

Manual de instalación y uso del proyecto AIRE

Guillermo Román Ferrero

Javier Gutiérrez Navío

Contenido

Aviso	1
Arquitectura del sistema	2
Instalación del sistema	2
Puesta en marcha de Kibana y ElasticSearch	2
Instalación y arranque de contenedores Kibana y ElasticSearch	2
Configuración de Kibana	3
Puesta en marcha del proyecto	5
Inicialización de la BD y creación de un usuario	5
Introducción de claves de API y tokens.....	6
Arranque del servidor	6
Utilización del sistema.....	7
Subida de capturas.....	7
Ejecución de la extracción y el análisis.....	8
Visualización y exploración de los datos	9
Dashboard	9
Listado de capturas	12
Detalle de información de la captura.....	12
Buscador.....	15
Agregar nuevas aplicaciones a la detección.....	16

Aviso

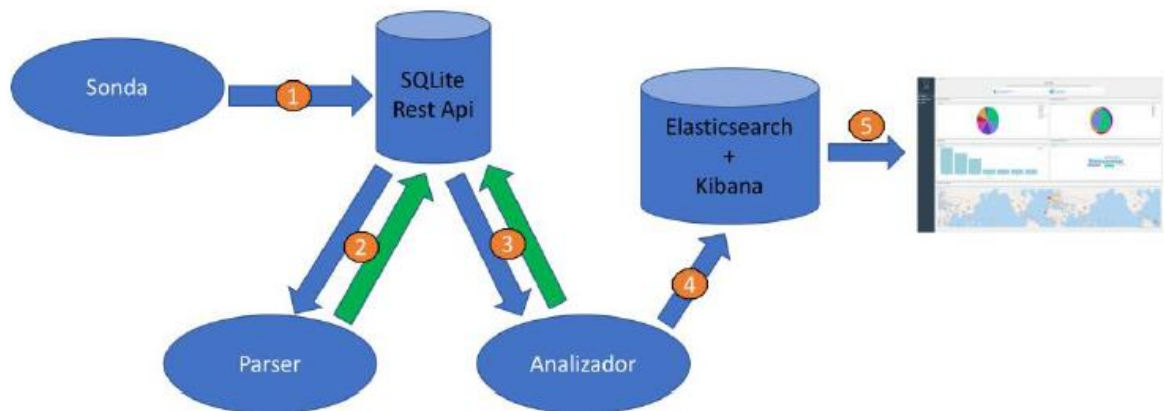
El proyecto se libera y se muestra como una prueba de concepto bajo la licencia GPL versión 3.

No debería usarse en sistemas de producción y solamente debería utilizarse para fines de aprendizaje.

Los autores no se hacen responsables, como establece la licencia, del uso que se le dé al proyecto.

Dicho esto, ¡gracias por mostrar interés en nuestro trabajo y esperemos que lo disfrutes!

Arquitectura del sistema



Antes de nada, conviene dar una pequeña explicación acerca de la estructura del sistema.

- Primero, la sonda se dedica a recoger paquetes de datos *pcap* de distintos dispositivos, y los agrupará por *MAC*. Estos datos serán enviados vía *API Rest* a nuestro servidor de Django para que se guarden en una base de datos de *SQLite*.
- Un script se encargará de pedir los datos recientes a el servidor, mediante la *API Rest*, y extraerá los datos que se han considerado relevantes, guardándolos en un *JSON* y reenviándolo al *webservice* a través de la *API*, actualizando su estado.
- El analizador, al igual que el *parseador*, obtendrá de la *API* los datos que ya han sido *parseando*, y a partir del *JSON* de datos, les dará valor para luego guardarlos de nuevo en el servidor.
- A la vez que lo guarda en el servidor, realizará la indexación de los datos, guardándolos en la base de *Elasticsearch* a través del *SDK*.
- El frontal sacará los datos de la base de datos, y los mostrará para la visualización por parte del usuario.

Instalación del sistema

Puesta en marcha de Kibana y ElasticSearch

Instalación y arranque de contenedores Kibana y ElasticSearch

El proyecto se apoya en *Elasticsearch* para el almacenamiento y manejo de los datos ya analizados y en *Kibana* para su visualización.

Podemos realizar una instalación limpia de ambas utilidades siguiendo los pasos necesarios que se pueden encontrar en la documentación oficial. Sin embargo, nosotros adoptamos la solución utilizar contenedores *Docker* para simplificar su instalación y puesta en marcha.

En caso de preferir realizar una instalación nativa, consultar:

- <https://www.elastic.co/guide/en/elasticsearch/reference/5.5/install-elasticsearch.html>
- <https://www.elastic.co/guide/en/kibana/5.5/install.html>

Para la instalación de estos servicios utilizando *Docker*, necesitaremos tener instalados los paquetes tanto de *Docker* como de *docker-compose*. Para la instalación de dichos paquetes podemos consultar:

- <https://docs.docker.com/install/linux/docker-ce/ubuntu>
- <https://docs.docker.com/compose/install/#install-compose>

Una vez preparados estos requisitos, podemos ejecutar el siguiente comando desde la raíz del proyecto *AIRE* descargado:

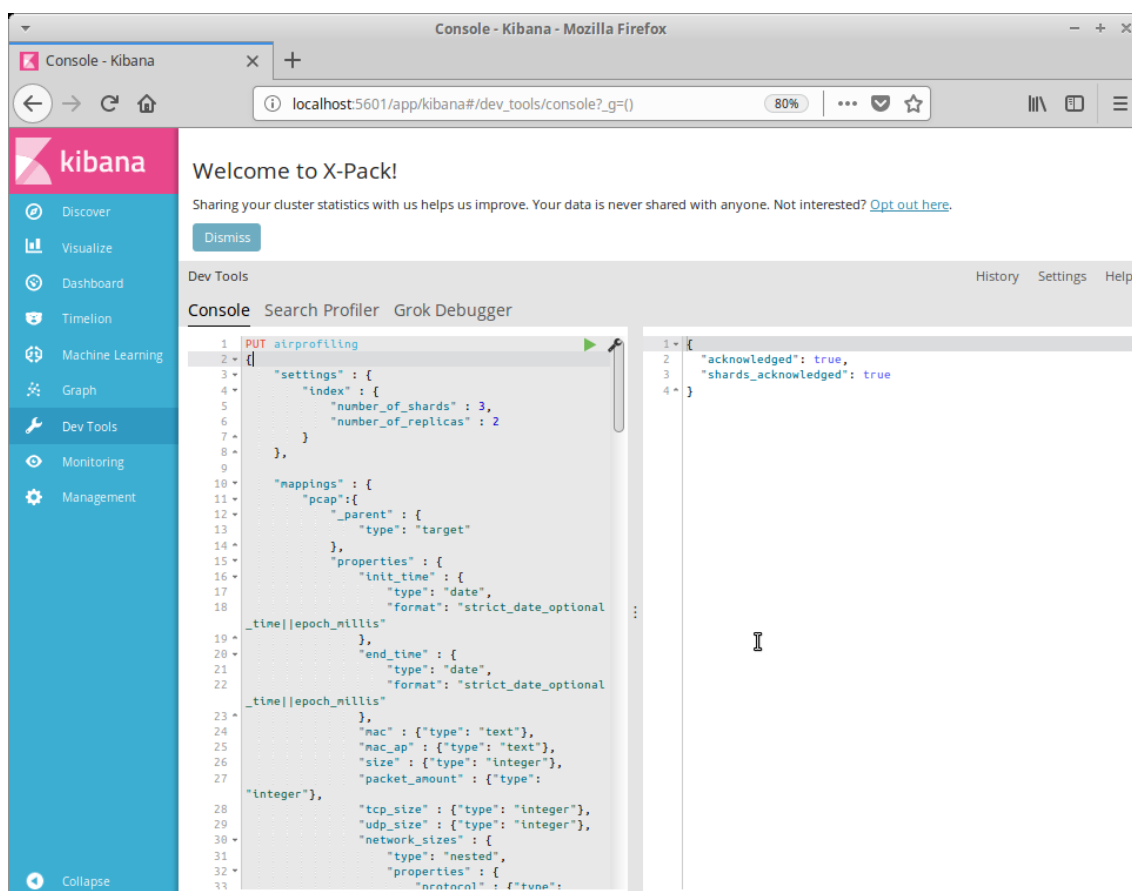
```
sudo docker-compose up -d
```

El comando empezará a descargar y levantar las máquinas necesarias para *Kibana* y *ElasticSearch* de forma automática leyendo el fichero *docker-compose.yml* del proyecto, tras lo cual las mantendrá en segundo plano.

Configuración de Kibana

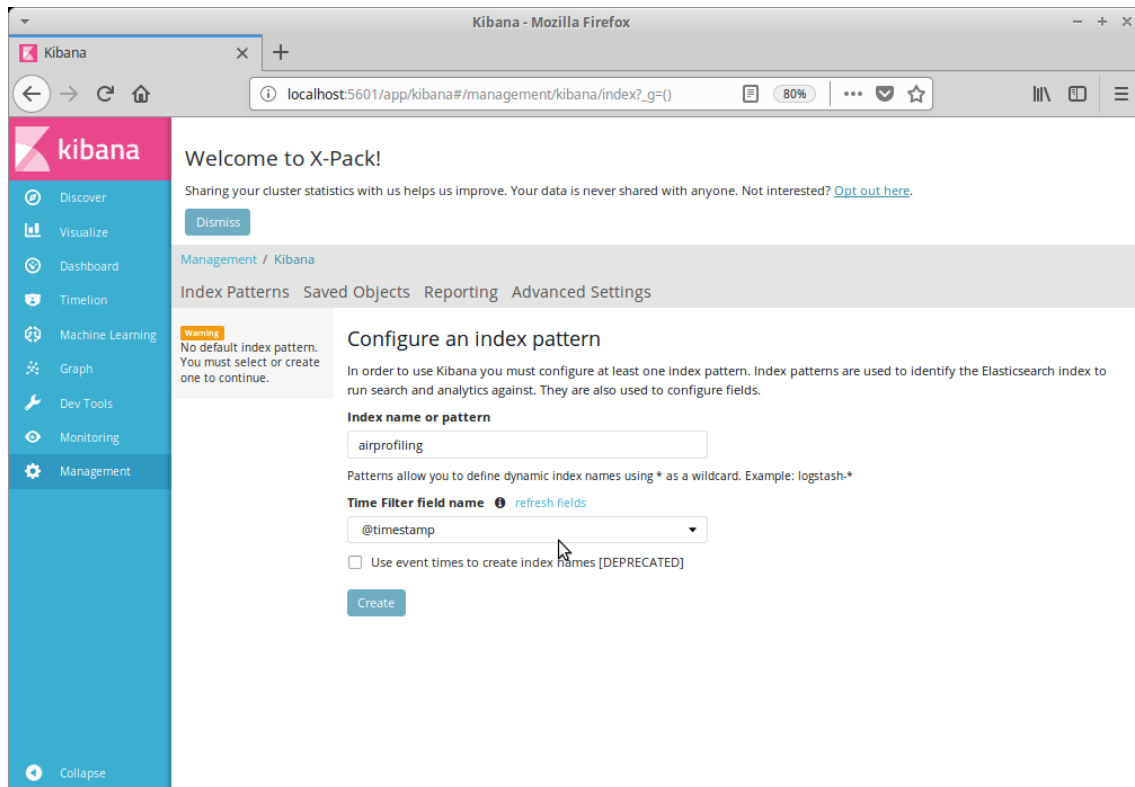
Debemos acceder ahora desde el explorador a la URL <http://localhost:5601/app/kibana> para realizar la configuración de *Kibana*.

En el panel de la izquierda haremos click en *Dev Tools*. Se nos mostrará una consola en la que deberemos introducir el contenido del fichero *airprofiling-index.json* que puede encontrarse en la carpeta *elasticsearch* del proyecto. Haremos click en el icono de ejecutar para así crear la configuración de índices de *ElasticSearch*.

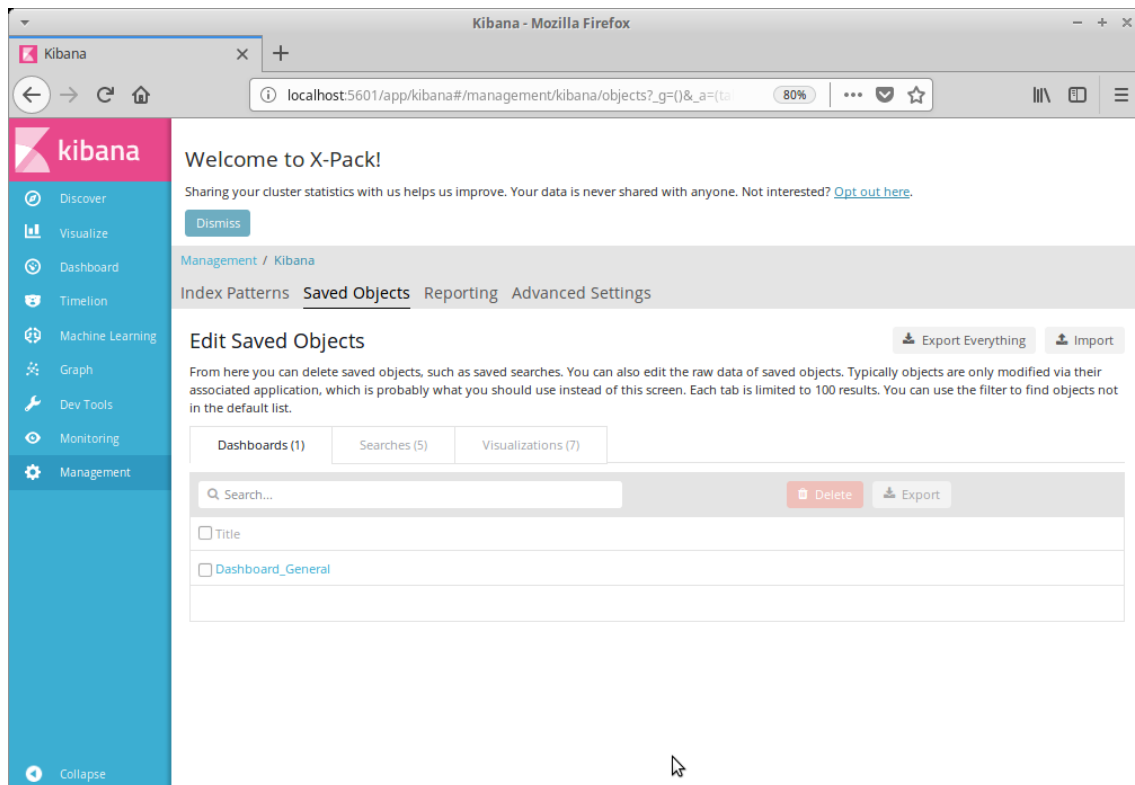


Hecho esto, haremos click en el panel izquierdo en *Management*. En la página que aparece, haremos click en *Index Patterns* dentro del apartado de *Kibana*. Introduciremos dentro del

campo *Index name or pattern* el valor *airprofiling*, y en el campo *Time Filter field name* seleccionaremos *@timestamp*. Por último, haremos click en *Create*.



Vemos como ha quedado creado el índice de forma correcta. Como último paso, haremos click en *Saved Objects*, y en la página que se muestra haremos click en *Import*. Debemos seleccionar el fichero *export_kibana_objects.json* que puede encontrarse en la carpeta *elasticsearch* del proyecto. Confirmaremos la sobreescritura en el diálogo de confirmación. De esta forma importaremos la configuración para la generación de vistas y gráficas para la página del proyecto.



En este punto tendremos *Kibana* y *ElasticSearch* correctamente configurados para su utilización por *AIRE*.

Puesta en marcha del proyecto

Inicialización de la BD y creación de un usuario

El primer paso será crear un usuario que será el que acceda al panel de la aplicación. Lo primero será ejecutar los comandos necesarios para crear la base de datos y su estructura. Ejecutaremos los siguientes comandos desde la raíz del proyecto:

```
python3 airprofiling/manage.py makemigrations
```

```
python3 airprofiling/manage.py migrate
```

Abriremos ahora una consola de *Django* ejecutando el siguiente comando desde la raíz del proyecto:

```
python3 airprofiling/manage.py shell
```

Nos aparecerá una consola de comandos para la administración del servidor. Debemos introducir los siguientes comandos, donde *<usuario>* y *<contraseña>* serán las credenciales que elijamos:

```
from django.contrib.auth.models import User  
  
user=User.objects.create_user("<usuario>",password="<contra  
seña>")  
  
user.is_superuser=True  
  
user.is_staff=True  
  
user.save()
```

```
exit()
```

Introducción de claves de API y tokens

Primero debemos ejecutar el siguiente comando para obtener el token de acceso del usuario que hemos creado anteriormente, utilizando sus credenciales de acceso:

```
http 127.0.0.1:8000/api-token-auth username="<usuario>"  
password="<contraseña>"
```

Nos devolverá una respuesta de esta forma:

```
HTTP/1.0 200 OK  
Allow: POST, OPTIONS  
Content-Length: 52  
Content-Type: application/json  
Date: Wed, 16 May 2018 18:07:57 GMT  
Server: WSGIServer/0.2 CPython/3.5.2  
Vary: Cookie  
X-Frame-Options: SAMEORIGIN  
  
{  
  "token": "c73b8e92e4ac8a2964cec350da1b9674ff831590"  
}
```

Debemos copiar el token y pegarlo en el fichero localizado en la dirección *parsing/conf/ApiRestConfiguration.py* desde la raíz del proyecto, como valor en la línea *AUTHTOKEN*. Esto nos permitirá que los scripts de extracción y análisis puedan actuar contra nuestro servidor.

Seguidamente, debemos introducir un token de la API de [Fono](#) en el fichero de configuración localizado en *parsing/conf/tokens.py* como valor de la línea *FONO_API_TOKEN*. El proyecto utiliza esta API para obtener características de modelos de terminales móviles. Podemos obtener el token en la siguiente dirección:

1. <https://fonoapi.freshpixl.com/token/generate>

En ese mismo fichero debemos introducir una *Access Key ID* y una *Secret Key* válidas de acceso de para las API de *Amazon*. En caso del proyecto, se hace uso de la API de *AWIS* (*Amazon Web Information Service*) para la categorización de páginas web. En la siguiente página puede consultarse cómo obtenerlas:

2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Arranque del servidor

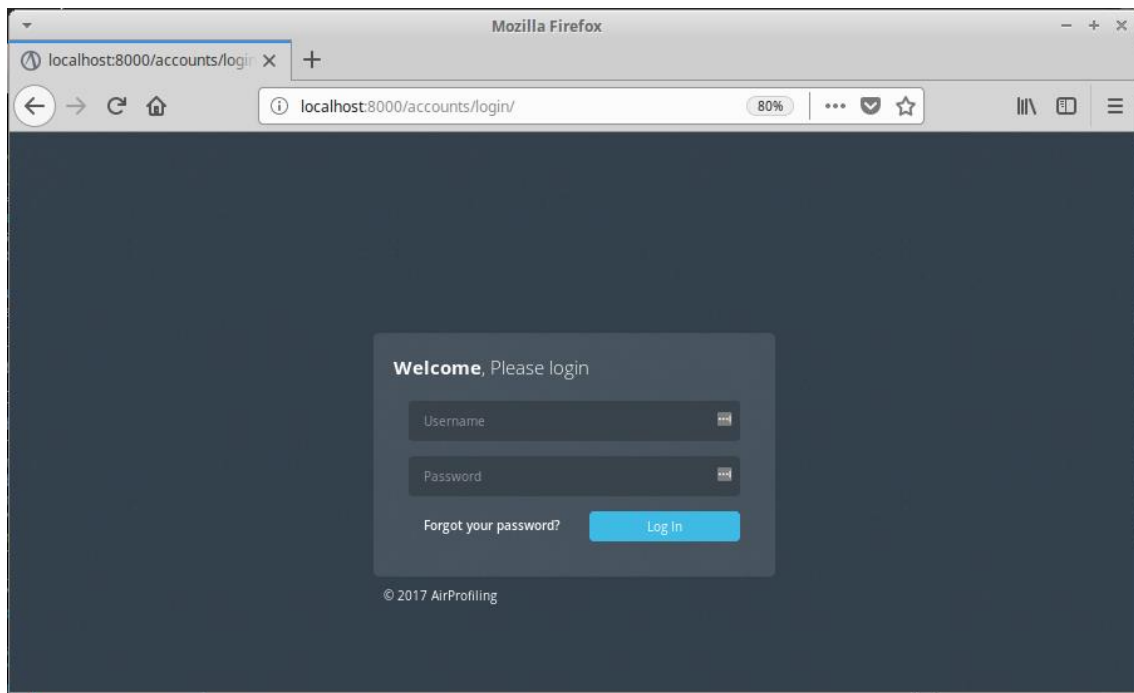
Por último, podremos por fin levantar el servidor de aplicación y dejarlo ejecutándose en una terminal.

```
python3 airprofiling/manage.py runserver
```

Podemos opcionalmente dejarlo ejecutándose en segundo plano:

```
python3 airprofiling/manage.py runserver &
```

En la URL <http://localhost:8000/airprofiling> podremos acceder a la aplicación web. Nos reenviará al principio a la pantalla de inicio de sesión, donde podremos introducir las credenciales anteriormente creados para acceder a la aplicación.



Utilización del sistema

Subida de capturas

La subida de las capturas al sistema puede hacerse mediante una llamada a su API. Podemos hacerlo desde diferentes alternativas, por ejemplo, desde *curl* o *httpie*.

Necesitaremos o bien el usuario y contraseña anteriormente creados o bien el token de acceso que obtuvimos para poder interactuar con la API y realizar la subida. Debemos además conocer tanto la dirección IP (local o remota) donde está localizado el servidor y el puerto del servicio, y la dirección local de la captura en nuestro sistema de ficheros.

El comando que ejecutar con *curl* es el siguiente:

```
curl -X POST -S -F
"file=@<direccion_completa_captura>;type=application/cap" -
F "description=pcap file"
http://<direccion_ip>:<puerto>/pcapfiles/pcap-api/ -H -u
"<usuario>:<contraseña>"
```

O bien:

```
curl -X POST -S -F
"file=@<direccion_completa_captura>;type=application/cap" -
F "description=pcap file"
http://<direccion_ip>:<puerto>/pcapfiles/pcap-api/ -H
"Authorization: Token <token>"
```

Se nos devolverá una respuesta de esta forma, a modo de ejemplo una vez formateado el *json*:

```
{
  "id": 2,
  "file": "http://<direccion_ip>:<puerto>/pcapfiles/pcap-
api/uploads/<captura>.pcap",
  "name": "<captura>",
```

```

    "description": "pcap file",
    "created": "2018-05-18T18:05:18.061937Z",
    "updated": "2018-05-18T18:05:18.061972Z",
    "size": 67250409,
    "owner": 1,
    "status": 0,
    "parsed_json": "",
    "analyzed_date": null,
    "analyzed_json": ""
}

```

```

hartek@ubuntu:~/AIRE$ curl -X POST -S -F "file=@/home/hartek/captura.pcap;type=application/cap" -F "description=pcap file" http://127.0.0.1:8000/pcapfiles/pcap-api/ -H "Authorization: Token f197e52f5904783834aed4e20c5b2b87ac04fcfa"
{"id":1,"file":"http://127.0.0.1:8000/pcapfiles/pcap-api/uploads/captura.pcap","name":"captura","description":"pcap file","created":"2018-05-18T18:36:42.654654Z","updated":"2018-05-18T18:36:42.654686Z","size":28870792,"owner":1,"status":0,"parsed_json":"","analyzed_date":null,"analyzed_json":""}hartek@ubuntu:~/AIRE$

```

Ejecución de la extracción y el análisis

Tanto la extracción como el análisis se realizan invocando un *script* del proyecto desde la máquina servidor en la que esté corriendo el servicio. Estos *scripts* están localizados en *parsing/bin/parsePcapService.py* (extracción) y en *parsing/bin/analyzeAndIndexData.py* (análisis).

Pueden invocarse puntualmente de forma manual para realizar la extracción y posteriormente el análisis de las nuevas capturas subidas, o bien de forma automatizada mediante invocación desde otros *scripts* o desde *cron*.

De cualquier manera, pueden invocarse desde la misma raíz del proyecto con los siguientes comandos:

```
python3 parsing/bin/parsePcapService.py
```

```
python3 parsing/bin/analyzeAndIndexData.py
```

```

hartek@ubuntu:~/AIRE$ python3 parsing/bin/parsePcapService.py
2018-05-18 20:37:18,242 INFO Iniciando el programa de parseo de pcaps. HOST-> localhost:8000/pcapfiles/pcap-api/
2018-05-18 20:37:18,253 INFO - Archivos pcap a parsear nuevos: 1
2018-05-18 20:37:18,270 INFO - Archivos pcap a parsear con error: 0
2018-05-18 20:37:18,270 INFO Parseando el archivo [#1]: http://localhost:8000/pcapfiles/pcap-api/uploads/captura.pcap
2018-05-18 20:37:18,337 INFO - Tamaño del archivo: 27.53125Mb
2018-05-18 20:37:21,836 INFO - Archivo parseado correctamente en 3.5653951550011698s
2018-05-18 20:37:21,854 INFO - Archivo subido correctamente en 0.01817727100024058s
2018-05-18 20:37:21,855 INFO ----
2018-05-18 20:37:21,855 INFO Proceso finalizado con exito para 1/1 paquetes
hartek@ubuntu:~/AIRE$ python3 parsing/bin/analyzeAndIndexData.py
2018-05-18 20:37:42,495 INFO Iniciando el programa de análisis e indexación de datos. HOST-> localhost:8000/pcapfiles/pcap-api/. ELKSERVER -> localhost:9200/airprofiling
2018-05-18 20:37:42,509 INFO Cantidad de paquetes a analizar e indexar: 1
2018-05-18 20:37:42,510 INFO - Analizando paquete [#1]
2018-05-18 20:37:58,018 INFO - [OK] Paquete analizado en 15.50827215299978s
2018-05-18 20:37:58,254 INFO - [OK] Datos indexados correctamente en 0.23546704000000318s
2018-05-18 20:37:58,360 INFO - [OK] ApiRest actualizada correctamente en 0.10588267899947823s
2018-05-18 20:37:58,360 INFO - Total: 15.850712595000005s
2018-05-18 20:37:58,361 INFO ----
2018-05-18 20:37:58,361 INFO Proceso finalizado con exito para 1/1 paquetes en 15.850997072999235s
hartek@ubuntu:~/AIRE$

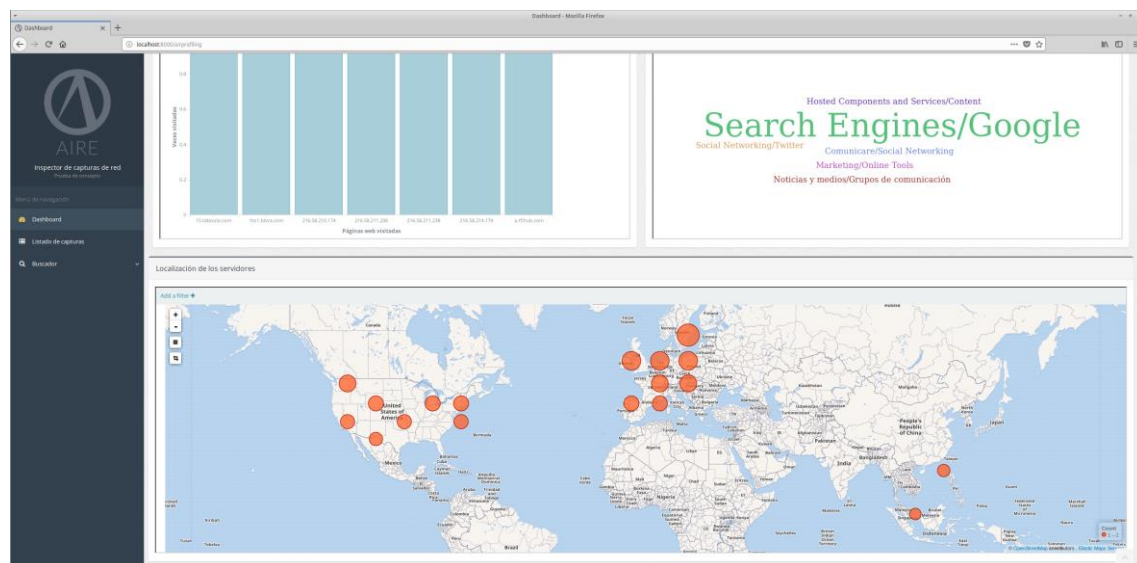
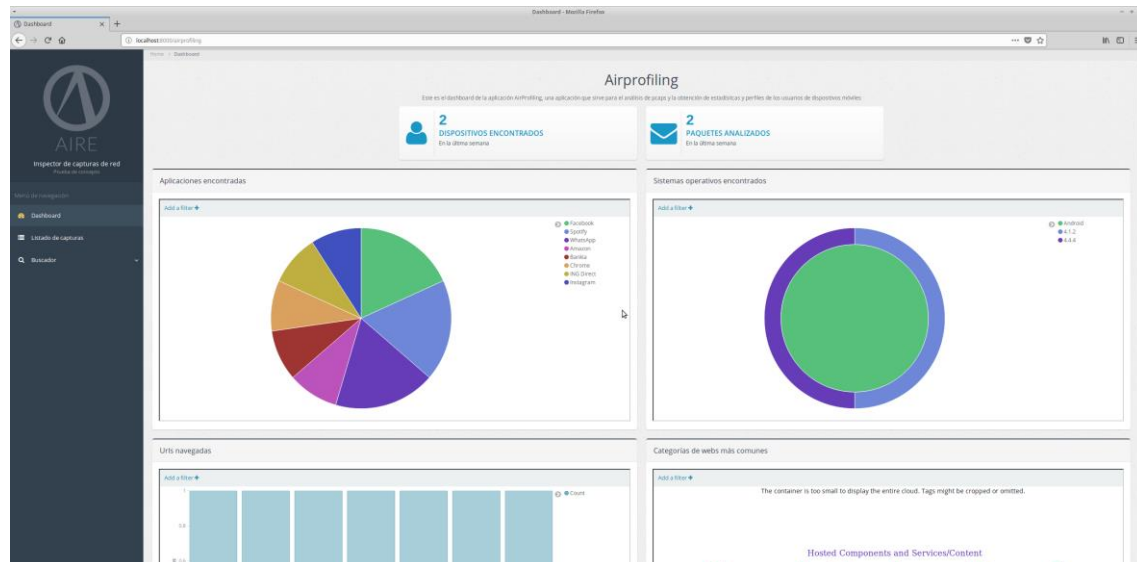
```

Los *scripts* además reintentarán el proceso sobre capturas que anteriormente hayan fallado por errores de conexión o cualquier otra causa.

Visualización y exploración de los datos

Una vez extraídos y analizados los datos, podremos ir a nuestra aplicación web alojada en el servidor, en la dirección <http://localhost:8000/airprofiling> si accedemos desde el mismo. Podemos navegar desde el menú lateral por las diferentes secciones de la aplicación.

Dashboard

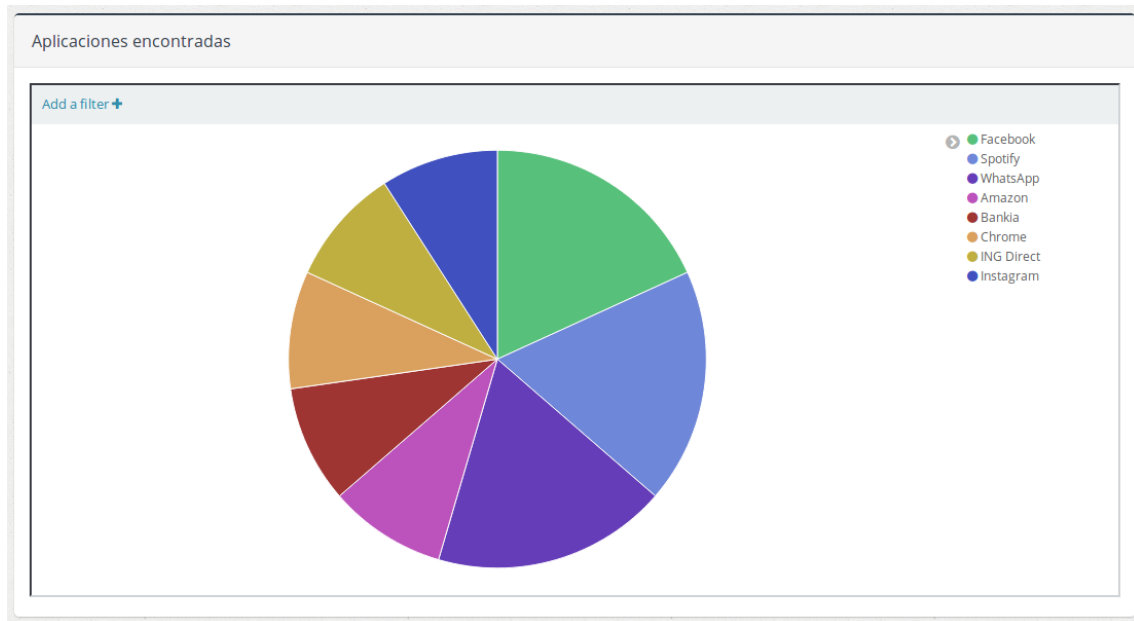


El panel principal incluye varias gráficas que compilan datos de todas las capturas analizadas hasta el momento.

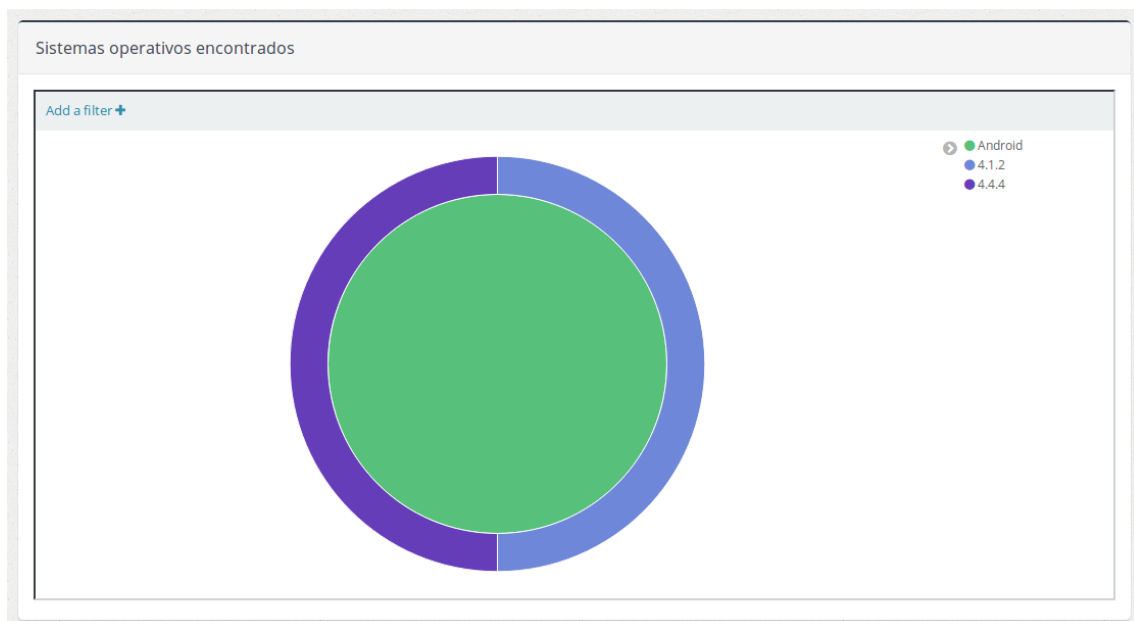
3. En la parte superior podemos encontrar el número de dispositivos encontrados en las capturas, y el número de paquetes analizados.



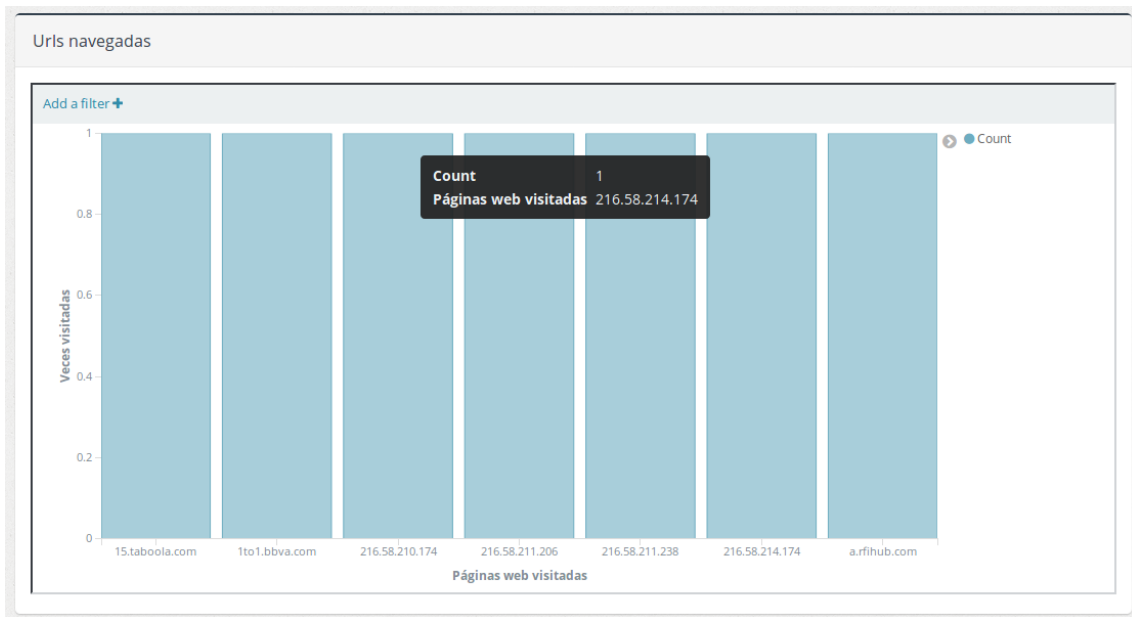
4. La gráfica de *aplicaciones encontradas* muestra el total de aplicaciones detectadas hasta el momento según apariciones.



5. En la gráfica de *sistemas operativos encontrados* podemos ver los sistemas operativos que se han detectado en las capturas, agrupados por tipo (Android o IOS).



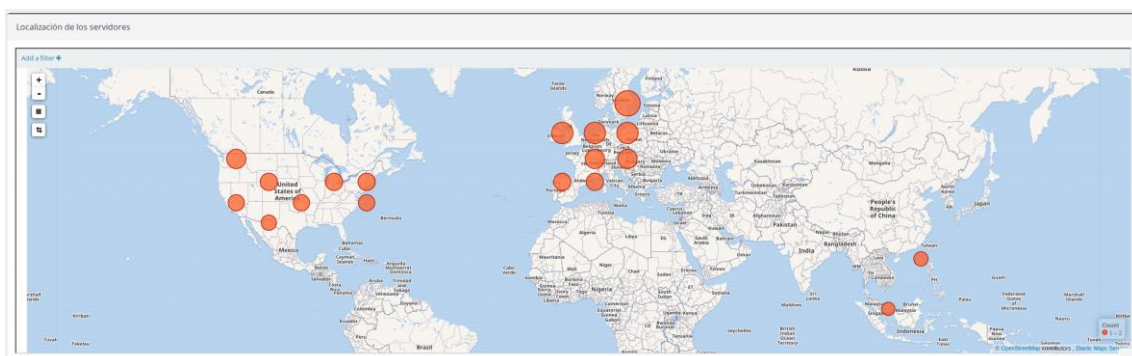
6. En la gráfica de *Urls navegadas* podemos ver las páginas web (o simplemente accesos por HTTP) más visitadas en las capturas.



7. En las *categorías de webs más comunes* podemos ver las temáticas más vistas dentro de las capturas en forma de nube de etiquetas.

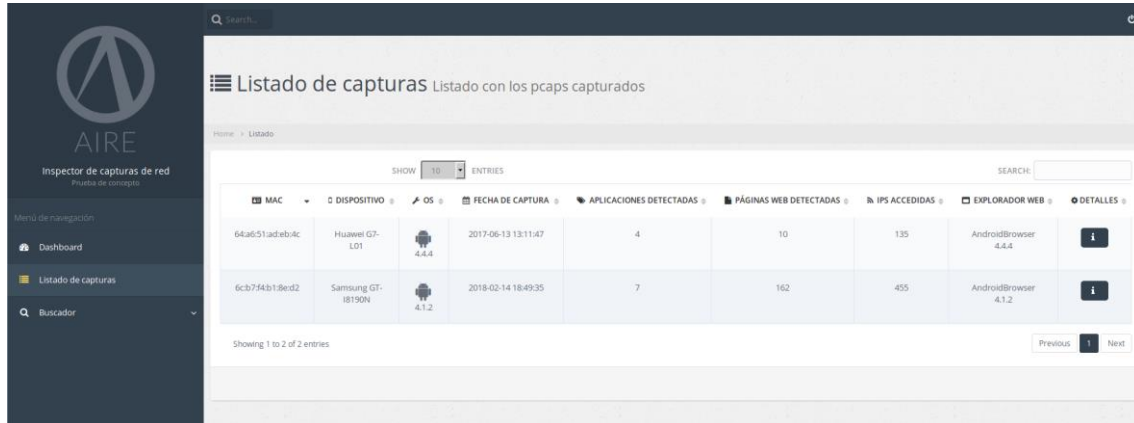


8. En el mapa de *localización de los servidores* podemos ver la ubicación de los servidores accedidos en las capturas y el número de accesos según el tamaño de las marcas.



Listado de capturas

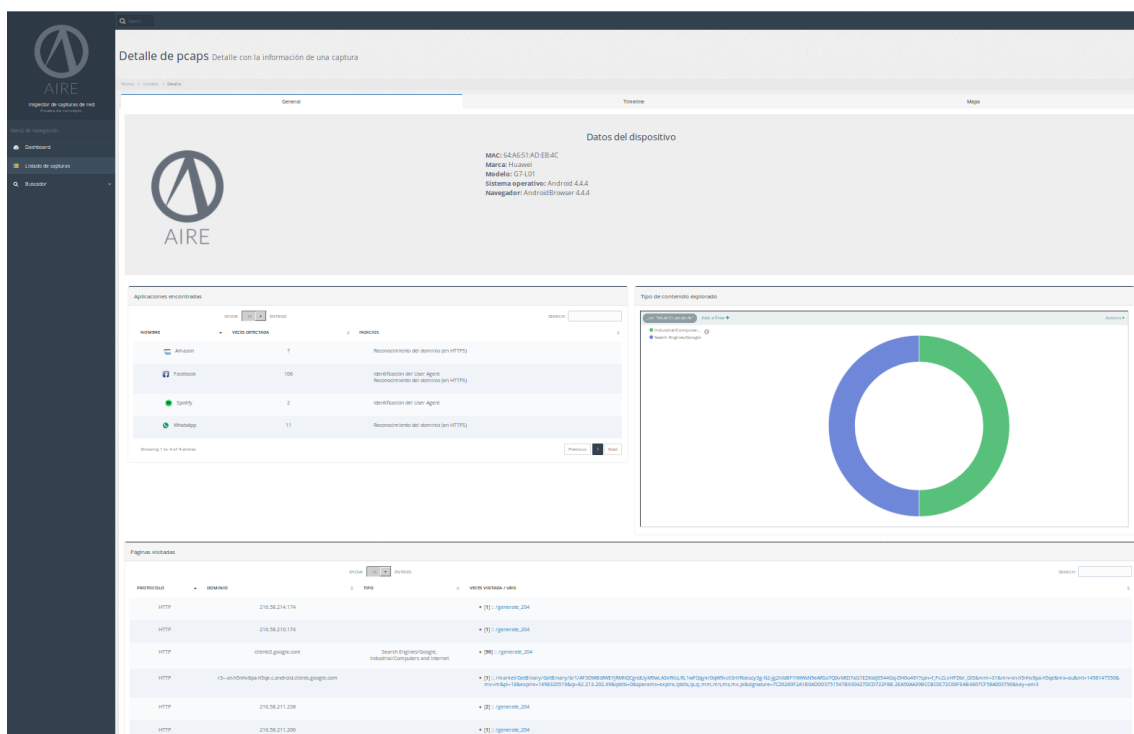
En el listado de capturas podemos ver una lista con datos básicos acerca de las capturas analizadas hasta el momento, como puede ser el dispositivo mayoritario detectado, la fecha y hora de la captura y otros datos más específicos como aplicaciones y páginas web detectadas, direcciones IP accedidas y el explorador web utilizado.



Podemos hacer click en el botón de detalles para ver la página específica de los datos de cada captura.

Detalle de información de la captura

En la página de detalle de las capturas analizadas tenemos tres diferentes pestañas disponibles: *General*, *Timeline* y *Mapa*.



Dentro de la pestaña general podemos ver, de forma parecida al *Dashboard*, varios elementos y gráficas:





9. Arriba tenemos algunos datos básicos acerca del dispositivo del que hemos realizado una captura.



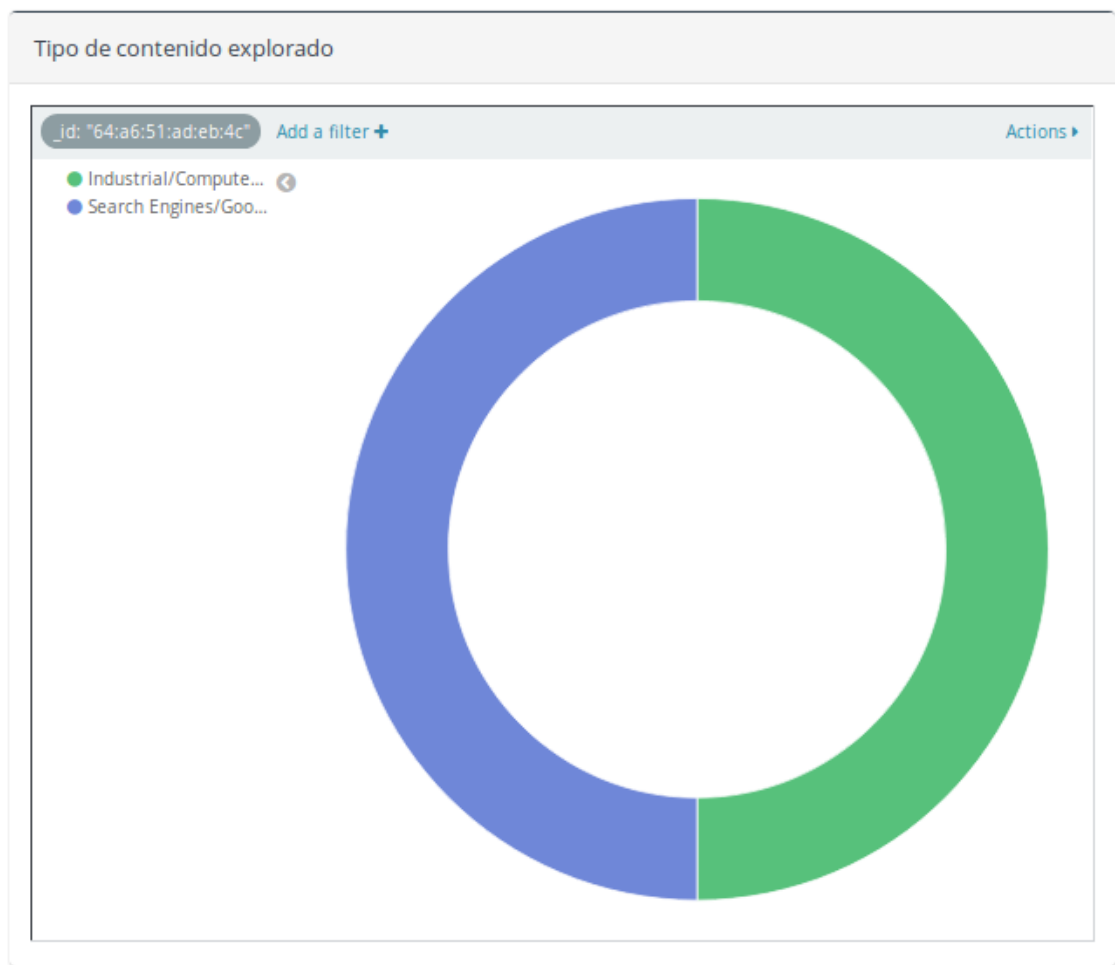
Datos del dispositivo

MAC: 64:A6:51:AD:EB:4C
Marca: Huawei
Modelo: G7-L01
Sistema operativo: Android 4.4.4
Navegador: AndroidBrowser 4.4.4

10. Tenemos a continuación el panel de *aplicaciones encontradas*, con una lista de aplicaciones que se han localizado dentro de la captura, las veces que se ha detectado y el tipo de indicio o indicios que han disparado la detección.

Aplicaciones encontradas		
SHOW 10 ENTRIES		SEARCH: <input type="text"/>
NOMBRE	VECES DETECTADA	INDICIOS
 Amazon	7	Reconocimiento del dominio (en HTTPS)
 Facebook	106	Identificación del User Agent Reconocimiento del dominio (en HTTPS)
 Spotify	2	Identificación del User Agent
 WhatsApp	11	Reconocimiento del dominio (en HTTPS)
Showing 1 to 4 of 4 entries		Previous 1 Next

11. En la gráfica de *tipo de contenido explorado* veremos las temáticas detectadas en las visitas web extraídas de la captura.



12. Por último, podemos ver las *páginas visitadas*, con una lista de los dominios y URIs visitados en la captura y el número de accesos a las mismas, así como la temática en caso de detectar alguna.
















Páginas visitadas

SHOW 10 ENTRIES SEARCH:

PROTOCOLO	DOMINIO	TIPO	VECES VISITADA / URIS
HTTP	216.58.214.174		[1] :: /generate_204
HTTP	216.58.210.174		[1] :: /generate_204
HTTP	clients3.google.com	Search Engines/Google, Industrial/Computers and Internet	[90] :: /generate_204
HTTP	r3--sn-h5nhv8pa-h5qe.c.android.clients.google.com		[1] :: /market/GetBinary/GetBinary/b/1/AF3DWBWWEY/RMhQCgndUyM9wLAsK0cLR1wFQykrQgW9vcl3nYRoecy3gN2-Jg2iibBFYIWWuN9eAR5o7QovMID7aG1EZKslj0544GgDH046Y7cpn-f_Fu2LVHFDbr_GIS8mm=31&mm=sn-h5nhv8pa-h5qe&ms=au&mt=1498147550&mv=m&pl=18&expire=1498320519&p=82.213.202.49&ipbits=0¶ms=expire.jpbits.jpq:mmmmms.mupl&signature=7C26260F2A1B0AD0D37515478630427DCD722F8B.2E450AA39BCBC0C72C68FEA84867CF58A003796&key=am3
HTTP	216.58.211.238		[2] :: /generate_204
HTTP	216.58.211.206		[1] :: /generate_204
HTTP	android.clients.google.com		[1] :: /market/download/Download?packageName=com.android.vending&versionCode=80793400&ch=zen2i1nk15s2swLcN16w&ssl=0&token=AOTCm0T1dVtEqy9ITbGk2ZghjFluoyKXuhBHFvesVSMET7_MRZAQb-DY6Q7Pj-J-weDcuU5iEqWT1CTBA86vbwzeinUq59bnMhnpTFF_SnUFUJdgsId97Eyle__ff_z7ghBmsEIh0nH9k3Mmf-78Mj2bzCzU95rZQnA2KZCukaQNSXZYG6IXDNX1B7UJVD-nzLtwig5XUmmms58NkryH4LoiSRATPWljw4C167OvF2wFQzQzoivOB8Z2Ww4D1ieBNDduvORgmWKAuh5IZWotMZRoPz524PlwaokrZee8Ob15Aj6jETJN-YatRRdip_ALyJND00CuwYofQVR41LUSLr9g8baseVersion=80793200&pf=3&cpn=f_Fu2LVHFDbr_GIS
HTTP	uts.kingapp.com		[2] :: /uts/install-channel-report?lang=es_ES&lang=es&client-version=2.0.5 [8] :: /report/apps/status
HTTP	api-v0.blockfolio.com		[13] :: /rest/updateAllCoins_v2?coin=WAVES&base=BTC&exchange=bittrex&coin=ETH&base=BTC&exchange=poloniex&coin=BTC&base=USD&exchange=bitfinex&token=c1on4FVlMwA.APA91bHuaU9woLyNaaBwjKjKq771MQJEML_EGROVSNYzMGyKdxD6B3LooQ6Dzhj0IMFwl0LE039GzdHUS0sB7omyH-F855Ruvca8pGGoPGH_qCVWoe92HNoTgfs-jdvoD [2] :: /rest/ico [21] :: /rest/announcements

General
Timeline
Mapa

Timeline del usuario

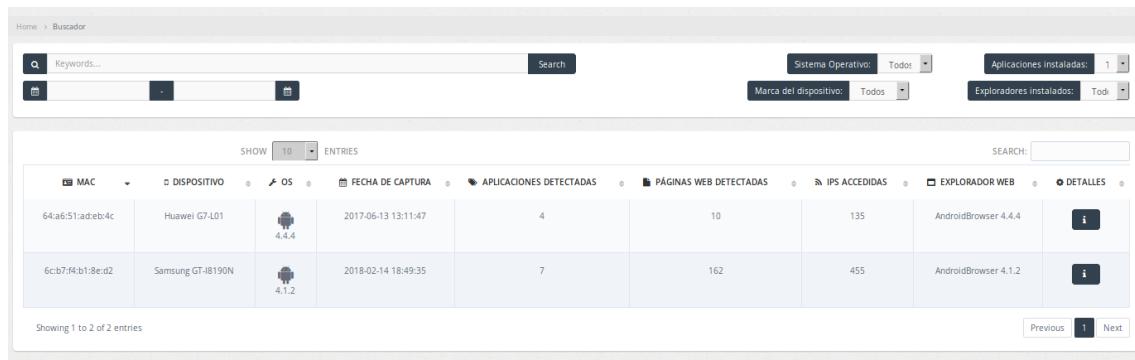
 Conexión de la aplicación Facebook 2017/06/22 14:09:57	 Conexión de la aplicación Facebook 2017/06/22 14:09:53
 Conexión de la aplicación Facebook 2017/06/22 14:09:49	 Conexión de la aplicación Facebook 2017/06/22 14:09:48
 Conexión de la aplicación Amazon 2017/06/22 14:09:47	 Conexión de la aplicación Facebook 2017/06/22 14:09:35
 Conexión de la aplicación Facebook 2017/06/22 14:09:33	 Conexión de la aplicación Amazon 2017/06/22 14:09:21
 Conexión de la aplicación WhatsApp 2017/06/22 14:09:20	 Conexión de la aplicación Facebook 2017/06/22 14:09:08
 Conexión de la aplicación WhatsApp 2017/06/22 14:09:07	 Conexión de la aplicación Facebook 2017/06/22 14:09:05
 Conexión de la aplicación Facebook 2017/06/22 14:08:55	 Conexión de la aplicación Facebook 2017/06/22 14:08:44
 Conexión de la aplicación Facebook 2017/06/22 14:08:42	Conexión con el dominio http://r3--sn-h5nhv8pa-h5qe.c.android.clients.google.com Se ha realizado una conexión con http://r3--sn-h5nhv8pa-h5qe.c.android.clients.google.com/market/GetBinary... 2017/06/22 14:08:40
Conexión con el dominio http://android.clients.google.com Se ha realizado una conexión con http://android.clients.google.com/market/download?packageName=com... 2017/06/22 14:08:38	Conexión con el dominio http://clients3.google.com 2017/06/22 14:08:37

General Timeline Mapa

Mapa de geolocalización de las conexiones

Por último, podemos acceder desde el panel lateral a un buscador que nos permitirá mostrar solamente las capturas que coincidan con unos ciertos criterios de búsqueda que podemos encontrar y modificar en la parte superior.

Podremos buscar las capturas, por ejemplo, en las que se haya detectado una cierta aplicación o que estén en un rango específico de fechas.



The screenshot shows a web interface for application detection. At the top, there's a search bar and filters for 'Sistema Operativo' (Todos), 'Aplicaciones instaladas' (1), 'Marca del dispositivo' (Todos), and 'Exploradores instalados' (Todos). Below this is a table with columns: MAC, DISPOSITIVO, OS, FECHA DE CAPTURA, APLICACIONES DETECTADAS, PAGINAS WEB DETECTADAS, IPS ACCEDIDAS, EXPLORADOR WEB, and DETALLES. The table contains two entries:

MAC	DISPOSITIVO	OS	FECHA DE CAPTURA	APLICACIONES DETECTADAS	PAGINAS WEB DETECTADAS	IPS ACCEDIDAS	EXPLORADOR WEB	DETALLES
64a651adeb4c	Huawei G7-L01	4.4.4	2017-06-13 13:11:47	4	10	135	AndroidBrowser 4.4.4	[i]
6cb7f4b18ed2	Samsung GT-I8190N	4.1.2	2018-02-14 18:49:35	7	162	455	AndroidBrowser 4.1.2	[i]

At the bottom, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' buttons.

Agregar nuevas aplicaciones a la detección

El proyecto se apoya en una serie de ficheros de configuración contenidos en el mismo para realizar la detección de aplicaciones. Estos ficheros se localizan en la dirección *parsing/analyzers/data_files* desde la raíz del proyecto y pueden editarse con el fin de añadir más aplicaciones que quieran detectarse.

Asimismo, otra dirección relevante para esto es *airprofiling/airprofiling/static/img/app_icons*, que contiene los iconos de las aplicaciones a detectar, simplemente para que se muestren de forma distintiva en la aplicación.

A la hora de agregar una aplicación a detectar, podemos seguir estos pasos:

1. En el fichero de configuración *domain.json*, podemos encontrar un fichero formateado como *JSON*. Podemos agregar a la lista que puede verse bajo *app_domain* nuevos valores, que son el nombre de la aplicación como clave y una lista de los dominios que queramos relacionar con la misma como valor.

```
1 {
2   "app_domain": {
3     "Spotify": ["spotify.com", "scdn.co"],
4     "Facebook": ["fbcdn.net", "facebook.com", "fb.com"],
5     "Instagram": ["cdninstagram.com", "instagram.com"],
6     "WhatsApp": ["whatsapp.net"],
7     "SoundCloud": ["soundcloud.com"],
8     "Pinterest": ["pinterest.com"],
9     "Wallapop": ["wallapop.com"],
10    "Apple Maps": ["gspe19.ls.apple.com", "gspe19.ls.apple.com"],
11    "Habitacalia": ["habitacalia.com"],
12    "Idealista": ["idealista.com"],
13    "LINE": ["line.me"],
14    "AliExpress": ["aliexpress.com"],
15    "Amazon": ["amazon.com"],
16    "BBVA": ["movil.bbva.es", "openapi.bbva.com", "servicios.bbva.es",
17             "APIM-openbbva-LB-1325968154.eu-west-1.elb.amazonaws.com", "bbva.d3.sc.omtrdc.net",
18             "api.bbva.es", "api.grupobbva.com", "bancamovil.grupobbva.com"],
19    "Bankia": ["m.bankia.es", "smetrics.bankia.es",
20              "bankia.es.ssl.d3.sc.omtrdc.net", "bankia.es"],
21    "EspanaDuero": ["mapas.espanaduro.es", "espanaduro.es"],
22    "EVO Movil": ["serviciosmoviles.evobanco.com"],
23    "ING Direct": ["metrics.ing.es", "ingdirectspain.demdex.net", "ing.ingdirect.es",
24                  "ing.es.ssl.sc.omtrdc.net", "ing.es"],
25    "Caixabank": ["webm.caixabank.es", "webm.lacaixa.es", "m.caixabank.es",
26                  "m.lacaixa.es", "m8.lacaixa.es", "m3.lacaixa.es", "app.caixabank.com",
27                  "m6.lacaixa.es", "m3.caixabank.es", "lacaixa.es", "caixabank.es", "caixabank.com"],
28    "Banc Sabadell": ["bancsabadel1.mobi", "ems.bancsabadel1.com", "bancsabadel1.com"],
29    "Banco Santander": ["bancosantander.es.edgekey.net", "microsite.bancosantander.es", "ban
30  }
31 }
```


2. De forma similar, en el fichero *geoip.json* podemos encontrar en la estructura *JSON* una lista de aplicaciones a detectar y una palabra contenida en el número de sistema autónomo detectado por la base de datos *GeoIP*.

Por ejemplo, la aplicación *Bankia* es reconocida por *GeoIP* como parte del sistema autónomo *AS20748 Bankia S.A.* Podemos introducir simplemente la palabra *Bankia*, contenida en el nombre del AS.

```
1  {
2      "app_geoip": {
3          "Spotify": ["Spotify"],
4          "Facebook": ["Facebook"],
5          "WhatsApp": ["Softlayer"],
6          "Twitter": ["Twitter"],
7          "Telegram": ["Telegram"],
8          "SoundCloud": ["Soundcloud"],
9          "Pinterest": ["Pinterest"],
10         "Netflix": ["Netflix",
11                     "NETFLIX-ASN"],
12         "Bankia": ["Bankia"],
13         "ING Direct": ["ING Direct"],
14         "Banc Sabadell": ["Banco Sabadell"]
15     }
16 }
```

3. En el fichero de configuración *user_agent.json*, de forma parecida, podemos introducir una palabra contenida en el agente de usuario que queramos relacionar con la aplicación.

```

1  {
2      "app_ua": {
3          "Spotify": ["Spotify"],
4          "Facebook": ["facebook", "Facebook"],
5          "Instagram": ["instagram", "Instagram"],
6          "Facebook Messenger": ["orca", "MessengerForiOS"],
7          "SoundCloud": ["SoundCloud"],
8          "WhatsApp": ["whatsapp", "WhatsApp", "WChat"],
9          "Chrome": ["Chrome"],
10         "Firefox": ["Firefox"],
11         "Pinterest": ["Pinterest"],
12         "Wallapop": ["Wallapop"],
13         "Atresplayer": ["Atresplayer"],
14         "Apple Maps": ["com.apple.Maps"],
15         "Safari": ["com.apple.mobilesafari"],
16         "Netflix": ["netflix"],
17         "Habitaclick": ["Habitaclick"],
18         "Idealista": ["idealista"],
19         "LINE": ["LINE"],
20         "Samsung Browser": ["SamsungBrowser"],
21         "AliExpress": ["AliExpress"],
22         "Amazon": ["Amazon"]
23     }
24 }

```

4. Por último, en la carpeta es `airprofiling/airprofiling/static/img/app_icons` anteriormente nombrada podemos introducir una imagen en formato *PNG* cuyo nombre será el de la aplicación que hemos utilizado al agregarla en los ficheros de configuración en minúsculas. El sistema la relacionará de forma automática.

