

## Tema 3.Tarea 2 Disuasión y Honeypots

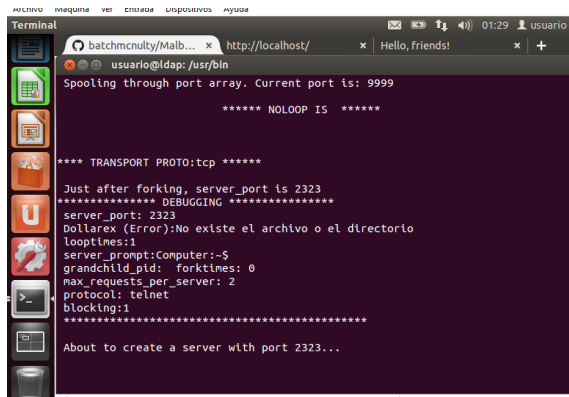
### Índice de Contenidos

1. Disuasión y Honeypots
  2. Instalación de Malbait desde GitHub
  3. Ejecución de Malbait
- Los mecanismos más habituales de disuasión son los Honeypots, esto es, los Tarros de Miel.

- Se trata de herramientas que permiten simular de forma atractiva el comportamiento de entornos u objetivos reales, de forma que el atacante se dirija hacia ellas en primer lugar, en vez de atacar los objetivos valiosos, que se ocultarán en un segundo plano.
- Por lo general, los honeypots se disponen en un primer plano y se les dota de un acceso aparentemente fácil, para que los posibles atacantes caigan en la trampa sin problemas.
- En cualquier caso, cuando el atacante se da cuenta del engaño, por lo general continúa con el ataque hacia otros objetivos de la red, no obstante, ya ha perdido el factor sorpresa, pues al haberse detectado su primer ataque en el honeypot, se levantan las correspondientes alarmas en el SOC y se disparan inmediatamente los refuerzos del blindaje de las máquinas importantes, que están siempre en un segundo plano.

## Disuasión y Honeypots

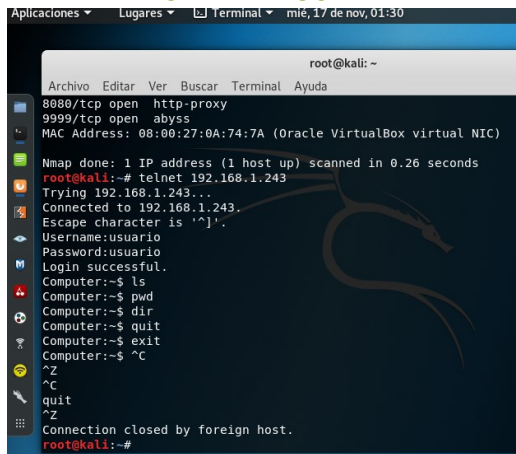
- Así pues, el primer consejo es disponer los honeypots siempre en una máquina separada de los objetivos principales, para poder organizar esta estrategia de defensa con dos planos, aprovechando esta valiosa demora entre el primer y el segundo ataque.



```
Terminal
batchmculy/Mal... x http://localhost/ x Hello, friends! x +
usuario@ldap: /usr/bin
Spooling through port array. Current port is: 9999
***** NOLOOP IS *****
**** TRANSPORT PROTO:tcp ****
Just after forking, server_port is 2323
***** DEBUGGING *****
server_port: 2323
Dollarx (Error):No existe el archivo o el directorio
looptimes:1
server_prompt:Computer:~$
grandchild_pid: forktimes: 0
max_requests_per_server: 2
protocol: telnet
blockings:1
*****
About to create a server with port 2323...
```

Ubuntu 14 metasploitable 3 : Honeypot

## Disuasión y Honeypots



```
Aplicaciones ▾ Lugares ▾ Terminal ▾ mié, 17 de nov, 01:30
root@kali: ~
Archivo Editar Ver Buscar Terminal Ayuda
8080/tcp open  http-proxy
9999/tcp open  abyss
MAC Address: 08:00:27:0A:74:7A (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
root@kali:~# telnet 192.168.1.243
Trying 192.168.1.243...
Connected to 192.168.1.243.
Escape character is '^]'.
Username:usuario
Password:usuario
Login successful.
Computer:~$ ls
Computer:~$ pwd
Computer:~$ dir
Computer:~$ quit
Computer:~$ exit
Computer:~$ ^C
^Z
quit
^Z
Connection closed by foreign host.
root@kali:~#
```

Kali 2019: Atacante

Ambas máquinas tienen que estar en la misma red interna inet1 dentro de la subred 192.169.1.0/24

## Disuasión y Honeypots

- Existen multitud de herramientas de código abierto para implementar honeypots de toda índole, desde los más simples hasta los más complejos.
- Los más sencillos permiten simular un entorno rico en potenciales objetivos, que se implementan mediante una ligera capa de funcionalidad muy realista. Su misión es hacer que los objetivos reales se confundan con los falsos y que el atacante pierda la mayor

cantidad de tiempo posible. Son muy útiles frente a los ataques con Fuerza Bruta que, por lo general, sólo buscan averiguar claves y entrar en los entornos para luego vender las credenciales obtenidas, pero después no se entretienen trabajando dentro de las sesiones que consiguen abrir.

- Los más complejos son auténticos Entornos de Desarrollo de máquinas virtuales, pues permiten simular completamente un host, con su sistema de ficheros real, y son editables, manipulables e incluso ejecutables.

- Ambos niveles de implementación tienen un detalle en común: toda la actividad del ataque se graba en detalle, para poder analizarla detenidamente a posteriori y trazar al atacante.
- En nuestra práctica utilizaremos un honeypot del primer tipo, esto es, de funcionalidad simple pero efectiva, que requiere poca configuración y brinda soluciones rápidas y muy operativas.
- Dentro de GitHub se pueden encontrar decenas de honeypots de código abierto de la más diversa



índole, y pertenecientes a ambos tipos, esto es, de funcionalidad simple y compleja.

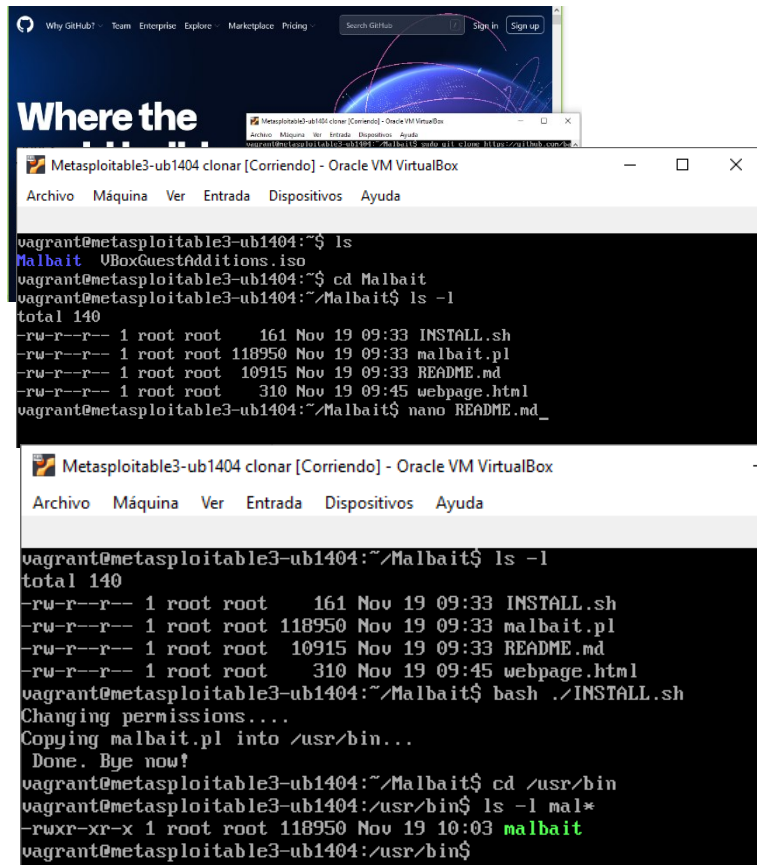
- Los honeypots complejos son tanto más difíciles de configurar cuanto mejor se quiera que simulen un servidor real, por lo que su parametrización suele ser un proceso largo, complicado y además constante, pues un honeypot complejo pero estático se detecta muy fácilmente por las trazas evidentes de inactividad (fechas antiguas de los ficheros y los logs, etc.).

- Para esta práctica utilizaremos el honeypot Malbait, que se utiliza mucho tal cual está programado y también como base de construcción de otros honeypots, mediante la modificación de su código abierto.

## Instalación de Malbait desde GitHub

- En primer lugar, localizaremos el código de Malbait en GitHub, para clonarlo en nuestro servidor Linux.
- Una vez clonado Malbait, entraremos en el directorio homónimo, localizaremos el fichero README y seguiremos las instrucciones detalladas en el mismo para su correcta instalación, poniendo especial atención en los requisitos previos.
- Por ejemplo, al tratarse de una aplicación desarrollada en PERL, deberemos comprobar en primer lugar que dicho lenguaje de programación está correctamente instalado y operativo.
- Hecho esto, ejecutaremos el script de instalación INSTALL.sh mediante una bash shell.

Nota.- Comprobar que se ha ejecutado correctamente el comando `chmod +777` incluido en el fichero `INSTALL.sh`. Esto se sabe porque el mensaje que aparece es el mismo que la captura.



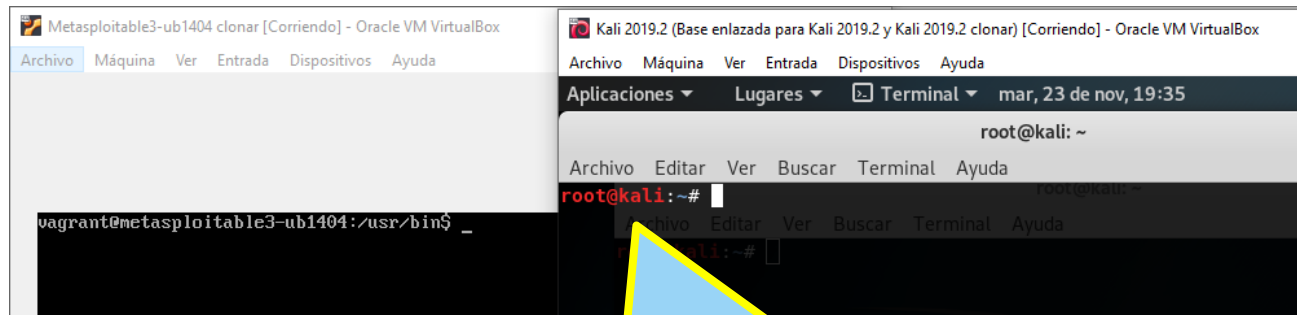
```
Where the
Metasploitable3-ub1404 clonar [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

vagrant@metasploitable3-ub1404:~$ ls
Malbait VBoxGuestAdditions.iso
vagrant@metasploitable3-ub1404:~$ cd Malbait
vagrant@metasploitable3-ub1404:~/Malbait$ ls -l
total 140
-rw-r--r-- 1 root root 161 Nov 19 09:33 INSTALL.sh
-rw-r--r-- 1 root root 118950 Nov 19 09:33 malbait.pl
-rw-r--r-- 1 root root 10915 Nov 19 09:33 README.md
-rw-r--r-- 1 root root 310 Nov 19 09:45 webpage.html
vagrant@metasploitable3-ub1404:~/Malbait$ nano README.md

Metasploitable3-ub1404 clonar [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

vagrant@metasploitable3-ub1404:~/Malbait$ ls -l
total 140
-rw-r--r-- 1 root root 161 Nov 19 09:33 INSTALL.sh
-rw-r--r-- 1 root root 118950 Nov 19 09:33 malbait.pl
-rw-r--r-- 1 root root 10915 Nov 19 09:33 README.md
-rw-r--r-- 1 root root 310 Nov 19 09:45 webpage.html
vagrant@metasploitable3-ub1404:~/Malbait$ bash ./INSTALL.sh
Changing permissions....
Copying malbait.pl into /usr/bin...
Done. Bye now!
vagrant@metasploitable3-ub1404:~/Malbait$ cd /usr/bin
vagrant@metasploitable3-ub1404:/usr/bin$ ls -l mal*
-rwxr-xr-x 1 root root 118950 Nov 19 10:03 malbait
vagrant@metasploitable3-ub1404:/usr/bin$
```

# Entorno de pruebas listo



- A continuación, prepararemos el entorno de pruebas, que estará compuesto por una máquina atacante basada en Kali Linux (VB Linux 2019), que es lo habitual en el pentesting o en el hacking, y por nuestra máquina atacada que estará basada en la distribución Ubuntu (Metasploitable 3). Simularemos esta última con Ubuntu 14-04.

## Mapeo de puertos

```
Nmap done: 1 IP address (1 host up) scanned in 4.64 seconds
root@kali:~# nmap 192.168.1.10
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-23 19:53 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.10
Host is up (0.00016s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
389/tcp    open  ldap
443/tcp    open  https
5037/tcp   open  pptp
MAC Address: 08:00:27:0A:74:7A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
root@kali:~#
```

- En primer lugar, lo habitual es lanzar un escaneo y mapeo de puertos contra la dirección IP de la máquina objetivo, para saber qué puertos están a la escucha, y con qué protocolos.
- Como se puede ver, sólo hay tres puertos abiertos con los protocolos (ssh), http (web), ldap, https y pptp.

usuario@ldap: /usr/bin



Inicio

```
usuario@ldap:/usr/bin$ sudo malbait -defaults
```

## Primera prueba

- Efectuamos una primera prueba, lanzando Malbait con privilegios de root y utilizando sus opciones por defecto.
- En este escenario, Malbait elige un grupo de puertos y protocolos por su cuenta y lanza un nutrido grupo de procesos que simulan un conjunto de aplicaciones operativas.
- Esta opción se suele utilizar para pruebas o cuando se desea simular con rapidez un entorno complejo si se ha detectado un ataque inminente o que ya está produciéndose.

# Procesos en diferentes puertos que simulan estar a la escucha

Archivo Máquina Ver Entrada Dispositivos Ayuda

usuario@ldap: /usr/bin

```
usuario@ldap: /usr/bin$ sudo netstat -an|grep perl
[sudo] password for usuario:
tcp        0      0 0.0.0.0:69          0.0.0.0:*          ESCUCHAR   2581/perl
tcp        0      0 0.0.0.0:7           0.0.0.0:*          ESCUCHAR   2606/perl
tcp        0      0 0.0.0.0:2345        0.0.0.0:*          ESCUCHAR   2590/perl
tcp        0      0 0.0.0.0:11          0.0.0.0:*          ESCUCHAR   2605/perl
tcp        0      0 0.0.0.0:587         0.0.0.0:*          ESCUCHAR   2594/perl
tcp        0      0 0.0.0.0:107         0.0.0.0:*          ESCUCHAR   2585/perl
tcp        0      0 0.0.0.0:13          0.0.0.0:*          ESCUCHAR   2604/perl
tcp        0      0 0.0.0.0:110         0.0.0.0:*          ESCUCHAR   2600/perl
tcp        0      0 0.0.0.0:9999        0.0.0.0:*          ESCUCHAR   2591/perl
tcp        0      0 0.0.0.0:143         0.0.0.0:*          ESCUCHAR   2577/perl
tcp        0      0 0.0.0.0:8080        0.0.0.0:*          ESCUCHAR   2598/perl
tcp        0      0 0.0.0.0:81          0.0.0.0:*          ESCUCHAR   2599/perl
tcp        0      0 0.0.0.0:465         0.0.0.0:*          ESCUCHAR   2593/perl
tcp        0      0 0.0.0.0:179         0.0.0.0:*          ESCUCHAR   2603/perl
tcp        0      0 0.0.0.0:2323        0.0.0.0:*          ESCUCHAR   2589/perl
tcp        0      0 0.0.0.0:115         0.0.0.0:*          ESCUCHAR   2582/perl
tcp        0      0 0.0.0.0:20          0.0.0.0:*          ESCUCHAR   2579/perl
tcp        0      0 0.0.0.0:21          0.0.0.0:*          ESCUCHAR   2580/perl
tcp        0      0 0.0.0.0:23          0.0.0.0:*          ESCUCHAR   2584/perl
tcp        0      0 0.0.0.0:152         0.0.0.0:*          ESCUCHAR   2583/perl
tcp        0      0 0.0.0.0:25          0.0.0.0:*          ESCUCHAR   2592/perl
tcp        0      0 0.0.0.0:7547        0.0.0.0:*          ESCUCHAR   2602/perl
tcp        0      0 0.0.0.0:123         0.0.0.0:*          ESCUCHAR   2586/perl
tcp        0      0 0.0.0.0:2525        0.0.0.0:*          ESCUCHAR   2595/perl
tcp        0      0 0.0.0.0:2526        0.0.0.0:*          ESCUCHAR   2596/perl
tcp        0      0 0.0.0.0:992         0.0.0.0:*          ESCUCHAR   2587/perl
tcp        0      0 0.0.0.0:993         0.0.0.0:*          ESCUCHAR   2578/perl
tcp        0      0 0.0.0.0:995         0.0.0.0:*          ESCUCHAR   2601/perl
tcp        0      0 0.0.0.0:12323       0.0.0.0:*          ESCUCHAR   2588/perl
usuario@ldap: /usr/bin$
```

- Como se puede ver, Malbait lanza de una vez decenas de procesos en puertos diferentes, todos ellos aparentemente a la escucha.
- Esta situación puede dar al posible hacker una primera impresión falsa de que la máquina merece la pena por tratarse de un servidor que debe de estar soportando muchas aplicaciones críticas.

# El Hacker inicia el scaneo :

ubuntu32 squid3 malbait [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

About to create a server with port 11...

Creating TCP server.....

Listening on port 11 using tcp protocol \$!:\*\*Des

\*\*\*\*

Just after forking, server\_port is 1

- El hacker inicia el escaneo/mapeo de la máquina objetivo con `nmap` desde su máquina `kali`, descubriendo que hay muchos puertos a la escucha, con muchos protocolos diferentes, lo cual confirma sus sospechas de que se trata de un servidor central.
- Por su parte, en la máquina atacada, `Malbait` ya comienza a registrar la actividad del ataque en sus logs.

Creating TCP server.....

Kali 2019.2 (Base enlazada para Kali 2019.2 y Kali 2019.2 clonar) [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Aplicaciones ▾ Lugares ▾ Terminal ▾ mar, 23 de nov, 20:15

root@kali: ~

Archivo Editar Ver Buscar Terminal Ayuda

Try using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 192.168.1.10

Host is up (0.00042s latency).

Not shown: 976 closed ports

STATE SERVICE

tcp open echo

13/tcp open daytime

20/tcp open ftp-data

21/tcp open ftp

tcp open ssh

open telnet

open smtp

open http

open hosts2-ns

p open pop3

p open imap

p open bgp

p open ldap

p open https

p open smtps

p open submission

tcp open telnets

993/tcp open imaps

995/tcp open pop3s

1723/tcp open pptp

2323/tcp open 3d-nfsd

2525/tcp open ms-y-world



# Process Status: procesos arrancados en cada puerto.

- `ps -ef |grep perl`

```
usuario@ldap:/usr/bin$ ps -ef|grep perl
root      2577      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2578      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2579      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2580      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2581      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2582      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2583      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2584      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2585      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2586      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2587      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2588      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2589      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2590      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2591      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2592      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2593      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2594      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2595      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2596      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2598      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2599      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2600      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2601      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2602      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2603      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2604      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2605      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root      2606      1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
usuario   2872    2608  0 20:33 pts/2    00:00:00 grep --color=auto perl
```

- Si ejecutamos un "Process Status" podremos ver los procesos PERL arrancados para cada puerto, con sus correspondientes identificadores de proceso.
- Esto nos será útil más tarde, dado que para parar los procesos PERL de Malbait será preciso matarlos con "killall malbait".

# Matamos todos los procesos

killall malbait

```
archivo  maquina  ver  entrada  dispositivos  ayuda
usuario@ldap: /usr/bin
root    2580    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root    2581    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait -defaults
root    2582    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2583    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2584    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2585    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2586    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2587    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2588    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2589    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2590    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2591    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2592    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2593    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2594    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2595    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2596    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2598    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2599    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2600    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2601    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2602    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2603    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2604    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2605    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
root    2606    1  0 19:54 pts/0    00:00:00 /usr/bin/perl -w /usr/bin/malbait
usuario 2872 2608 0 20:33 pts/2    00:00:00 grep --color=auto
usuario@ldap: /usr/bin$ sudo killall malbait
[sudo] password for usuario:
usuario@ldap: /usr/bin$
```

Archivo Editar Ver Buscar Terminar  
Try using --system-dns or spec  
168.1.1

3080/tcp open http-proxy  
9999/tcp open abyss  
MAC Address: 08:00:27:0A:74:7A

- Matamos todos los procesos de Malbait para pasar a la segunda parte de la práctica, en la que nos centraremos en un par de protocolos en particular, y en el comportamiento simulado que éstos presentan ante el ataque.
- Como hemos comentado al principio, se trata de una herramienta de tipo simple, por lo que su profundidad funcional no es muy grande, pero sí lo es su efectividad a la hora de crear confusión y despistar a los posibles atacantes.

## Ejecutamos de nuevo nmap

- Volvemos de nuevo al escenario antes de ejecutar Malbait.

- Volvemos a ejecutar nmap en kali y observamos que ya no están a la escucha los puertos que había abierto Malbait, presentándose como activos los mismos puertos reales que estaban operativos al principio de la práctica.

```
root      2604          0        0        0 pts/0    00:00:00 /usr/bin/sshd -p /etc/ssh/sshd_config
root      2605          1        0        0 pts/0    00:00:00 /usr/sbin/sshd -p /etc/ssh/sshd_config
root      2606          1        0        0 pts/0    00:00:00 /usr/bin/ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=quiet -o PasswordAuthentication=no -o BatchMode=yes root@172.17.0.1
usuario   2872      2608          0        0        0 pts/0    00:00:00 /usr/bin/greasybot --no-sandbox --disable-features=IsolateOrigins,site-per-process,enable-automation --remote-debugging-port=9222 --headless --incognito --load-extension=/tmp/.chromium-devtools
usuario@ldap:/usr/bin$ sudo killall sshd
[sudo] password for usuario:
usuario@ldap:/usr/bin$
```


MAC Address: 08:00:27:0A:74:7A (Oracle VirtualBox virtual NIC)

- Antes de seguir con la práctica, instalamos un servidor Apache, para ocupar el puerto 80 y comprobar que Malbait no arranca ningún proceso en él si ya está ocupado.
- Para visualizar la página de arranque de apache, simplemente rellenamos en el navegador del PC la dirección IP de la máquina acabada en :80 (o sin especificar el puerto, por tratarse del puerto web por defecto).

← → ↻ 🏠 No es seguro | http://192.168.1.76 ☆

WORK seguridad Educación Oficial Telefonica Sociales Finanzas Comercial Salud Seguros Suministros Nubes Multimedia Lectura

## Apache2 Debian Default Page

 **It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

training Kali Tools

t has been added,

```
root@ldap:/usr/bin# sudo malbait -proto:http
```

```
Protocol selection:http
```

```
Assuming we're running, we've selected protocol http, overriding defaults
Will write report to malbait-report-http.txt
```

```
Whois scan will be performed...
```

```
nmap_scan is not enabled, so client will not be nmaped.
-noloop option is enabled, this program will loop infinitely until you break
out of it with Ctrl+C. All it (but why would you wanna do that?)
```

```
*****
```

```
* TRANSPORT *
```

- Arrancamos Malbait sólo con el protocolo http, para que sea capaz de simular una página web.
- Si no especificamos ningún puerto, Malbait arranca sus procesos simuladores de webserver sobre los puertos 8080, 81 y sobre el puerto clásico por defecto 80, en caso de que éste último no esté ocupado por un webserver real dentro de la máquina atacada (como es nuestro caso).
- Tras rearrancar Malbait, podremos comprobar en la máquina kali que ya aparecen dos puertos a la escucha con protocolos http.

```
Aplicaciones Lugares Terminal mar, 23 de nov, 21:12
```

```
Archivos
```

```
root@kali: ~
```

```
root@kali:~# nmap 192.168.1.10
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-23 21:
mass_dns: warning: Unable to determine any DNS servers. R
Try using --system-dns or specify valid servers with --d
Nmap scan report for 192.168.1.10
Host is up (0.00013s latency).
```

```
Not shown: 993 closed ports
```

```
PORT      STATE SERVICE
```

```
22/tcp    open  ssh
```

```
80/tcp    open  http
```

```
81/tcp    open  hosts2-ns
```

```
389/tcp   open  ldap
```

```
443/tcp   open  https
```

```
1723/tcp  open  pptp
```

```
8080/tcp  open  http-proxy
```

```
MAC Address: 08:00:27:0A:74:7A (Oracle VirtualBox virtual
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.17 second
```

```
root@kali:~#
```

## Malbait: Arrancamos solo el protocolo http

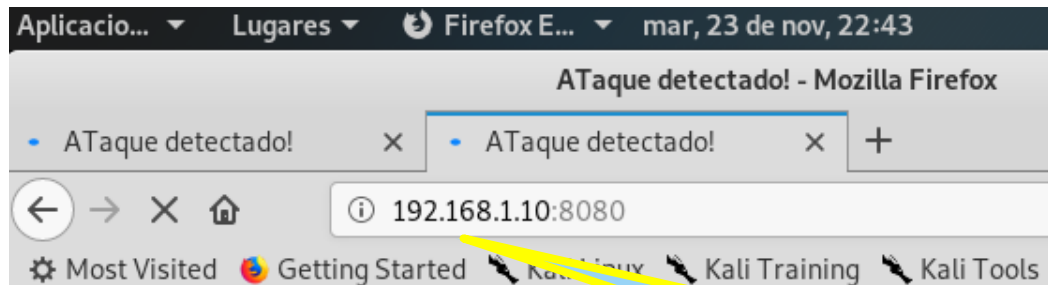
Sudo malbait -proto:http

```
root@ldap:/usr/bin# netstat -anp|grep perl
tcp        0      0 0.0.0.0:8080          0.0.0.0:*            ESCUCHAR   3319/perl
tcp        0      0 0.0.0.0:81           0.0.0.0:*            ESCUCHAR   3320/perl
root@ldap:/usr/bin#
```

## Puertos abiertos alternativos al 80

- Vemos que, en esta ocasión, sólo hay dos procesos PERL simulando los dos webservers a la escucha en los puertos 8080 y 81.

```
netstat -anp|grep perl
```



Navegamos puerto 8080

¡ATAQUE DETECTADO! /home/usuario/malbait

A un panal de rica miel, dos mil moscas acudieron, que por golosas murieron, presas de patas en él.

```
Archivo Editar Ver Buscar Terminal Ayuda
root@kali:~# curl 192.168.1.10:81
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html> <title> ATAque detectad
o!</title>

<body bgcolor="blue">
  <font color="yellow">
    <P>
    <B> ¡ATAQUE DETECTADO! /home/usuario/malbait</B>
    <P>
    A un panal de rica miel, dos mil moscas acudieron, que por golosas murieron, pre
sas de patas en él.
    <P>
    <B>
    Capturados datos del atacante
    </B>
    <P>
    <I> INICIAMOS ACCIONES LEGALES</I>
    </font>
  </HTML>

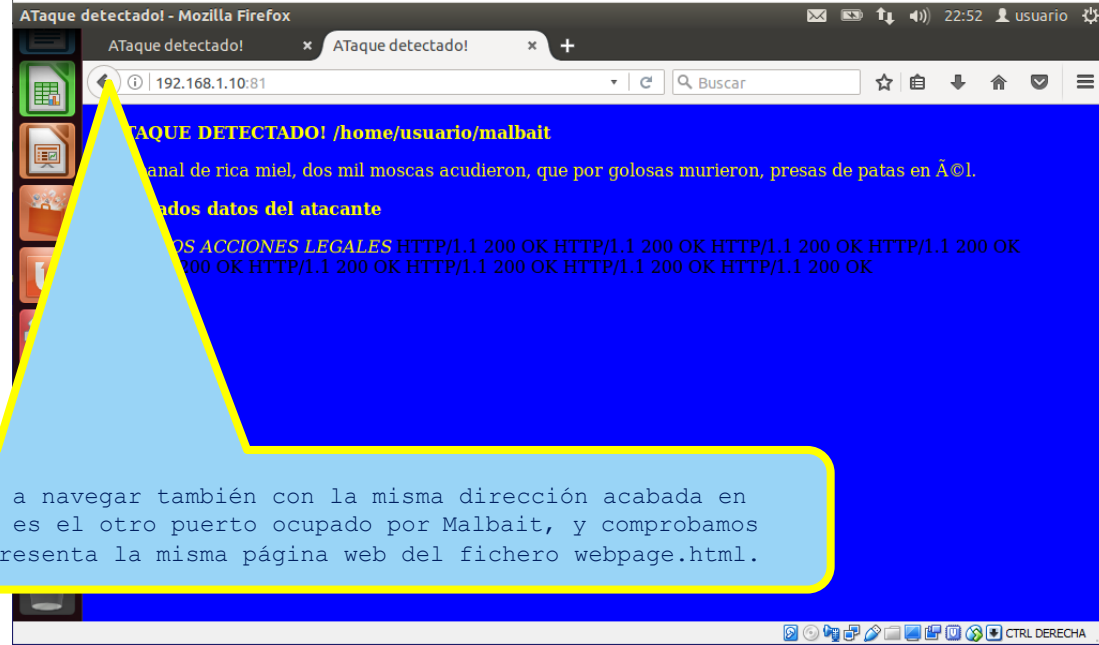
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
curl: (18) transfer closed with 431 bytes remaining to read
root@kali:~#
```

- Navegamos con la misma dirección, pero acabada en :8080, que es uno de los puertos ocupados por Malbait.
  - En este caso, el honeypot simula un webserver con el código del fichero webpage.html situado en el directorio Malbait.
  - El fichero por defecto que viene con la instalación es una página que simula un ransomware, pero nosotros lo hemos modificado para que sea más agradable.
  - Lo más recomendable es simular una página web realista de nuestra empresa, pues así entretendremos más tiempo al hacker
- También podemos ver la página con curl:

curl 192.168.1.10:8080



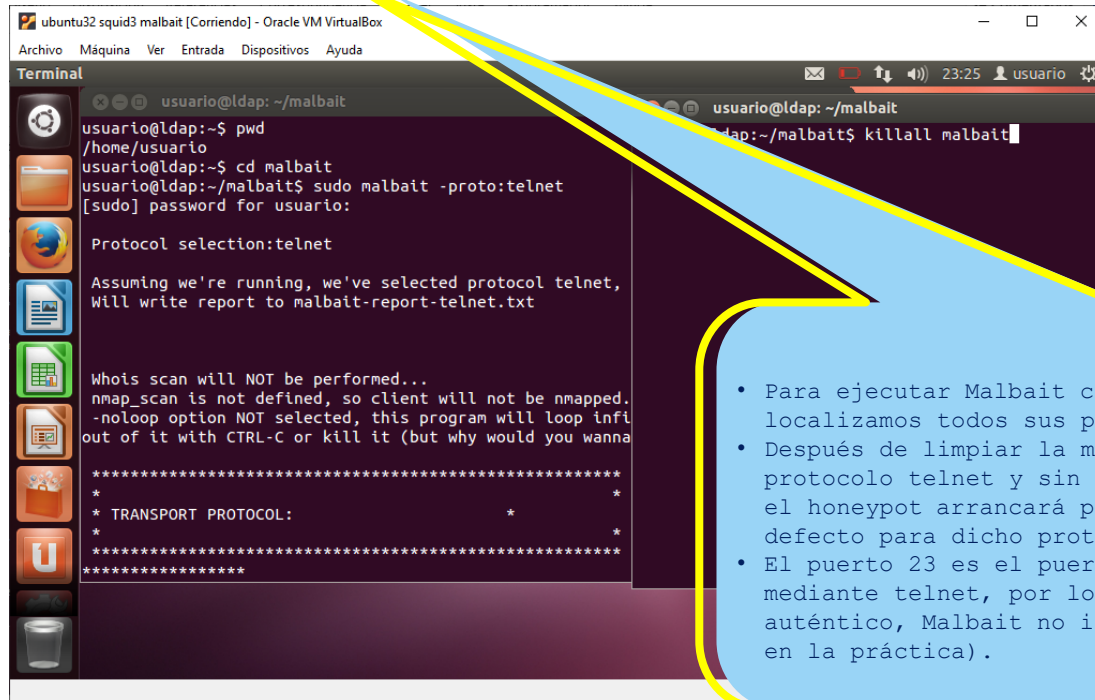
## Puerto 81





# Ejecutamos Malbait con otro protocolo

• malbait -proto:telnet



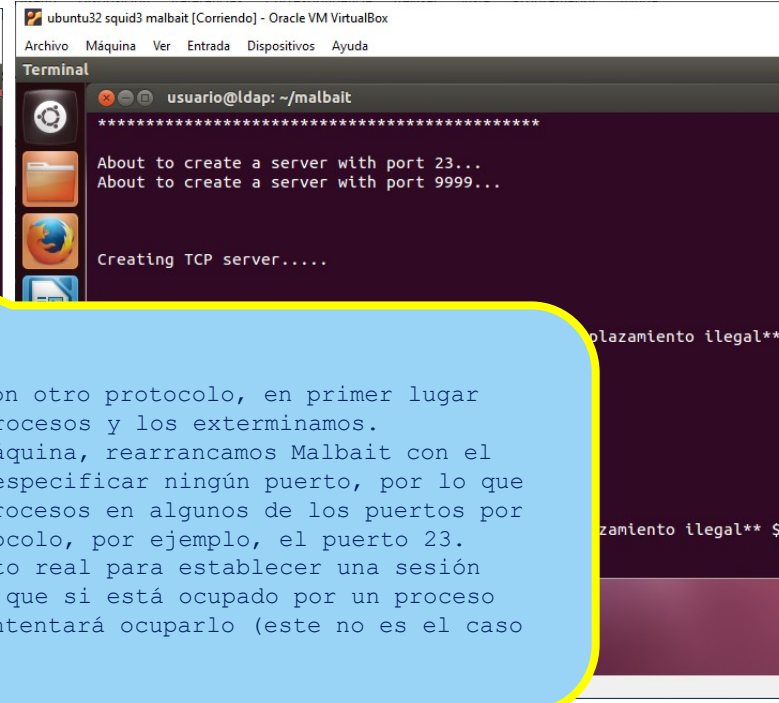
```
usuario@ldap: ~/malbait
usuario@ldap:~$ pwd
/home/usuario
usuario@ldap:~$ cd malbait
usuario@ldap:~/malbait$ sudo malbait -proto:telnet
[sudo] password for usuario:

Protocol selection:telnet

Assuming we're running, we've selected protocol telnet,
Will write report to malbait-report-telnet.txt

Whois scan will NOT be performed...
nmap_scan is not defined, so client will not be nmapped.
-noloop option NOT selected, this program will loop infi
out of it with CTRL-C or kill it (but why would you wanna

*****
*
* TRANSPORT PROTOCOL:
*
*****
```



```
usuario@ldap: ~/malbait
*****
About to create a server with port 23...
About to create a server with port 9999...

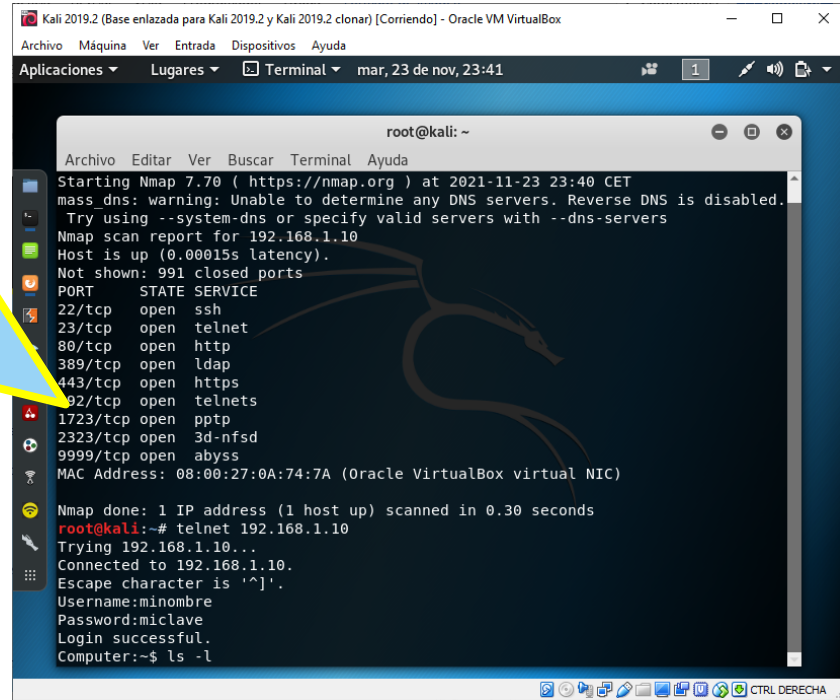
Creating TCP server.....
```

- Para ejecutar Malbait con otro protocolo, en primer lugar localizamos todos sus procesos y los exterminamos.
- Después de limpiar la máquina, rearrancamos Malbait con el protocolo telnet y sin especificar ningún puerto, por lo que el honeypot arrancará procesos en algunos de los puertos por defecto para dicho protocolo, por ejemplo, el puerto 23.
- El puerto 23 es el puerto real para establecer una sesión mediante telnet, por lo que si está ocupado por un proceso auténtico, Malbait no intentará ocuparlo (este no es el caso en la práctica).

# Descubriendo y atacando

- telnet 192.168.1.10

- El hacker que opera desde la máquina kali sigue utilizando nmap para mapear puertos, y descubre que hay un proceso telnet a la escucha en el puerto habitual, esto es, el 23.
- Dado que telnet es un protocolo con ciertas debilidades (de hecho, se usa cada vez menos, utilizándose en su lugar ssh), el hacker decide atacarlo usando claves obtenidas mediante ingeniería social, o simplemente por fuerza bruta.
- El proceso de Malbait simula un login, que acepta cualquier usuario y clave, permitiendo al hacker entrar en una sesión de una máquina inexistente, denominada "Computer".
- Ninguno de los comandos que se tecleen en dicha sesión tendrá ningún efecto, pero quedará todo convenientemente registrado en un log, incluidas las credenciales que el hacker haya utilizado para entrar (esto último es muy útil para saber cómo se han averiguado las claves).



The screenshot shows a Kali Linux terminal window titled "Kali 2019.2 (Base enlazada para Kali 2019.2 y Kali 2019.2 clonar) [Corriendo] - Oracle VM VirtualBox". The terminal displays the output of an Nmap scan for 192.168.1.10, showing several open ports including 23/tcp (telnet). Following the scan, the user runs the command "telnet 192.168.1.10", which successfully connects to the target. The telnet session shows a prompt for a username, followed by "minombre", and a prompt for a password, followed by "miclave". The connection is successful, and the user is prompted with a command prompt "Computer:~\$".

```
root@kali: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-23 23:40 CET  
mass dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.  
Try using --system-dns or specify valid servers with --dns-servers  
Nmap scan report for 192.168.1.10  
Host is up (0.00015s latency).  
Not shown: 991 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
23/tcp    open  telnet  
80/tcp    open  http  
389/tcp   open  ldap  
443/tcp   open  https  
492/tcp   open  telnet  
1723/tcp  open  pptp  
2323/tcp  open  3d-nfsd  
9999/tcp  open  abyss  
MAC Address: 08:00:27:0A:74:7A (Oracle VirtualBox virtual NIC)  
  
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds  
root@kali:~# telnet 192.168.1.10  
Trying 192.168.1.10...  
Connected to 192.168.1.10.  
Escape character is '^]'.  
Username:minombre  
Password:miclave  
Login successful.  
Computer:~$ ls -l
```

# Entrenteniendo al atacante

- Cierra la sesión de forma inesperada

- Después de entretener un rato al hacker, haciéndole teclear comandos inútiles, Malbait temporiza y cierra la falsa sesión.

```
Dottarex (Error):Llamada al siste...
looptimes:1
server_prompt:Computer::~$
grandchild_pid:2740 forktimes: 0
max_requests_per_server: 2
protocol: telnet
blocking:1
*****

About to create a server with port 23...

Creating TCP server.....

Listening on port 23 using tcp protocol $!:**Desplazamiento
**16**
```

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

root@kali:~# telnet 192.168.1.10
Trying 192.168.1.10...
Connected to 192.168.1.10.
Escape character is '^J'.
Username:minombre
Password:micontraseña
Login successful.
Computer::~$ ls -l
Computer::~$ pwd
Computer::~$ dir
Computer::~$ cd ..
Computer::~$ list
Computer::~$ Connection closed by foreign host.
root@kali:~#
```

- Traza del origen del paquete y credenciales usadas.

- Traza del origen del paquete y credenciales usadas.

The screenshot shows a terminal window titled "ubuntu32 squid3 malbait [Corriendo] - Oracle VM VirtualBox". The user is logged in as "usuario@ldap: ~/malbait\$".

```

usuario@ldap:~/malbait$ ls
INSTALL.sh          malbait-report-telnet-table.csv  TEMP-3298
malbait.pl          malbait-report-telnet.txt       TEMP-4238
malbait-report-http-table.csv    malbait-report-.txt            webpage.html
malbait-report-http.txt         README.md                      TEMP-3252
malbait-report--table.csv
usuario@ldap:~/malbait$ cat malbait-report-telnet.txt
##### tcp server: AN EVIL HACKER CONNECTED TO US! #####
#####
*** LoopTimes:1 *** Time: Tue Nov 23 23:41:25 2021 ***
Then: 192.168.1.20 port 58360
Us: (our own IP) port 23

The client has contacted us 1 times this session
Server:Username:

Client :♦♦♦♦♦♦♦♦♦♦ ♦♦♦♦♦♦♦♦♦♦#minombre

Analysis:

countchr:0      currentchr:♦ 255
countchr:1      currentchr:♦ 253
countchr:2      currentchr:▢ 3
countchr:3      currentchr:♦ 255
countchr:4      currentchr:♦ 251
countchr:5      currentchr:▢ 24
countchr:6      currentchr:♦ 255
countchr:7      currentchr:♦ 251
countchr:8      currentchr:▢ 31
countchr:9      currentchr:♦ 255
countchr:10     currentchr:♦ 251
countchr:11     currentchr: 32
countchr:12     currentchr:♦ 255
countchr:13     currentchr:♦ 251
countchr:14     currentchr:! 33
  
```

- Tras finalizar el ataque, en el directorio de Malbait se habrán grabado ficheros .csv y .txt con todos los detalles del ataque.
- Esta información resulta muy útil para el análisis forense del honeypot pues, en muchas ocasiones, permite trazar el origen del ataque, y además permite prevenir futuros ataques.
- Estos ficheros de log se suelen conectar con una pila ELK (Elastic Search, Logstash, Kibana) en el SOC para alertar a los operadores de un ataque en primer plano, y preparar rápidamente la estrategia defensiva del segundo plano, que es donde están situadas las máquinas de verdad.

## Bibliografía

- <https://github.com/batchmcnulty/Malbait>
- Transcripción del Readme.rm de Malbait  
<https://github.com/desarrolloweb/Malbait>