

Manuel Complet de THC Hydra

Introduction à THC Hydra

THC Hydra, souvent simplement appelé Hydra, est un outil de craquage de mots de passe par force brute et par dictionnaire, open-source et largement utilisé. Développé par The Hacker's Choice (THC), il est conçu pour tester la sécurité des systèmes en tentant de deviner les identifiants de connexion (nom d'utilisateur et mot de passe) pour une multitude de services réseau. Hydra est un outil essentiel pour les professionnels de la cybersécurité, les testeurs d'intrusion (pentesters) et les auditeurs de sécurité afin d'évaluer la robustesse des mécanismes d'authentification.

Qu'est-ce que la force brute ?

La force brute est une méthode d'attaque qui consiste à essayer toutes les combinaisons possibles de caractères pour un mot de passe jusqu'à ce que la bonne combinaison soit trouvée. Les attaques par dictionnaire, quant à elles, utilisent une liste de mots de passe couramment utilisés ou de mots provenant de dictionnaires pour tenter de se connecter.

Pourquoi utiliser Hydra ?

Hydra est apprécié pour sa rapidité, sa flexibilité et sa capacité à prendre en charge un grand nombre de protocoles. Il peut être utilisé pour tester la sécurité de divers services, notamment :

- **Web** : HTTP/HTTPS (GET/POST), formulaires web
- **Base de données** : MySQL, PostgreSQL, MSSQL, Oracle
- **Fichiers** : FTP, SMB, SSH, Telnet
- **Mail** : POP3, IMAP, SMTP
- **Autres** : VNC, RDP, IRC, SNMP, Cisco, etc.

Installation de THC Hydra

THC Hydra est préinstallé sur de nombreuses distributions Linux orientées sécurité, comme Kali Linux et Parrot OS. Si ce n'est pas le cas, ou si vous utilisez un autre système d'exploitation, voici les méthodes d'installation courantes :

Installation sur Linux (Debian/Ubuntu)

Sur les systèmes basés sur Debian ou Ubuntu, vous pouvez installer Hydra via le gestionnaire de paquets `apt` :

```
sudo apt update  
sudo apt install hydra
```

Installation sur Linux (Arch Linux)

Sur Arch Linux, vous pouvez l'installer depuis les dépôts officiels :

```
sudo pacman -S hydra
```

Installation depuis les sources

Si vous souhaitez compiler Hydra depuis les sources (par exemple, pour obtenir la dernière version ou pour une distribution non prise en charge), vous pouvez suivre ces étapes :

1. Cloner le dépôt GitHub :

```
bash git clone https://github.com/vanhauser-thc/thc-hydra.git cd  
thc-hydra
```

2. Installer les dépendances (les noms des paquets peuvent varier selon la distribution) :

```
bash sudo apt install libssl-dev libssh-dev libidn11-dev  
libpcre3-dev libpq-dev libmysqlclient-dev libfreerdp-dev libncp-  
dev libmemcached-dev libgnutls28-dev libgcrypt-dev
```

3. Compiler et installer : `bash ./configure make sudo make install`

Installation sur Windows

Bien que Hydra soit principalement un outil Linux, il existe des versions compilées pour Windows. Vous pouvez souvent trouver des binaires précompilés sur des sites tiers ou des dépôts GitHub non officiels. Soyez prudent lors du téléchargement de binaires depuis des sources non officielles.

Une autre approche consiste à utiliser le Sous-système Windows pour Linux (WSL) et d'installer Hydra comme sur une distribution Linux.

Vérification de l'installation

Après l'installation, vous pouvez vérifier que Hydra fonctionne correctement en exécutant la commande suivante dans votre terminal :

```
hydra --version
```

Ceci devrait afficher la version de Hydra installée, confirmant que l'installation a réussi.

Commandes et Exemples d'Utilisation de THC Hydra

Hydra est un outil en ligne de commande puissant et flexible. Voici les options et les exemples d'utilisation les plus courants pour vous aider à démarrer.

Syntaxe Générale

La syntaxe de base de Hydra est la suivante :

```
hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr]
[-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f]
[-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [-m
MODULE_OPT] [service://server[:PORT][/OPT]]
```

Options Clés

- `-l LOGIN` ou `-L FILE` : Spécifie un nom d'utilisateur unique (`LOGIN`) ou un fichier (`FILE`) contenant une liste de noms d'utilisateur à tester.
- `-p PASS` ou `-P FILE` : Spécifie un mot de passe unique (`PASS`) ou un fichier (`FILE`) contenant une liste de mots de passe à tester.
- `-C FILE` : Spécifie un fichier au format "login:pass" (utilisateur:motdepasse) pour tester des paires prédéfinies.
- `-e nsr` : Options supplémentaires pour les mots de passe :
 - `n` : Essayer un mot de passe nul (vide).
 - `s` : Essayer le nom d'utilisateur comme mot de passe.
 - `r` : Essayer le nom d'utilisateur inversé comme mot de passe.
- `-o FILE` : Enregistre les identifiants trouvés dans un fichier.
- `-t TASKS` : Nombre de tâches parallèles par cible (par défaut : 16).
- `-f` : Arrête l'attaque dès qu'une paire login/mot de passe est trouvée pour une cible.
- `-s PORT` : Spécifie un port différent du port par défaut pour le service.

- `-v` / `-V` / `-d` : Mode verbeux / afficher login+pass pour chaque tentative / mode débogage.
- `server` : La cible (adresse IP ou nom d'hôte).
- `service` : Le service à attaquer (ftp, ssh, http-get, pop3, etc.).

Exemples d'Utilisation Courants

Voici quelques exemples pratiques pour illustrer l'utilisation de Hydra sur différents services.

1. Attaque FTP

Pour attaquer un serveur FTP avec un nom d'utilisateur spécifique et une liste de mots de passe :

```
hydra -l utilisateur -P /chemin/vers/liste_mots_de_passe.txt  
ftp://192.168.1.100
```

Pour attaquer un serveur FTP avec une liste d'utilisateurs et une liste de mots de passe :

```
hydra -L /chemin/vers/liste_utilisateurs.txt -P /chemin/vers/  
liste_mots_de_passe.txt ftp://192.168.1.100
```

2. Attaque SSH

Pour attaquer un serveur SSH avec un nom d'utilisateur spécifique et une liste de mots de passe :

```
hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://  
192.168.1.101
```

3. Attaque HTTP (Formulaire POST)

Pour attaquer un formulaire de connexion HTTP qui utilise la méthode POST. Vous devrez analyser le formulaire pour trouver les noms des champs utilisateur et mot de passe, ainsi que l'URL d'action.

Exemple de commande pour un formulaire avec les champs `user` et `pass` et une page de succès `Welcome.php` :

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.1.102 http-post-form "/login.php:user=^USER^&pass=^PASS^:S=Welcome.php"
```

- `/login.php` : Le chemin vers la page de connexion.
- `user=^USER^&pass=^PASS^` : Les noms des champs du formulaire et les marqueurs pour l'utilisateur et le mot de passe.
- `S=Welcome.php` : Indique à Hydra de rechercher la chaîne "Welcome.php" dans la réponse pour déterminer une connexion réussie.

4. Attaque RDP (Remote Desktop Protocol)

Pour attaquer un service RDP :

```
hydra -L /chemin/vers/liste_utilisateurs.txt -P /chemin/vers/liste_mots_de_passe.txt rdp://192.168.1.103
```

5. Attaque SMB (Server Message Block)

Pour attaquer un partage SMB :

```
hydra -L /chemin/vers/liste_utilisateurs.txt -P /chemin/vers/liste_mots_de_passe.txt smb://192.168.1.104
```

6. Utilisation d'un fichier de paires utilisateur:motdepasse

Si vous avez un fichier contenant des paires utilisateur:motdepasse (par exemple, `credentials.txt` avec `user1:pass1`, `user2:pass2`, etc.) :

```
hydra -C /chemin/vers/credentials.txt ssh://192.168.1.101
```

7. Spécifier un port non standard

Si le service tourne sur un port non standard (par exemple, SSH sur le port 2222) :

```
hydra -l user -P passlist.txt -s 2222 ssh://192.168.1.101
```

8. Arrêter après la première paire trouvée

Pour arrêter le scan dès qu'une paire valide est trouvée :

```
hydra -f -l user -P passlist.txt ftp://192.168.1.100
```

9. Mode verbeux

Pour afficher plus de détails sur chaque tentative :

```
hydra -V -l user -P passlist.txt ssh://192.168.1.101
```

Fonctionnalités Avancées de THC Hydra

Hydra offre des fonctionnalités avancées pour affiner les attaques par force brute et les rendre plus efficaces ou plus discrètes.

Utilisation de Proxies

Hydra peut utiliser des serveurs proxy pour masquer l'adresse IP de l'attaquant ou pour contourner les restrictions réseau. Cela peut être utile pour les tests d'intrusion où la discrétion est importante.

- **Proxy HTTP** : `bash hydra -l user -P passlist.txt -s 8080 http-get://192.168.1.100/index.html -x http://proxy_ip:proxy_port` Ou en utilisant une variable d'environnement : `bash export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080 hydra -l user -P passlist.txt http-get://192.168.1.100/index.html`
- **Proxy SOCKS5** : `bash export HYDRA_PROXY=socks5://login:pass@proxy_ip:proxy_port hydra -l user -P passlist.txt ssh://192.168.1.101`

Vous pouvez également spécifier une liste de proxies dans un fichier texte :

```
export HYDRA_PROXY=proxylist.txt  
hydra -l user -P passlist.txt ssh://192.168.1.101
```

Modes d'Attaque Spécifiques

Hydra prend en charge divers modes d'attaque adaptés à des scénarios spécifiques :

- **Attaque par dictionnaire (Dictionary Attack)** : C'est le mode le plus courant, où Hydra utilise des listes de mots de passe (dictionnaires) pour tenter de se

connecter. C'est ce que nous avons vu dans la plupart des exemples précédents avec l'option `-P FILE`.

- **Attaque par force brute pure (Brute-Force Generation)** : Hydra peut générer des mots de passe en fonction de critères définis (longueur minimale/maximale, jeu de caractères). Cela est utile lorsque les dictionnaires ne sont pas suffisants ou pour des mots de passe plus complexes.
 - `-x MIN:MAX:CHARSET` : Génère des mots de passe de `MIN` à `MAX` caractères en utilisant le jeu de caractères `CHARSET`.
 - `a` : minuscules
 - `A` : majuscules
 - `1` : chiffres
 - `%` : caractères spéciaux
 - `^` : tous les caractères imprimables

Exemple : Générer des mots de passe de 4 à 6 caractères, composés de minuscules et de chiffres : `bash hydra -l admin -x 4:6:a1 ssh://192.168.1.101`

- **Attaque par masque (Mask Attack)** : Similaire à la force brute, mais permet de définir un masque pour les caractères inconnus. Par exemple, si vous savez qu'un mot de passe commence par

`pass` et se termine par un chiffre, vous pouvez utiliser un masque comme `pass?d.*?`
`l` : minuscule * `?u` : majuscule * `?d` : chiffre * `?s` : caractère spécial * `?a` : tous les caractères alphanumériques

Exemple : Attaquer un mot de passe qui commence par "admin" et se termine par 3 chiffres :

```
```bash
hydra -l user -x admin?d?d?d ssh://192.168.1.101
```
```

Support SSL/TLS

Hydra prend en charge les connexions sécurisées SSL/TLS pour les services qui les utilisent (par exemple, HTTPS, SMTPS, POP3S, IMAPS, etc.). L'option `-S` est utilisée pour forcer une connexion SSL.

Exemple : Attaquer un serveur POP3S :

```
hydra -L users.txt -P passwords.txt -S pop3s://192.168.1.105
```

Options de Performance et de Temporisation

Pour optimiser la vitesse de l'attaque ou pour éviter la détection, Hydra offre des options de performance et de temporisation :

- **-t TASKS** : Définit le nombre de tâches parallèles par cible. Augmenter cette valeur peut accélérer l'attaque, mais peut aussi rendre l'attaque plus bruyante et potentiellement bloquée par des systèmes de détection d'intrusion.
- **-w TIME** : Temps d'attente en secondes pour une réponse du serveur. Utile pour les connexions lentes ou instables.
- **-W TIME** : Temps d'attente en secondes entre les connexions par thread. Peut être utilisé pour ralentir l'attaque et la rendre moins détectable.

Exemple : Utiliser 32 tâches parallèles et attendre 5 secondes entre chaque connexion :

```
hydra -t 32 -W 5 -L users.txt -P passwords.txt ssh://  
192.168.1.101
```

Options de Sortie

Hydra permet de contrôler la manière dont les résultats sont affichés et enregistrés :

- **-o FILE** : Enregistre les paires login/mot de passe trouvées dans un fichier spécifié. C'est une bonne pratique pour conserver une trace des identifiants compromis.
- **-b FORMAT** : Spécifie le format de sortie pour le fichier **-o** . Les formats disponibles incluent **text** (par défaut), **json** , **jsonv1** .

Exemple : Enregistrer les résultats au format JSON :

```
hydra -L users.txt -P passwords.txt -o results.json -b json  
ssh://192.168.1.101
```

Mode Verbeux et Débogage

Pour obtenir plus d'informations pendant l'exécution de l'attaque, vous pouvez utiliser les options de verbosité :

- **-v** : Mode verbeux. Affiche plus de détails sur le processus de l'attaque.
- **-V** : Mode très verbeux. Affiche chaque tentative de login/mot de passe.
- **-d** : Mode débogage. Fournit des informations très détaillées, utiles pour le dépannage ou la compréhension approfondie du fonctionnement d'Hydra.

Exemple : Exécuter une attaque en mode très verbeux :

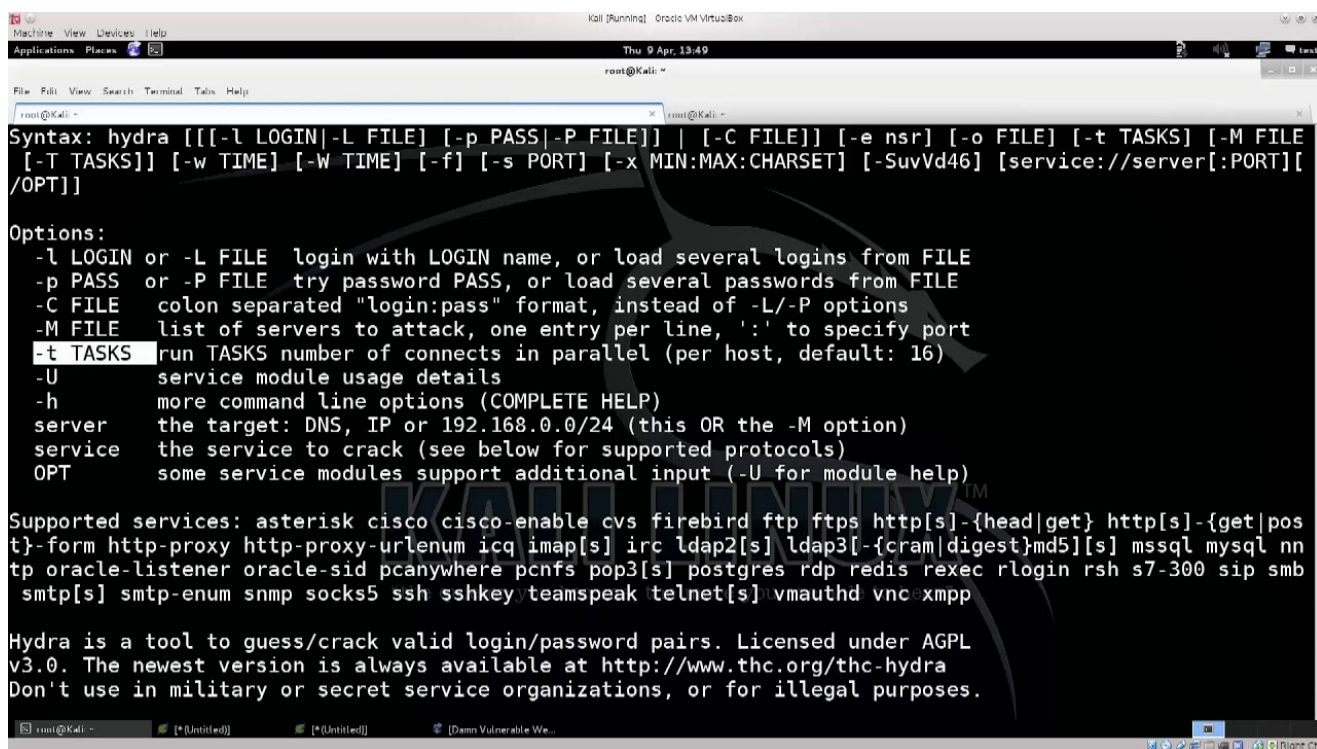
```
hydra -V -l user -P passlist.txt ftp://192.168.1.100
```

Conclusion

THC Hydra est un outil indispensable dans l'arsenal de tout professionnel de la cybersécurité. Sa capacité à effectuer des attaques par force brute et par dictionnaire sur une multitude de protocoles en fait un outil précieux pour tester la robustesse des mécanismes d'authentification. Ce manuel a couvert les aspects fondamentaux de Hydra, de son installation à ses fonctionnalités avancées, en passant par des exemples d'utilisation concrets. Il est crucial de se rappeler que Hydra est un outil puissant qui doit être utilisé de manière éthique et légale, uniquement sur des systèmes pour lesquels vous avez l'autorisation explicite de réaliser des tests. Une utilisation responsable garantit que cet outil reste un atout pour la sécurité, et non une menace.

Captures d'écran et Exemples Visuels

Bien que Hydra soit un outil en ligne de commande, il est utile de visualiser son fonctionnement à travers des captures d'écran. Ces images montrent des exemples de l'exécution de commandes Hydra et de leurs sorties.



```
Machine View Devices Help
Applications Places
Thu 9 Apr, 13:49
root@Kali ~
File Edit View Search Terminal Tabs Help
root@Kali ~
Syntax: hydra [[[ -l LOGIN | -L FILE ] [-p PASS | -P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE]
[-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvVd46] [service://server[:PORT]]
/OPT]]
Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-C FILE colon separated "login:pass" format, instead of -L/-P options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel (per host, default: 16)
-U service module usage details
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)

Supported services: asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get} http[s]-{get|pos
t}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest|md5}[s] mssql mysql nn
tp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres rdp redis rexec rlogin rsh s7-300 sip smb
smtp[s] smtp-enum snmp socks5 ssh sshkey teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.
```

Cette capture d'écran illustre la syntaxe générale de Hydra et certaines de ses options principales, telles qu'affichées dans un terminal Linux. Elle met en évidence la flexibilité de l'outil pour cibler différents services et utiliser diverses méthodes d'authentification.