Universidad de Murcia - Facultad de Informática

Propuesta de un sistema para identificar y clasificar ciberataques a través de honeypots SSH

Autor: Pablo José Rocamora Zamora
Tutor: Gregorio Martínez Pérez
Tutor: Manuel Gil Pérez

18 de septiembre de 2019

Contexto y problemática

- Sistemas de seguridad actuales basados en firmas son **poco eficientes**.
- Un honeypot es un software diseñado con el propósito de **atraer a atacantes**, simulando que es un sistema real y vulnerable a posibles ataques.
- Los honeypot se pueden clasificar según su **nivel de interacción**.
- Este proyecto resuelve el **reto propuesto por INCIBE** en el Track de Transferencia de las Jornadas Nacionales de Investigación en Ciberseguridad.
- El objetivo del reto es diseñar e implementar un sistema que permita clasificar las sesiones registradas por el honeypot Cowrie en función de su comportamiento.

Motivación

- No es posible realizar un análisis manual de los datos.
- Un honeypot solo registra datos.
- Gran cantidad de dispositivos expuestos a internet (Ej: IoT).
- Antecedentes de ataques a IoT (Ej: 2016 Mirai y 2018 Sora).
- Necesidad de sistema capaz de clasificar ataques en base al comportamiento.

Cowrie

- Es un honeypot SSH de media interacción.
- Es el más usado actualmente.

Características

- Shell emulada con múltiples comandos implementados.
- Registro en formato textual y JSON.
- Sistema de archivos falso, similar a Debian 5.
- Almacenamiento de archivos descargados con wget, curl, SFTP o scp.
- Soporte para comandos exec SSH.

Cowrie

- Es un honeypot SSH de media interacción.
- Es el más usado actualmente.

Características

- Shell emulada con múltiples comandos implementados.
- Registro en formato textual y JSON.
- Sistema de archivos falso, similar a Debian 5.
- Almacenamiento de archivos descargados con wget, curl, SFTP o scp.
- Soporte para comandos exec SSH.

Objetivos principales

Fase inicial

- Estudio de los logs generados por Cowrie para analizar cuáles son las características más relevantes de estos.
- Creación de una herramienta capaz de convertir esos datos no estructurados en datos estructurados (formato JSON).

Fase de clasificación

- Clasificación manual de los datos almacenados en la base de datos.
- Selección de las features y uso de técnicas Machine Learning para la clasificación de los datos.

Objetivos principales

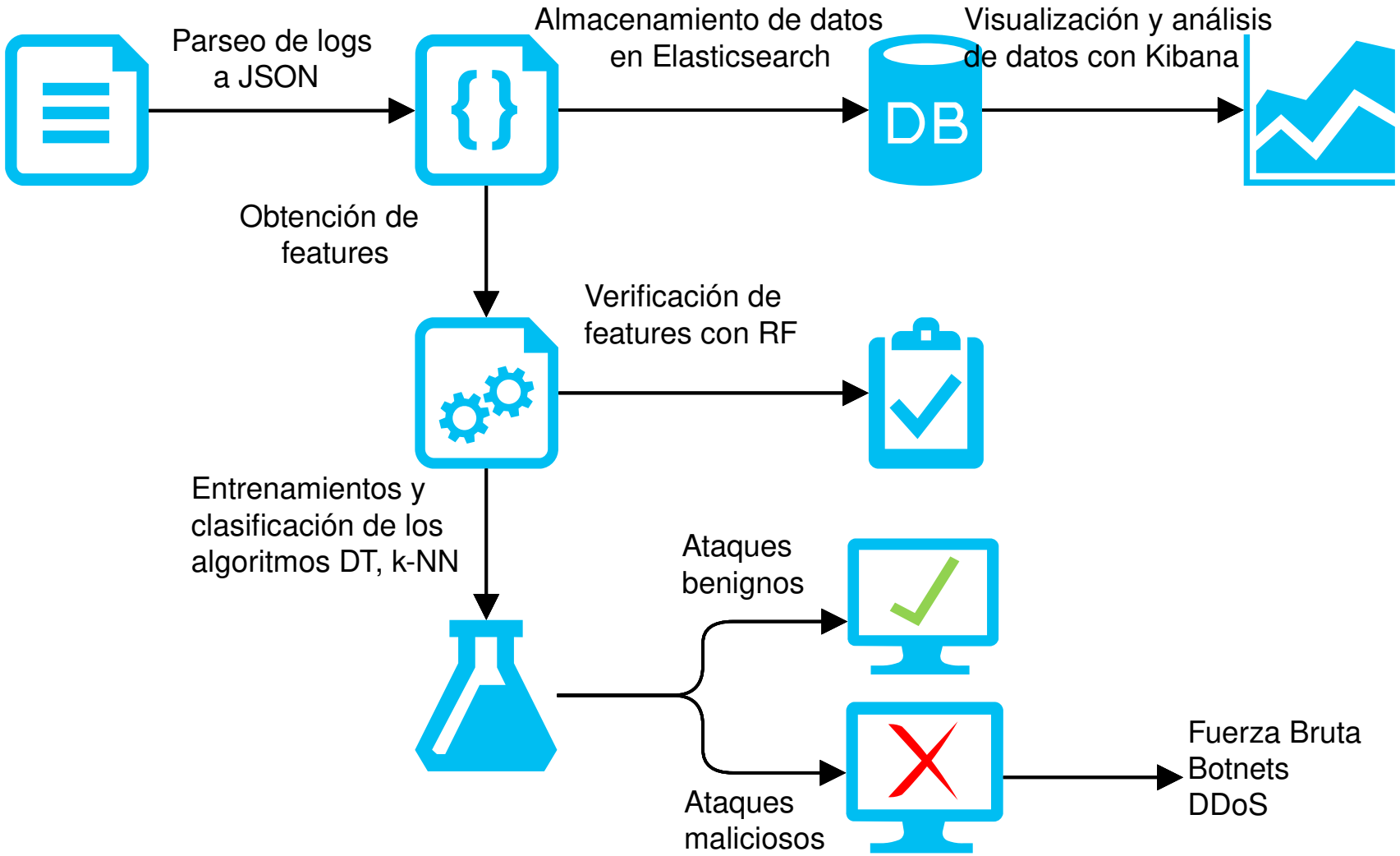
Fase inicial

- Estudio de los logs generados por Cowrie para analizar cuáles son las características más relevantes de estos.
- Creación de una herramienta capaz de convertir esos datos no estructurados en datos estructurados (formato JSON).

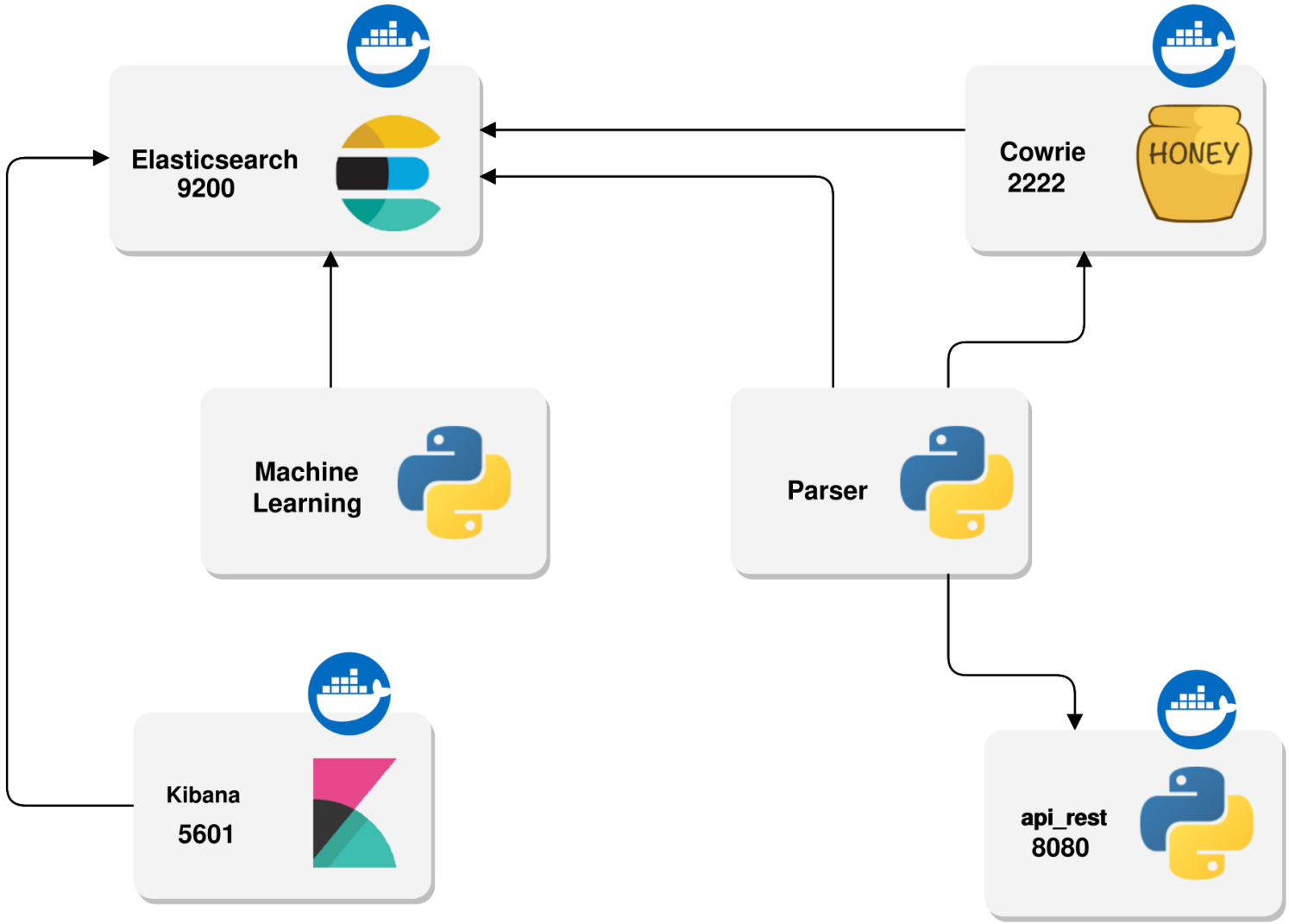
Fase de clasificación

- Clasificación manual de los datos almacenados en la base de datos.
- Selección de las features y uso de técnicas Machine Learning para la clasificación de los datos.

Diseño de la solución propuesta



Arquitectura del escenario diseñado



Cowrie. Instalación, configuración y logs

Eventos registrados

- Inicio de una sesión.
- User-Agent del cliente SSH.
- Dirección IP del atacante.
- Algoritmos de intercambio de claves y algoritmos de cifrado de datos.
- Credenciales con el resultado del login.
- Lista de comandos utilizados con el resultado de la ejecución del comando.
- Lista de descargas realizadas con su hash y ruta de descarga.
- Fin de la sesión con la duración total.

Ejemplo de log

```
[HT] New connection: 172.22.0.1:59128 [session: ea75ae90eab4]
[HT,1,172.22.0.1] Remote SSH version: b'SSH-2.0-OpenSSH_8.0'
[HT,1,172.22.0.1] login attempt [b'root'/b'test1'] succeeded
[HT,1,172.22.0.1] CMD: wget https://procamora.io/code.sh
[HT,1,172.22.0.1] Command found: wget https://procamora.io/code.sh
[HTTPProgressDownloader#info] Starting factory
  <HTTPProgressDownloader: b'https://procamora.github.io/code.sh'>
```

Parser

Características

- Parseo de logs en crudo en formato JSON.
- Recomponer sesiones partidas en ficheros diferentes.
- Obtención de la reputación SSH y geolocalización de las IP registradas.
- Etiquetado de las conexiones usando un sistema experto basado en reglas.
- Script para enviar los datos a Elasticsearch con el mapping necesario para cada tipo de datos.
- Script para convertir logs antiguos de Cowrie en logs actuales y poder parsearlos.

API REST

Motivación y características

- Limitación peticiones por minuto.
- Conexión con VirusTotal.
- Base de datos local.
- Cola peticiones pendientes.

HTTP	Recurso	Descripción
GET	/analizeHash	Método para analizar un hash
POST	/analizeUrl	Método para analizar una URL
POST	/downloadUrl	Método para descargar el fichero asociado a una URL
POST	/getHash	Método para obtener el hash asociado a una URL
GET	/getReputationIp	Método para obtener la reputación SSH de una IP

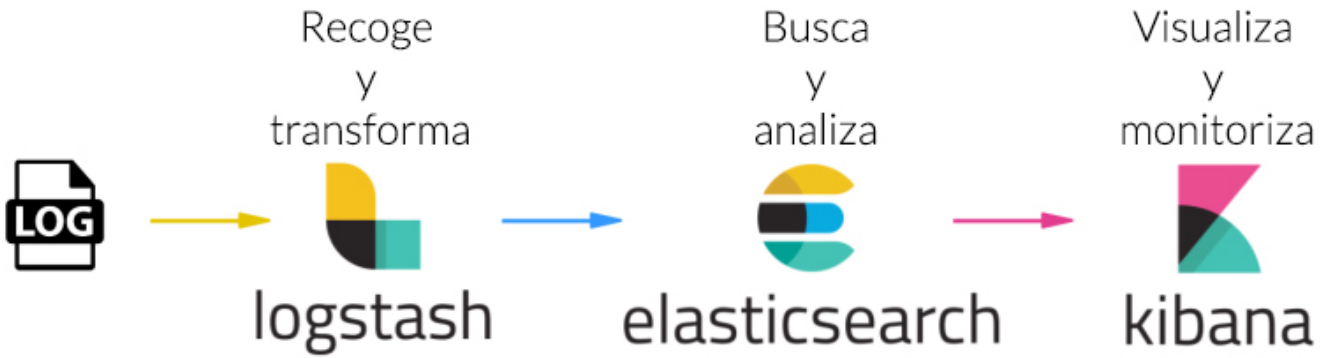
Elasticsearch y Kibana

Motivación

- Orientado a documentos JSON.
- Búsqueda de texto completo.
- Optimizado para búsquedas.

Configuración

- Indice distinto por cada honeypot.
- Mapping predefinido.



Análisis de logs generados por Cowrie. Mapa de calor

- 6 honeypots.
- 82 GBs de datos.
- 16.967.189 conexiones.
- 66.209 direcciones IP únicas.

Figura: Mapa de calor de ataques

Análisis de logs generados por Cowrie

Dirección IP	Frecuencia	País
5.188.86.174	1.128.71	Irlanda
5.188.86.211	1.009.42	Irlanda
5.188.87.53	681.125	Irlanda
5.188.87.55	671.261	Irlanda
5.188.87.51	654.598	Irlanda
5.188.87.52	614.103	Irlanda
5.188.87.54	612.463	Irlanda
5.188.87.49	603.271	Irlanda
109.248.9.102	603.007	UK
109.248.9.101	599.346	UK

Cuadro: Tabla de frecuencia de IPs

Whois 5.188.86.174

- inetnum: 5.188.86.0 - 5.188.87.255
- descr: pool for VPS and Cloud hosting
- mnt-domains: GLOBALLAYER

Global Layer B.V. es una empresa de alojamiento web.

Análisis de logs generados por Cowrie. Mapa de calor

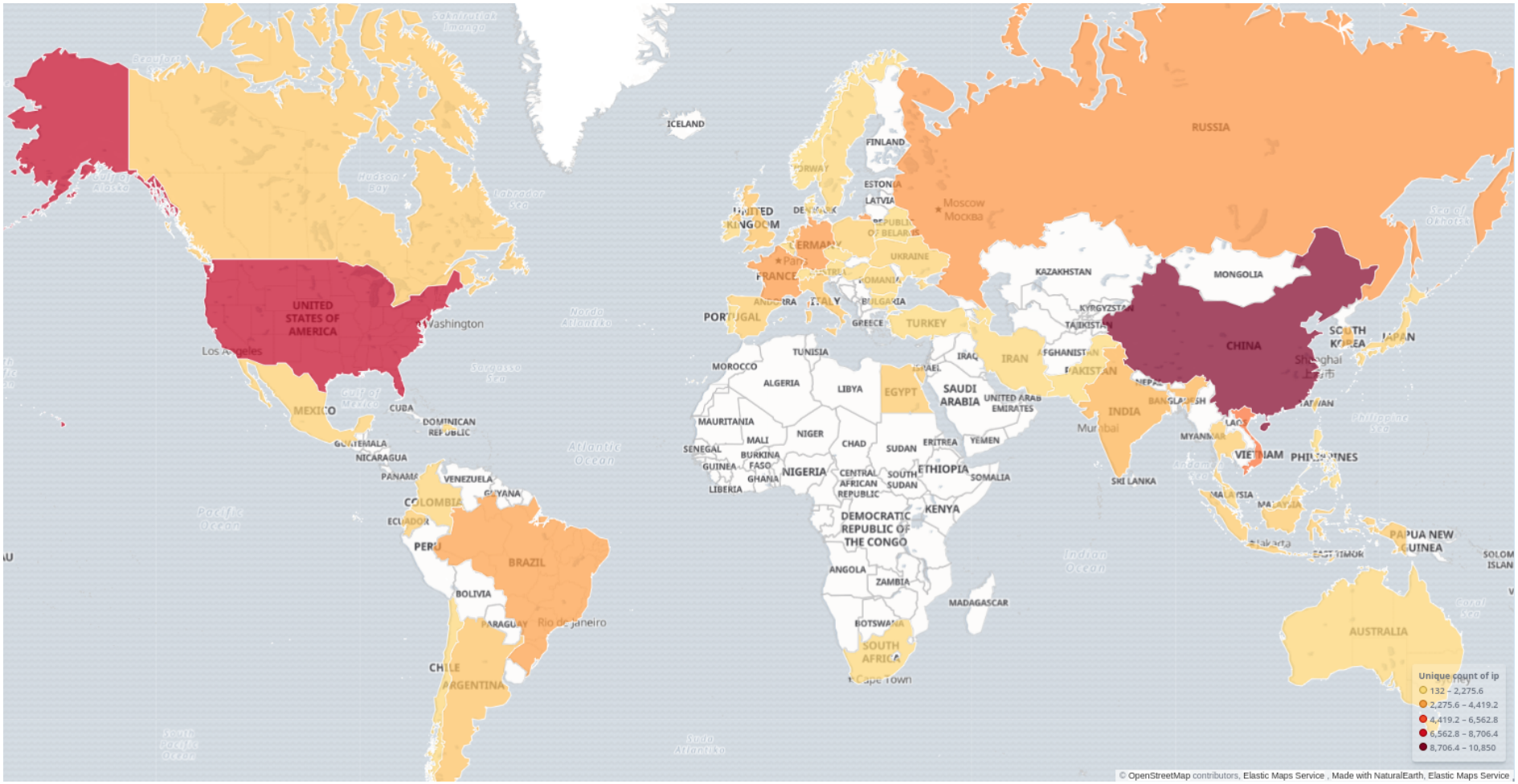


Figura: Mapa de calor de ataques con IP única

Análisis de logs generados por Cowrie

Credenciales	Frecuencia
root/admin	14.161.712
admin/pfsense	25.047
root/changeme	23.025
admin/admin	21.619
admin/password	21.025
nproc/nproc	17.656
root/123456	15.659
root/root	14.221
support/support	10.000

(a) Tabla de credenciales

Comandos	Frecuencia
cat /proc/cpuinfo	22.687
grep name	19.906
wc -l	15.837
/gisdfoewrsfdf	14.753
uname -a	11.379
cd	7.819
cat //.nippon	6.813
rm -f //.nippon	6.700
free -m	6.521

(b) Tabla de comandos ejecutados

Análisis de logs generados por Cowrie

Clientes SSH	Frecuencia
SSH-2.0-OpenSSH_7.3	7.517.861
SSH-2.0-Go	7.254.738
SSH-2.0-libssh-0.6.3	474.818
SSH-2.0-PUTTY	337.872
SSH-2.0-libssh2_1.7.0	255.016
SSH-2.0-check_ssh_2.2.1	228.501
SSH-2.0-Granados-1.0	95.159
SSH-2.0-libssh-0.2	59.435
SSH-2.0-libssh2_1.4.3	52.652

(a) Tabla de clientes SSH

Clientes SSH	Frecuencia
OpenSSH	7.575.363
Go	7.254.866
libssh	608.540
libssh2	399.261
PUTTY	337.872
check_ssh	228.501
Granados	95.159
PuTTY	46.998
paramiko	33.878

(b) Tabla de clientes SSH sin versión

Análisis de logs generados por Cowrie

Buffer overflow

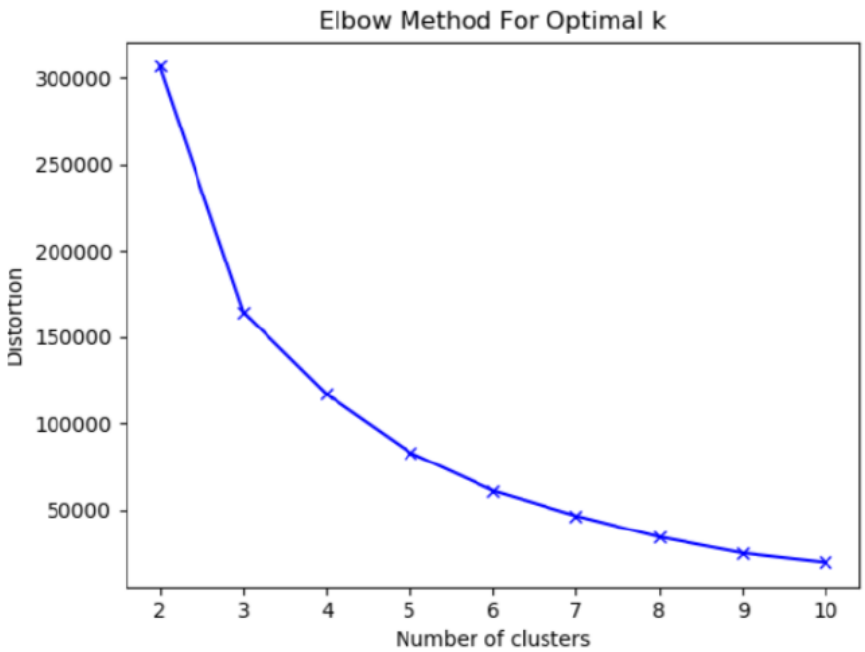
```
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x03\x00\x00\x00'
MAIL FROM:<AnabelleDarstt@aaifgg.com>

\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00-tcpCONNECT 211.231.108.47:25 HTTP/1.0\x00

\x16\x03\x01\x00\x9a\x01\x00\x00\x96\x03\x03\xd5
\xfd>@\xd5t\xea'\xd0\x03

\xc8\x00\x00D<\x7f\x00\x00\xe0*E<\x7f\x00\x00\x98vx
\xfd~\xb8c\xdf\x90\x0e\x9f><\x7f\x00\x00>B\xb6D3
\xa1h\xcc\xcd\xf8|'\x04e\xd4\x07\xd1\x0b\xbcB\x8f
\xb1\xc1&\x0c\xfcq\xf4\xee}\x00\x91;*\x7f'\x18\x95
\xdb\x07\xaa}\xc1,\xe6\xa3Y\x9cSSH-2.0-OpenSSH_7.3

\x00\x00\x8d>\x00\x00\x00'250 Ok, message accepted for delivery
```

Análisis de logs generados por Cowrie

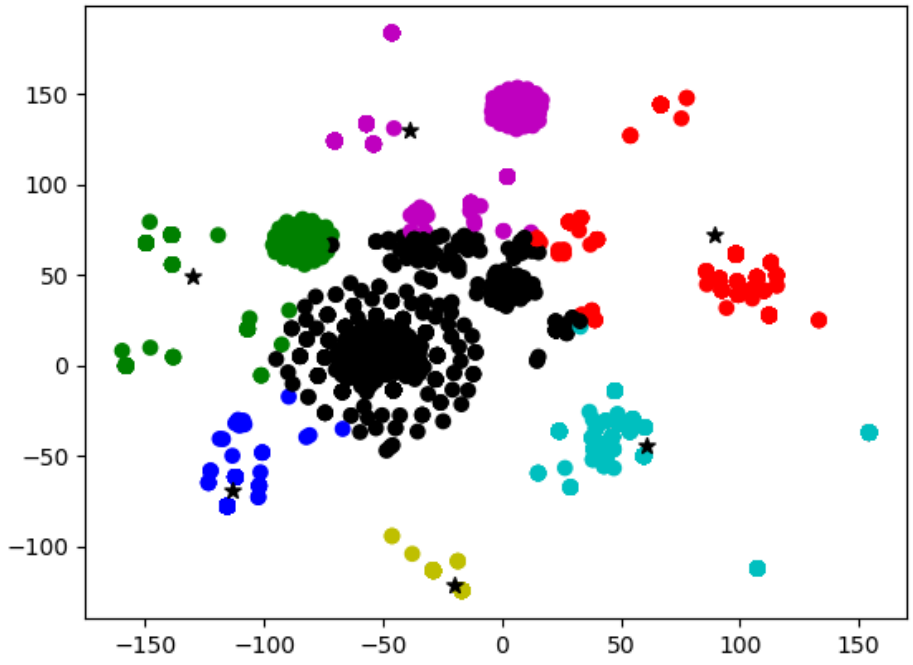
Ficheros descargados	Peligrosidad	Frecuencia
85.159.237.19/bins.sh	5	437
195.22.126.16/ssh1.txt	8	414
185.234.217.21/ssh1.txt	6	357
google.com	0	337
107.174.34.70/Swag.sh	3	313
222.186.139.216:9960/chongfu.sh	2	217
203.146.208.208/drago/images/.ssh/y.txt	9	214
51.15.48.238/armex.sh	1	188
77.247.178.189/bins.sh	3	144
107.174.34.68/Swag.sh	9	127

Cuadro: Tabla de ficheros descargados

Resultados de las técnicas de Machine Learning



(a) Cálculo del k óptimo a través del método del codo



(b) Resultado del clustering (k-means) aplicado a las sesiones

Resultados de las técnicas de Machine Learning

Aprendizaje supervisado (scikit-learn)

- k-Nearest Neighbors (k-NN)
- Decision Tree (DT)
- 80 % entrenamiento
- Random Forest (RF)
- Support Vector Machines (SVM)
- 20 % evaluación

Experimento 1

- Features: comandos ejecutados etiquetado según el nivel de severidad.
- Dataset: propio.
- $\text{Precisión}_{\text{max}}$: 49 %.
- F1-Score: 47 %.
- Niveles de severidad:
  - Nivel 1: Descarga.
  - Nivel 2: Modificar datos.
  - Nivel 3: Obtener datos.

Resultados de las técnicas de Machine Learning

Experimento 2

- Features: comandos ejecutados etiquetado según la acción del comando.
  - Lectura en disco.
  - Escritura en disco.
  - Obtención de información del SO.
  - Conexiones a internet.
  - Compilación o instalación de programas.
  - Ejecución de programas.
  - Eliminar procesos del SO.
- Dataset: propio y scriptzteam.
- $\text{Precisión}_{\text{max}}$: 82 % y 98 %.
- F1-Score: 82 % y 97 %.

Resultados de las técnicas de Machine Learning

Experimento 3

- Features: comandos ejecutados etiquetado según la acción del comando.
  - Tiempo total de la sesión.
  - Análisis de ficheros.
  - País asociado a la IP.
  - Cliente SSH.
- Dataset: scriptzteam.
- $\text{Precisión}_{\text{max}}$: 99 %, F1-Score: 99 %.

Algoritmo	Precision	Recall	F1-Score	Accuracy
k-NN	0,9666	0,9642	0,9652	0,9775
Decision Tree	0,9968	0,9968	0,9968	0,9979
Random Forest	**0,9982**	**0,9973**	**0,9977**	**0,9989**
SVM	0,9917	0,9883	0,9900	0,9948

Conclusiones

Objetivos conseguidos

- Se ha diseñado un sistema capaz de almacenar sesiones y generar gráficas con los datos.
- Se ha diseñado un sistema capaz de clasificar las sesiones en función del comportamiento.

- El protocolo SSH es el que **más ataques recibe**.
- Es necesario una **configuración correcta** y un **formato estructurado** para los logs.
- Realizar un **análisis en tiempo real de los ficheros descargados**.
- El **83 % de las sesiones** han usado las **mismas credenciales**.

Vías futuras

- Usar un honeypot de alta interacción (HIH) que no esté **limitado por los comandos** programados.
- Cowrie es **fácilmente detectable** como honeypot.
- Usar herramientas como **fail2ban** para bloquear aquellas conexiones que están conectándose cada poco tiempo
- Mezclar **distintos tipos de honeypots** para realizar un análisis de varios protocolos, no centrándose únicamente en uno.

Universidad de Murcia - Facultad de Informática

Propuesta de un sistema para identificar y clasificar ciberataques a través de honeypots SSH

Muchas gracias por su atención

18 de septiembre de 2019

Parser

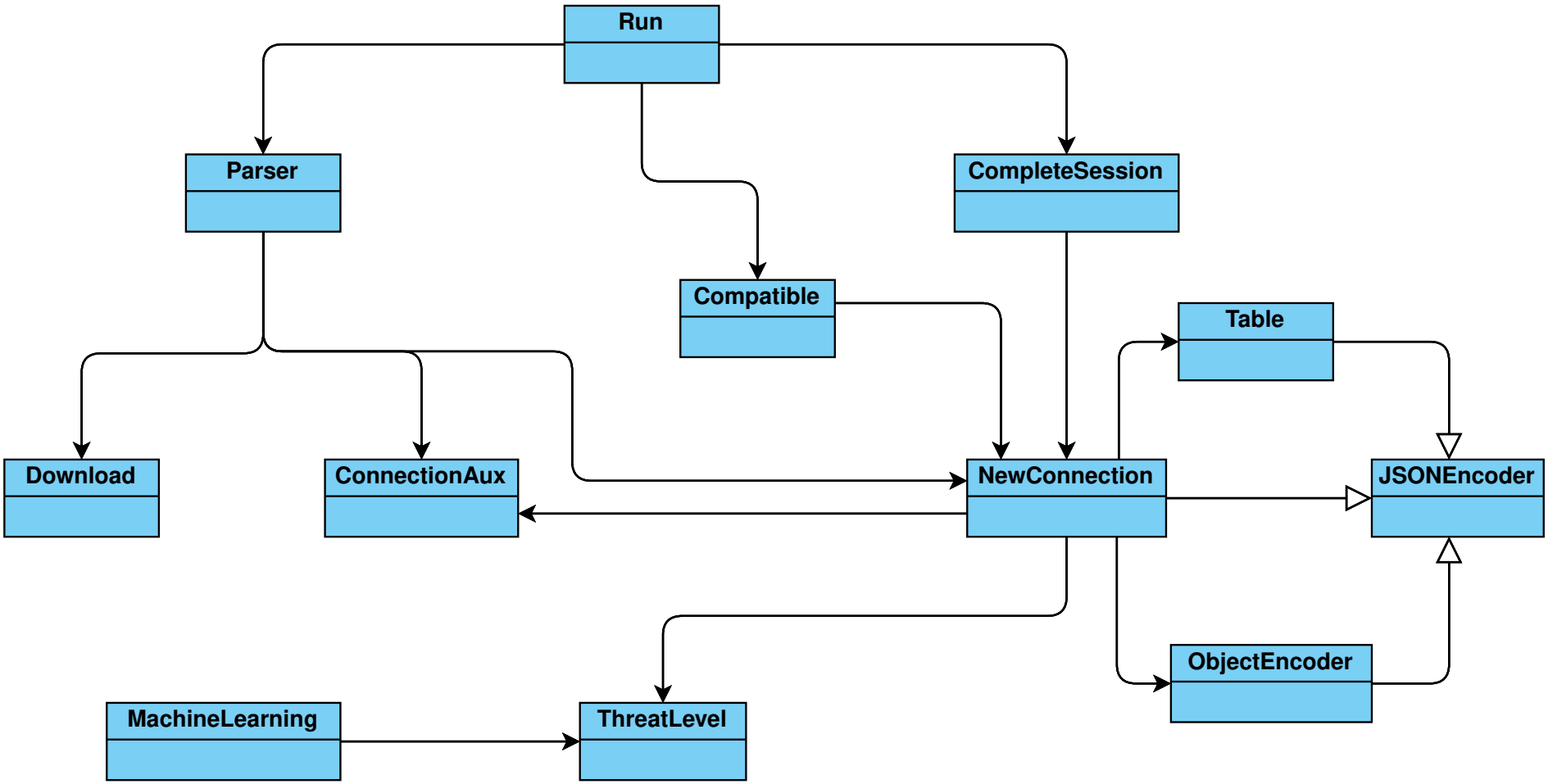


Figura: Diagrama de clases del Parser

Parser

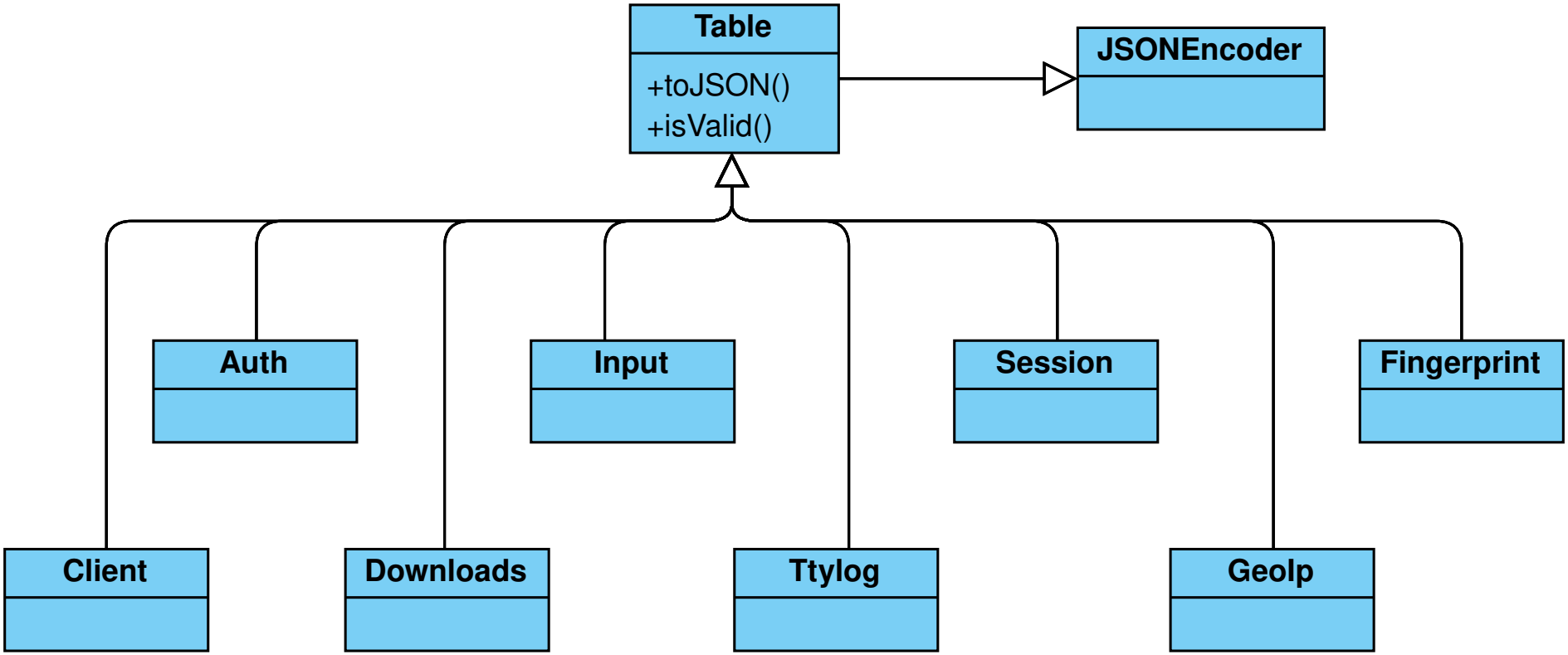


Figura: Diagrama de clases del módulo Tables

Experimentos

Algoritmo	Precision	Recall	F1-Score	Accuracy
k-NN	**0,4991**	**0,4662**	**0,4709**	**0,4698**
Decision Tree	0,3287	0,3298	0,3276	0,3312
Random Forest	0,3285	0,3299	0,3275	0,3309
SVM	0,3286	0,3298	0,3276	0,3311

Cuadro: Resultados del experimento 1 con dataset propio

Algoritmo	Precision	Recall	F1-Score	Accuracy
k-NN	0,7674	0,8103	0,7526	0,7515
Decision Tree	0,8297	**0,8392**	**0,8205**	**0,8609**
Random Forest	0,8297	**0,8392**	**0,8205**	**0,8609**
SVM	**0,8298**	0,8391	0,8204	0,8607

Cuadro: Resultados del experimento 2 con dataset propio