Lycée Jean ROSTAND

BTSSIO

Bloc 3

TD-01
DVWA
Session 2

Durée : 1h30

Auteur : NGO

Sommaire

Le TD est à réaliser individuellement.

Vous devez rendre les réponses de ce TD sur un document libre office normalisé et rédigé avec des copies d'écrans des résultats obtenu à déposer sur le NAS de la section :

DEPOT/TD-01_Session2/nom_prenom.odt

PRESENTATION (RAPPEL)

Damn Vulnerable Web App (DVWA) est une application Web PHP/MySQL qui est sacrément vulnérable. Ses principaux objectifs sont d'aider les professionnels de la sécurité à tester leurs compétences et leurs outils dans un environnement légal, d'aider les développeurs Web à mieux comprendre les processus de sécurisation des applications.

Les principaux objectifs de DVWA

- Apprendre à identifié les vulnérabilités des sites et des applications web,
- Tester les techniques d'exploitation et d'intrusion,
- Apprendre les méthodes de correction pour mieux sécuriser des systèmes.

Les failles web disponible dans l'application DVWA

- Attaque par force brute
- Exécution de commande via shell_exec en PHP
- Attaques CSRF
- Faille Include
- Attaques par SQL Injection
- Faille upload
- Attaques XSS

Le login de l'application c'est : **admin** et le mot de passe : **password**

Aide pour les commandes linux : https://doc.ubuntu-fr.org/accueil

MISSION 3 : File Upload

Le partage de fichiers est une technique de transfert de fichier consistant à distribuer ou à donner accès, à distance, à des données numériques à travers un réseau informatique.

Les failles d'upload sont une vulnérabilité courante sur les sites web qui permettent aux utilisateurs de télécharger des fichiers. Les attaquants peuvent exploiter cette vulnérabilité en téléchargeant des fichiers malveillants sur le serveur, ce qui peut entraîner des compromissions de sécurité

1. Choisir une image puis appuyer sur le bouton Upload, que constatez-vous ?

2. Cliquer sur le lien de l'image, que visualisez-vous ?

3. Créer un fichier phpinfo.php avec le contenu suivant

```
< ?php
      phpinfo();
?>
```

4. Uploader ce fichier, indiquer si l'upload a fonctionné

5. Saisir sur l'URL, le chemin et nom du fichier où le fichier a été stocké, que constatez-vous ?

6. Indiquer les risques d'un fichier phpinfo (ou fichier malveillant) en accès public

7. Consulter le code source et indiquer quelle vérification est effectuée en medium / high pour sécuriser les upload de fichier

8. En médium, indiquer si vous pouvez charger une image avec une extension « .jpg ». Justifier

9. En medium, indiquer si vous pouvez charger.PNG (extension en majuscule). Justifier

10. Indiquer à quoi sert la fonction `strtolower` et indiquer pourquoi elle est nécessaire ici

11. Indiquer les recommandations pour se protéger de cette faille en analysant les différents niveaux du code source.

MISSION 4 : CSRF

En sécurité des systèmes d'information, le cross-site request forgery, abrégé CSRF ou XSRF, est un type de vulnérabilité des services d'authentification web.

L'objectif de cette mission sera de faire en sorte que l'utilisateur actuel modifie lui-même son mot de passe, sans qu'il soit au courant de ses actions, en utilisant une attaque CSRF.

1. En difficulté « Low », tester la fonctionnalité du formulaire en essayer de changer votre mot de passe par « password123 » et indiquer ce que vous constatez.

2. Indiquer selon vous et par rapport à vos éventuels changements de mot de passe sur d'autres applications, la première faille du formulaire en termes de bonne pratique.

3. Consulter l'URL tel qu'elle apparait dans votre navigateur. Indiquer la faille de sécurité.

4. Déduiser à quoi correspond « password_new » et « password_conf »

5. Expliquer la démarche qui pourrait permettre une réinitialisation du mot de passe de l'utilisateur sans action de sa part sur le formulaire

6. créer une nouvelle URL
http://IP_VM/dvwa/vulnerabilities/csrf/?password_new=[votre_nouveau_mot_de_passe]&password_conf=[votre_nouveau_mot_de_passe]&Change=Modifier# puis valider, que constatez vous ?

7. Déconnecter vous, puis reconnectez-vous avec le nouveau mot de passe pour vérifier si l'attaque a fonctionné.

8. Indiquer quel technique peut-on utiliser inciter un utilisateur à cliquer sur le lien malveillant à votre place.

9. Indiquer les étapes (4) démarche pour que cette attaque puisse fonctionner.

10. Indiquer les recommandations pour se protéger de cette faille au niveau utilisateur puis au niveau développement en analysant les différents niveaux du code source.

11. La mission est terminé vous pouvez remettre le mot de passe par défaut pour les utilisations futures.