Réseau et Sécurité
TP 4 - Attaques web

1 Analyse de logs

Un service web est disponible sur le serveur central de l'entreprise après analyse du réseau. Il y a quelques années, les prestataires de services ont mis en place un ERP. Bien qu'il n'ait pas été mis à jour depuis, le directeur ne vous l'avait pas mentionné car il le considérait comme intouchable. Il semble pourtant qu'il ait été compromis, découvrez comment !

— Téléchargez le logs de l'attaque suivant et ouvrez la avec votre éditeur de texte favoris :

`https://perso.isima.fr/~chaolivi/reseausecu/dolibarr_access.txt`

1. A quelle heure commence l'attaque bruteforce précédente ?
2. Une fois authentifié l'attaquant à cherché une vulnérabilité sur une page php du serveur cloud de l'entreprise victime. Pour cela il essaie plusieurs extensions jusqu'à en trouver une vulnérable. Quelles extensions a-t-il testé ?
3. Quelle est l'extension qu'il utilise par la suite de l'attaque ?
4. Dans la suite de l'attaque (lignes 1340-1343), l'attaquant se sert du fichier php vulnérable pour exécuter des commandes shell sur le serveur. Que fait-il ?

2 Failles XSS

Préparation : cela peut prendre un peu de temps, commencez le reste du TP en parallèle.

— Dans votre machine virtuelle, téléchargez l'image docker suivante : `https://perso.isima.fr/~chaolivi/reseausecu/lab5.tar.gz`

— Entrez la commande suivante : docker load -i lab5.tar.gz

— Entrez la commande suivante :

`sudo docker run --rm --add-host=host.docker.internal:host-gateway -p 5000:5000 lab5`
La commande tourne en continu, laissez le terminal tel quel et ouvrez en un autre.

Les attaques de type Cross-Site Scripting (XSS) sont un type d'injection, dans lequel des scripts externes et malveillants sont injectés dans des sites web de confiance. Les failles XSS réfléchies, aussi qualifiées de "non permanentes", apparaissent lorsque des données fournies par un client web sont utilisées telles quelles par les scripts du serveur pour produire une page de résultats.
Les failles XSS stockées, aussi qualifiées de "permanentes", se produisent quand les données fournies par un utilisateur sont stockées sur un serveur (dans une base de données, des fichiers, ou autre), et ensuite ré-affichées.

<script> commande a executer </script>

1. Accédez à la page suivante : `http://127.0.0.1:5000` et réalisez une attaque XSS réfléchie. Comment avez-vous fait ?
2. Accédez à la page suivante : `http://127.0.0.1:5000` et réalisez une attaque XSS stockée. Comment avez-vous fait ?

3 Failles LFI et RFI

Sur la page web, nettoyez les commentaires existants s'il y en a avec le bouton "Clear" de la page web. Le code Python permettant le filtrage des commentaires par le serveur est le suivant :

```
def get_comments(search_query=None):
```

```
db = connect_db()
results = []
get_all_query = 'SELECT comment FROM comments'
for (comment,) in db.cursor().execute(get_all_query).fetchall():
    if search_query is None or search_query in comment:
        results.append(comment)

if search_query and results == []:
    if search_query.endswith(".pyb"):
        results = [execute(search_query)]
    elif search_query.startswith("http"):
        results = [requests.get(search_query).text]
    else:
        try:
            results = [open(search_query, "rb").read()]
        except FileNotFoundError:
            pass

return results
```

1. Réalisez une attaque LFI sur le fichier /etc/shadow du serveur web. Comment avez-vous fait ? Expliquez la vulnérabilité dans le code. Trouvez le mot de passe du compte root du serveur avec john ou équivalent.
2. Réalisez une attaque RFI incluant la page `http://perdu.com`. Comment avez-vous fait ? Expliquez la vulnérabilité dans le code. Essayez avec d'autres pages web.
3. Réalisez une attaque RFI exécutant la page

   `https://perso.isima.fr/~chaolivi/reseausecu/attack.pyb`.

   Comment avez-vous fait ? Expliquez la vulnérabilité dans le code.

4 SQL Injection

L'objectif est de trouver des attaques SQL à la main dans un premier temps sur une base de données et dans un second temps de se servir des outils `SQLMAP` [1], `JohnTheRipper` [2] et `HashCat` [3] pour hacker une autre base de données.

Télécharger le fichier suivant avec :

```
wget https://perso.isima.fr/~chaolivi/reseausecu/SQLIA.tar
```

Pour lancer docker aller dans le répertoire `SQLIA` et faites `docker-compose up` ce qui va lancer le site `http://172.19.19.19:8080/`.

Vérifiez que le site est accessible avec `links http://172.19.19.19:8080` (en ligne de commande), avec `curl http://172.19.19.19:8080/` (aussi en ligne de commande) ou avec Firefox.

Il faudra peut-être `unset` le proxy et/ou utiliser `links`.

Une fois docker quitté il faut faire `docker-compose down --volumes` pour arrêter proprement le système.

1. `https://sqlmap.org/`
2. `https://www.openwall.com/john/`
3. `https://hashcat.net/wiki/doku.php?id=example_hashes`

1. Aller sur la page `level0` et monter une attaque par SQL injection sur le site `myblog.com` pour se connecter comme `user` et comme `admin`. Sachant que la requête à la base de données est :

   ```
   SELECT * FROM users
   WHERE ‘email‘ = ’" . $email . "’ AND ‘password‘ = ’" . $password . "’";
   ```

   Quand le nombre de résultats de la requête est 1 alors l'utilisateur choisi peut se connecter.
2. Installer SQLMAP en faisant un git clone de :

   `https://github.com/sqlmapproject/sqlmap.git --depth=1 --single-branch --branch 1.5.9`

   Aller sur la page `level1` et utiliser l'outil `SQLMAP` avec les options `-data` et ensuite `-dump` (cette deuxième option est lente mais très efficace). Pour récuperer la totalité de la BD.

   Une fois le contenu de la BD obtenu, utiliser le dictionnaire `rockyou.txt` disponible à

   `https://perso.isima.fr/~chaolivi/reseausecu/rockyou.txt.tar.gz`

   avec le logciel JohnTheRipper ou Hashcat pour retoruver les mots de passe des deux utilisateurs de ce niveau.

Remarque pour effacer les résultats de `sqlmap` il faut faire `sqlmap --purge`.

5 Failles CSRF

1. Télécharger le fichier suivant : `https://perso.isima.fr/~chaolivi/reseausecu/CSRF_bad_server.tar` Placez ce fichier dans votre `public_html` ou bien lancez le serveur avec la commande `php`. Authentifiez-vous sur le site avec le login `etudiant` et le mot de passe `securepassword`. Validez que vous pouvez afficher et modifier le mot de passe. Attention, si votre session PHP est détruite, le mot de passe redeviendra celui par défaut.
2. Quels champ de formulaire pouvez-vous trouver dans le formulaire de changement de mot de passe ?
3. Regardez le code du serveur et notez qu'il ne présente aucune protection contre les attaques par CSRF. Créez une page `malicious.html` qui permet de modifier le mot de passe en `motdepasspirate` une fois le client authentifié.
4. Proposez une correction du code du serveur implémentant une protection à base de token CSRF aléatoires et vérifiez que votre page d'attaque ne fonctionne plus.