# Honeynet para el análisis del tráfico y muestras de malware

Santiago de Diego de Diego
Estudiante de doble grado de matemáticas e ingeniería informática.
Gustavo Romero López
Departamento de Arquitectura y Tecnología de los Computadores
Universidad de Granada
santidediego@gmail.com
gustavo@ugr.es

**Resumen.** En este proyecto nos hemos propuesto desplegar varios honeypots en dos dispositivos del tipo Raspberry Pi a fin de analizar ataques dirigidos a la red de la UGR. Presentamos a continuación un breve resumen del experimento. Por un lado hemos obtenido resultados de un honeypot de tipo Kippo, relacionados con ataques del tipo fuerza bruta, procedentes de varias direcciones IP, la mayoría de ellas de la zona de Asia. Además mostraremos los resultados del análisis de las muestras de malware obtenidas mediante Kippo. Por otro lado tenemos analizaremos los resultados obtenidos sobre ataques de tipo web recibidos por otro honeypot de baja interacción, Glastopf. No obstante, el principal objetivo del proyecto es identificar y clasificar diferentes muestras de malware así como proporcionar al lector una receta para este fin.

**Palabras Clave:** seguridad, honeypots, honeynet, malware, análisis del tráfico

**Abstract.** In this project we are about to deploy several honeypots in two Raspberry PI devices in order to analyze attacks directed to the UGR network. We present here a brief resume of the results of the experiment. On the one hand, we have results from a Kippo honeypot related to brute force attacks from several IP directions, most of them coming from Asia. In addition we show the results of a malware analysis of samples obtained from Kippo. On the other hand, we will analyse results related to web attacks with another low/medium interaction honeypot, Glastopf. In this particular project, the main purpose is to identify and classify several samples of malware as well as to show to the reader a recipe to achieve this goal.

**Keywords:** Security, honeypots, honeynet, malware, traffic analysis

# 1 Introducción

Antes del surgimiento de los honeypots, la seguridad era principalmente defensiva y se basaba en técnicas para parar a los atacantes. Existía mucha información sobre los elementos de un ataque tales como exploits, metodología del ataque… pero poca sobre los atacantes en sí. Por aquel entonces los administradores de sistemas no tenían ni tiempo ni los recursos necesarios para analizar todos los ataques.

Alrededor de 1999 la situación empezó a cambiar y se comenzó a estudiar también a los atacantes. Se desarrollaron los primeros dispositivos que se centraban la monitorización de sistemas, lo que dio lugar al *Honeypot Project* [1]. Algunos trabajos, [2] y [3], utilizan análisis de componentes principales para intentar caracterizar a los atacantes, mientras que otros, [4], se centran en elaborar una clasificación de atacantes según su comportamiento. Al principio los honeypots eran sistemas reales pero gracias a la virtualización, a lo largo de los años se fueron implementando soluciones más flexibles.

Actualmente, constituyen una solución ampliamente utilizada para el análisis de los diversos factores que componen un ataque. En [5] y [6] se centran en el análisis del malware, el primero de ellos proporciona un procedimiento genérico empleando análisis estático y dinámico para lidiar con esta problemática, del cual se han aplicado algunas nociones para realizar el presente trabajo. Otros como [7] se centran en explicar las técnicas más comunes empleadas para la ofuscación del malware, a fin de dificultar su desensamblado.

Según su interacción con el usuario podemos clasificarlos en honeypots de **baja, media** o **alta interacción.** Los más interesantes de cara a la obtención de información son los últimos, pero también son los más complicados de implementar. Además las posiciones donde podemos colocar un honeypot son:

- **Detrás del firewall:** en esta posición se encuentra protegido por las reglas de filtrado del firewall y por tanto deberemos configurar este para que no bloquee ataques del exterior. Tiene la ventaja de que permite detectar ataques internos, además de la posibilidad de comprometer la red interna.
- **Delante del firewall:** en esta posición se encuentra expuesto directamente a internet por lo que no es necesario configurar el firewall. Por contra, tiene la desventaja de que no permite detectar ataques internos.
- **En una zona desmilitarizada:** el honeypot se encuentra en una zona donde se encuentran los servidores pero separada de la red interna. De esta forma permite recibir ataques tanto internos como externos sin comprometer la red interna. Tiene la desventaja de que será necesario también configurar el firewall.

## 2 Escenario

Para dicha investigación se han desplegado dos honepots de baja/media interacción (Kippo y Glastopf) en una posición delante del firewall a fin de poder recibir ataques procedentes del exterior. Para este propósito se han empleado dos dispositivos del tipo Raspberry Pi 3, uno de ellos con sistema operativo Raspbian y el otro con sistema operativo Honeeepi, una distribución especial que viene con varios honeypots preinstalados, lo cual facilita enormemente la labor de configuración. Se ha prestado especial atención a las medidas de seguridad de la honeynet a fin de no comprometer la red de la UGR, ya que al ser dispositivos vulnerables por definición y además el objetivo es que sean atacados, este punto es crucial.

Para el análisis de las muestras de malware se han empleado máquinas virtuales de las mismas características que el objetivo de las muestras, las cuales veremos en la sección correspondiente. Posteriormente, se ha procedido a la eliminación de dichas máquinas virtuales.

## 3 Resultados generales

En este momento se han recibido **29342** ataques dirigidos al puerto 22, todos ellos con patrones muy similares (ver Fig. 1). La mayoría de ellos proceden de China, y en menor medida de Polonia, Vietnam y Holanda. Sin embargo, a pesar de la enorme tasa de ataques, resulta sorprendente la poca tasa de éxito de los mismos, ya que solamente el **12.62%** de los mismos ha resultado existoso. Este dato resulta aún más sorprendente si tenemos en cuenta que se han empleado credenciales de acceso realmente sencillas (admin-admin, root-root…).

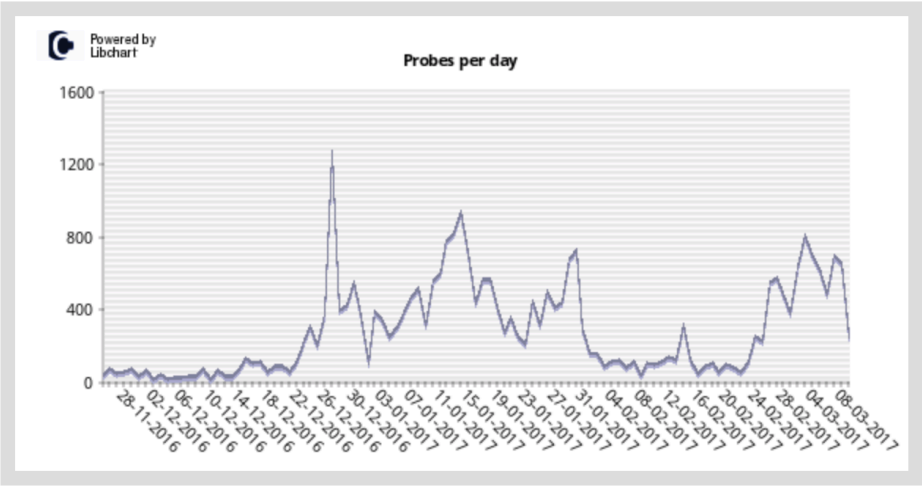



**Figura 1.** Pruebas por día

En la imagen puede verse un incremento repentino cerca del 22 de Diciembre, que seguramente haya sido debido a que en esa fecha es cuando se terminó de ocultar el honeypot como tal, de forma que ya simulaba ser un servidor real. Además, al principio el servidor no era muy conocido pero en unas pocas semanas pudimos comprobar como apareció en el buscador *Shodan* y por tanto es lógico esperar que el número de ataques recibidos se incremente enormemente desde entonces.

Por si fuera poco, se ha grabado las sesiones de los atacantes una vez entraban en el honeypot, a fin de poder descubrir patrones en su comportamiento. Por ejemplo, uno de los patrones observado es que todos ellos escribían a gran velocidad, así como que no cometían errores de escritura, por lo que es seguro que emplearon scripts automatizados para realizar los ataques. Podemos ver una de estas sesiones en la siguiente imagen:

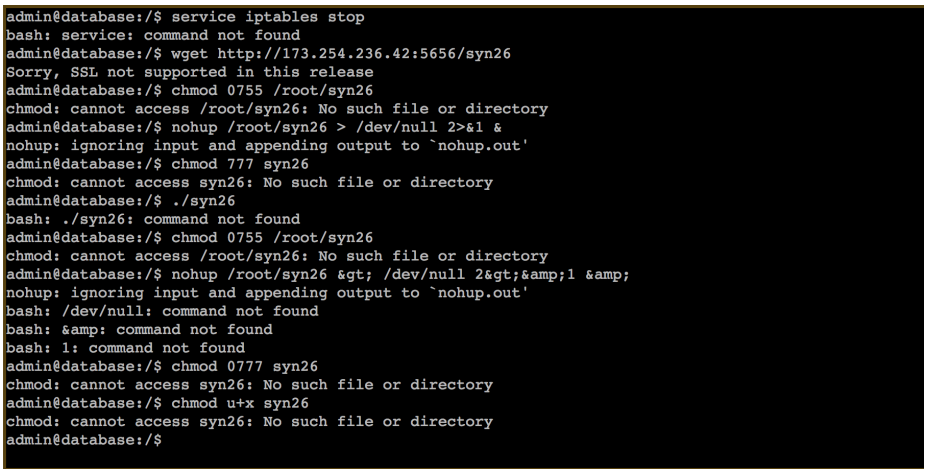



**Figura 2.** Sesión grabada de un atacante

Además hemos encontrado un patrón común a todos ellos:

1. El atacante accede al sistema por el puerto 22
2. Desactiva las reglas de filtrado del *firewall*
3. Descarga un fichero de una dirección IP remota
4. Ejecuta el fichero mediante el comando *nohup,* de forma que la ejecución continue cuando salga de la sesión

A raíz del análisis de estos *logs*, se ha descubierto además que los atacantes siempre realizan acciones similares sin tener en cuenta la situación, por ejemplo, es común ver como un atacante intenta ejecutar un fichero que no ha sido descargado correctamente, o incongruencias similares (ver imagen anterior) y por tanto tenemos otro indicio de que los ataques son realmente automatizados.

Además se ha procedido a clasificar a los atacantes por países, a fin de poder realizar estadísticas. Por ejemplo, podemos ver en la siguiente imagen una lista de los 10 países más beligerantes, de forma que se puede obtener una clasificación por países de los ataques recibidos.

| ID | IP Address | Probes | City | Region | Country Name | Code | Latitude | Longitude | Hostname |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 116.229.239.244 | 1511 | Shanghai | Shanghai Shi | China | CN | 31.0456 | 121.3997 | 116.229.239.244 |
| 2 | 202.109.143.116 | 708 | Nanchang | Jiangxi Sheng | China | CN | 28.55 | 115.9333 | 202.109.143.116 |
| 3 | 109.236.91.85 | 423 | | | Netherlands | NL | 52.3667 | 4.9 | customer.worldstream.nl |
| 4 | 122.227.189.222 | 348 | Ningbo | Zhejiang Sheng | China | CN | 29.8782 | 121.5495 | 122.227.189.222 |
| 5 | 217.23.10.181 | 305 | | | Netherlands | NL | 52.3667 | 4.9 | customer.worldstream.nl |
| 6 | 93.190.143.155 | 229 | | | Netherlands | NL | 52.3667 | 4.9 | customer.worldstream.nl |
| 7 | 202.109.143.111 | 185 | Nanchang | Jiangxi Sheng | China | CN | 28.55 | 115.9333 | 202.109.143.111 |
| 8 | 123.31.34.215 | 183 | Hanoi | Thanh Pho Ha Noi | Vietnam | VN | 21.0333 | 105.85 | localhost |
| 9 | 121.18.238.99 | 179 | Hebei | Hebei | China | CN | 39.8897 | 115.275 | 121.18.238.99 |
| 10 | 183.91.14.188 | 133 | Hanoi | Thanh Pho Ha Noi | Vietnam | VN | 21.0333 | 105.85 | static.cmcti.vn |

**Figura 3.** Evaluación por países

Otra información útil puede ser ver qué clientes ssh han sido los más utilizados en los ataques. Podemos ver también una clasificación de los 10 clientes más utilizados en la figura 4:

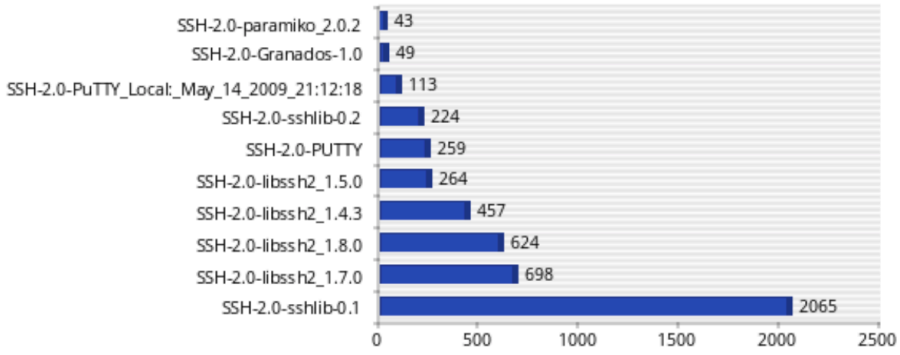



**Figura 4.** Top 10 de clientes ssh

Centrándonos ahora en Glastopf, hemos recibido numerosos ataques del tipo inyección SQL, así como búsqueda de los ficheros *robots.txt* o *sitemap.xml*, ambos de ellos conocidos por contener información valiosa sobre la estructura de un sitio web. Además hemos encontrado varias pruebas hacia la ruta */wp-login.php*.

En otro ataque muy común, el atacante trata de acceder a la carpeta */shell*, la cual no existe, lo que nos hace suponer que los ataques dirigidos al nodo Glastopf también son automatizados. El atacante escribe en dicha ubicación una cadena muy larga de caracteres, por ejemplo una como la siguiente (la más común):

*/ s h e l l ? % 6 3 % 6 4 % 2 0 % 2 F % 7 4 % 6 D % 7 0 % 3 B %77%67%65%74%20%68%74%74%70%3A%2F%2F %36 %31 %2E %31 %36 %30 %2E%32%31%33 %2E %32 %38 %3A %35 %34 %33 %32 %31 %2 F %64 %6C %72 %2E %61 %72 %6D %3B %63 %68 %6D %6F%64%20%37%37 %37 %20 %2A %3 B %2E %2F %64 %6C %72 %2E %61 %72 %6D*

La cual puede traducirse como:

*cd /tmp && wget http ://61.160.213.28:54321/ dlr .arm; chmod 777 ∗;./ dlr.arm*

Se han encontrado varias cadenas de este tipo y todas ellas presentan un patrón similar:

1. El atacante se mueve a la carpeta */tmp* ya que es el único lugar donde tiene permiso de escritura con el usuario *www*.
2. Descarga un binario malicioso de una dirección IP
3. Le asigna los permisos necesarios y lo ejecuta

Además hemos observado que el origen de los ataques es, a diferencia de en Kippo, bastante diverso. Podemos encontrar ataques de países muy diferentes, como podemos observar en la figura 5.

Se han empleado para el experimento **865** muestras, en lugar del enorme número de ataques registrados en Kippo. Esto es debido a que el ratio de ataques es mucho más bajo en el protocolo HTTP que en el SSH, debido a el tipo de explotación, ya que la segunda se presta más a su automatización por *script kiddies*[1] que la primera.

[1] Término despectivo aplicado a atacantes con poca experiencia

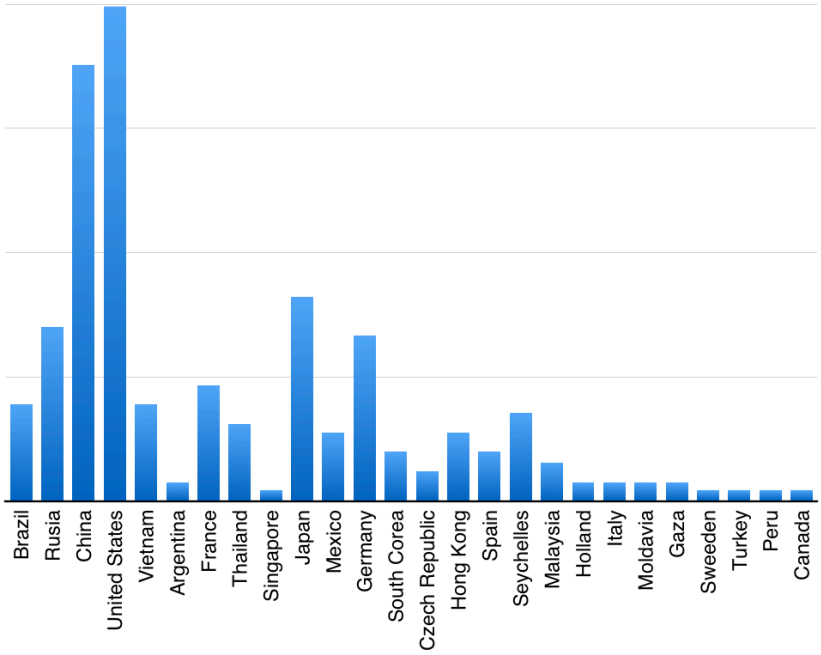



**Figura 5.** Pruebas por día en Glastopf

## 4 Análisis de las muestras de malware obtenidas

Para tal fin se han empleado herramientas tanto manuales como automatizadas (*objdump, gdb, radare, API de VirusTotal*…).

Las muestras recogidas son muy similares aunque con ligeras variaciones, como por ejemplo la IP a la que se conectan. En la siguiente tabla podemos ver un resumen con algunas de las muestras encontradas y analizadas, siendo el resto similares a estas. Todas ellas están escritas con objetivo sistemas Linux con arquitectura de 32 bits siendo el resto de características las que vemos en la tabla:

**Table 1.** Muestras de malware analizadas

| **Muestra** | **Procesador** | **Tipo de ejecutable** | **Lenguaje** | **Información de depuración** | **Enlazado** |
|---|---|---|---|---|---|
| Z | Intel 80386 | ELF | C++ | Sí | Estático |
| Pomo | Intel 80387 | ELF | C | No | Estático |
| Udp26 | Intel 80388 | ELF | C++ | No | Estático |
| Syn26 | Intel 80389 | ELF | C++ | Sí | Estático |
| Wrt | Intel 80390 | ELF | C++ | Sí | Estático |
| Xudp | Intel 80391 | ELF | C | Sí | Estático |
| Mips | Intel 80392 | ELF | C++ | Sí | Estático |

Hemos descubierto que las muestras pertenecen a un tipo muy particular de malware chino, muy común y potencialmente dañino. Las muestras tienen funcionalidad de *backdoor* y abren un puerto elegido aleatoriamente para conectarse a una IP remota. Además, se ha descubierto que son capaces de realizar ataques de denegación de servicio a otras máquinas una vez que han comprometido el host objetivo. Algo que nos ha llamado mucho la atención es que algunas muestras presentan información de depuración, por lo que nos hemos centrado en estas para poder obtener el máximo posible de información.

Sospechamos que dichas muestras pueden pertenecer a la botnet *BillGates,* calificada como de alto riesgo, a causa de las similitudes encontradas con otras muestras de malware de dicha botnet. Por ejemplo, un patrón en común es que todas ellas almacenan información en la librería */usr/libamplify.so* de sistemas Linux. Para realizar el análisis hemos empleado técnicas de ingeniería inversa, tales como el análisis estático y dinámico, todo ello en un entorno aislado y controlado. Si el lector quiere profundizar en estas tácticas de análisis, puede consultar [9], donde se proporcionan ejemplos y procedimientos para este fin.

Analizando la entropía de los binarios con *radare* hemos descubierto que no han sido encriptados, ya que dicha entropía es menor del **60%.** Las muestras son bastante indetectables, ya que todas ellas fueron subidas a VirusTotal utilizando un script en Python y solamente cerca del **15%** de los antivirus las detectaban como malware.

Hemos descubierto varias funciones y variables que nos han permitido afianzar nuestras sospechas sobre el comportamiento del malware. Podemos ver una captura de las funciones de una de las muestras, extraídas con ddd en la figura 6:

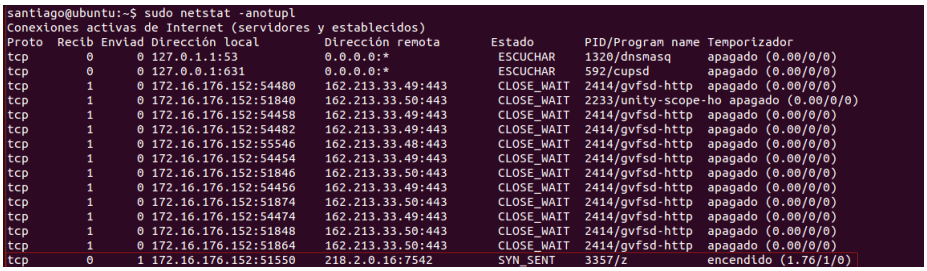

**Figura 6.** Funciones usadas por una de las muestras

Algunas herramientas de línea de comandos permiten recuperar este tipo de información por separado, como pueden ser *file, strings* o *strace* y pueden ser muy útiles para complementar la información obtenida examinando el código ensamblador. Un examen más exhaustivo del código, como mencionábamos anteriormente, revela

funcionalidad de DOS. Podemos encontrar varias cadenas relacionadas con este tipo de ataque como son:

- *AttackBase*
- *PacketAttack*
- *AttackUDP*
- *AttackSyn*
- *AttackICMP*
- *AttackDNS*
- *Tcpattack*

Un ejemplo de análisis dinámico muestra que las muestras de malware son completamente funcionales y funcionan como se espera de ellas. En la última línea de la imagen podemos ver la conexión con una ip desconocida:



```
santiago@ubuntu:~$ sudo netstat -anotupl
Conexiones activas de Internet (servidores y establecidos)
Proto  Recib Enviad Dirección local         Dirección remota       Estado       PID/Program name Temporizador
tcp        0      0 127.0.1.1:53            0.0.0.0:*              ESCUCHAR     1320/dnsmasq     apagado (0.00/0/0)
tcp        0      0 127.0.0.1:631           0.0.0.0:*              ESCUCHAR     592/cupsd        apagado (0.00/0/0)
tcp        1      0 172.16.176.152:54480    162.213.33.49:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:51840    162.213.33.50:443      CLOSE_WAIT   2233/unity-scope-ho apagado (0.00/0/0)
tcp        1      0 172.16.176.152:54458    162.213.33.49:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:54482    162.213.33.49:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:55546    162.213.33.48:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:54454    162.213.33.49:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:51846    162.213.33.50:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:54456    162.213.33.49:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:51874    162.213.33.50:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:54474    162.213.33.49:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:51848    162.213.33.50:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        1      0 172.16.176.152:51864    162.213.33.50:443      CLOSE_WAIT   2414/gvfsd-http  apagado (0.00/0/0)
tcp        0      1 172.16.176.152:51550    218.2.0.16:7542        SYN_SENT     3357/z           encendido (1.76/1/0)
```

**Figura 7.** Malware **z** conectándose a una IP remota

## 5 Conclusiones y trabajos futuros

La principal conclusión que podemos extraer es que es esencial asegurar nuestros sistemas, ya que mientras no nos damos cuenta estamos recibiendo numerosos ataques. Además los atacantes disponen de mucho más tiempo y mejores recursos económicos que nosotros por lo que no debemos subestimarlos. Por tanto es imprescindible el sentido común para evitar ataques de ingeniería social, además de una sólida política de seguridad ya que, como hemos visto, la mayoría de los ataques son automatizados y pueden ser evitados. Enfatizamos en el empleo de contraseñas seguras y en evitar el uso de configuraciones “por defecto” para prevenir este tipo de ataques.

Las futuras ampliaciones del trabajo, ordenadas por prioridad son:

1. Creación de una receta de despliegue automático
2. Creación de una interfaz web que simplifique la configuración y el uso de la honeynet

3. Creación de nuevos nodos, con diferentes honeypots, algunos de ellos de alta interacción, a fin de obtener más información
4. Empleo de técnicas de clasificación automática usando técnicas de inteligencia artificial

## Referencias

[1] "The honeynet project,"http://www.honeynet.org/project,accessed:2017-05-24.

[2] S.Almotairi, A.Clark, G.Mohay, and J.Zimmermann,"Characterization of attackers' activities in honeypot traffic using principal component analysis," in *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. IEEE, 2008, pp. 147–154.

[3] ——, "A technique for detecting new attacks in low-interaction honeypot traffic," in Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference on. IEEE, 2009, pp. 7–13.

[4] G. Salles-Loustau, R. Berthier, E. Collange, B. Sobesto, and M. Cukier, "Characterizing attackers and attacks: An empirical study," in *Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on*. IEEE, 2011, pp. 174–183.

[5] K. Kendall and C. McMillan, "Practical malware analysis," in *Black Hat Conference, USA*, 2007, p. 10.

[6] D. A. Quist and L. M. Liebrock, "Visualizing compiled executables for malware analysis," in *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*. IEEE, 2009, pp. 27–32.

[7] R. Harwood and M. Serrano, "Lecture 26: Obfuscation," 2013, car- negie Mellon University, https://www.cs.cmu.edu/~fp/courses/15411-f13/ lectures/26-obfuscation.pdf.

[8] "Shodan is the world's first search engine for internet-connected devices," https://www.shodan.io, accessed: 2017-05-25.

[9] H.project,KnowYourEnemy:LearningAboutSecurityThreats. Addison Wesley, 2004.