection de vulnérabilités avec Damn Vulnerable Web Application(DVWA)

❑ Installation de DVWA



The screenshot displays the DVWA web application interface. The browser address bar shows the URL `127.0.0.1/DVWA-master/vulnerabilities/sqli/`. The DVWA logo is visible at the top. On the left sidebar, the 'SQL Injection' menu item is highlighted. The main content area is titled 'Vulnerability: SQL Injection' and features a form with a 'User ID:' label, an input field, and a 'Submit' button. Below the form, there is a 'More Information' section with a list of links to external resources. At the bottom left, the user's current session information is displayed: 'Username: admin', 'Security Level: impossible', and 'PHPIDS: disabled'. At the bottom right, there are buttons for 'View Source' and 'View Help'.

127.0.0.1/DVWA-master/vulnerabilities/sqli/

DVWA

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
DVWA Security
PHP Info
About
Logout

Vulnerability: SQL Injection

User ID:

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.cwasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

Username: admin
Security Level: impossible
PHPIDS: disabled