Honeynet para el análisis del tráfico y muestras de malware

Santiago de Diego, *Estudiante UGR* Gustavo Romero, *Profesor UGR*

***Resumen*—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 obtain several results related to web attacks with another low/medium interaction honeypot, Glastopf.

The purposes of this work are to identify and classify several samples of malware to show to the reader a general method to achieve this goal.

***Index Terms*—Security, honeypots, honeynet, malware, traffic analysis**

I. 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.

En 1999 esto empezó a cambiar y se comenzó a estudiar también a los atacantes. Se desarrollaron los primeros dispositivos que se centraban en monitorizar los sistemas, lo que dio lugar al Honeypot Project. En algunos trabajos ([**?**] y [**?**]) se emplea el uso de análisis de componentes principales para la caracterización de atacantes, mientras que otros ([**?**]) se centran en elaborar una escala de 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, por ejemplo algunos trabajos ([**?**] y [**?**]) 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 [**?**] se centran en explicar las técnicas más comunmente empleadas para la ofuscación del malware, a fin de dificultar su desensamblado.

Santiago de Diego es estudiante del doble grado en matemáticas e ingeniería informática por la Universidad de Granada

Gustavo Romero es profesor en la Universidad de Granada, en el departamento de Arquitectura y Tecnología de los Computadores

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 gestionar.

Podemos colocar un honeypot en tres posiciones distintas principalmente:

- **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.

II. Escenario

En este trabajo 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 con la misma arquitectura que el objetivo de las muestras, las cuales veremos en la sección correspondiente. Posteriormente, se ha procedido a la eliminación de dichas máquinas virtuales.

III. Resultados obtenidos en el nodo Kippo

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…).

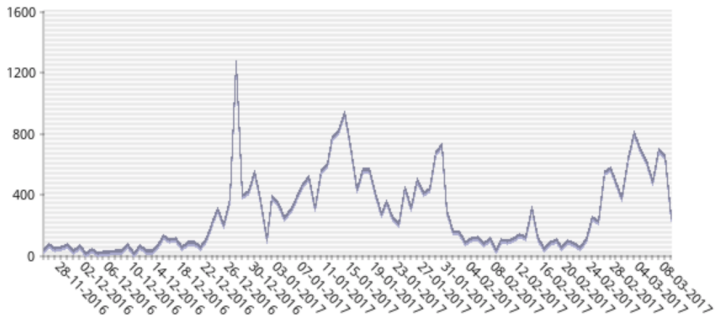

Figura 1. Pruebas por día

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 era lógico esperar que el número de ataques recibidos se incrementase 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:

```
admin@database:/$ service iptables stop
bash: service: command not found
admin@database:/$ wget http://173.254.236.42:5656/syn26
Sorry, SSL not supported in this release
admin@database:/$ chmod 0755 /root/syn26
chmod: cannot access /root/syn26: No such file or directory
admin@database:/$ nohup /root/syn26 > /dev/null 2>&1 &
nohup: ignoring input and appending output to `nohup.out'
admin@database:/$ chmod 777 syn26
chmod: cannot access syn26: No such file or directory
admin@database:/$ ./syn26
bash: ./syn26: command not found
admin@database:/$ chmod 0755 /root/syn26
chmod: cannot access /root/syn26: No such file or directory
admin@database:/$ nohup /root/syn26 &gt; /dev/null 2&gt;&amp;1 &amp;
nohup: ignoring input and appending output to `nohup.out'
bash: /dev/null: command not found
bash: &amp: command not found
bash: 1: command not found
admin@database:/$ chmod 0777 syn26
chmod: cannot access syn26: No such file or directory
admin@database:/$ chmod u+x syn26
chmod: cannot access syn26: No such file or directory
admin@database:/$
```

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 Fig. 3 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 paises

Otra información útil puede ser ver qué clientes ssh han sido los más utilizados en los ataques, ya que esto puede ayudar a establecer clasificaciones de los diferentes atacantes según el tipo de máquina o sistema operativo que utilicen. Podemos ver también una clasificación de los 10 clientes más utilizados en la figura 4:

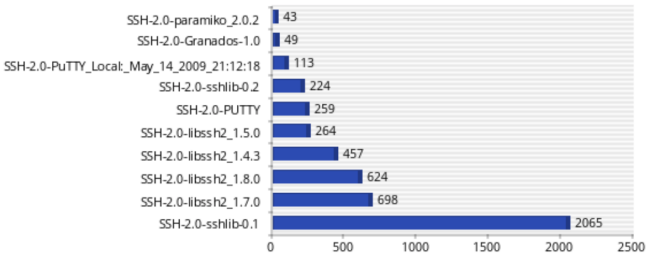

Figura 4. Top 10 clientes ssh

IV. Resultados obtenidos en el nodo Glastopf

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*, debido a que el atacante ha debido creer que tenemos un *wordpress* funcionando en el servidor y ha

tratado de localizar el login.

En otro ataque muy común, el atacante trata de acceder a la carpeta */shel*l, 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):

```
/shell?  %63  %64  %20  %2F  %74  %6D  %70  %3B
     %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  %2F  %64  %6C  %72  %2E
     %61  %72  %6D  %3B  %63  %68  %6D  %6F  %64
     %20  %37  %37  %37  %20  %2A  %3B  %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 kiddies1 que la primera.

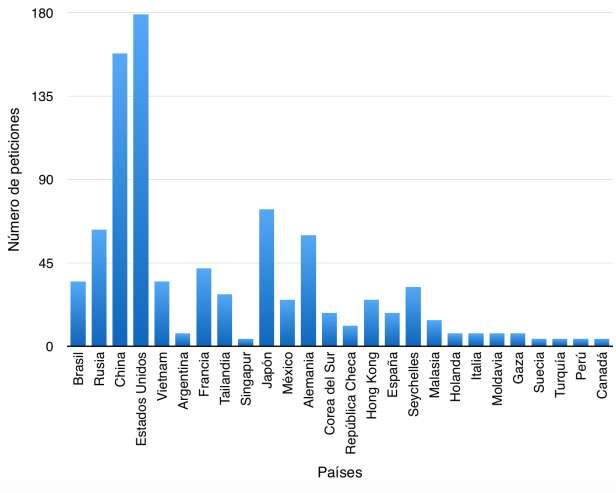

Figura 5. Número de ataques por día hacia Glastopf

V. Análisis de las muestras de malware obtenidas

En esta sección analizaremos las diferentes muestras de malware que hemos obtenido mediante el nodo Kippo. 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. La mayoría de ellas con binarios de tipo ELF [1], están escritas en C/C++, para arquitectura x86 y su sistema operativo objetivo es Linux. La mayoría de ellas tienen como objetivo un procesador Intel 80386 Processor, pero algunas de ellas tienen como objetivo un MIPS R3000.

Hemos descubierto que las muestras pertenecen a un tipo muy particular de malware chino, muy común y potencialmente dañino. Las muestras tienen funcionalidad de puerta trasera 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 [**?**], donde se proporcionan ejemplos y procedimientos para este fin.

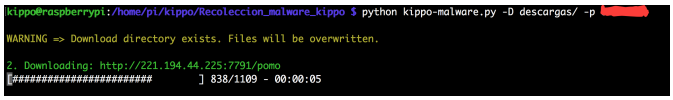

Figura 6. Descarga de malware desde uno de los honeypots

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 Fig. 7:

[1]Executable and Linkable Format

Figura 7. Funciones empleadas por una de las muestras de malware

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 corrobora que las muestras de malware son completamente funcionales y hacen lo que se espera de ellas. A modo de ejemplo, en la última línea de la imagen podemos ver la conexión con una ip desconocida:

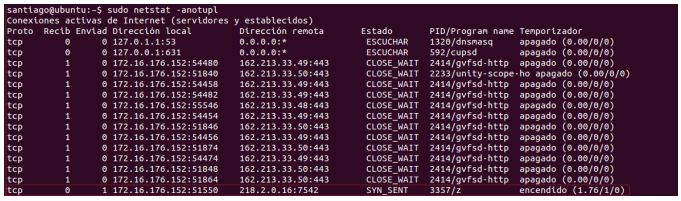

Figura 8. Malware **z** conectándose con una IP remota

VI. Conclusiones y trabajo futuro

La principal conclusión que podemos extraer es que es esencial asegurar nuestros sistemas, ya que, aunque no somos conscientes, estamos recibiendo ataques constantemente. Además los atacantes disponen de mucho más tiempo y recursos 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. Despliegue automático de la herramienta
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

Agradecimientos

Primero de todo, a mi tutor, Gustavo Romero del departamento de Arquitectura y Tecnología de los Computadores, de la Universidad de Granada, por ayudarme en todo lo que ha podido y por recordarme conocimientos que había olvidado y han sido cruciales para el desarrollo de este proyecto. Por último pero no por ello menos importante, a mis padres por su constante apoyo y porque sin ellos nada de esto sería posible.

Referencias

[1] Lukas Rist, Sven Vetsch, Marcel Koßin, Michael Maue, *A dynamic, low-interaction web application honeypot*. The Honeynet Project, 2010. Disponible en: https://www.honeynet.org/sites/default/files/files/KYT-Glastopf-Final_v1.pdf

[2] The honeypot project, *Know your enemy: learning about security threats*, 2nd ed. Addison-Wesley, 2004

[3] Niels Provos, Thorsten Holz, *Virtual Honeypots : from botnet tracking to intrusion detection*, 1st ed. Addison-Wesley, 2007.

[4] Peter Kálnai, Jaromír Hořejší, *DDoS Trojan: A Malicious Concept that Conquered the ELF Format*. Virus Bulletin, 2016. Disponible en: https://www.virusbulletin.com/virusbulletin/2016/06/vb2015-paper-ddos-trojan-malicious-concept-conquered-elf-format/

[5] Mikhail Kuzinší, *Versatile DDoS Trojan for Linux*. SecureList, 2014. Disponible en: https://securelist.com/analysis/publications/64361/versatile-ddos-trojan-for-linux/

[6] Akamai's Security Intelligence Research Team, *Threat Advisory: "BillGates" Botnet*. Akamai's [state of the internet] / security, 2016. Disponible en: https://www.akamai.com/kr/ko/multimedia/documents/state-of-the-internet/bill-gates-botnet-threat-advisory.pdf

[7] Robbie Harwood and Maxime Serrano, *Lecture 26: Obfuscation*. Carnegie Mellon University, 2013. Disponible en: https://www.cs.cmu.edu/~fp/courses/15411-f13/lectures/26-obfuscation.pdf

[8] Lance Spitzner, *Honeypots: tracking hackers*, 1st ed. Addison-Wesley, 2002.

[9] R.C. Joshi, Anjali Sardana, *Honeypots:A New Paradigm to Information Security*. CRC Press, 2011

[10] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann, *Characterization of Attackers' Activities in Honeypot Traffic Using Principal Component Analysis*. IFIP International Conference on Network and Parallel Computing, 2008.

[11] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann, *A Technique for Detecting New Attacks in Low-Interaction Honeypot Traffic*. Fourth International Conference on Internet Monitoring and Protection, 2009

[12] Gabriel Salles-Loustau, Robin Berthier, Etienne Collange, Bertrand Sobesto, and Michel Cukier, *Characterizing Attackers and Attacks: An Empirical Study*. 17th IEEE Pacific Rim International Symposium on Dependable Computing, 2011.

[13] Kris Kendall, *Practical malware analysis*. 17th IEEE Pacific Rim International Symposium on Dependable Computing, 2011.

[14] Quist, Daniel A. Liebrock, Lorie M., *Visualizing compiled executables for malware analysis*. 6th International Workshop on Visualization for Cyber Security, 2009.