Cómo crear un Honeypot paso a paso

Autor: Rafael Ruda

Versión: 1.0 (20130727)

Edición y publicación: www.segu-info.com.ar

Introducción

Es un placer tener la oportunidad de realizar esta pequeña introducción a los Honeypot.

En este caso voy a explicar cómo desarrollar un pequeño sistema de seguridad que he configurado en mi casa. Hay personas con las que he hablado sobre este tema e incluso en foros especializados y me han preguntado si ese esfuerzo es necesario. Solo puedo decirles que después de muchas horas instalando, configurando y monitorizando el sistema, el esfuerzo ha valido mucho la pena.

¿Cuántos de ustedes que se dedican a la seguridad alguna vez ha pensado en la cantidad de ataques que recibimos al estar conectados a Internet?

¿Cuántos de ustedes que no han tenido la oportunidad de introducirse en este maravilloso mundo de la seguridad informática son desconocedores de todo esto que sucede?

Hace unas pocas semanas en un foro de seguridad abrí un debate sobre una noticia que a mí personalmente no me causó ninguna sorpresa. El debate tenía cómo argumento si estamos espiados en el mundo virtual, por quién y porqué.

Cómo es de suponer había opiniones de todos los colores. Y eso me hizo reflexionar sobre este tema. Creo que ahora más que nunca es necesario disponer de un control total sobre nuestras conexiones y los ataques que recibimos.

Como ustedes verán más adelante la cantidad de ataques que recibimos es enorme, así como el malware que nos inyectan para poder tomar el control de nuestra computadora.

Según mi opinión (esto lo digo para que no quede ninguna duda al respecto) deberíamos trabajar en el camino de poder ayudar a las personas que no pueden por desconocimiento dar a conocer todo lo anteriormente expuesto. Es por esto por lo que me he animado a escribirles este pequeño diario personal sobre cómo realizar una puesta en marcha de un Honeypot desde el punto de vista de las personas sin grandes conocimientos y que les sirva de guía si lo consideran necesario.

Las capturas muestran la cantidad de ataques que he sufrido en unas pocas horas de funcionamiento del Honeypot y la cantidad de binarios que he conseguido con malware.

Instalación de la máquina virtual

He realizado este trabajo de dos formas diferentes:

- La primera instalando un Honeypot en una máquina física e instalando **Stratagem**. Esta es una distribución basada en **Linux Mint** y en la cual se encuentra instalado la base de los honeypots mencionados. Por ejemplo Dionaea está instalado y es plenamente funcional pero no tiene el entorno gráfico. Este debemos de instalarlo nosotros. He de comentar que no hubo ningún problema al instalar estos componentes. Me decanté por esta versión ya que tiene un honeypot para sistemas SCADA que actualmente estoy monitorizando y estudiando para poder mejorarlo.
- La segunda y la que voy a explicar aquí se refiere a la instalación de una distribución llamada **Honeydrive** en una máquina virtual ya preparada para funcionar con **Virtualbox**.

Me he decantado a explicarles este segundo tema porque me parece interesante que ustedes tomen una primera toma de contacto con esta clase de programas ya preconfigurados y sin perderse en la instalación de uno desde el principio.

Les recomiendo poner en marcha esta distribución en un sistema totalmente parcheado y con una suite de seguridad con el antivirus actualizado.

Mi instalación la realicé bajo un sistema con Windows 7, totalmente actualizado y por supuesto con mi suite de seguridad totalmente actualizada.

Para realizar la instalación de la máquina virtual debemos de ir a la página https://www.virtualbox.org/wiki/Downloads y descargarnos el software para nuestra plataforma.

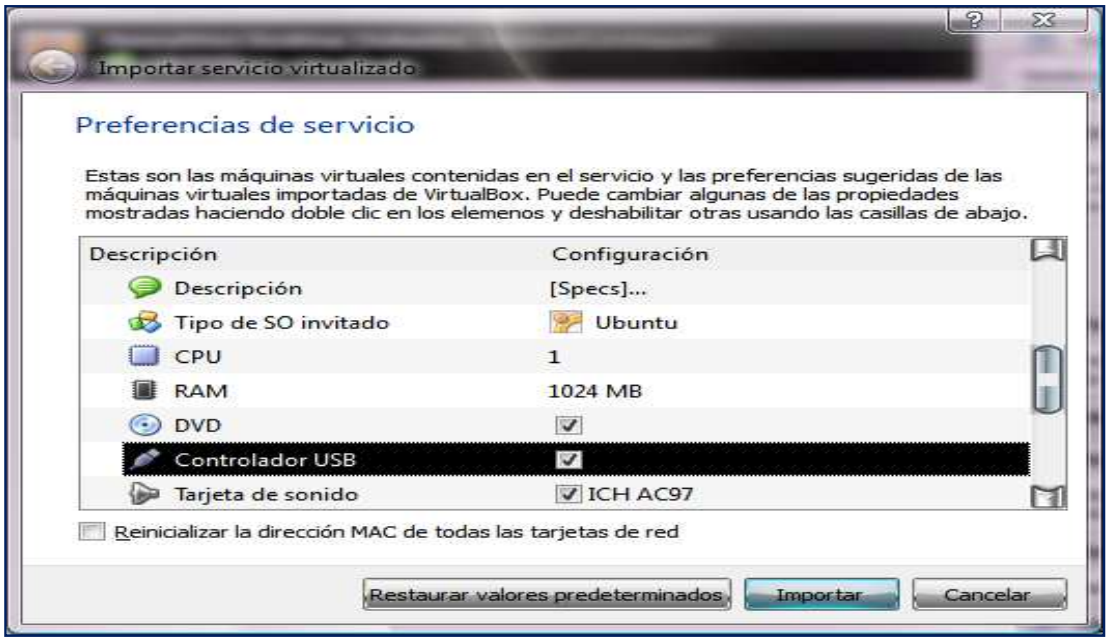
Una vez descargado lo instalamos, no creo que haga falta explicar cómo instalar VirtualBox en nuestra computadora.

Una vez instalado nos vamos a bajar la máquina virtual Honeydrive que alberga nuestros Honeypot: http://bruteforce.gr/honeydrive

Entre sus características podemos encontrar:

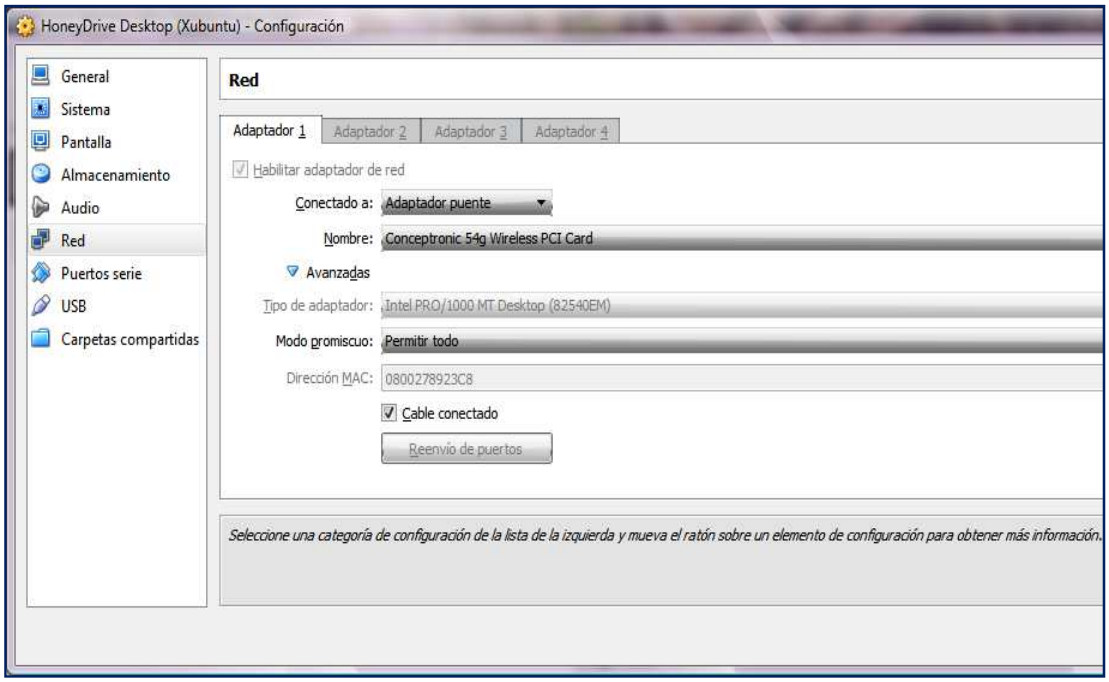
- Corre con Xubuntu Desktop 12.04 32 bits
- Servidor LAMP con PHPMyAdmin
- Kippo SSH, Kippo-Graph y Kippo2MySQL
- Dionaea Malware + phpLiteAdmin
- Honeyd + Honeyd2MySQL y Honeyd-Viz
- LaBrea, Tiny Honeypot, IIS Emulator, INetSim y SimH
- Varias utilidades para el análisis de malware, PDFs, etc

La imagen ocupa más de 2 Gb y recomiendo que la instalen bajo Virtual Box ya que está optimizada para ello. Se puede correr bajo VMWare pero nosotros vamos a seguir la recomendación y la utilizaremos bajo VirtualBox. Una vez descargado todo lo necesario lo vamos a poner en marcha y seguimos con la configuración predeterminada.

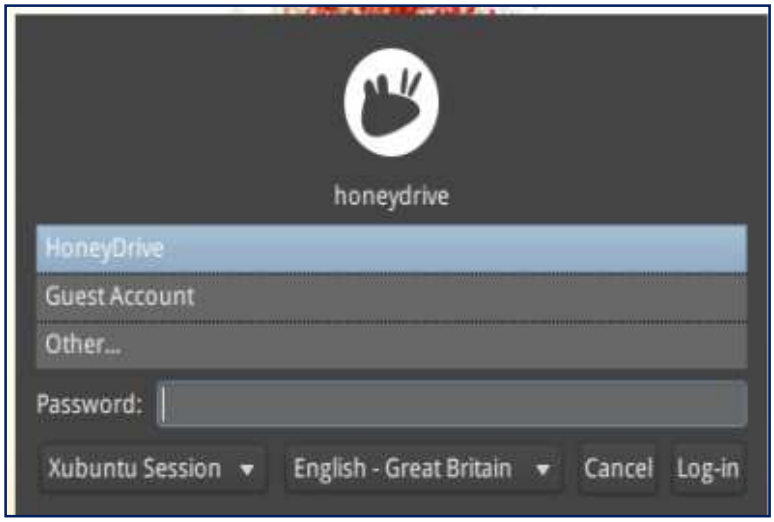


Podemos dejarla como está o modificarla a nuestro gusto. Yo le pongo 1024 Megas de RAM. Le damos a importar y nos saldrá un aviso de licencia que debemos aceptar.

Ahora importará la imagen de la máquina virtual. Una vez importada se debe configurar de red y poner la placa de red en modo promiscuo. Puede que informe sobre que la tarjeta de red no es la misma. En este caso simplemente se debe aceptar y poner la que tenga el sistema.

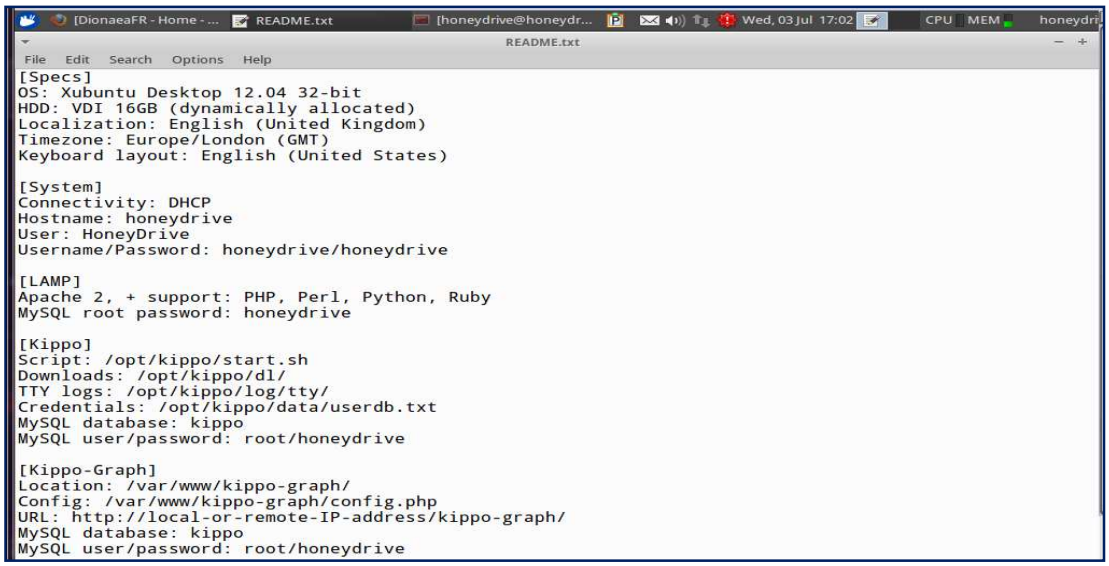


Le damos a iniciar la máquina virtual y esperamos a que cargue la imagen de Honeydrive. Cuando el proceso termine se debe ver esta pantalla.



La contraseña es *honeydrive*. Luego de ingresar recomiendo cambiar la contraseña a través del comando `sudo passwd`.

Lo primero que vamos a realizar es abrir el fichero *readme.txt*. En este archivo nos encontramos una aclaración de los honeypots que podemos utilizar y la ubicación de los archivos de configuración de cada uno de ellos. Esto va a ayudar mucho en nuestra labor de configuración.



Nosotros vamos a poner en marcha dos honeypots, **Dionaea** y **Kippo**. Creo que con estos dos tenemos más que suficiente y cubre nuestras necesidades. Aconsejo apuntar dónde están ubicadas las rutas de los archivos de configuración.

Llegados a este punto tengo que hacer una pequeña introducción de que es un honeypot y una breve explicación de cada honeypot que he elegido para que tengáis claro el funcionamiento y para que lo vamos a utilizar.

Honeypot

Debido a la cantidad de ataques que sufrimos en Internet desde hace años las empresas, estamentos gubernamentales, usuarios avanzados, etc. han tenido que adoptar medidas para incrementar la seguridad de los sistemas. Una de las maneras de implementar esta seguridad es la utilización de sistemas trampa para observar el comportamiento de un ciberataque y analizar la intrusión y el método utilizado.

Estos sistemas simulan ser equipos vulnerables y que son perceptibles de ser atacados. Los especialistas en seguridad tienen una gran ventaja al utilizar estos métodos ya que les da la posibilidad de aprender métodos de intrusión, captura de malware entre los que puede haber un *zero-day* y por supuesto analizar con tiempo la intrusión y poder aplicar fórmulas de seguridad que permitan detectar y rechazar estos ataques.

Estos sistemas trampa muchas veces nos proporcionan los programas que han utilizado los ciberdelincuentes y como todos sabemos eso es oro en nuestro trabajo. Una vez obtenidos estos programas podemos realizar un análisis en profundidad para observar el comportamiento y tomar las medidas oportunas. En algunos casos son ataques con mucho ingenio ayudados por creaciones de malware que te dan una idea de la magnitud y dedicación de estas personas al desarrollar sus armas de ataque.

Las funciones principales de un Honeypot son:

- Desviar la atención del atacante con este tipo de sistemas para salvar el sistema principal y en muchos casos en el que el atacante pueda determinar que se trata de un honeypot poder disuadirle de seguir adelante con la intrusión o ganar un tiempo muy valioso que nos permita reaccionar y tomar las medidas oportunas para frenar el ataque.
- Capturar nuevo tipo de malware para el estudio del mismo.
- Poder obtener una base de datos con direcciones de atacantes y métodos de ataques desconocidos.
- Una de las más importantes. Poder conocer nuevas vulnerabilidades y de esta manera poder aplicar medidas para que no afecten a la seguridad de nuestros sistemas.

Clasificación

Para poder implementar debidamente un honeypot debemos tener clara la idea de qué tipo de información queremos obtener. Una vez tengamos claro este concepto solo nos queda llevar a cabo la idea. El típico honeypot y, a mí personalmente el que más me gusta, es aquel que nos permite analizar el comportamiento de un atacante y las herramientas

que está utilizando para poder descubrir nuevos tipos de ataque y vulnerabilidades en el sistema.

Creo que este es el que más nos puede interesar de cara a si somos administradores de sistemas o simplemente usuarios con inquietudes y ganas de aprender.

Según lo anterior, se debe remarcar que un honeypot no es una medida de seguridad en un sistema. Es una implementación en nuestro sistema de seguridad, es decir que por sí solo no sirve para poder frenar un ataque y asegurar de manera efectiva el sistema que tenemos que administrar.

Una de las clasificaciones que podemos llevar a cabo al implementar un honeypot es según el nivel de interacción que un atacante puede tener con él.

Lo importante de nuestro honeypot es que el atacante no se dé cuenta de que es un sistema trampa y que genere mucha actividad dentro del mismo. De esta manera estará facilitándonos información que puede resultar muy valiosa y aprenderemos mucho sobre sus métodos de ataque y el tipo de herramientas que ha utilizado.

De todas maneras, cuanta más interacción tenga el atacante con el honeypot más expuestos estaremos a que logre tener acceso a otros equipos “reales” de la red.

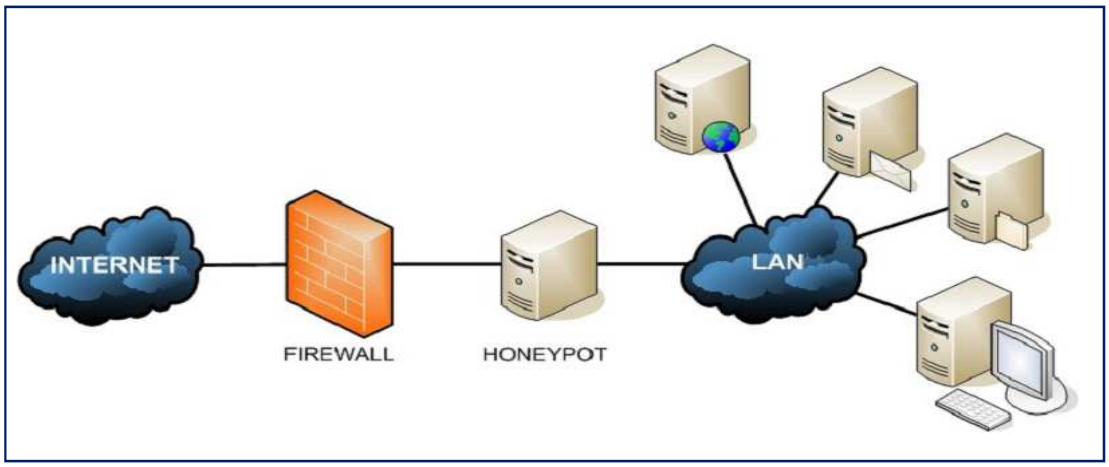
Según el nivel de interacción podemos clasificarlos.

Baja Interacción	Son los menos complicados de implantar. La funcionalidad de la que disponen es muy reducida y lo que realizan prácticamente es la emulación de servicios que van a permitir a un atacante realizar interactuar sin poder afectar al sistema en el que está implementado. Por ejemplo podemos observar con esta clase de honeypot los *login* de sesiones y la detección de escaneos. Un ejemplo del mismo puede ser Specter, HoneyBot, etc.
Media Interacción	Un atacante tiene mayor libertad de interactuar con nuestro honeypot. El honeypot es capaz de generar respuestas ante la interacción del atacante con un servicio en concreto. Puede emular servicios cómo la anterior clase que hemos comentado anteriormente y puede emular el comportamiento de software. Como ejemplo podemos mencionar la emulación de un servidor web y capturar el tráfico que genera el atacante hacia él. Cómo ejemplo podemos poner el honeypot Kippo.

Alta Interacción	Este tipo proporciona mucha información sobre los atacantes y sus métodos. Por el contrario requiere de mucho más tiempo para su implementación y mantenimiento. Suelen ser riesgosos si no están bien configurados. Aquí no tenemos ningún servicio ni software emulado. Nuestro atacante tiene el sistema operativo para interactuar con él. Cómo punto positivo tenemos que la información que nos proporciona es muy valiosa. Hay gran cantidad de información, nuevas herramientas de ataque, nuevo tipo de malware, nuevas vulnerabilidades, etc. Dado el nivel de riesgo que implican este tipo de honeypots se ponen en ambientes controlados, ya que si se logra el control del mismo podemos poner todo el sistema bajo control del atacante. Para mitigar estos riesgos, los Honeypots de alta interacción suelen colocarse detrás de un firewall. La principal función del firewall será el permitir todo el tráfico de entrada hacia los Honeypots, pero impedir que desde el Honeypot se ataque a otros equipos de la red. Una honeypot de este tipo es Honeynet.

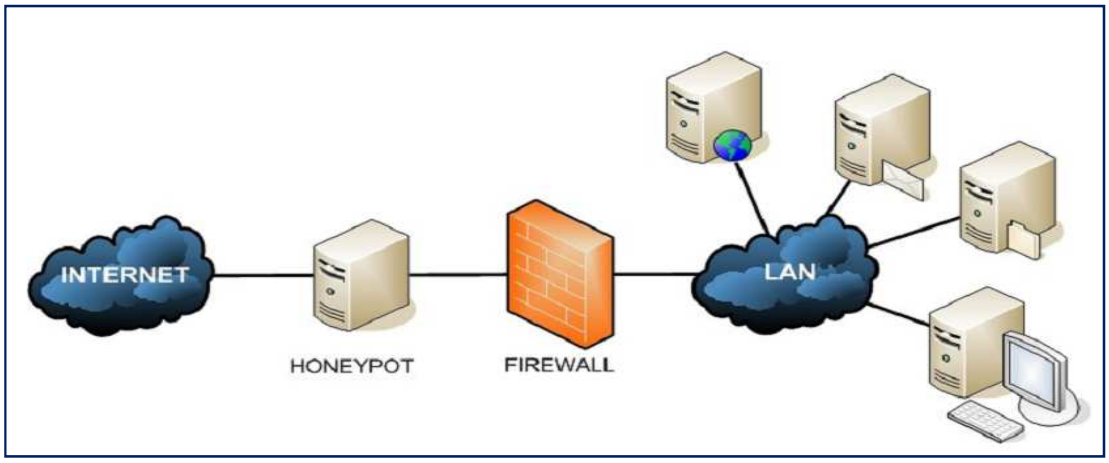
Ubicación de un honeypots

Antes del firewall (Front of firewall): esta localización permite evitar el incremento del riesgo inherente a la instalación del honeypot. Como este se encuentra fuera de la zona protegida por el firewall, puede ser atacado sin ningún tipo de peligro para el resto de la red.



Esta configuración evitara las alarmas de otros sistemas de seguridad de la red (IDS) al recibir ataques en el honeypot. Sin embargo, existe el peligro de generar mucho tráfico debido precisamente a la facilidad que ofrece el honeypot para ser atacado.

Detrás del firewall (Behind the firewall): en esta posición, el Honeypot queda afectado por las reglas de filtrado del firewall. Por un lado se tiene que modificar las reglas para permitir algún tipo de acceso al honeypot por posibles atacantes externos, y por el otro lado, al introducir un elemento potencialmente peligroso dentro de la red se puede permitir a un atacante que gane acceso al honeypot y a la red.



Cualquier atacante externo será lo primero que encuentra y esto generará un gran consumo de ancho de banda y espacio en los ficheros de log. Por otro lado, esta ubicación evita la detección de atacantes internos.

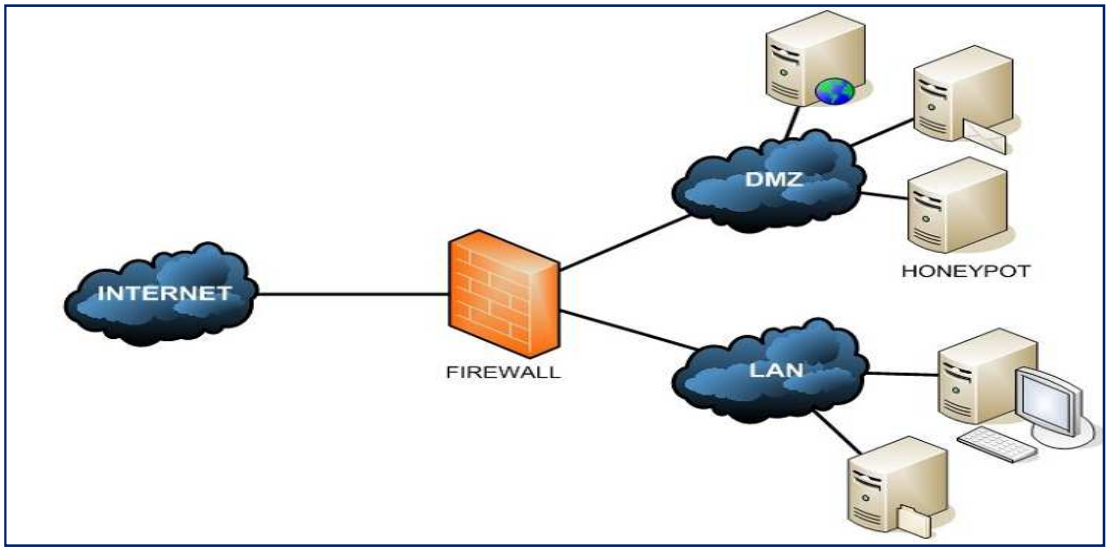
La ubicación tras el firewall permite la detección de atacantes internos así como firewalls mal configurados, máquinas infectadas por gusanos o virus e incluso atacantes externos.

Sin embargo las contrapartidas más destacables son la gran cantidad de alertas de seguridad que generarán otros sistemas de seguridad de la red (Firewalls, IDS). Al recibir ataques el honeypot se ve la necesidad de asegurar el resto de nuestra red contra el honeypot mediante el uso de firewalls extras o sistemas de bloqueo de acceso, ya que si un atacante logra comprometer el sistema tendrá vía libre en su ataque a toda la red.

Hay varias circunstancias que obligan a este tipo de arquitectura, como por ejemplo la detección de atacantes internos o la imposibilidad de utilizar una dirección IP externa para el honeypot.

En la zona desmilitarizada (DMZ): la ubicación en la zona desmilitarizada permite por un lado juntar en el mismo segmento a los servidores de producción con el honeypot y por el

otro controlar el peligro que añade su uso, ya que tiene un firewall que lo aísla de resto de la red local.



Esta arquitectura nos permite tener la posibilidad de detectar ataques externos e internos con una simple reconfiguración del sistema de firewall puesto que se encuentra en la zona de acceso público.

Además, se elimina las alarmas de los sistemas internos de seguridad y el peligro que supone para la red al no estar en contacto directo con esta. La detección de atacantes internos se ve algo debilitada, puesto que al no compartir el mismo segmento de red que la LAN, un atacante local no accederá al Honeypot.

Sin embargo, desde la red local sí es posible acceder al Honeypot, con lo que un atacante interno que intente atacar a los servidores públicos u otros sistemas externos, por ejemplo un gusano, muy probablemente acabe siendo detectado.

Se puede obtener mayor cantidad de información en la tesis “Diseño y desarrollo de Honeypots virtuales utilizando VMWare para la detección de intrusos informáticos” de Ventura Penado, Yesenia Lisseth Rodríguez Campos y Nelson Alfredo (marzo 2008).

Conociendo todas las posibilidades, yo he decidido colocar este sistema trampa en una DMZ creada exclusivamente para su uso.

Instalación de los honeypots

Es importante el tema de la configuración de red. Aconsejo editar el fichero de configuración de Honeydrive y poner una IP estática para no tener cambiar la configuración de las interfaces cada vez que arranque el sistema. El archivo en cuestión se encuentra en la ruta `/etc/network/interfaces`.

Bien echa estas aclaraciones vamos a empezar a instalar los honeypots.

Ante todo un consejo: cuando pongan en marcha estos honeypot practiquen con ellos unos días para ver su funcionamiento, siempre monitorizando el host que alberga la máquina virtual. Es lo que yo he realizado.

Kippo

Aquí presento a Kippo, un honeypot de media interacción:
http://code.google.com/p/kippo/

Este tipo de honeypot nos va a permitir emular un servicio SSH con un *login* y *pass* poco seguras, reproducir un sistema de ficheros en el que incluso podremos integrar un sistema operativo, poner documentos y ejecutar comandos.

En este tipo de honeypot aprenderemos mucho sobre un atacante ya que quedan registrados los comandos utilizado por un atacante y se pueden reproducir de manera automática la sesión que se ha establecido. Nos muestra estadísticas, localización GeoIP, un top de comandos introducidos, comandos "curiosos", etc.

Directorios importantes

- *dl/:* donde se guarda los ficheros descargados mediante wget
- *log/kippo.log:* donde se guarda información de uso y depuración
- *log/tty/: logs* de las sesiones
- *utils/playlog.py:* herramienta para reproducir los *logs* de sesión
- *utils/createfs.py:* utilizado para crear *fs.pickle*
- *fs.pickle:* falso sistema de ficheros
- *honeyfs/:* contenido del falso sistema de ficheros. Aquí podemos poner una copia de un sistema real.

Archivo de configuración:

```
[Kippo]
Script: /opt/kippo/start.sh
Downloads: /opt/kippo/dl/
TTY logs: /opt/kippo/log/tty/
Credentials: /opt/kippo/data/userdb.txt
MySQL database: kippo
MySQL user/password: root/honeydrive

[Kippo-Graph]
Location: /var/www/kippo-graph/
Config: /var/www/kippo-graph/config.php
URL: http://local-or-remote-IP-address/kippo-graph/
MySQL database: kippo
MySQL user/password: root/honeydrive

[Kippo2MySQL]
Location: /opt/kippo2mysql/
MySQL database: kippo2mysql
MySQL user/password: root/honeydrive
```

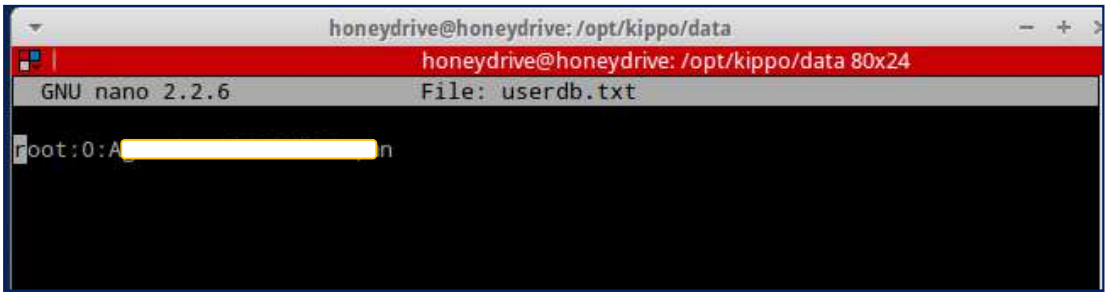
```
[Kippo-Scripts]
Location: /opt/kippo-scripts/
+ kippo-sessions
+ kippo-stats
+ kippo2wordlist
```

Vamos a ponerlo en marcha. Abrimos el Terminal que hay en el escritorio llamado *"terminator".* Lo primero que vamos a realizar es cambiar la contraseña por defecto de SSH para no dejar la proporcionada por la distribución y que el atacante no entre a las primeras de cambio. Para eso tecleamos lo siguiente para ir a la ruta donde se encuentra el fichero de configuración.

```
cd /opt/kippo/data
```

Como vemos tenemos un fichero llamado *userdb.txt.* Este lo vamos a editar para poder cambiar el pass por defecto. El nombre de usuario lo voy a dejar cómo lo tengo por defecto para no poner las cosas tampoco muy complicadas. Con una buena contraseña es suficiente siempre y cuando no pongan en marcha la honey de manera continua. En caso contrario cambiar tanto el *login* como el *pass*.

```
nano userdb.txt
```



Bien eso que veis es el login "root" y el pass"123456" es el que viene por defecto. Una vez terminado de editar pulsamos **Crt + X**, presionamos **Y** para salvarlo.

Ahora solo nos queda ponerlo en marcha. Nuevamente en la consola vamos a la ruta donde ejecutaremos el script de Kippo para ponerlo en funcionamiento y a la escucha.

```
cd /opt/kippo/
./start.sh
```

Si todo ha ido de manera correcta ya tenemos **kippo** corriendo y con el servicio **SSH** a la escucha para posibles atacantes.

```
honeydrive@honeydrive:~$ cd /opt/kippo
honeydrive@honeydrive:/opt/kippo$ ./start.sh
Starting kippo in background...Loading dblog engine: mysql
honeydrive@honeydrive:/opt/kippo$
```

Comprobamos que el servicio está a la escucha con el comando:

```
sudo netstat -atnp | grep 22
```

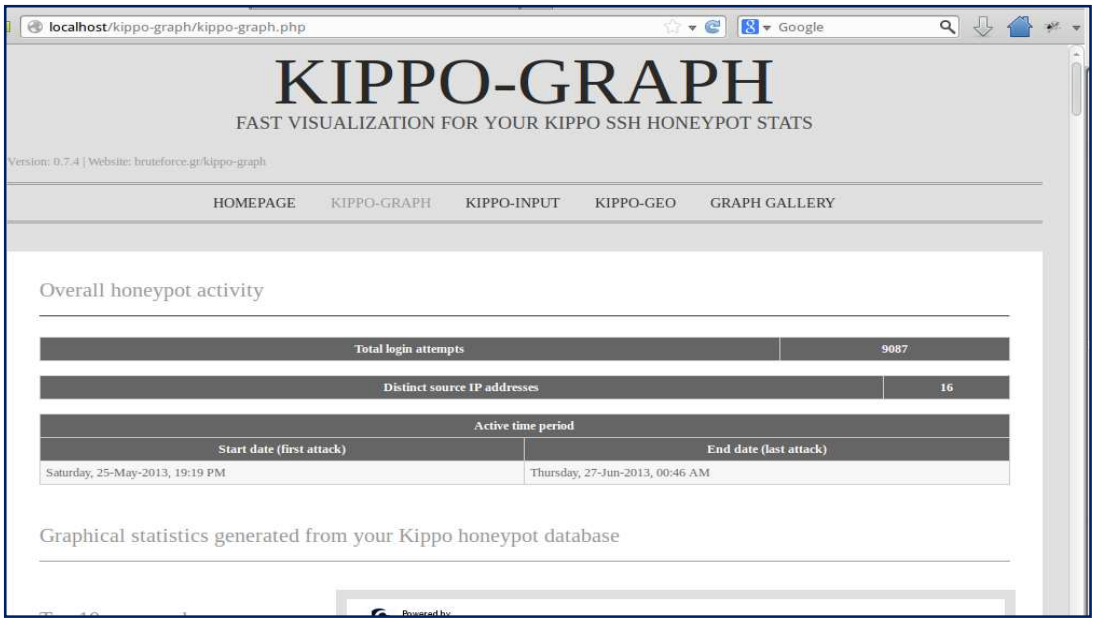
Si todo esta correcto nos saldrá lo siguiente:

```
honeydrive@honeydrive:/opt/kippo$ sudo netstat -atnp | grep 2
2
[sudo] password for honeydrive:
tcp        0      0 0.0.0.0:22              0.0.0.0:*
       LISTEN      2139/python
honeydrive@honeydrive:/opt/kippo$
```

Ahora ya lo tenemos en marcha y hay que esperar a posibles ataques. Normalmente no tardan mucho en producirse pero hay que tener paciencia. Podemos consultar la base de datos para mirar los ataques, comandos introducidos, etc. O bien podemos echar mano de lo fácil y bonito, el entorno gráfico de Kippo.

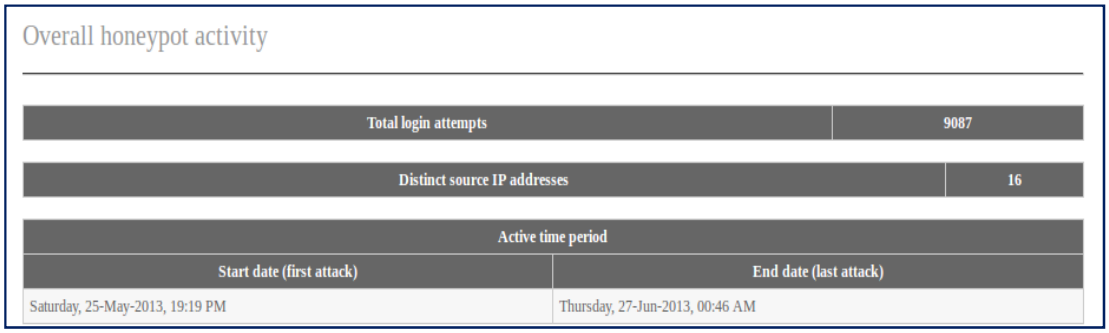
Para poder arrancarlo, simplemente se debe ingresar al navegador, según nos indica el fichero de configuración *"readme.txt"*:

http://local-or-remote-IP-address/kippo-graph/



Si abrimos KIPPO-GRAPH veremos si hemos sufrido ataques, cuando se han producido, IPs del ataque, gráficos, comandos, etc. Os dejo unas capturas de pantalla para que lo veáis.

Vemos el número total de logs que se ha producido, las diferentes IP que han intentado entrar. También observamos la fecha y hora del primer ataque registrado y del último registrado.

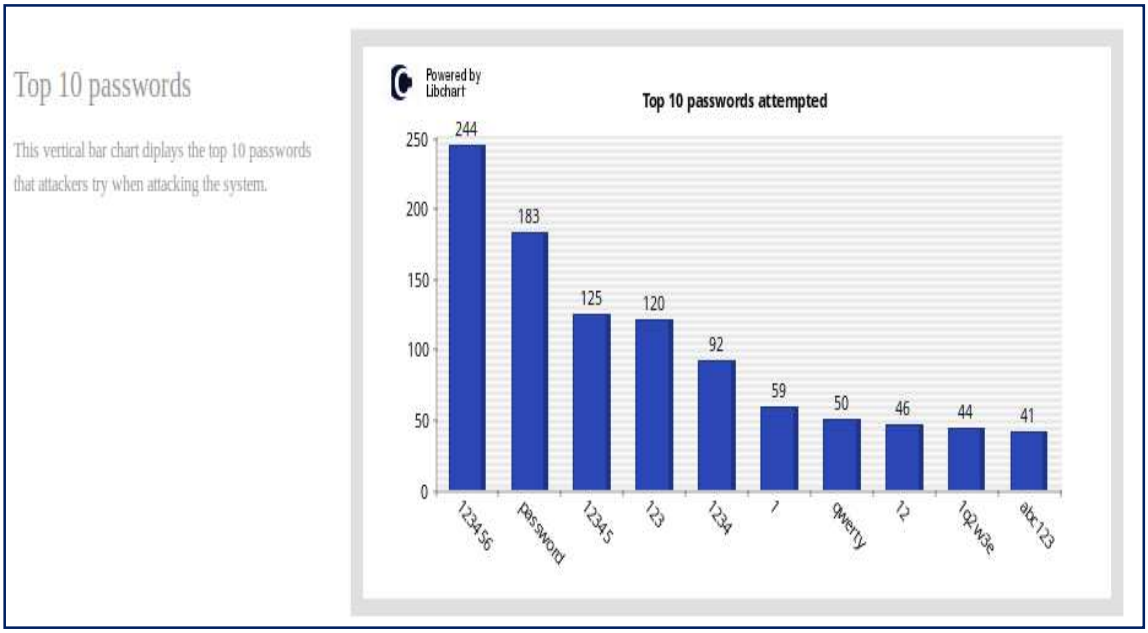


Overall honeypot activity

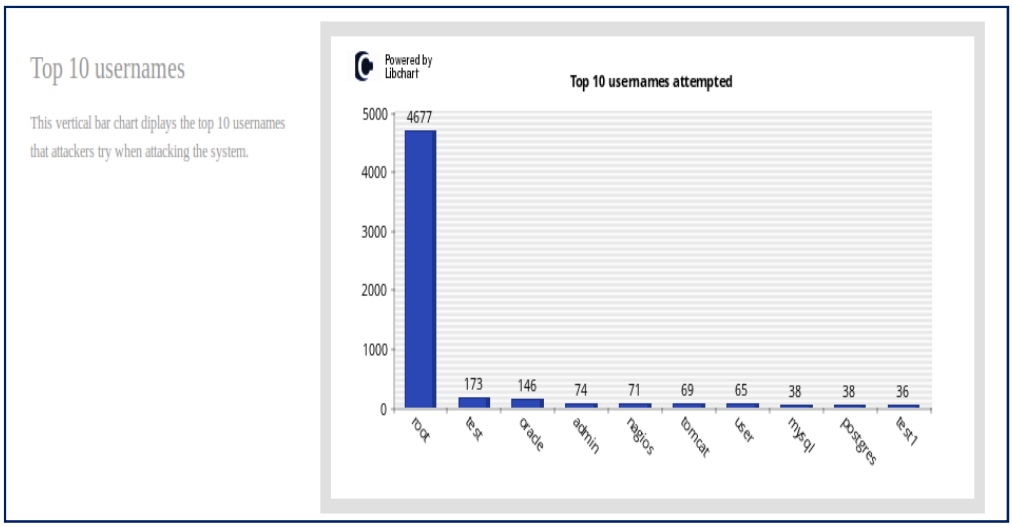
Total login attempts	9087
Distinct source IP addresses	16

Active time period	
Start date (first attack)	End date (last attack)
Saturday, 25-May-2013, 19:19 PM	Thursday, 27-Jun-2013, 00:46 AM

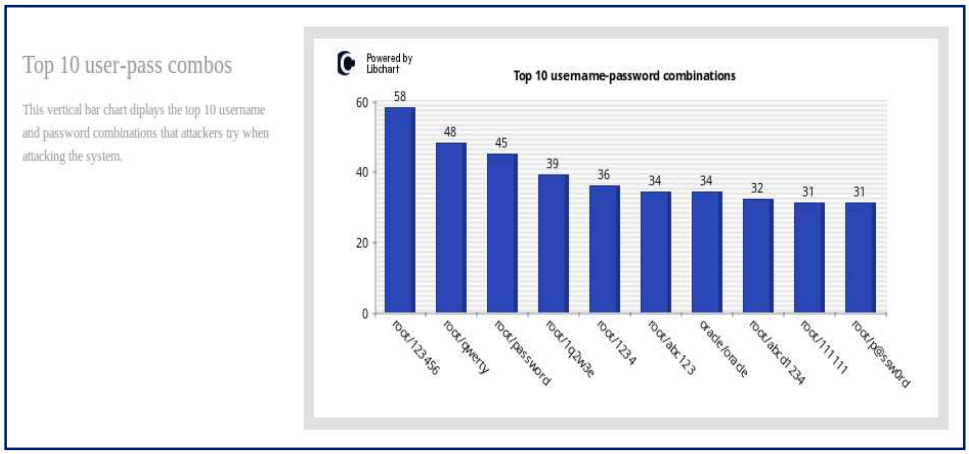
Aquí vemos el top 10 de los *passwords* introducidos para poder entrar en mi servicio SSH.

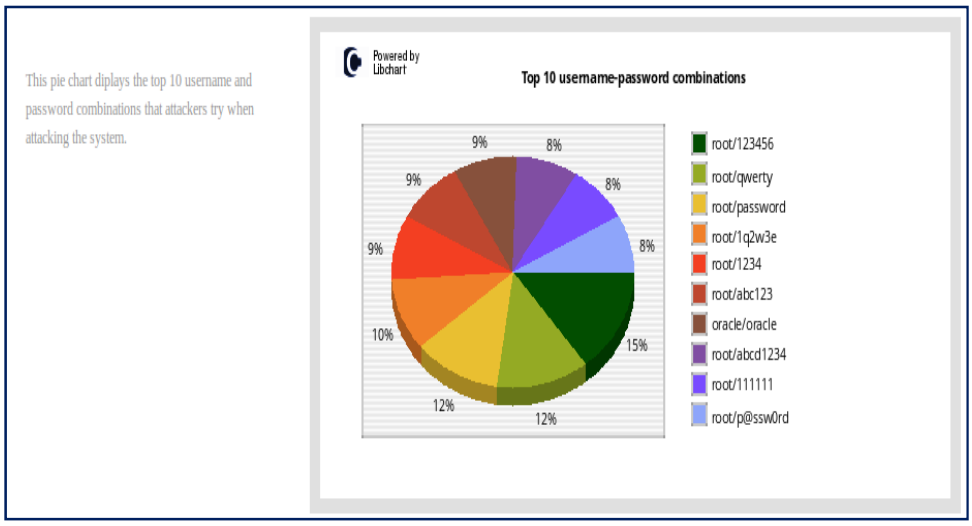


En este gráfico vemos el top 10 de los *login* realizados.

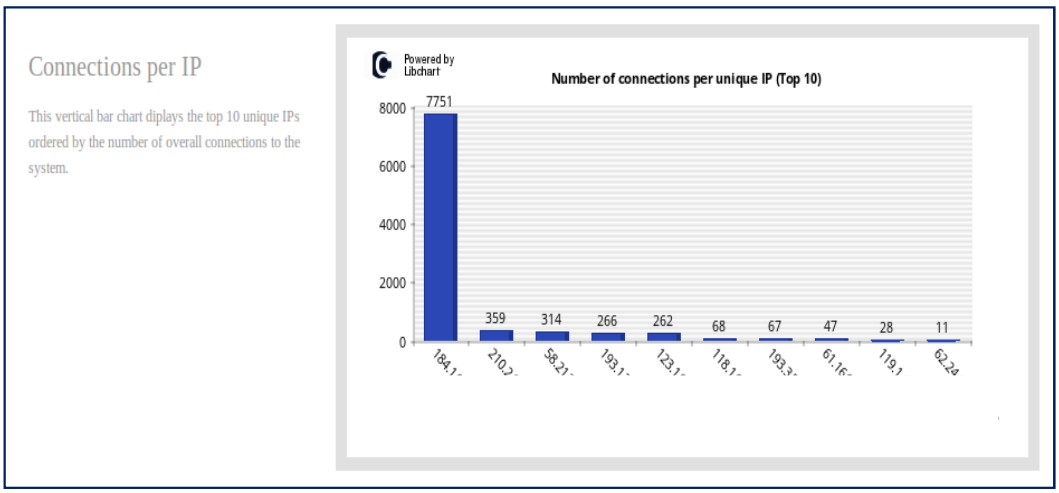


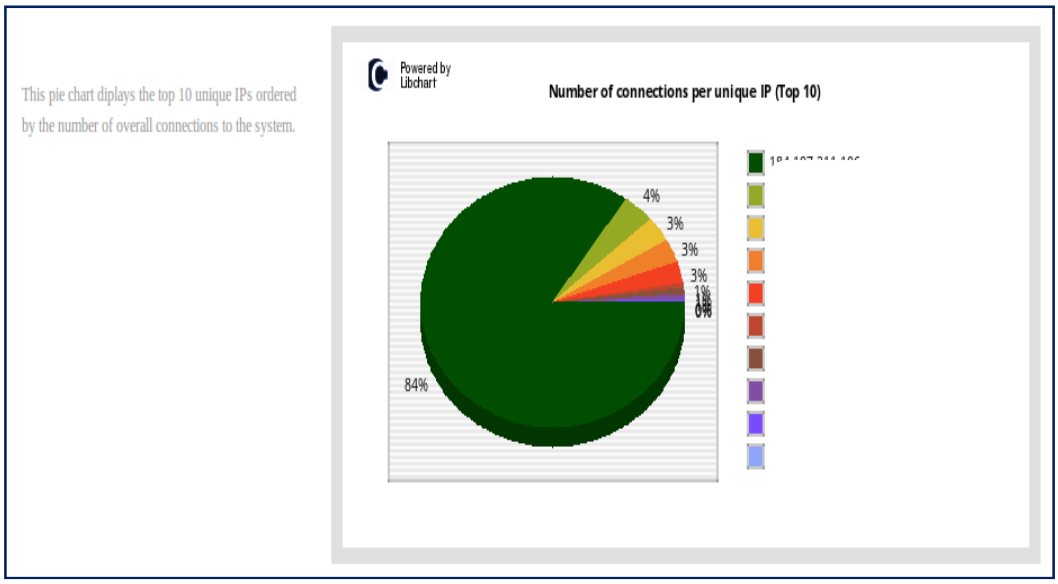
En esta otra captura vemos la combinación de *login* y *pass* utilizados.



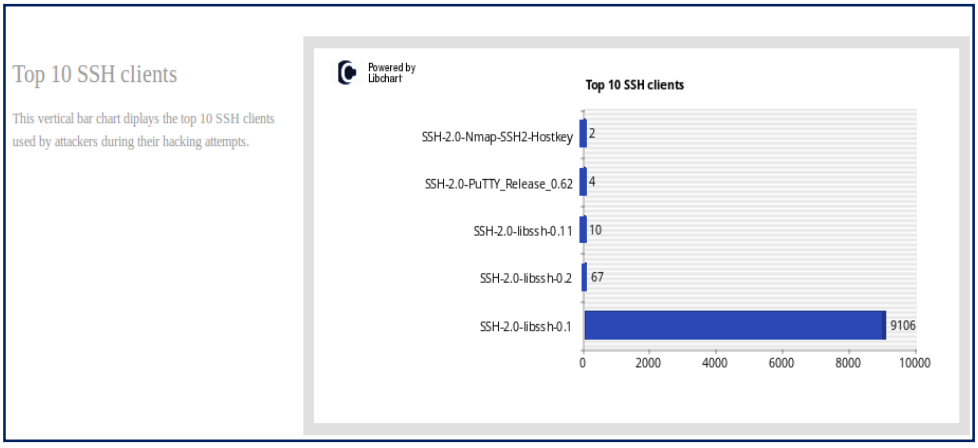


Ahora veremos las IPs que se han intentado conectar por SSH a nuestro honeypot.

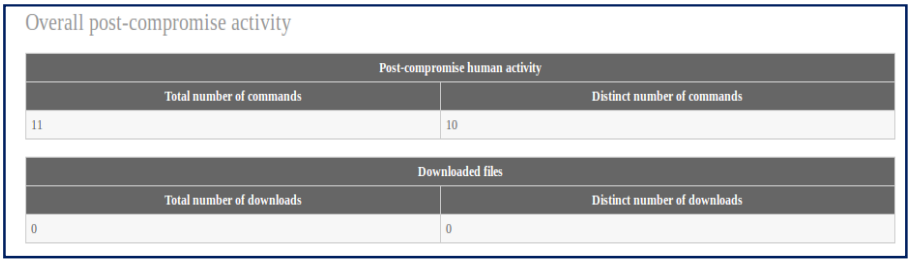




En esta otra vemos el cliente que han utilizado para intentar la conexión.



En la pestaña KIPPO-INPUT input podemos ver lo siguiente:



Overall post-compromise activity

Post-compromise human activity	
Total number of commands	Distinct number of commands
11	10

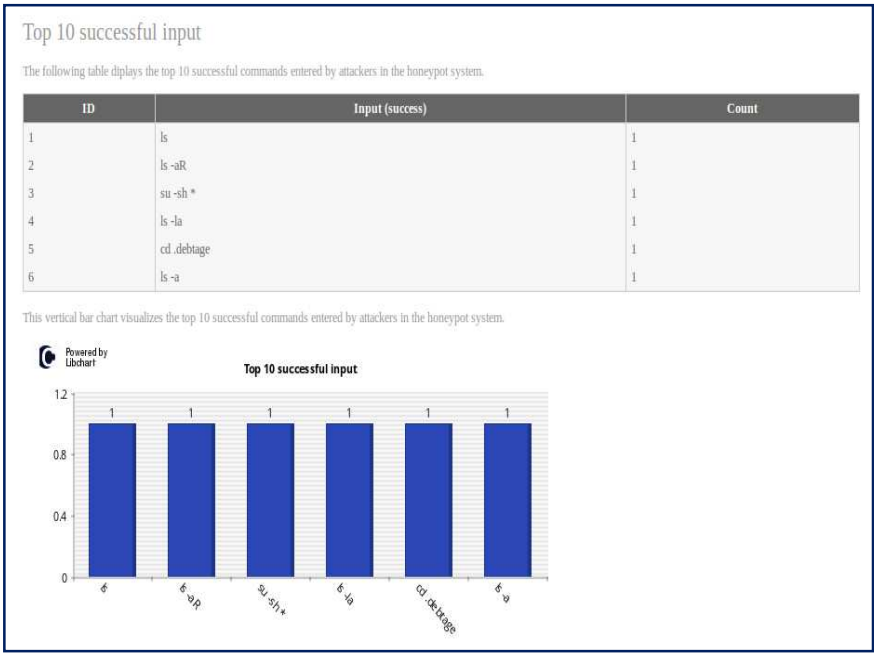
Downloaded files	
Total number of downloads	Distinct number of downloads
0	0

Top 10 input (overall)

The following table diplays the top 10 commands (overall) entered by attackers in the honeypot system.

ID	Input	Count
1	dir	2
2	ls	1
3	ls -aR	1
4	.	1
5	su -sh *	1
6	du -sh *	1
7	quit	1
8	ls -la	1
9	cd .debtage	1
10	ls -a	1

This vertical bar chart visualizes the top 10 commands (overall) entered by attackers in the honeypot system.

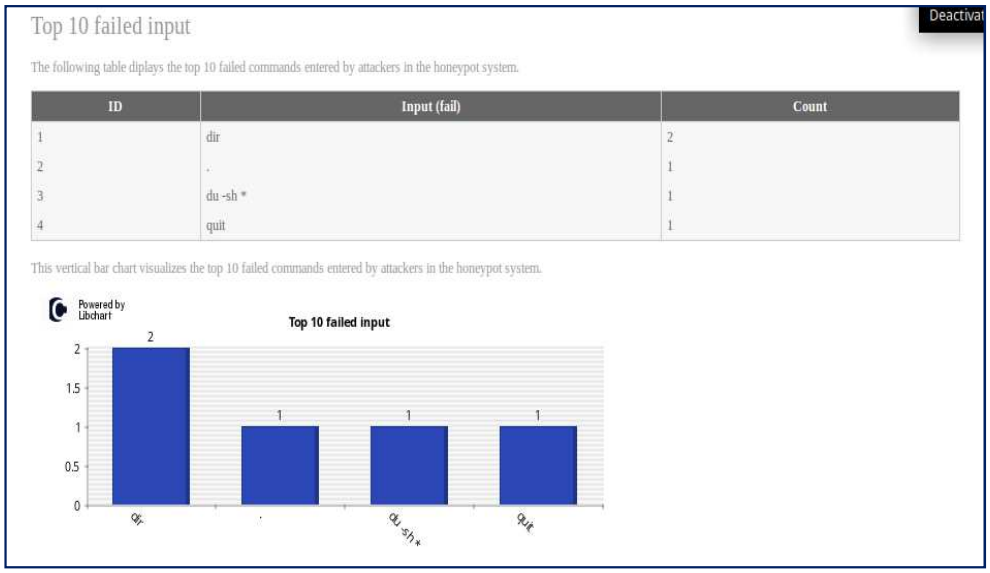


Top 10 successful input

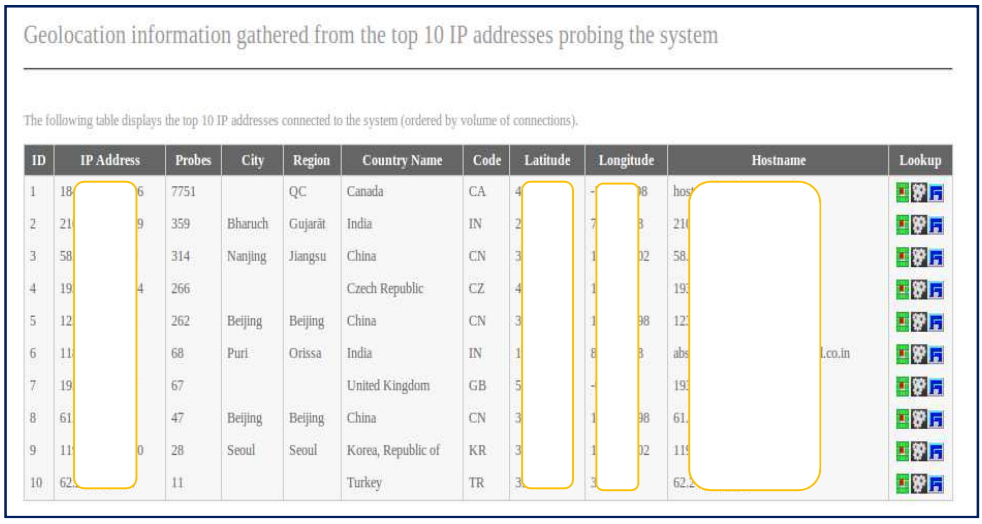
The following table diplays the top 10 successful commands entered by attackers in the honeypot system.

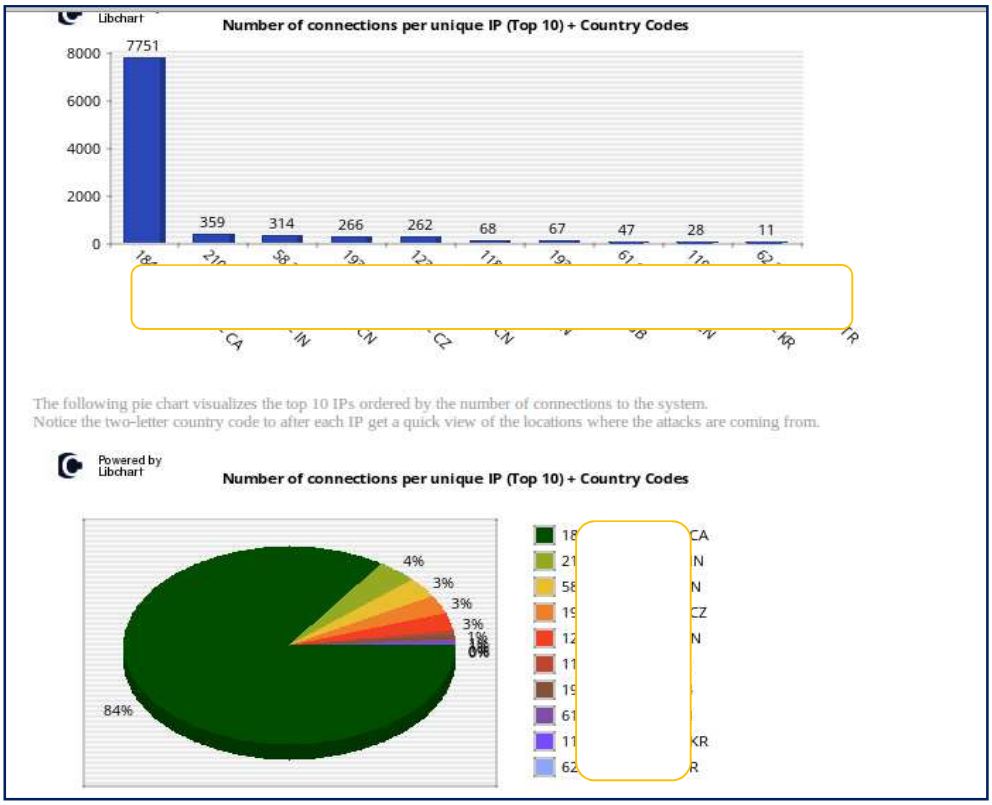
ID	Input (success)	Count
1	ls	1
2	ls -aR	1
3	su -sh *	1
4	ls -la	1
5	cd .debtage	1
6	ls -a	1

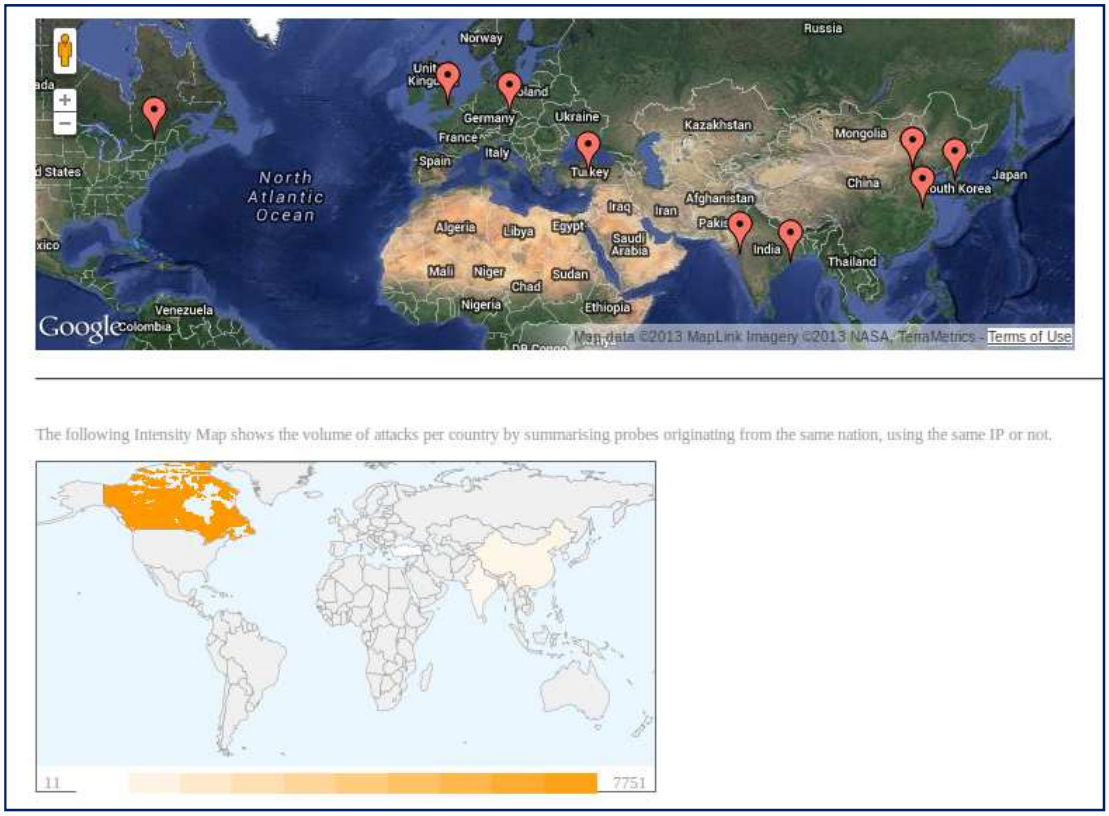
This vertical bar chart visualizes the top 10 successful commands entered by attackers in the honeypot system.



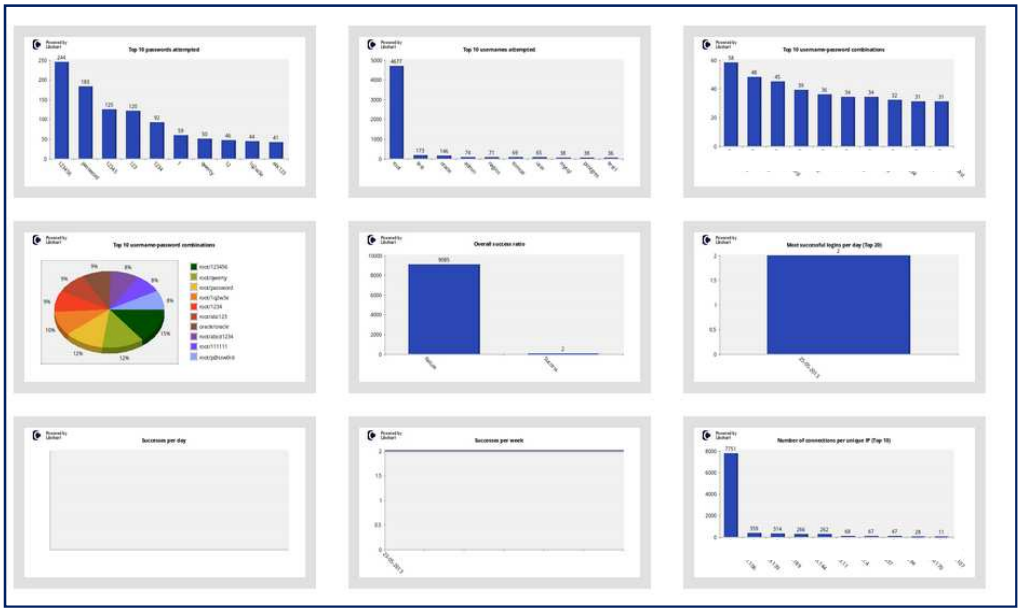
En la pestaña KIPPO-GEO tenemos la localización de procedencia de las IP y algunos datos más, muy interesante esta pantalla que además nos brinda de herramientas de localización.




Libchart
Number of connections per unique IP (Top 10) + Country Codes
8000
6000
4000
2000
0
7751
359
314
266
262
68
67
47
28
11
CA
IN
CN
CZ
CN
IR
The following pie chart visualizes the top 10 IPs ordered by the number of connections to the system.
Notice the two-letter country code to after each IP get a quick view of the locations where the attacks are coming from.
Powered by Libchart
Number of connections per unique IP (Top 10) + Country Codes
4%
3%
3%
3%
84%
CA
N
N
CZ
N
KR
R


Norway
Russia
Germany
Ukraine
Kazakhstan
Mongolia
France
Italy
Spain
Turkey
China
Japan
North Atlantic Ocean
Iraq
Afghanistan
Iran
Algeria
Libya
Egypt
Saudi Arabia
India
Thailand
Mali
Niger
Chad
Sudan
Nigeria
Ethiopia
Venezuela
Colombia
Google
Map data ©2013 MapLink Imagery ©2013 NASA, TerraMetrics
Terms of Use
The following Intensity Map shows the volume of attacks per country by summarising probes originating from the same nation, using the same IP or not.
11
7751

Por último tenemos la pestaña GRAPH GALLERY que nos muestra todos los datos en una sola ventana y podemos acceder a ellas pulsando en las mismas.



Si queremos ver los logs generados podemos ir a */opt/kippo/log* y dentro del directorio *tty* tenemos *logs* generados que nos van a servir para poder simular lo que ha realizado un atacante mediante el script *playlog.py* que se encuentra en la ruta */opt/kippo/utils.* Nos fijamos en el nombre de los logs y vamos a la ruta mencionada para ejecutar el script.

```
sudo python playlog.py -b -m 2
/opt/kippo/log/tty/20130525-214559-6881.log 0
```

Vemos como nos va saliendo todo lo realizado por el intruso.

```
honeydrive@honeydrive:/opt/kippo/utils$ sudo python playlog.p
y -b -m 2 /opt/kippo/log/tty/20130525-214559-6881.log 0
webserver:~# llss  --llaa
drwxr-xr-x 1 root root 4096 2013-05-25 21:46 .
drwxr-xr-x 1 root root 4096 2013-05-25 21:46 ..
drwxr-xr-x 1 root root 4096 2009-11-06 11:16 .debtags
-rw------- 1 root root 5515 2009-11-20 09:08 .viminfo
drwx------ 1 root root 4096 2009-11-06 11:13 .aptitude
-rw-r--r-- 1 root root  140 2009-11-06 11:09 .profile
-rw-r--r-- 1 root root  412 2009-11-06 11:09 .bashrc
webserver:~# ccdd  ..ddeebbttaaggee
bash: cd: .debtage: No such file or directory
webserver:~# ddiirr
bash: dir: command not found
webserver:~# llss  --aa
.       ..       .debtags  .viminfo  .aptitude .profile  .
bashrc
webserver:~# honeydrive@honeydrive:/opt/kippo/utils$
```

En último lugar, podemos ver el log que se genera en tiempo real. Para esto nos situamos en el directorio donde se guardan los *logs*:

```
tail -f kippo.log
```

Se podría decir muchas más cosas de Kippo pero creo que lo mencionado hasta aquí es suficiente para que podáis vosotros mismos ponerlo en funcionamiento y comprobar todas sus posibilidades.

Dionaea

Este honeypot me ha parecido fascinante. Vamos a ponerlo en funcionamiento y se podrá ver la cantidad de escaneos y ataque que sufrimos al conectarnos a Internet.

Dionaea está diseñado principalmente para recoger malware a través de las vulnerabilidades de seguridad que ofrece. Gracias a esto un atacante nos descargará malware pensando que de esa manera el ordenador está bajo su poder y utilizarlo para el fin que el tenga pensado. Dioanea está compuesto de módulos que emulan a protocolos.

Prácticamente se pueden emular todos los que queramos aunque ya viene bien configurado para este aspecto. Por ejemplo el protocolo SMB está activo para que sufra ataques. Los gusanos existentes Internet suelen atacar este protocolo al igual que los. Otro de los módulos activos es el de SIP que es capaz de establecer sesiones.

Otra de las características de Dionaea es su capacidad de escuchar en varias interfaces de red y recoge información de muchas IPs de forma simultánea.

Gracias a sus características y a las vulnerabilidades expuestas podemos observar los *exploits* que han utilizado, analizar el código introducido en una *shell*, recuperación de binarios y por supuesto analizar los logs que nos van a revelar mucha información.

Tenemos el registro detallado de los ataques y esto incluye cualquier acción que se puedan realizar en el sistema. El fin último para lo que está programado este honeypot tan fantástico es el de la obtención de malware para su posterior análisis.

Entre los servicios vulnerables que mantiene a la espera para ser atacado nos encontramos con SMB, SIP, MYSQL, Ftpd, epmapper, etc. Utiliza para almacenar todos estos resultados una base de datos SQLite. Otra de sus características es que es capaz de almacenar el *login* y *pass* que el atacante ha utilizado para intentar obtener un acceso.

Ante todo el archivo de configuración:

```
[Dionaea]
Location: /opt/dionaea/
Bin: /opt/dionaea/bin/dionaea
Config: /opt/dionaea/etc/dionaea/dionaea.conf
Logs: /opt/dionaea/var/log/
SQLite database: /opt/dionaea/var/dionaea/logsql.sqlite
Malware samples: /opt/dionaea/var/dionaea/binaries/
+ phpLiteAdmin: /var/www/phpliteadmin,
+ password: honeydrive,
+ URL: http://local-or-remote-IP-
address/phpliteadmin/phpliteadmin.php
```

```
[DionaeaFR]
Location: /opt/dionaeaFR/
Script: /opt/dionaeaFR/manage.py
[Dionaea-Scripts]
Location: /opt/dionaea-scripts/
+ mimic-nepstats
+ dionaea-sqlquery
```

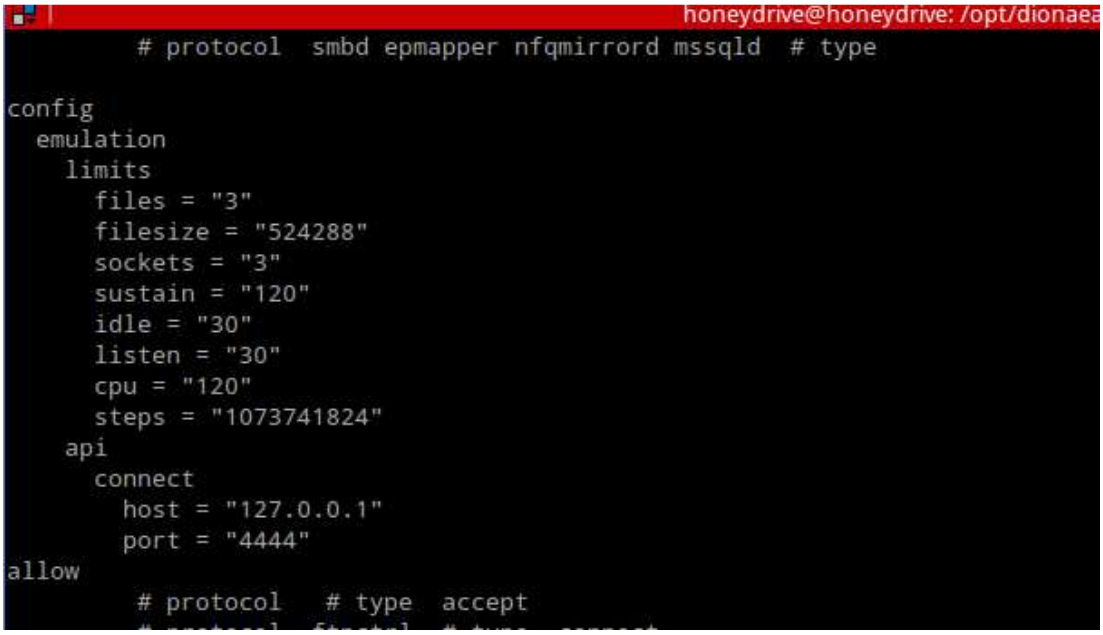
Para ponerlo en marcha vamos a una shell en nuestra distribución. Vamos a ir a la ruta donde se encuentra el binario para poner en marcha el honeypot.

```
/opt/dionaea/bin
```

Una vez en este directorio pasamos a ejecutar el *script* para cazar todo tipo de ataques y malware.

```
./dionaea -l all,-debug -L '*'
```

Si todo ha funcionado bien, se ejecuta el script y se verá cómo va cargando el fichero de configuración y abriendo los puertos que emulan los servicios.



Ya tenemos nuestro honeypot corriendo, ahora queremos visualizar todo lo que pasa ¿verdad? Para esto se debe poner en marcha el fabuloso entorno gráfico que se llama **DionaeaFR**. Para eso vamos a otra *shell* y nos vamos a la ruta donde está el *script* que lanza este entorno y así poder verlo desde un navegador.

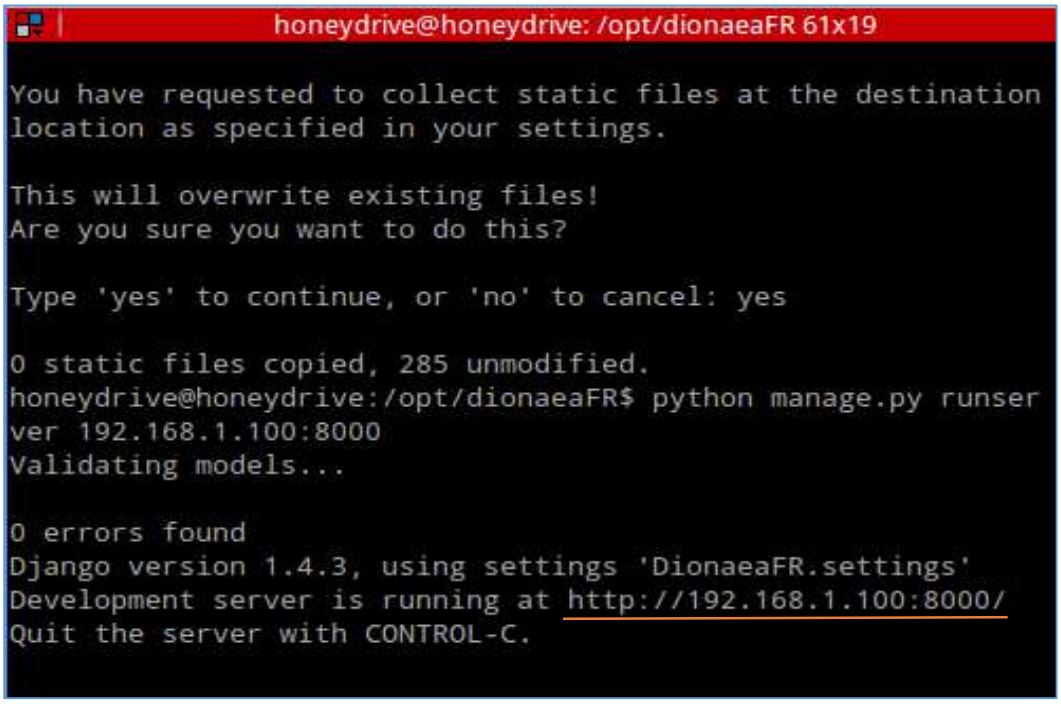
```
/opt/dionaeaFR
```

Ahora ejecutamos los siguientes scripts para poder ponerlo en marcha.

```
python manage.py collectstatic
```

Nos pedirá que confirmemos con *"Yes"* para importar todos los registros. Ahora ejecutamos el siguiente script que nos va a lanzar el entorno gráfico y la ruta con el puerto que queramos.

```
python manage.py runserver 192.168.1.100:8000
```

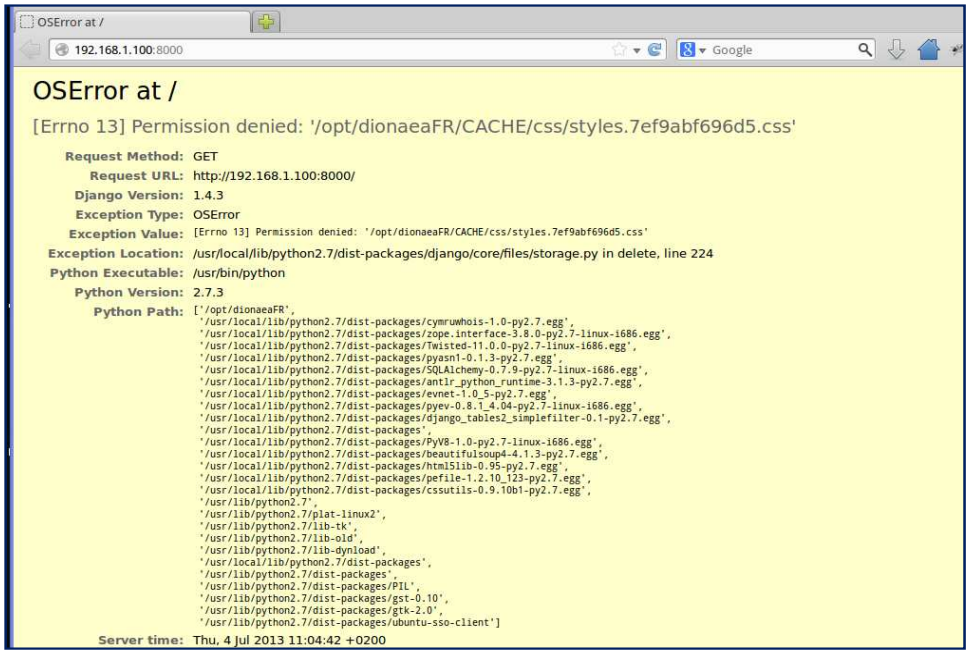


Pues ya tenemos nuestro honeypot en marcha, ahora abrimos en el navegador la dirección que hemos puesto con ese puerto y se nos abrirá la visualización:

http://192.168.1.100:8000

Ups ¿qué es esto?

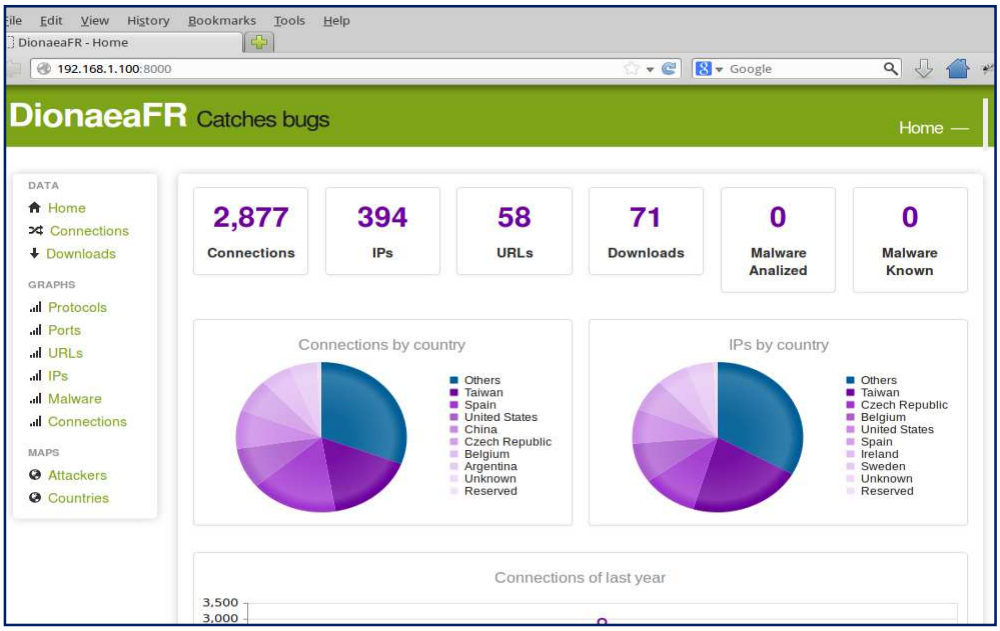
Bien hasta aquí quería llegar. **Parece ser que necesitamos ser root para poder visualizar el entorno gráfico de Dionaea.**



Vamos a ejecutar los scripts como administradores esta vez.

```
sudo python manage.py collectstatic
sudo python manage.py runserver 192.168.1.100:8000
```

Vamos a ver qué pasa ahora.

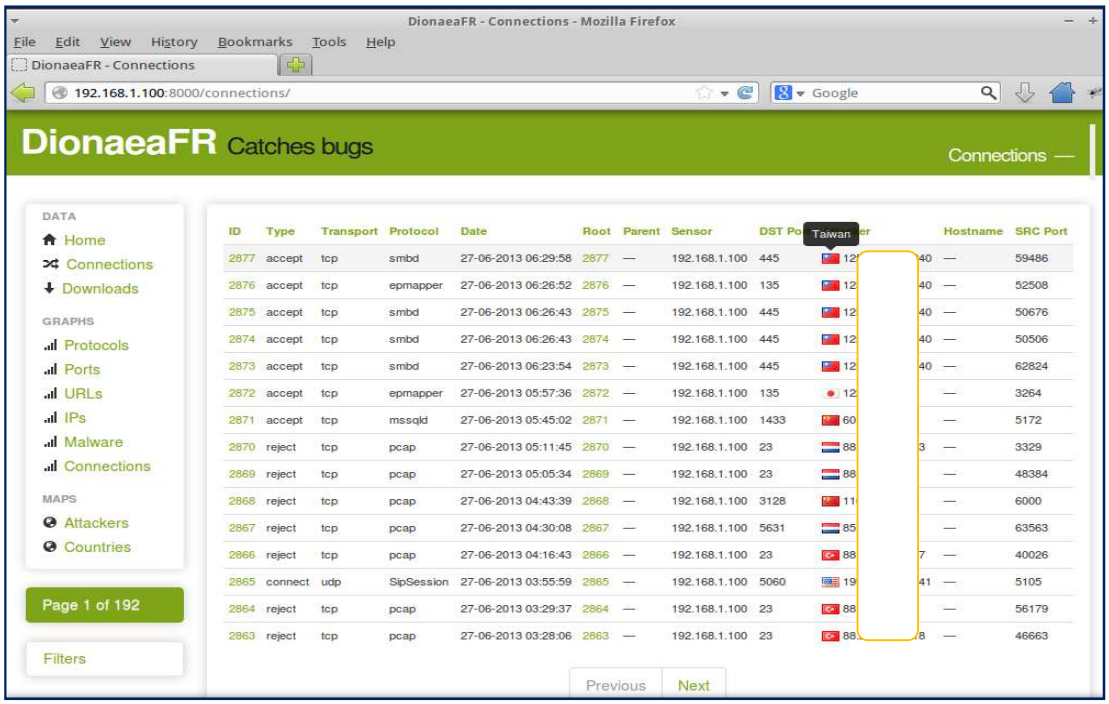


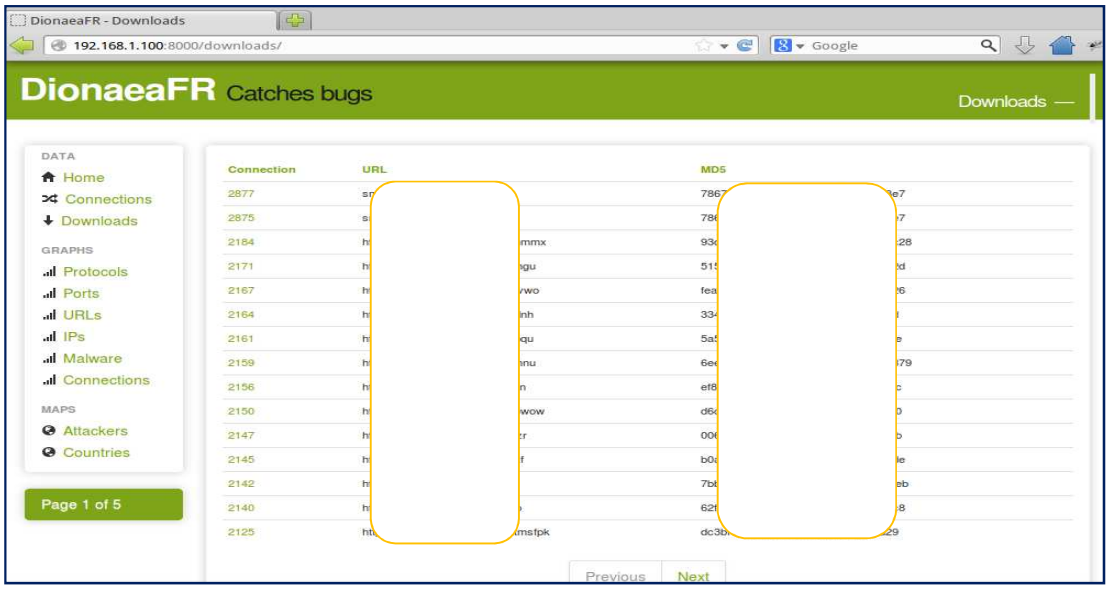
Parece que ahora sí que nos deja ver el entorno gráfico de Dionaea. Cómo veis está muy logrado y es muy fácil de interpretar que es en verdad lo importante para poder entender de una manera rápida lo que está sucediendo. A vosotros todos estos valores os saldrán a cero por supuesto.

En esta pantalla, de manera intuitiva y rápida podemos ver:

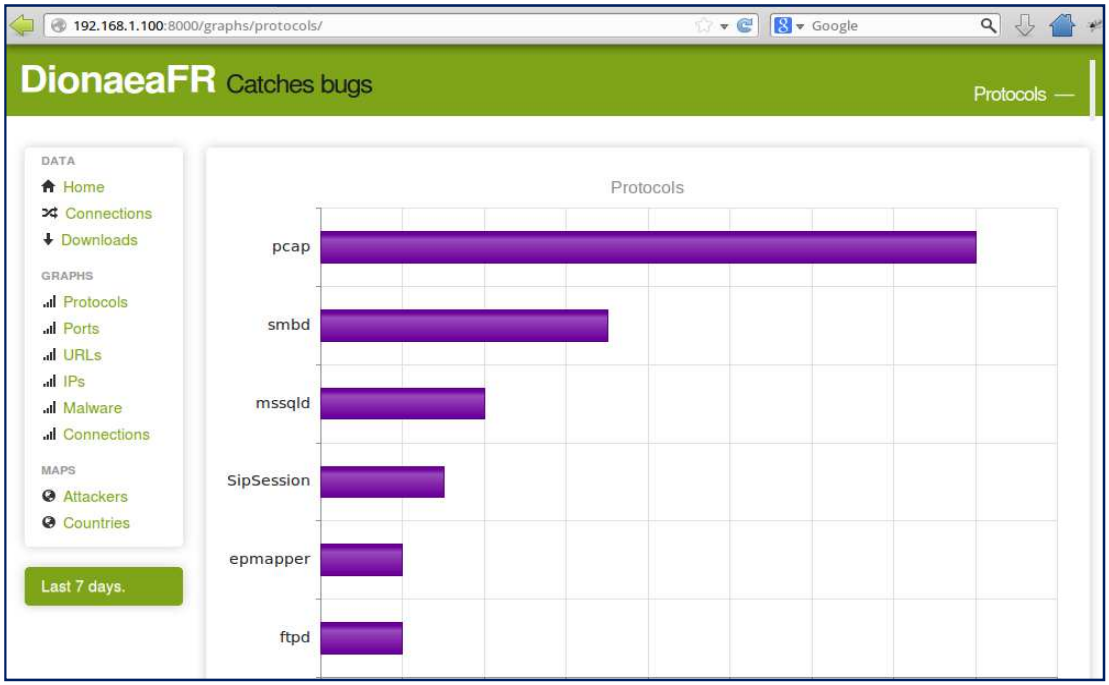
- Conexiones realizadas a nuestro honeypot.
- IPs diferentes que han realizado esas conexiones.
- URLs que nos han atacado.
- Binarios que nos han descargado y hemos capturado.
- Malware analizado.
- Malware desconocido.
- Conexiones por país = 0
- IPs por país = 0

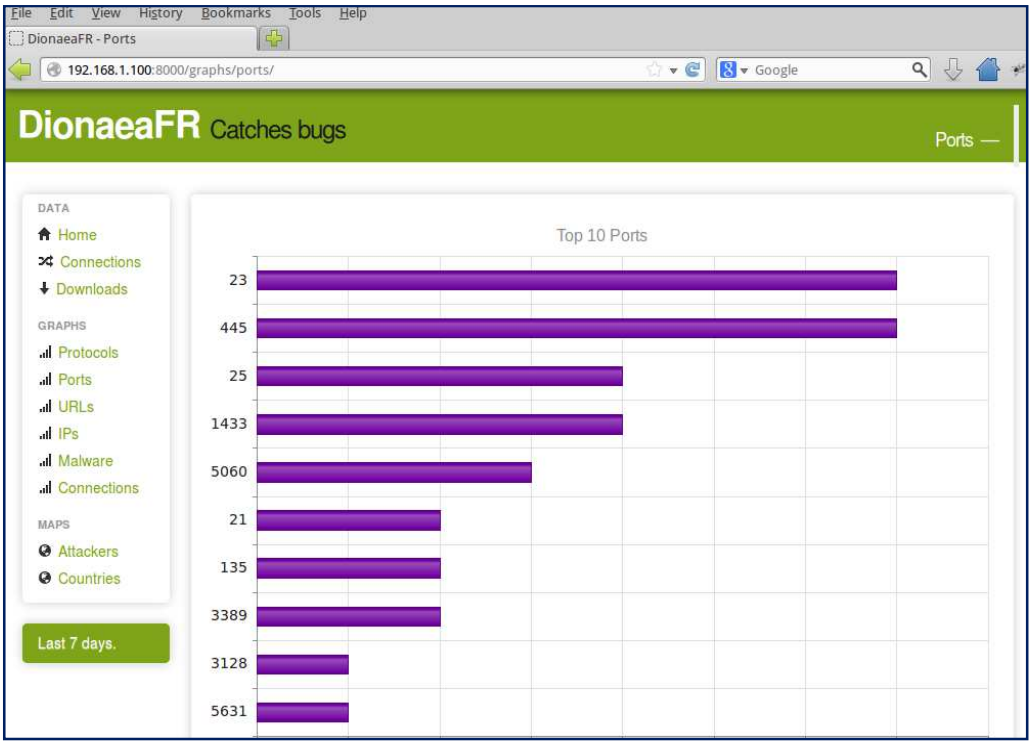
Estos dos últimos valores que se ven con valor cero están así de manera predeterminada y no he utilizado API para subir las muestras a Virustotal para analizarlos. Ahora, vamos a seguir viendo resultados obtenidos con este magnífico honeypot. Creo que para entender esta pantalla no hace falta ninguna explicación.



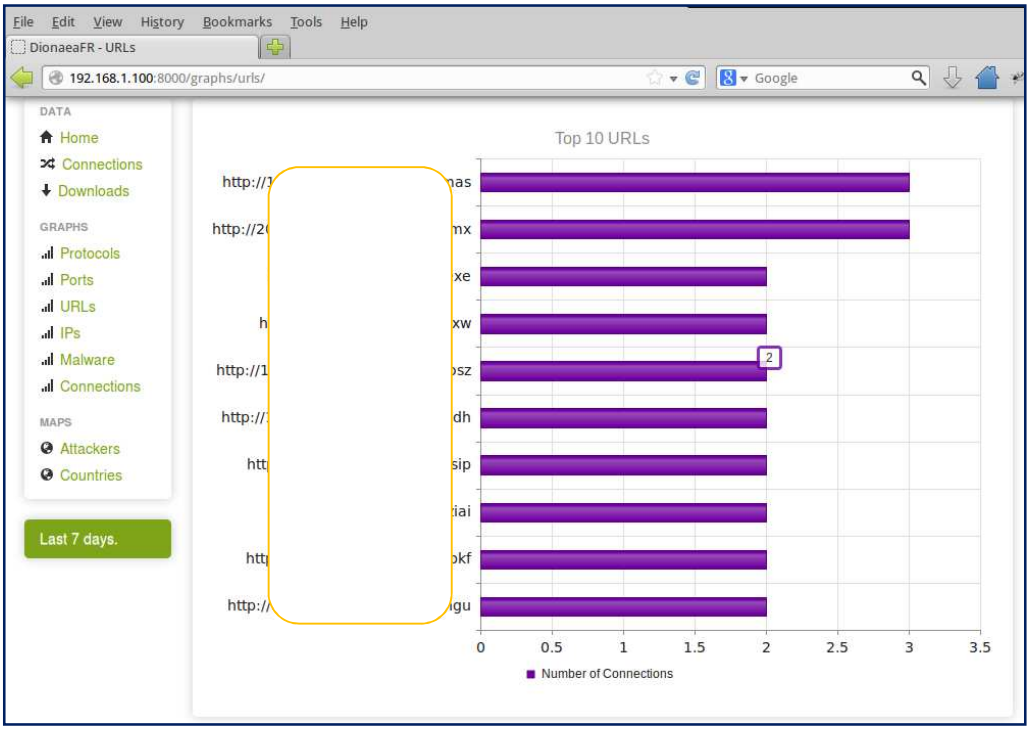


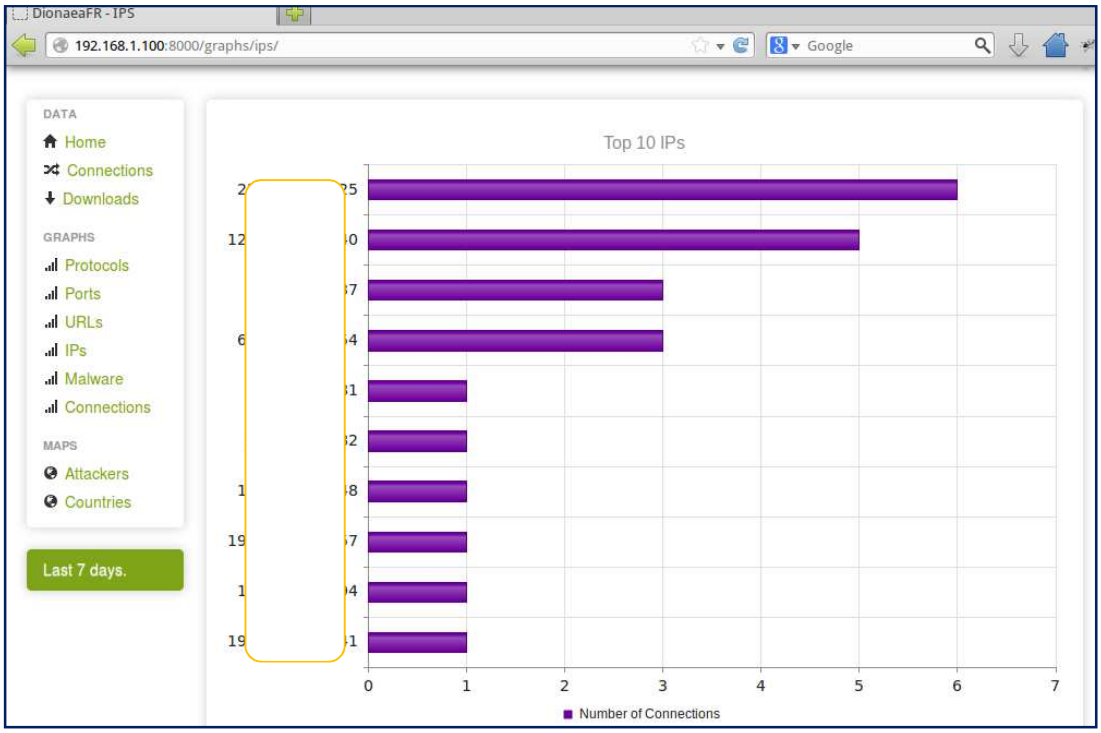
Ahora vemos los protocolos y los puertos que más se han utilizado a la hora de atacar el honeypot.



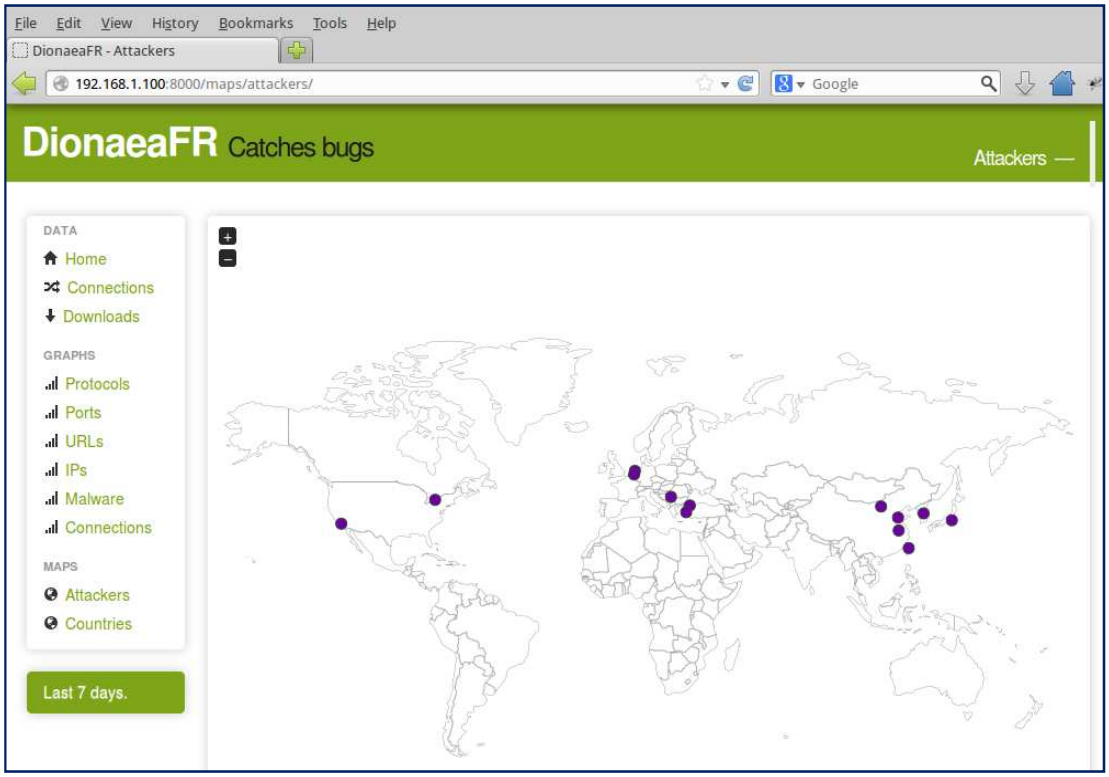


A continuación aparecen las URL e IPs de los ataques:

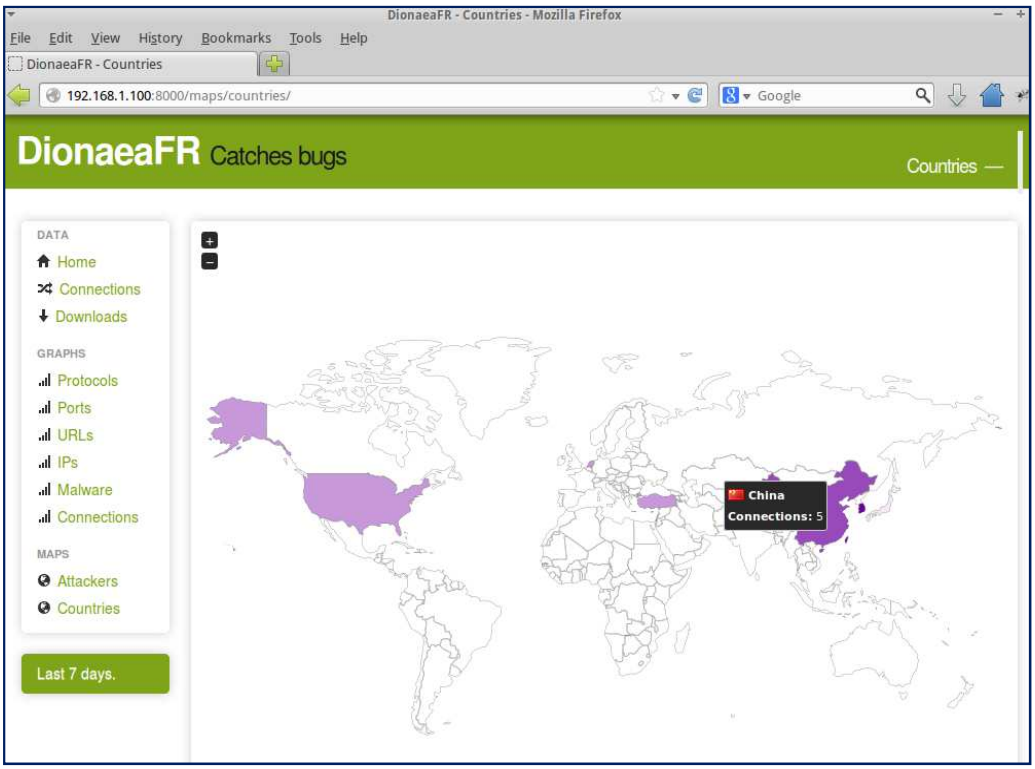




La siguiente pantalla nos informa la localización de los ataques. El mapa no está tan logrado cómo en Kippo pero es más que suficiente.



Vemos el mismo mapa pero con países específicos.

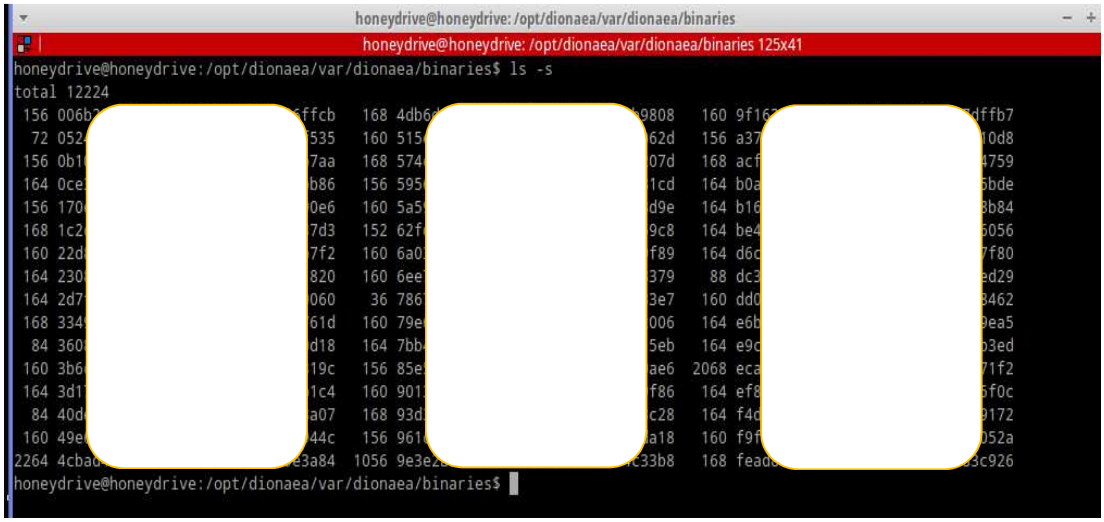


En cuanto al entorno gráfico ya hemos visto todo lo que nos ofrece. Como veis está muy detallado y no hay problemas para interpretar los datos que nos muestra.

Ahora si queremos ver los binarios descargados nos vamos a la ruta:

`/opt/dionaea/var/dionaea/binaries/`

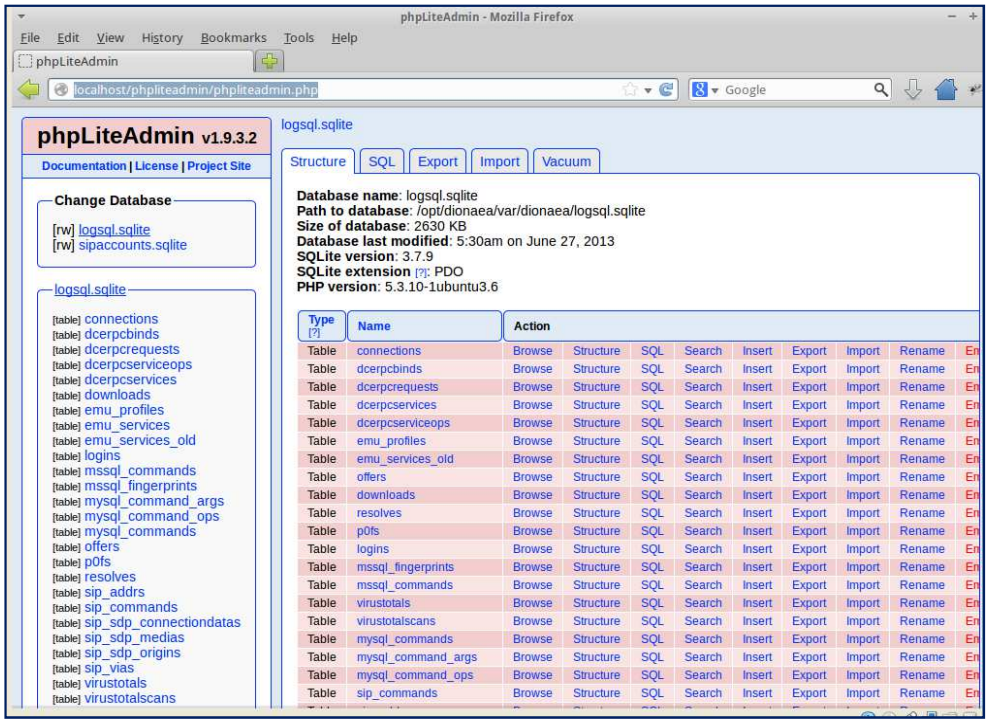
Si hacemos un listado de lo que tenemos dentro veremos los binarios descargados:



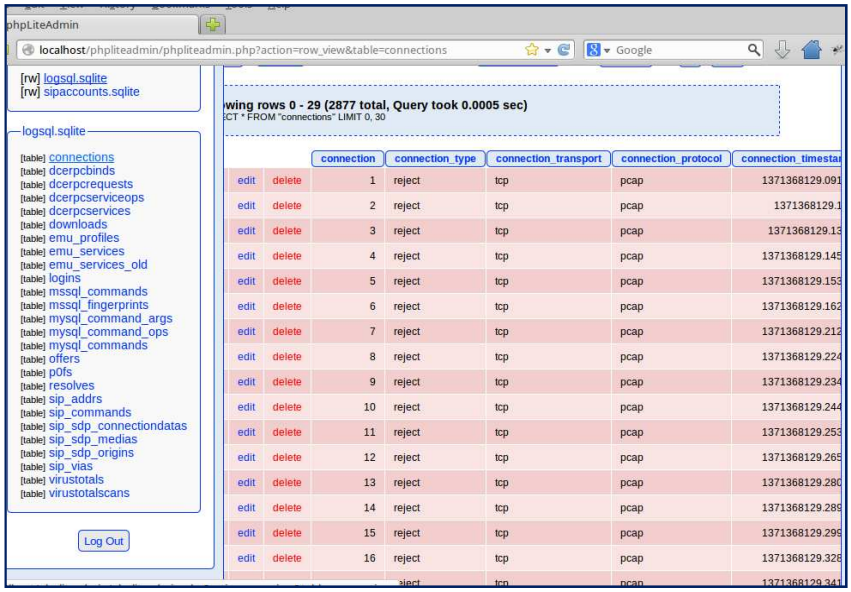
También podemos ver lo mismo mediante la base de datos:

http://localhost/phpliteadmin/phpliteadmin.php

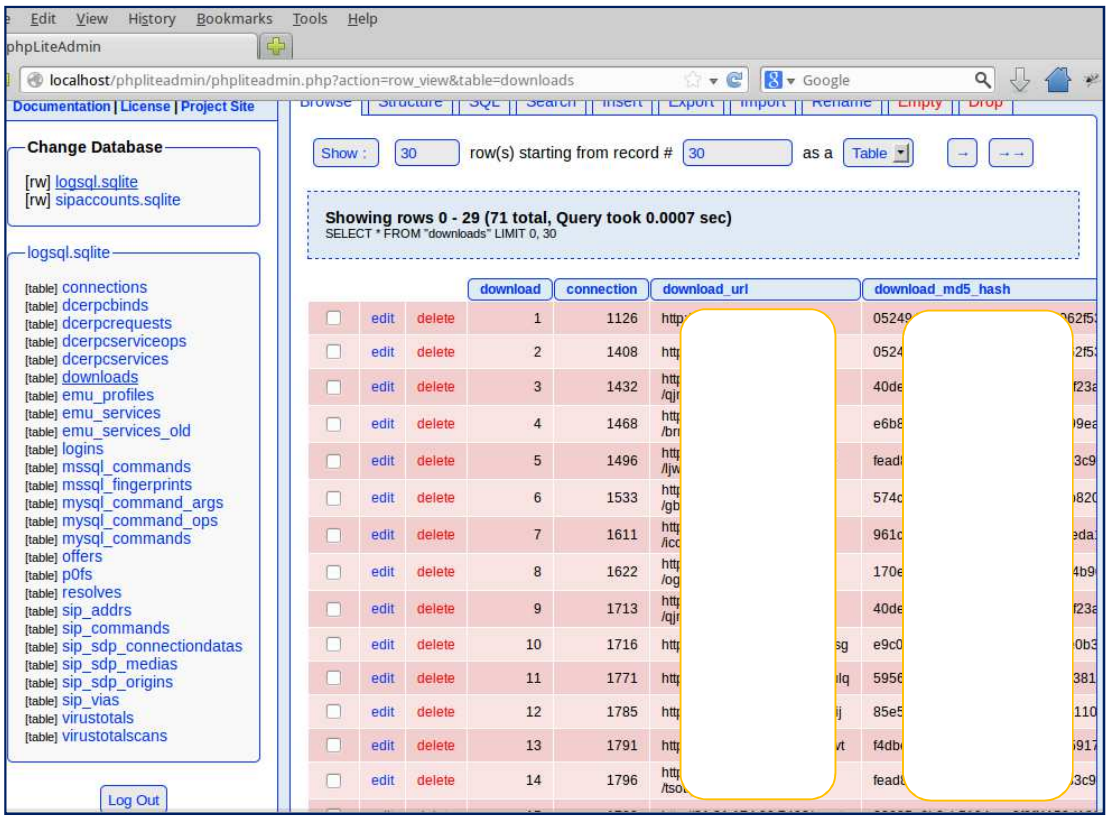
Ahora veréis la base de datos y desde esta pantalla podemos interactuar con ella. Podemos borrar, vaciar, exportar resultados.



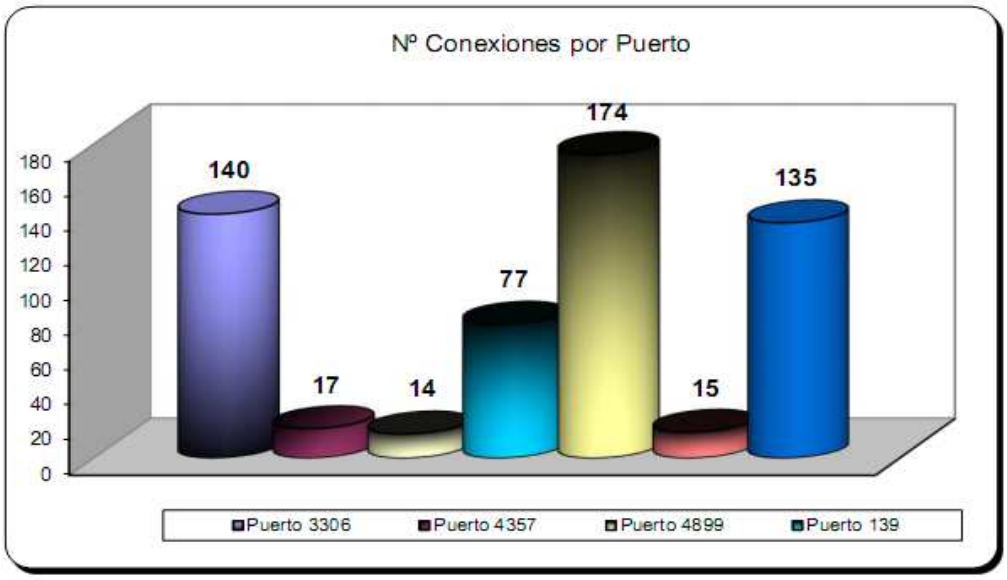
Aquí tenemos por ejemplo la tabla donde se almacenan las conexiones que luego **DioaneaFr** nos muestra.

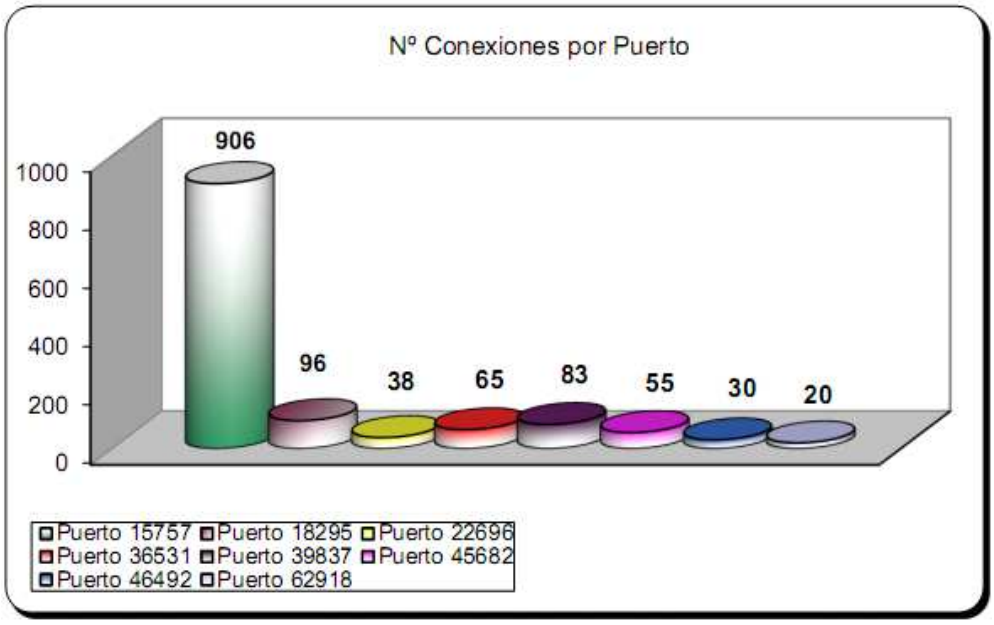


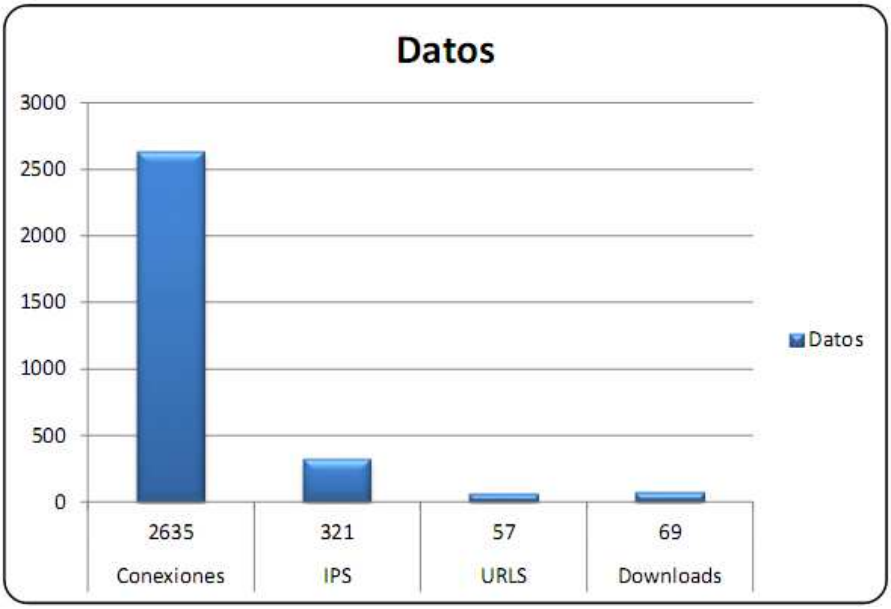
Aquí la tabla que contiene el malware recogido.



Ahora muestro las estadísticas que estoy realizando en Excel para tener un mejor control de todos los datos que recogemos y tener una mejor visión a la hora de su análisis.







Ahora, solo queda mirar un poco por vuestra cuenta y mirar todos los datos que recogemos y se almacenan en esta base de datos.