Servicio FTP sobre Ubuntu Server 18.04

Práctica 03 – Servicios de Redes e Internet

Gonzalo Tudela Chavero

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ENUNCIADO

Se quiere habilitar un servidor FTP para los usuarios de la intranet de una organización.
Los requisitos impuestos sobre el servicio FTP son los siguientes:

- Se usará el paquete vsftpd para proporcionar el servicio.
- El usuario anonymous sólo podrá descargar ficheros, nunca crearlos, modificarlos, etc.
- Existirá un usuario local, Fernando, que podrá revisar el contenido de los directorios de los usuarios locales que usan el servicio de FTP. Del resto de los usuarios locales no.
- Los usuarios que usan el servicio FTP no podrán hacer uso del comando chmod para modificar sus permisos e impedir a Fernando no poder acceder a sus directorios FTP.
- Como es lógico, los usuarios locales que usen el servicio FTP no podrán salir de su "directorio home" ni ver ni modificar el contenido de los directorios personales de otros usuarios.
- Los usuarios locales que no usan el servicio de FTP tampoco podrán acceder al "directorio home" del resto de usuarios.

Los usuarios locales con permiso para poder usar el servicio FTP son:

username	*password*	*group*	*directorio home*
fernando	123456	ftp, admin	/home/ftp/fernando
magiones	qwerty	ftp	/home/ftp/magiones
alcasec	*7¡Vamos!	ftp	/home/ftp/alcasec

Los usuarios locales sin permiso para el uso del servicio ftp son:

username	*password*	*group*	*directorio home*
root	a6_123	root, admin	/root
lobo	ie168	lobo	/home/lobo

Requisitos impuestos por el jefe del proyecto sobre el servidor:

- El servidor FTP estará dentro de la red 10.0.2.0/24.
- El servidor FTP sólo aceptará conexiones desde la dirección IPv4 10.0.2.254/24.
- El servidor FTP no aceptará escaneos de tipo ACK para intentar detectar la existencia de un firewall. Como elemento de defensa perimetral se usará IPTables.
- Sólo podrán utilizar el comando ifconfig el usuario root y el usuario fernando. Además, estos usuarios serán los únicos que puedan realizar modificaciones sobre el fichero de configuración de las interfaces de red del servidor. El resto de los usuarios no podrán consultar la configuración lógica de red (comando ifconfig) ni el contenido del fichero de configuración de las interfaces de red. Tampoco podrá modificar el fichero.
- Los clientes del servicio FTP accederán al servidor a través de su FQDN: ftp.tnt.hack
- El administrador del sistema comprobará la fortaleza de las claves (herramienta John The Ripper) y para ello usará el diccionario "rockyou.txt".

ESQUEMA DE RED

Para la realización de la práctica utilizaremos un servidor con el sistema operativo Ubuntu Server 18.04 que tendrá los servicios de FTP y DNS, el equipo remoto desde el que simularemos las conexiones al FTP será una maquina Kali.

El esquema de red para la práctica será el siguiente:

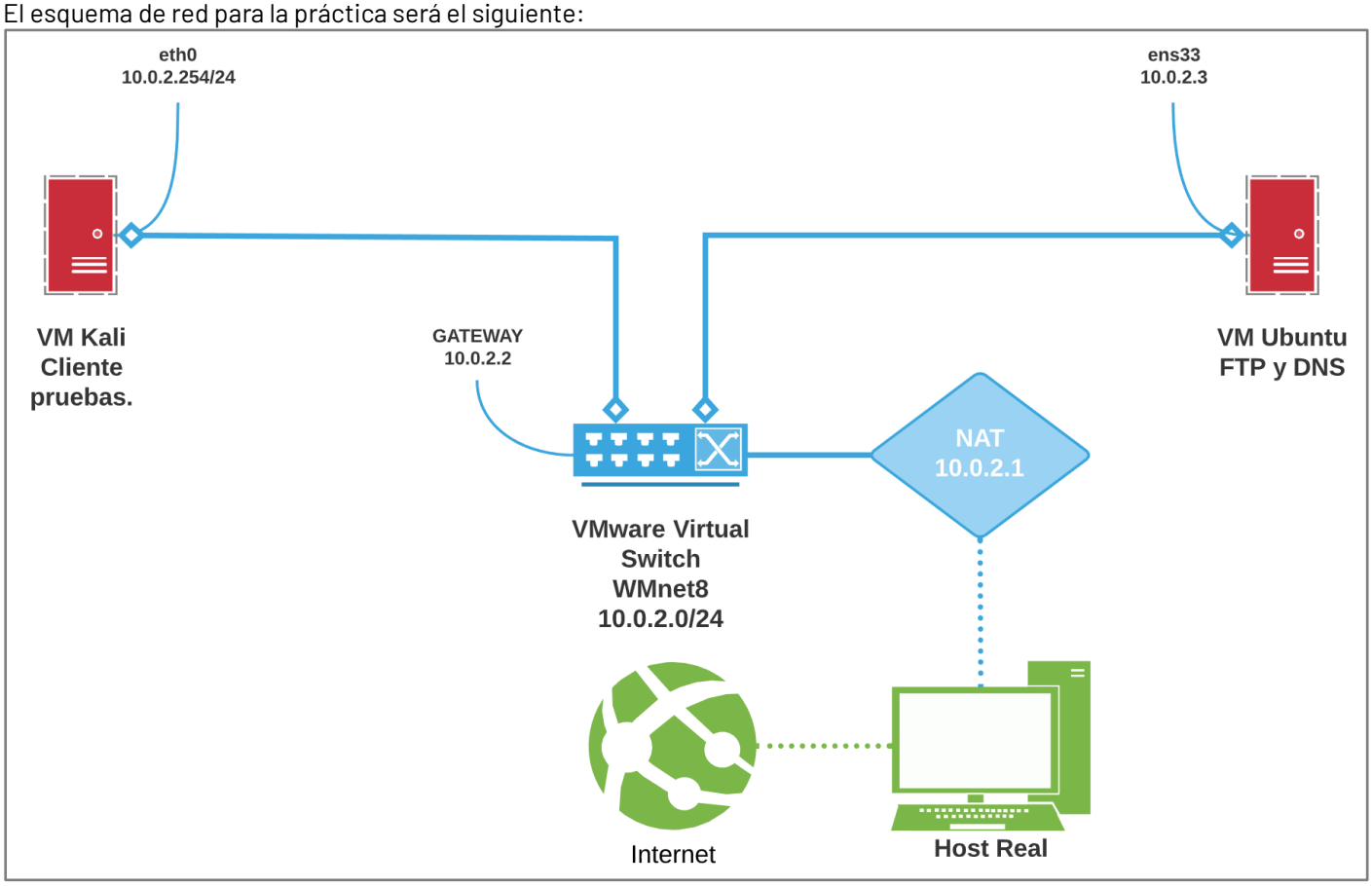

Ilustración 1 - Esquema de red para la práctica.

INSTALACION DE LOS SERVICIOS

Antes de instalar estos procedo a configurar la interfaz de red como corresponde al esquema de red anterior.

```
nano /etc/netplan/50-cloud-init.yaml
```

```
  GNU nano 2.9.3                        /etc/netplan/50-cloud-init.yaml

# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: false
      addresses: [10.0.2.3/24]
      gateway4: 10.0.2.2
      nameservers:
        addresses: [192.168.100.100,1.1.1.1]
```

Ilustración 2 - Configuración de la interfaz de red en el servidor.

Comprobación de conectividad:

```
ping www.google.es
```

```
root@gon-ubuntu-server:/home/gon# ping www.google.es
PING www.google.es (172.217.168.163) 56(84) bytes of data.
64 bytes from mad07s10-in-f3.1e100.net (172.217.168.163): icmp_seq=1 ttl=128 time=19.0 ms
64 bytes from mad07s10-in-f3.1e100.net (172.217.168.163): icmp_seq=2 ttl=128 time=23.0 ms
^C
--- www.google.es ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 19.080/21.041/23.003/1.966 ms
```

Ilustración 3 - La máquina tiene conectividad con internet.

Instalación del servicio vsftpd y bind9:

```
apt install vsftpd
apt install bind9
```

Comprobación del estado del paquete:

```
dpkg -s vsftpd
```

```
root@gon-ubuntu-server:/home/gon# dpkg -s vsftpd
Package: vsftpd
Status: install ok installed
Conflicts: ftp-server
Conffiles:
 /etc/ftpusers 839f3157aad792bafbbdcd932a95a345
 /etc/init.d/vsftpd 189fccd73e2600c50de8066582d58a39
 /etc/logrotate.d/vsftpd dac2cb7b9cfd8a03b4fa9ca3601a43a6
 /etc/pam.d/vsftpd e75200b7896d8b2c2f2590d0e3d4a6ef
 /etc/vsftpd.conf 0ed7ed3a33022af132b878c8c937bad9
Description: lightweight, efficient FTP server written for security
 This package provides the "Very Secure FTP Daemon", written from
 the ground up with security in mind.
```

Ilustración 4 - Estado del paquete, paquetes conflictivos con este y archivos de configuración.

```
dpkg -s bind9
```

```
Package: bind9
Status: install ok installed
Suggests: bind9-doc, dnsutils, resolvconf, ufw
Conffiles:
 /etc/apparmor.d/usr.sbin.named 2554ab429effae02d3ab07de11adf762
 /etc/bind/bind.keys 4c562437426d0569ce142a0f0e20f020
 /etc/bind/db.0 8aba258068c8c60a7ade3952a285f57d
 /etc/bind/db.127 64f5cf50e8d8192109dad43b779e5e36
```

Ilustración 5 - Estado de la instalación y algunos de sus archivos de configuración.

CREACIÓN Y PERMISOS DE USUARIOS LOCALES

Creamos los usuarios locales con los siguientes comandos:
Creamos el usuario Fernando indicando su directorio home y el grupo al que pertenecerá.

```
adduser fernando --home /home/ftp/fernando --ingroup ftp
```

Agregamos el grupo admin:

```
addgroup admin
```

Metemos a Fernando en el grupo admin:

```
usermod fernando -G admin
```

Añadimos al usuario magiones:

```
adduser magiones --home /home/ftp/magiones --ingroup ftp
```

Añadimos al usuario alcasec:

```
adduser alcasec --home /home/ftp/alcasec --ingroup ftp
```

Modificamos la contraseña de root:

```
passwd
```

Agregamos a root al grupo admin:

```
usermod root -G admin
```

Creamos el usuario lobo:

```
adduser lobo
```

Restringir el uso de chroot solo al propietario y al grupo propietario (root y root respectivamente):
Comprobamos que el propietario y grupo de chroot es root en ambos casos.

```
ls /bin/chmod -l
```

Eliminamos los permisos para otros en los siguientes archivos en /bin.

```
chmod o=--- chmod
chmod o=--- chown
chmod o=--- chgrp
```

Ahora nadie podrá utilizar el comando `chmod` salvo root y los usuarios en el grupo root.
Eliminamos los permisos para ejecutar (acceder) a las carpetas de los usuarios del servicio ftp:

```
chmod  o-x /home/ftp/* -v
```

El usuario lobo no es un usuario del grupo FTP así que probamos a ver si puede acceder a las carpetas home de los que si están en FTP.

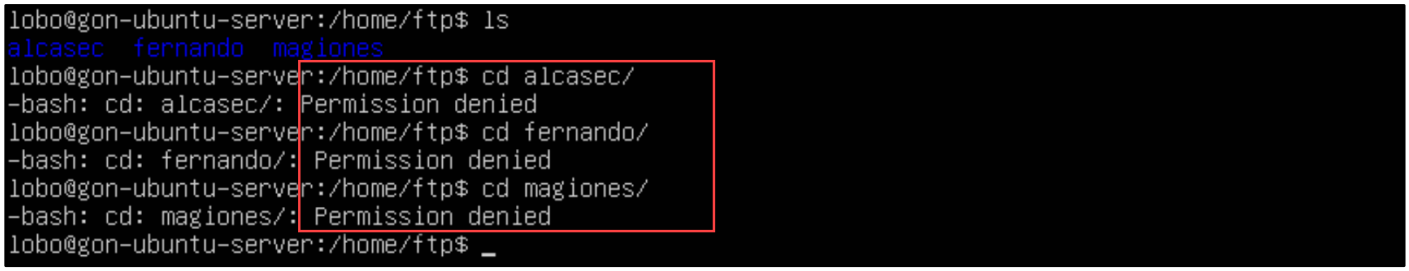

Ilustración 6 - Prueba de acceso de lobo a las carpetas de los usuarios del ftp.

Eliminamos el permiso de acceso para otros a la carpeta de usuario de *lobo* y *gon* (usuario de la instalación).

```
chmod -o-x /home/lobo/ -v
chmod -o-x /home/gon/ -v
chmod -o-x /home/ftp/ -v
```

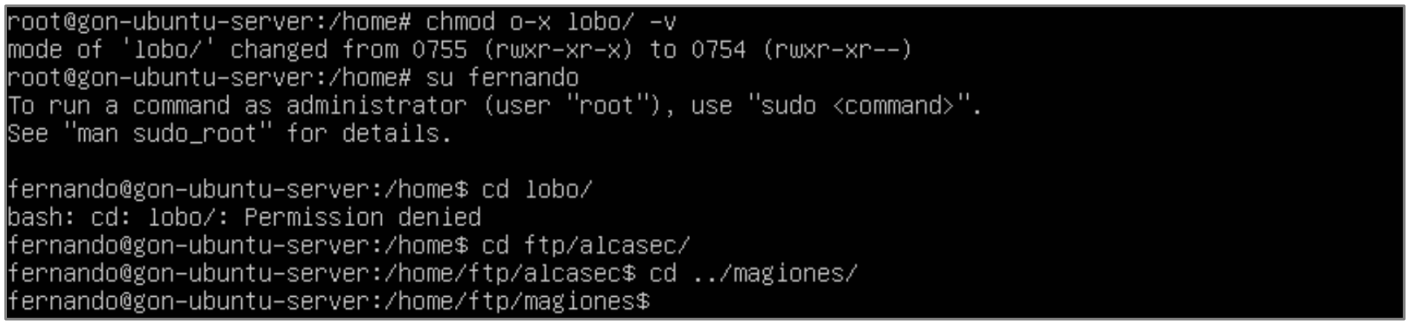

Ilustración 7 - Prueba de que fernando accede a las carpetas de los usuarios de FTP, pero no las del resto.

Establecemos el grupo para ifconfig en admin y eliminamos el permiso de ejecución para otros, lo que equivale a que solo root y el grupo admin pueden ejecutar este comando.

```
chgrp /sbin/ifconfig admin
chmod /sbin/ifconfig o-x -v
```

Modificamos los permisos de 50-init-cloud.yaml para que pertenezca al grupo admin y este tenga permisos de escritura.

```
chgrp admin /etc/netplan/50-init-cloud.yaml
chmod g+w /etc/netplan/50-init-cloud.yaml
```

CONFIGURACIÓN DEL SERVICIO VSFTPD

El contenido que hemos modificado de /etc/vsftpd.conf es el siguiente:

```
anonymous_enable=YES #permite login con ftp o anonymous
local_enable=YES #permite login de cuentas locales
write_enable=YES #permite operaciones de escritura del ftp
anon_upload_enable=NO #deniega upload de anonymous
anon_mkdir_write_enable=NO #deniega creacion de directorios y escritura de anon.
chroot_local_user=YES #enjaula los usuarios locales.
chroot_list_enable=YES #activa lista de usuarios no enjaulados
chroot_list_file=/etc/vsftpd.chroot_list #define archivo de usuarios enjaulados
userlist_enable=YES #activa lista de usuarios prohibidos
userlist_file=/etc/vsftpd.userlist #define archivo de usuarios prohibidos
```

El archivo vsftpd.chroot_list contendrá **fernando** para que este usuario no esté enjaulado, así como vsftpd.userlist contendrá **lobo** y **root** como usuarios no permitidos.
Crearemos un directorio que contendrá los directorios de usuarios que no son del sistema y le asignamos propietario y permisos:

```
mkdir /var/ftp/
chown ftp:ftp /var/ftp/
chmod 755 /var/ftp/
```

Crearemos un directorio dentro de /var/ftp/ desde donde el usuario anonymous / ftp pueda descargar archivos.

```
mkdir /var/ftp/anon/
chmod 775 /var/ftp/anon/
```

Los permisos de las carpetas de usuario se establecieron como se puede ver en la siguiente imagen:

```
root@gon-ubuntu-server:/home# ls -l
total 12
drwxr-xr-- 5 ftp  ftp  4096 Feb  7 19:57 ftp
drwxr-xr-- 5 gon  gon  4096 Feb  7 19:48 gon
drwxr-xr-- 4 lobo lobo 4096 Feb  7 21:06 lobo
root@gon-ubuntu-server:/home#
```

Ilustración 8 - Permisos y propietarios de las carpetas de usuario del sistema.

Los permisos de los directorios de los usuarios del servicio FTP:

```
root@gon-ubuntu-server:/home/ftp# ls -l
total 12
drwxr-x--- 2 alcasec  ftp 4096 Feb  7 19:57 alcasec
drwxr-x--- 4 fernando ftp 4096 Feb  8 12:11 fernando
drwxr-x--- 2 magiones ftp 4096 Feb  7 19:56 magiones
root@gon-ubuntu-server:/home/ftp# _
```

Ilustración 9 - Permisos y propietarios de los directorios de los usuarios del FTP.

IPTABLES

Como queremos que nuestras reglas sean permanentes en algún momento instalaremos el siguiente paquete:

```
apt install iptables-persistent
```

Agregamos una **chain* (cadena)** de reglas en la tabla INPUT para aceptar conexiones al puerto 21 local con origen en 10.0.2.254, tras esta regla añadiremos a la cadena de comprobaciones una regla que rechace conexiones al puerto 21 local desde cualquier origen, por lo que *si un paquete entra en la primera regla de la cadena se aceptará y si no cumple esta se comprobará la siguiente regla de la cadena en donde será rechazado.

```
iptables -A INPUT -s 10.0.3.254 -d 10.0.2.3 -p tcp --dport 21 -j ACCEPT
iptables -A INPUT -d 10.0.2.3 -p tcp --dport 21 -j DROP
```

Hacemos estos cambios permanentes:

```
iptables-save > /etc/iptables/rules.v4
```

Comprobamos su funcionamiento cambiando la IP de nuestro cliente por la .253 e intentando conectar al servidor.

```
root@lightsaber: ~
root@lightsaber:~# ifconfig eth0 10.0.2.253 netmask 255.255.255.0
root@lightsaber:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.253  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::20c:29ff:febb:75a1  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:bb:75:a1  txqueuelen 1000  (Ethernet)
        RX packets 30  bytes 2600 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 3287 (3.2 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 6  bytes 318 (318.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 318 (318.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@lightsaber:~# ftp 10.0.2.3
ftp: connect: Connection timed out
ftp>
```

Ilustración 10 - Comprobación de conexión al FTP desde otra IP.

Configuramos nuestro cliente con la IP aceptada y reintentamos:

```
root@lightsaber: ~
root@lightsaber:~# ifconfig eth0 10.0.2.254 netmask 255.255.255.0
root@lightsaber:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.254  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::20c:29ff:febb:75a1  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:bb:75:a1  txqueuelen 1000  (Ethernet)
        RX packets 34  bytes 2906 (2.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 50  bytes 3643 (3.5 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 6  bytes 318 (318.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 318 (318.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@lightsaber:~# ftp 10.0.2.3
Connected to 10.0.2.3.
220 Bienvenido a ECORP FTP.
Name (10.0.2.3:root): ftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Ilustración 11 - Comprobación de conexión desde la IP correcta.

Configuramos otra cadena en la tabla input para que se descarten todos los paquetes TCP con flag ACK que inicien una conexión nueva, (usamos -I para introducir esta regla en el inicio en vez de -A para añadir al final).

```
iptables -I INPUT -d 10.0.2.3 -p tcp --tcp-flags ALL ACK -m state --state NEW -j DROP
```

```
root@lightsaber:~# hping3 10.0.2.3 -S -p 21
HPING 10.0.2.3 (eth0 10.0.2.3): S set, 40 headers + 0 data bytes
len=46 ip=10.0.2.3 ttl=64 DF id=0 sport=21 flags=SA seq=0 win=29200 rtt=7.9 ms
len=46 ip=10.0.2.3 ttl=64 DF id=0 sport=21 flags=SA seq=1 win=29200 rtt=7.0 ms
len=46 ip=10.0.2.3 ttl=64 DF id=0 sport=21 flags=SA seq=2 win=29200 rtt=7.0 ms
^C
--- 10.0.2.3 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 7.0/7.3/7.9 ms
root@lightsaber:~# hping3 10.0.2.3 -A -p 21
HPING 10.0.2.3 (eth0 10.0.2.3): A set, 40 headers + 0 data bytes
^C
--- 10.0.2.3 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@lightsaber:~#
```

Ilustración 12 - Prueba de rechazo de los paquetes TCP con flag ACK.

BIND

Realizamos la configuración de BIND9 en Ubuntu Server editando los archivos en /etc/bind/

Contenido de named.conf.options: (opciones generales)

```
  GNU nano 2.9.3                          named.conf.options

options {
        directory "/var/cache/bind";
        listen-on port 53 { 10.0.2.3; };
        listen-on port 53 { 127.0.0.1; };
        allow-query { 10.0.2.0/24; };
        allow-query-cache { 10.0.2.0/24; };
        recursion yes;
        forwarders {
                10.0.2.2;
                8.8.8.8;
        };
};
```

Ilustración 13 - Archivo configuración DNS BIND9.

Contenido de named.conf.local: (zonas de búsqueda).

```
  GNU nano 2.9.3                            named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
// include "/etc/bind/zones.rfc1918";
// Zona TNT
zone "tnt.hack"{
        type master;
        file "/etc/bind/db.tnt.hack";
        allow-transfer {none;};
};
// Zona TNT inversa
zone "2.0.10.in-addr.arpa" {
        type master;
        file "/etc/bind/2.0.10.db";
        allow-transfer {none;};
};
```

Ilustración 14 - Configuración de las zonas de búsqueda en BIND.

Contenido del archivo de búsqueda inversa, 2.0.10.db:

```
  GNU nano 2.9.3                              2.0.10.db

$TTL 1D
@ IN SOA ns.tnt.hack. gondbt@gmail.com. (
08022020 ;serial como fecha
1D
1H
1W
3H
)
@ IN NS ns.tnt.hack.
1 IN PTR ns.tnt.hack.
2 IN PTR ftp.tnt.hack.
```

Ilustración 15 - Configuración de los registros de búsqueda inversa.

Contenido del archivo de búsqueda directa, db.tnt.hack:

```
  GNU nano 2.9.3                          db.tnt.hack

$TTL 1D
@ IN SOA ns.tnt.hack. gondbt@gmail.com. (
08022020
1D
1H
1W
3H
)
@ IN NS ns.tnt.hack.
@ IN A 10.0.2.3
ns IN A 10.0.2.3
ftp IN A 10.0.2.3
```

Ilustración 16 - Configuración de los registros de búsqueda directa.

Nos aseguramos de que no existe ningún problema de acceso a los archivos de los registros cambiando el propietario con:

```
chown bind:bind 2.0.10.db
chown bind:bind db.tnt.hack
```

Comprobación de que el cliente es capaz de conectar mediante FQDN del servidor FTP:

```
root@lightsaber:~# ping ftp.tnt.hack
PING ftp.tnt.hack (10.0.2.3) 56(84) bytes of data.
64 bytes from 10.0.2.3 (10.0.2.3): icmp_seq=1 ttl=64 time=0.113 ms
64 bytes from 10.0.2.3 (10.0.2.3): icmp_seq=2 ttl=64 time=0.192 ms
^C
--- ftp.tnt.hack ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1016ms
rtt min/avg/max/mdev = 0.113/0.152/0.192/0.039 ms
root@lightsaber:~# ftp
ftp> open ftp.tnt.hack
Connected to ftp.tnt.hack.
220 Bienvenido a ECORP FTP.
Name (ftp.tnt.hack:root): fernando
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Ilustración 17 - Prueba de funcionamiento del FQDN.

JOHN THE RIPPER

Para la instalación de esta herramienta en nuestro Ubuntu Server utilizaremos:

```
apt install john
```

Utilizamos el propio servicio FTP para enviar el archivo rockyou.txt al servidor:

```
root@lightsaber:~# ftp
ftp> open ftp.tnt.hack
Connected to ftp.tnt.hack.
220 Bienvenido a ECORP FTP.
Name (ftp.tnt.hack:root): fernando
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put /usr/share/wordlists/rockyou.txt ./rockyou.txt
local: /usr/share/wordlists/rockyou.txt remote: ./rockyou.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
139921507 bytes sent in 0.54 secs (245.3406 MB/s)
ftp>
```

Ilustración 18 - Envío de rockyou.txt mediante el servicio FTP.

Combinamos los archivos passwd y shadow para que john pueda utilizarlos:

```
unshadow /etc/passwd /etc/shadow > /root/unshadow.txt
```

Lanzamos john utilizando el diccionario rockyou y el archivo anterior:

```
john --wordlist=rockyou.txt --users=fernando,magiones,alcasec,lobo,root unshadow.txt
```

```
root@gon-ubuntu-server:~# john --wordlist=rockyou.txt fusion.txt
Loaded 6 password hashes with 6 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (fernando)
qwerty           (magiones)
```

Ilustración 19 - Usuarios con contraseñas débiles.

La comprobación de las contraseñas es un proceso largo, en nuestro caso tras llevar todo esto a una maquina no virtualizada y repetir el proceso el cálculo la estadística señalo 12 horas para comprobar los usuarios que se habían indicado.

En el caso de *alcasec* la contraseña tiene 9 caracteres lo que tomara mas tiempo que las de *root* o *lobo*, con 6 y 5 caracteres que son mucho más rápidas de obtener incluso por fuerza bruta.

HASHCAT

Extraemos el archivo shadow del servidor copiándolo a la carpeta personal de fernando, descargándolo mediante ftp y una vez en la maquina Kali lo copiamos directamente al escritorio del anfitrión donde tenemos una carpeta con una copia de hashcat en la que ejecutaremos lo siguiente:

```
.\hashcat64.exe -O -m 1800 -a 0 -o contra.txt --remove shadow.txt rockyou.txt
```

-O reduce el tamaño de salt y otras opciones para acelerar el proceso
-m 1800 indica el tipo de algoritmo de hash utilizado para resumir las contraseñas.
-o es la salida a un archivo del resultado.
-- remove elimina del archivo con los hashes los que ya han sido resueltos.

En este caso el tiempo estimado es de 6 minutos utilizando un i74770K a 4,8Ghz y una Nvidia GTX 1070.

```
Session..........: hashcat
Status...........: Running
Hash.Type........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: shadow.txt
Time.Started.....: Sun Feb 09 10:42:32 2020 (5 mins, 29 secs)
Time.Estimated...: Sun Feb 09 10:49:55 2020 (1 min, 54 secs)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    96492 H/s (8.01ms) @ Accel:128 Loops:64 Thr:32 Vec:1
Recovered........: 0/3 (0.00%) Digests, 0/3 (0.00%) Salts
Progress.........: 32018676/43033155 (74.40%)
Rejected.........: 315636/32018676 (0.99%)
Restore.Point....: 10672349/14344385 (74.40%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:640-704
Candidates.#1....: RocKsTar -> Pagent956
Hardware.Mon.#1..: Temp: 71c Fan: 70% Util: 99% Core:1860MHz Mem:3802MHz Bus:8

[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit =>
```

Ilustración 20 - Prueba de fortaleza con hashcat (CPU + GPU).

Los hashes para los que se encontró solución fueron:
6B/CCFKot$6KzJx9tGTCoV7o/VRuM6..Y0wWZPx89U28wdYQaNvdAJFJtIwMI5xR36rIjD0FtVPQzUHqE8TOQfMKD5NUpfL0:qwerty
$6$6UbqoVqt$bleaoyJJJ2/.zqKhmh6jSIIA4.IiGJkSFCH4sAiS3thk8t8PqwWxnCzOdRrFT2s1mHzX8FIyI5pRS/ZFKiAhg.:123456

El resto de los hashes no correspondían con ninguna solución del diccionario, por lo que en el archivo shadow.txt quedan 3 hashes sin resolver (la opción --remove eliminó los resueltos).

```
$6$4IaypwUW$Ex4I1ab/hM69OzqLTWAvLfMsWy1A0408003H1Yr8c0DnsYWWsDxc9crC/ewgTbRcmE5WyT5FI6eOTUt24pr7Z.
$6$sOTmCuH0$avO1EljhXeGsmLPVeh81LKPI4mkILxl0VgyBoUy4li0WFWPrDb8VW99vWSSo9Zlj1xKQeB/SX.o704mtkLCQL.
$6$K.IwhTde$VBVPIgJK9Mv.E4GzNsquf1D6.5IBbUI5mdvqNDJoAvVJVezdGeYvHM3cyZpSKoQMSyLODaSsjg0VmZRzKktFt/
```

Estos corresponden con LOBO, ROOT y ALCASEC respectivamente, atacaremos por fuerza bruta el de lobo (el más débil) probando con reglas de mas sencillas a más complejas, para ahorrar tiempo sabemos que la password de LOBO tiene 5 caracteres con letras (mayúsculas, minúsculas) y números.

```
hashcat64 -O -a3 -m 1800
$6$sOTmCuH0$avO1EljhXeGsmLPVeh81LKPI4mkILxl0VgyBoUy4li0WFWPrDb8VW99vWSSo9Zlj1xKQeB/SX.o704mtkLCQL. -1 ?l?d?u ?1?1?1?1?1
```

la opción interesante aquí es -1 que indica un charset mixto compuesto por loss grupos predefinidos L,D y U que son letras minúsculas, mayúsculas y números, tras esto especificamos la máscara que son 5 caracteres del tipo 1.

El tiempo máximo para todas las combinaciones será de 2 horas, pero en 2 minutos 30 segundos encontró la solución.

```
Session..........: hashcat
Status...........: Running
Hash.Type........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$sOTmCuH0$avO1EljhXeGsmLPVeh81LKPI4mkILxlOVgyBoUy...kLCQL.
Time.Started.....: Sun Feb 09 11:48:49 2020 (1 min, 46 secs)
Time.Estimated...: Sun Feb 09 14:07:12 2020 (2 hours, 16 mins)
Guess.Mask.......: ?1?1?1?1?1 [5]
Guess.Charset....: -1 ?l?d?u, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   110.3 kH/s (6.88ms) @ Accel:128 Loops:64 Thr:32 Vec:1
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 11673600/916132832 (1.27%)
Rejected.........: 0/11673600 (0.00%)
Restore.Point....: 184320/14776336 (1.25%)
Restore.Sub.#1...: Salt:0 Amplifier:4-5 Iteration:1728-1792
Candidates.#1....: bGpRI -> bWuna
Hardware.Mon.#1..: Temp: 71c Fan: 68% Util: 99% Core:1860MHz Mem:3802MHz Bus:8

$6$sOTmCuH0$avO1EljhXeGsmLPVeh81LKPI4mkILxlOVgyBoUy4liOWFWPrDb8VW99vWSSo9Zlj1xKQeB/SX.o704mtkLCQL.:ie168

Session..........: hashcat
Status...........: Cracked
Hash.Type........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$sOTmCuH0$avO1EljhXeGsmLPVeh81LKPI4mkILxlOVgyBoUy...kLCQL.
Time.Started.....: Sun Feb 09 11:48:49 2020 (2 mins, 30 secs)
Time.Estimated...: Sun Feb 09 11:51:19 2020 (0 secs)
Guess.Mask.......: ?1?1?1?1?1 [5]
Guess.Charset....: -1 ?l?d?u, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   110.4 kH/s (7.02ms) @ Accel:128 Loops:64 Thr:32 Vec:1
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 16465920/916132832 (1.80%)
Rejected.........: 0/16465920 (0.00%)
Restore.Point....: 245760/14776336 (1.66%)
Restore.Sub.#1...: Salt:0 Amplifier:19-20 Iteration:4992-5000
Candidates.#1....: iJtel -> iDpO1
Hardware.Mon.#1..: Temp: 72c Fan: 70% Util: 99% Core:1860MHz Mem:3802MHz Bus:8

Started: Sun Feb 09 11:48:44 2020
Stopped: Sun Feb 09 11:51:20 2020

D:\Desktop\hashcat-5.1.0>
```

Ilustración 21 - Resultado hashcat sobre el hash de la contraseña del usuario LOBO.

La conclusión es que de la misma forma que la dificultad para solucionar un hash por fuerza bruta crece exponencialmente esta también se reduce exponencialmente con pistas que se pueden obtener de formas mas mundanas como escuchar el numero de golpes en el teclado que dio el usuario, o observando que grupos de teclas se presionaron, así como cualquier técnica de hacking social que permita darnos pistas sobre la posible contraseña que nos permitiría generar una serie de reglas que reducirían tiempos inabarcables a tan solo días o minutos.