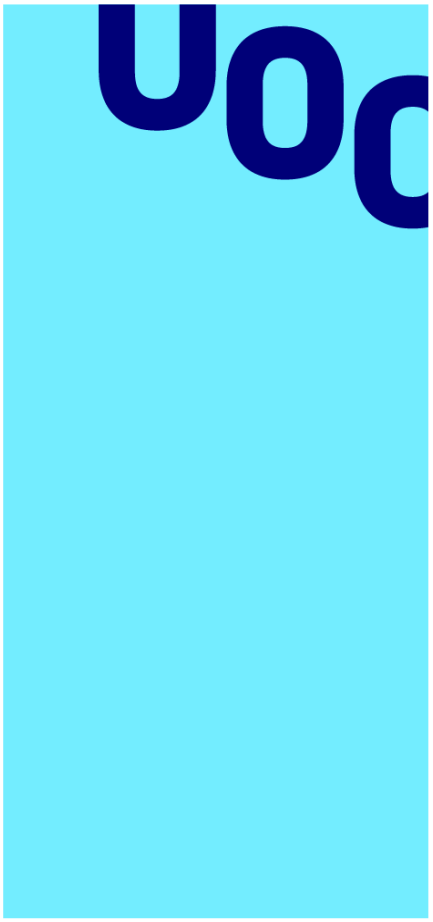
# OSINT: estudio, automatización e integración de diferentes herramientas para la obtención de información de fuentes abiertas

Un enfoque práctico para la eficiencia y efectividad



Universitat Oberta de Catalunya

**Brayan José Maeso Mateos**

Máster Universitario de Ciberseguridad y Privacidad
Trabajo de Fin de Master (Seguridad Empresarial)

**Nombre Tutor/a de TF**
Borja Guaita Pérez

**Profesor/a responsable de la asignatura**
Víctor García Font

**Fecha Entrega**
06/2023

Agradecimientos:

En primer lugar, quiero agradecer a mi familia y amigos por su apoyo incondicional y sus palabras de ánimo.

También, me gustaría expresar mi más sincero agradecimiento a todas las personas que contribuyeron de manera significativa en la realización de este Trabajo de Fin de Máster, en especial a mis tutores. Sin su apoyo, dedicación y aliento, este proyecto no habría sido posible.

**FICHA DEL TRABAJO FINAL**

| | |
|---|---|
| **Título del trabajo:** | *OSINT: estudio, automatización e integración de diferentes herramientas para la obtención de información de fuentes abiertas* |
| **Nombre del autor:** | *BRAYAN JOSÉ MAESO MATEOS* |
| **Nombre del consultor/a:** | *Borja Guaita Pérez* |
| **Nombre del PRA:** | *Víctor García Font* |
| **Fecha de entrega (mm/aaaa):** | *06/2023* |
| **Titulación o programa:** | Máster Universitario de Ciberseguridad y Privacidad |
| **Área del Trabajo Final:** | *Seguridad Empresarial* |
| **Idioma del trabajo:** | *Castellano* |
| **Palabras clave:** | *OSINT, Automation, Elastic Stack* |

**Resumen del Trabajo**

La finalidad de este trabajo es valorar las ventajas que ofrece la implantación, automatización y centralización de la información recopilada en un entorno enfocado en OSINT. Para ello se ha partido de la distribución Kali Linux sobre la cual se han desplegado las herramientas theHarvester, DMitry, ExifTool, Osintgram, OSRFramework y Spiderfoot. Se ha creado un script para poder delimitar los dominios sobre los que realizar las consultas y que estas se ejecuten automáticamente. Además, se ha incorporado un proceso de normalización de los datos obtenidos con el objetivo de desechar información irrelevante y centralizar todo el conocimiento en un mismo panel de forma visual. Después, se han utilizado las herramientas Filebeat, ElasticSearch y Kibana de la Pila Elastic para crear visualizaciones personalizadas de cada herramienta.

Finalmente, las conclusiones han sido satisfactorias en lo relativo a disponer de un entorno con estas características ya que, no solo posibilita ahorrar un tiempo y esfuerzo significativo, sino que permite adaptarse a las necesidades de cualquier tipo de usuario sin la necesidad de que este sea un experto en la materia. Además, siguiendo la metodología propuesta, la incorporación de nuevas herramientas se convierte en un proceso trivial.

**Abstract**

The purpose of this work is to assess the advantages offered by the implementation, automation, and centralization of the information collected in an OSINT-focused environment. To do this, we started with the Kali Linux

distribution, on which the tools theHarvester, DMitry, ExifTool, Osintgram, OSRFramework, and Spiderfoot have been deployed. A script has been created to delimit the domains on which to perform queries and to execute them automatically. In addition, a data normalization process has been incorporated to discard irrelevant information and centralize all the knowledge in a single visually accessible panel. Afterwards, the Filebeat, ElasticSearch, and Kibana tools from the Elastic Stack have been used to create custom visualizations for each tool.

Finally, the conclusions have been positive regarding the availability of an environment with these characteristics, as it not only allows significant time and effort savings but also enables adaptation to the needs of any type of user without requiring them to be an expert in the field. Furthermore, following the proposed methodology, the incorporation of new tools becomes a trivial process.

# Contenido

# Lista de figuras

# 1. Introducción

## 1.1. Contexto y justificación del Trabajo

En los últimos años, ha aumentado considerablemente la información que se expone en la red. Entre ella la información personal. Es decir, información que permite identificar a una persona ya sea con datos como su nombre, dirección, número de teléfono, etc. Son datos que un usuario común puede introducir en plataformas como son las redes sociales. Esto supone un grave riesgo para los individuos puesto que dicha información puede ser utilizada por cibercriminales para cometer delitos basados, por ejemplo, en la suplantación de identidad.

Por otra parte, la obtención de datos personales puede ser llevada a cabo para otro tipo de fines como son la elaboración de perfiles. Un ejemplo bastante representativo puede ser el que realizan las empresas para identificar posibles clientes potenciales o, con fines de investigación, para tratar de conocer mejor las actividades que realiza en la red un determinado candidato a un puesto.

Por tanto, es esencial proteger los datos personales en la era digital actual. Además, solo de esta forma es posible garantizar la seguridad y privacidad de los individuos.

De igual forma hay que añadir los recientes desarrollos relacionados con los dispositivos IoT y Big Data, los cuales han supuesto un aumento exponencial en la cantidad de datos disponibles. En consecuencia, los métodos de investigación basados en la recopilación y análisis de datos a partir de fuentes públicas (OSINT) se han convertido en un tema cada vez más relevante. Y, en parte, se debe a la gran accesibilidad que ofrecen dichos métodos puesto que no se requieren habilidades técnicas avanzadas ni costosos recursos para obtener información. Es decir, gracias a la gran cantidad de información disponible, OSINT se ha convertido en una herramienta muy útil para la investigación de muchos campos como son el periodismo, la inteligencia, los negocios, etc.

Este trabajo se centra en automatizar el proceso de recopilación de información mediante el uso de herramientas OSINT. Lo cual permite agilizar los procesos en el marco temporal y facilitar su uso para cualquier tipo de usuario sin conocimientos avanzados del tema.

Además, se pretende facilitar a un usuario común una visión global de la información que se facilita en la red. De esta forma, se consigue que cualquier usuario pueda tener conocimiento de la información pública que hay disponible en la red. Por tanto, esta funcionalidad es de suma importancia ya que, por ejemplo, una persona puede obtener la información que publica en la red y tomar las medidas que considere oportunas para evitar posibles riesgos de seguridad o privacidad.

No obstante, el resultado obtenido también puede ser de gran ayuda para otros ámbitos como el empresarial. Esto facilita procesos como son la investigación

de un determinado usuario para identificar si, efectivamente, se trata del candidato adecuado que la empresa puede buscar. Por ejemplo, la empresa podría obtener conocimiento acerca de los valores de una persona a través de lo que comparte en sus redes sociales y así determinar si le interesa incorporar o no a un empleado con dichos valores.

En definitiva, hay que tener bastante cuidado con la información que se sube a la red ya que, aunque pueda parecer que no es relevante, puede suponer el primer paso para obtener una gran información sobre uno mismo. Si a esto se le añade la capacidad de las nuevas herramientas para combinar información de diferentes fuentes, el resultado puede ser aún más preocupante. Además, a pesar de que estas herramientas se encuentran en constante desarrollo y sofisticación por el entorno tan cambiante en el que se enfocan, han demostrado que son realmente eficaces. Un claro ejemplo de que estos métodos llevan mucho tiempo utilizándose es que se pueden encontrar numerosos estudios e investigaciones realizadas sobre el tema. Concretamente, en el año 2013, se publicó un estudio en el que se advertía que es posible trazar un perfil detallado a partir de las publicaciones que le han gustado a un usuario [1] pudiendo, incluso, deducirse información relativa a la orientación sexual, etnia o preferencias políticas entre otras.

## 1.2. Objetivos del Trabajo

A continuación, se muestran los principales objetivos de este trabajo de fin de master:

**Objetivos de investigación:**

- ✓ Investigación de las técnicas y herramientas OSINT disponibles para la obtención de datos personales, información de redes sociales, etc.
- ✓ Investigación sobre las posibilidades que ofrecen los principales motores de búsqueda (Google, Bing…).
- ✓ Investigación de plataformas enfocadas en OSINT.
- ✓ Investigación de alternativas para la visualización de los datos.

**Objetivos de implantación:**

- ✓ Instalación y configuración de la distribución seleccionada y de las herramientas y dependencias necesarias.
- ✓ Aprender a utilizar las herramientas OSINT escogidas.
- ✓ Automatizar consultas e integrar las herramientas.
- ✓ Procesar y visualizar los resultados obtenidos a partir de las herramientas OSINT.
- ✓ Priorizar la eficiencia de los procesos involucrados y la eficacia de la información procesada y de los resultados obtenidos.

**Objetivos de Entrega:**

- ✓ Desarrollar y entregar los trabajos parciales en el periodo establecido.

- ✓ Desarrollar la memoria final del trabajo.
- ✓ Preparar una vídeo-presentación.

## 1.3. Impacto en sostenibilidad, ético-social y de diversidad

El compromiso ético y global resulta fundamental en este trabajo centrado en la OSINT puesto que, la información obtenida, puede tener implicaciones importantes en cuanto a la privacidad y los derechos humanos. En consecuencia, se ha establecido seguir las siguientes competencias:

- ✓ Hacer un uso adecuado de la información obtenida siguiendo las directrices y políticas éticas para investigadores.
- ✓ Sensibilidad en cuanto a las cuestiones de privacidad y seguridad, asegurando que la información obtenida no infrinja ningún derecho fundamental.
- ✓ Compromiso con la precisión y la imparcialidad en la recopilación y presentación de información.
- ✓ Habilidad para analizar y evaluar la información de manera crítica y reflexiva, y discernir la calidad y fiabilidad de las fuentes de información.
- ✓ Conocimiento y aplicación de técnicas de investigación ética, tales como la anonimización de datos y la obtención de consentimiento informado.

## 1.4. Enfoque y método seguido

El trabajo está enfocado en obtener toda la información disponible acerca de uno o varios objetivos determinados a través de técnicas de OSINT. Por tanto, la finalidad deseada es la de crear un informe lo más detallado posible y de fácil interpretación; el cual contenga datos como información de redes sociales, dominios, números de teléfono, etc. En definitiva, cualquier información disponible que pueda ser considerada de utilidad.

El proyecto está divido en dos partes:

- Parte teórica de investigación en la cual se realizará un estudio sobre conceptos como OSINT y SOCMINT (Social Media Intelligence) junto con las herramientas y distribuciones disponibles. Además, se investigarán las opciones de búsquedas avanzadas que ofrecen los motores de búsqueda convencionales (Google o Bing) y no tan convencionales (Búsquedas en la Deep Web). El resultado de esta fase se presentará en la PEC2.

- Parte práctica de instalación y configuración de la distribución y herramientas seleccionadas. También, se automatizarán las consultas y se integrarán las herramientas para obtener un resultado visual completo. Como resultado se presentará la PEC3.

Puesto que el tiempo es limitado para el desarrollo de todos los objetivos propuestos en la planificación presentada; puede darse el caso que coexistan

tareas del bloque teórico junto con tareas del bloque práctico. Lo cual, facilitará, en la medida de lo posible, ir avanzando en la fase de implementación.

Una vez realizada la fase de investigación, se analizará si la consecución de los objetivos propuestos es viable. Además, se podrá ajustar la planificación para que, en caso de que haya algún objetivo considerado inviable, siempre queden los objetivos de aprendizaje.

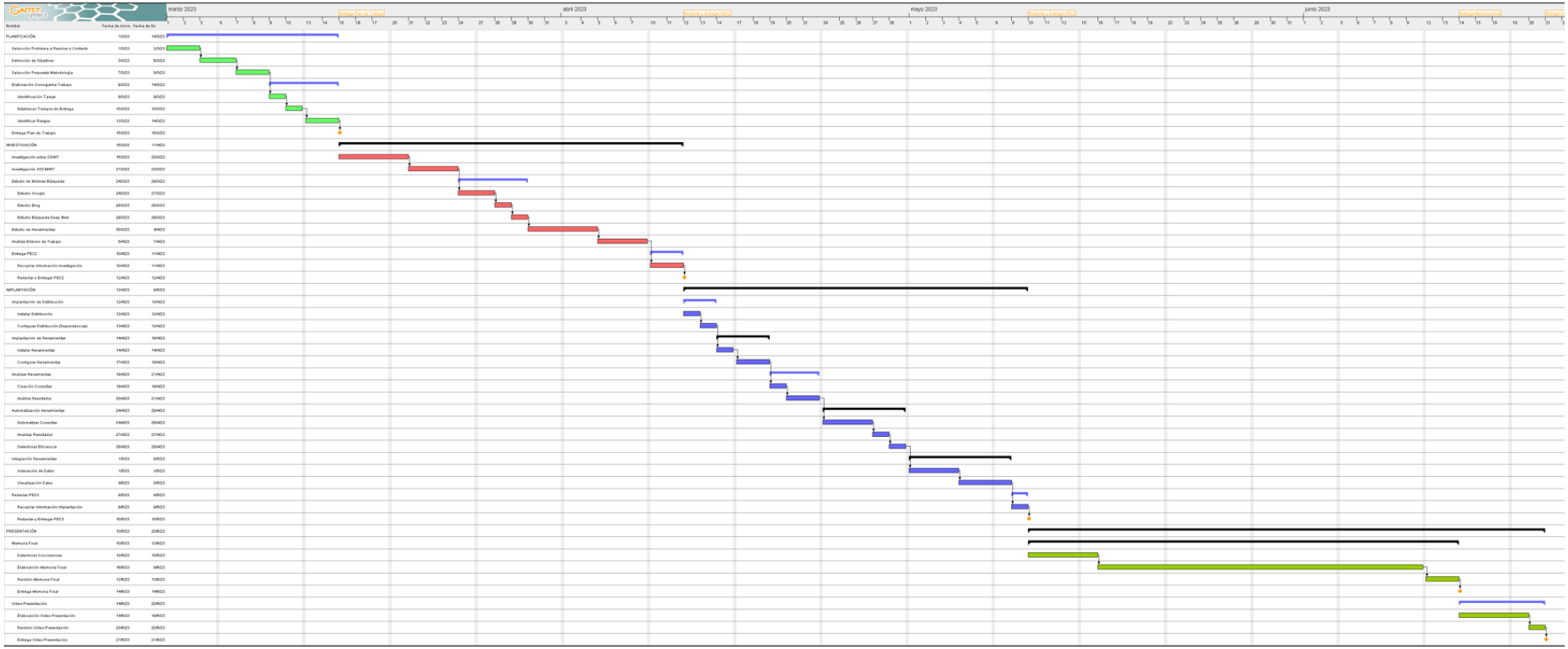
## 1.5. Planificación del Trabajo

| Id | Actividad | Inicio | Fin | Duración |
|---|---|---|---|---|
| 1 | Planificación | 1/3/23 | 14/3/23 | 10 |
| 1.1 | Selección Problema a Resolver y Contexto | 1/3/23 | 2/3/23 | 2 |
| 1.2 | Definición de Objetivos | 3/3/23 | 6/3/23 | 2 |
| 1.3 | Selección Propuesta Metodología | 7/3/23 | 8/3/23 | 2 |
| 1.4 | Elaboración Cronograma Trabajo | 9/3/23 | 13/3/23 | 3 |
| 1.4.1 | Identificación Tareas | 9/3/23 | 9/3/23 | 1 |
| 1.4.2 | Establecer Tiempos de Entrega | 10/3/23 | 10/3/23 | 1 |
| 1.4.3 | Identificar Riesgos | 13/3/23 | 14/3/23 | 2 |
| 1.5 | Entrega Plan de Trabajo | 14/3/23 | 14/3/23 | Hito |
| 2 | Investigación | 15/3/23 | 11/4/23 | 20 |
| 2.1 | Investigación sobre OSINT | 15/3/23 | 20/3/23 | 4 |
| 2.2 | Investigación SOCMINT | 21/3/23 | 23/3/23 | 3 |
| 2.3 | Estudio de Motores Búsqueda | 24/3/23 | 29/3/23 | 4 |
| 2.3.1 | Estudio Google | 24/3/23 | 27/3/23 | 2 |
| 2.3.2 | Estudio Bing | 28/3/23 | 28/3/23 | 1 |
| 2.3.3 | Estudio Búsqueda Deep Web | 29/3/23 | 29/3/23 | 1 |
| 2.4 | Estudio de Herramientas | 30/3/23 | 4/4/23 | 4 |
| 2.5 | Análisis Entorno de Trabajo | 5/4/23 | 7/4/23 | 3 |
| 2.6 | Entrega PEC2 | 10/4/23 | 11/4/23 | 2 |
| 2.6.1 | Recopilar Información Investigación | 10/4/23 | 11/4/23 | 2 |
| 2.6.2 | Redactar y Entregar PEC2 | 11/4/23 | 11/4/23 | Hito |
| 3 | Implantación | 12/4/23 | 9/5/23 | 20 |
| 3.1 | Implantación de Distribución | 12/4/23 | 13/4/23 | 2 |
| 3.1.1 | Instalar Distribución | 12/4/23 | 12/4/23 | 1 |
| 3.1.2 | Configurar Distribución (Dependencias) | 13/4/23 | 13/4/23 | 1 |
| 3.2 | Implantación de Herramientas | 14/4/23 | 18/4/23 | 3 |
| 3.2.1 | Instalar Herramientas | 14/4/23 | 14/4/23 | 1 |
| 3.2.2 | Configurar Herramientas | 17/4/23 | 18/4/23 | 2 |
| 3.3 | Analizar Herramientas | 19/4/23 | 21/4/23 | 3 |
| 3.3.1 | Creación Consultas | 19/4/23 | 19/4/23 | 1 |
| 3.3.2 | Análisis Resultados | 20/4/23 | 21/4/23 | 2 |
| 3.4 | Automatización Herramientas | 24/4/23 | 28/4/23 | 5 |
| 3.4.1 | Automatizar Consultas | 24/4/23 | 26/4/23 | 3 |
| 3.4.2 | Analizar Resultados | 27/4/23 | 27/4/23 | 1 |
| 3.4.3 | Determinar Eficiencia | 28/4/23 | 28/4/23 | 1 |
| 3.5 | Integración Herramientas | 1/5/23 | 8/5/23 | 6 |
| 3.5.1 | Indexación de Datos | 1/5/23 | 3/5/23 | 3 |
| 3.5.2 | Visualización Datos | 4/5/23 | 8/5/23 | 3 |
| 3.6 | Redactar PEC3 | 9/5/23 | 9/5/23 | 1 |
| 3.6.1 | Recopilar Información Implantación | 9/5/23 | 9/5/23 | 1 |
| 3.6.2 | Redactar y Entregar PEC3 | 9/5/23 | 9/5/23 | Hito |
| 4 | Presentación | 10/5/23 | 20/6/23 | 30 |
| 4.1 | Memoria Final | 10/5/23 | 13/6/23 | 25 |
| 4.1.1 | Determinar Conclusiones | 10/5/23 | 15/5/23 | 4 |

| | | | | |
|---|---|---|---|---|
| **4.1.2** | **Elaboración Memoria Final** | **16/5/23** | **9/6/23** | **19** |
| **4.1.3** | **Revisión Memoria Final** | **12/6/23** | **13/6/23** | **2** |
| **4.1.4** | **Entrega Memoria Final** | **13/6/23** | **13/6/23** | **Hito** |
| **4.2** | **Video-Presentación** | **14/6/23** | **20/6/23** | **5** |
| **4.2.1** | **Elaboración Video-Presentación** | **14/6/23** | **19/6/23** | **4** |
| **4.2.2** | **Revisión Video-Presentación** | **20/6/23** | **20/6/23** | **1** |
| **4.2.3** | **Entrega Video-Presentación** | **20/6/23** | **20/6/23** | **Hito** |

**Nota:** Para la planificación se han descartado los sábados y domingos como días de trabajo.

## 1.6. Planificación Temporal detalla en tarareas y procesos



## 1.7. Análisis de Riesgos

En el siguiente apartado se mencionan los posibles riesgos que pueden surgir durante la elaboración de este trabajo de fin de master. Además, estos riesgos pueden afectar al desarrollo correcto de la planificación establecida e, incluso, hacer que el proyecto fracase.

Riesgo 1: Objetivo demasiado ambicioso.

**Definición del riesgo:**
Puede darse la situación de que el trabajo no finalice en el tiempo establecido puesto que el objetivo del trabajo es bastante ambicioso para el plazo establecido. Prueba de ello es que trata temas que, por sí solos, pueden ser utilizados para la elaboración de un TFM completo como, por ejemplo:

- Investigación y configuración de herramientas dedicadas a OSINT.
- Estudio e implementación de una plataforma dedica a la búsqueda de información de fuentes abiertas.
- Investigación de las funciones avanzadas de los motores de búsqueda.

**Mitigación del Riesgo:**
Se debe tener especial cuidado en tratar de respetar los tiempos establecidos, acotando de manera precisa el alcance del trabajo.
En la parte de investigación no dedicar excesivo tiempo en valorar en demasiado detalle el alcance de cada herramienta disponible. En lugar de ello, mejor centrarse en determinar aquellas que ofrezcan una mayor compatibilidad y flexibilidad.
De ser necesario, se valorará reducir el alcance o abandonar la realización de un objetivo concreto. No obstante, se tratará de centrar dicho objetivo en su aprendizaje permitiendo finalizar el trabajo.

Riesgo 2: Problemas e incompatibilidad de las herramientas.

**Definición del riesgo:**
Hasta que no se realice la fase de investigación, no están definidas las herramientas sobre las que se va a trabajar. Esto puede suponer que se encuentren incompatibilidades entre las diferentes herramientas escogidas, lo cual pueda generar una pérdida de tiempo que repercuta de manera negativa en la planificación establecida.
Además, puede darse el caso de que se encuentren limitaciones técnicas en las herramientas seleccionadas como, por ejemplo, funciones reservadas únicamente para versiones de pago.

**Mitigación del riesgo:**
Asegurarse de en la fase de investigación que las herramientas seleccionadas ofrecen compatibilidad tanto con la distribución seleccionada como con las demás herramientas. De tal manera que se favorezca la posterior visualización de los datos obtenidos de manera conjunta.

Riesgo 3: Limitaciones técnicas.

**Definición del riesgo:**
Durante el manejo de las herramientas seleccionadas, puede darse el caso de que el funcionamiento de alguna herramienta no sea el esperado. Ya sea ofreciendo datos poco precisos en algunas situaciones o limitándose el uso de dicha herramienta a unas pocas búsquedas, que solo ofrezca características básicas en versiones gratuitas, etc.

**Mitigación del riesgo:**
Tratar de seleccionar herramientas que ofrezcan todas sus características de manera gratuita para limitar en la medida de lo posible encontrar limitaciones que solo están disponibles en versiones de pago.

Riesgo 4: Problemas relativos a la dependencia de datos.

**Definición del riesgo:**
El acceso a fuentes de datos confiables puede ser limitado o interrumpido durante la realización del trabajo, lo que puede afectar negativamente a la calidad de los resultados o al propio tratamiento de los datos.

**Mitigación del riesgo:**
Tratar de seleccionar fuentes de datos que ofrezcan resultados confiables en la medida de lo posible. Además, tener en cuenta que dichos datos pueden ser eliminados o modificados durante el desarrollo del trabajo lo que puede afectar a su posterior visualización.

Riesgo 5: Falta de tiempo para redactar la memoria.

**Definición del riesgo:**
Durante la elaboración del trabajo pueden darse muchas situaciones que conlleven un retraso en la planificación. De esta forma se podría acabar en una situación en la que el tiempo disponible para la redacción de la memoria final no fuera el suficiente.

**Mitigación del riesgo:**
Documentar, en la medida de lo posible, los avances que se vayan llevando a cabo para facilitar y agilizar la posterior elaboración de la memoria final.

## 1.8. Estado del Arte

A pesar de que este apartado se ampliará en la fase de investigación, se han realizado algunas búsquedas para profundizar más en el concepto de OSINT y SOCMINT. Además, se ha echado un vistazo a las distribuciones y herramientas que se encuentran disponibles y que pueden ser de utilizad para recabar información interesante.

En lo relativo a la búsqueda de información a través de los navegadores se ha investigado el uso de Dorks. Es decir, utilizar consultas de búsqueda avanzada que usan operadores específicos para obtener resultados más precisos y detallados. Esto supone una clara ventaja puesto que, gracias a estas consultas específicas, se consigue obtener información más precisa a la vez que se puede automatizar ciertas consultas. De esta manera, se puede ahorrar mucho tiempo en la realización de las búsquedas a la vez que se elimina información poco relevante.

Entre los navegadores más populares se encuentran Google, Bing, Yahoo, Baidu o DuckDuckGo entre otros. Concretamente, se centrará la investigación en los dos primeros ya que son los más utilizados actualmente.

Por otra parte, también se analizará la posibilidad de búsqueda de información a través de la Deep Web. Para ello, se ha valorará el uso de herramientas como TorBot, la cual es una herramienta escrita en Python que trata de obtener toda la información disponible de ciertos dominios con extensión onion.

En cuanto a las distribuciones que pueden ser de utilidad se han encontrado algunas como:

- Kali Linux: Se trata de una distribución basada en Debian GNU/Linux diseñada principalmente para realizar auditorías de seguridad. No obstante, trae incorporadas numerosas herramientas centradas en la obtención de información de fuentes abiertas como Maltego, Recon-ng, Shodan, etc.
- Buscador: Distribución basada en Ubuntu está enfocada en la recolección de información. Para ello incorpora una gran cantidad de herramientas OSINT.
- Osintux: Distribución basada en Ubuntu LTS y Debian. Es similar a “Buscador” con la diferencia de que está más enfocada para investigaciones de personas y organizaciones.
- Huron: Sistema basado en Linux que cuenta con las principales herramientas OSINT preinstaladas.

Para finalizar, muestran algunas herramientas que se han encontrado para valorar su posible uso en el trabajo:

- FOCA: Permite extraer metadatos de ficheros. Además, utiliza técnicas de Google y Bing hacking para descubrir archivos ofimáticos asociados a un determinado dominio.
- Osintgram: Se trata de una herramienta de OSINT para Instagram, la cual permite recopilar y analizar información de cuentas de usuarios de esta red social.
- Spiderfoot: Esta herramienta permite recopilar información de diferentes fuentes de datos. Entre la información que recaba se encuentran direcciones de correo electrónico, nombres de dominio, números de teléfono, etc.

- Maltego: Herramienta que trata de obtener toda la información disponible en internet sobre personas y empresas. Además, es ofrece la funcionalidad de cruzar datos para obtener perfiles de redes sociales.

Otras herramientas que se pretenden valorar son las siguientes: Recon-ng, Creepy, Metagoofil, MediaInfo, EmailHarvester, Aquatone, InstaLooter, Shodan etc.

## 1.9. Recursos necesarios y presupuesto del proyecto

| Recursos | Función | Coste |
|---|---|---|
| **Ordenador Personal** | • Redactar Memoria<br>• Instalar Distribución y Herramientas necesarias<br>• Desarrollar Scripts o Programas necesarios | 500€ |
| **Conexión a Internet** | • Permitir acceso a fuentes de datos (OSINT)<br>• Realizar búsquedas y consultas de datos | 30€/mes |
| **Distribución y Herramientas** | • Obtener información de fuentes abiertas<br>• Visualización de los datos obtenidos | 0€ |

**Nota:** La herramientas y distribución seleccionada no se incluyen como gasto en el presupuesto ya que se tratará de seleccionar aquellas que sean gratuitas. Además, de ser necesario, se podrá usar versiones gratuitas de las herramientas seleccionadas.

# 2. Fase de Investigación

## 2.1. Definición de OSINT y SOCMINT

El término OSINT, Open Source Intelligence, hace referencia al conjunto de técnicas y herramientas utilizadas para recopilar información de fuentes públicas. Todo ello, con la intención de analizar y convertir la información obtenida en conocimiento útil para una petición específica de inteligencia.

Además, al utilizar información abierta, existen una serie de ventajas muy a tener en cuenta como:

- ✓ Posibilidad de recopilar la información en tiempo real.
- ✓ Alta disponibilidad y fácil acceso a los datos.
- ✓ Bajo costo.

No obstante, se deben tener en cuenta algunos problemas que pueden surgir como:

- × Escasa importancia de la información obtenida.
- × Dificultad para determinar la credibilidad o fiabilidad de la información.

A continuación, se muestra la estructura general de OSINT en la que se agrupan los procesos que tienen lugar.

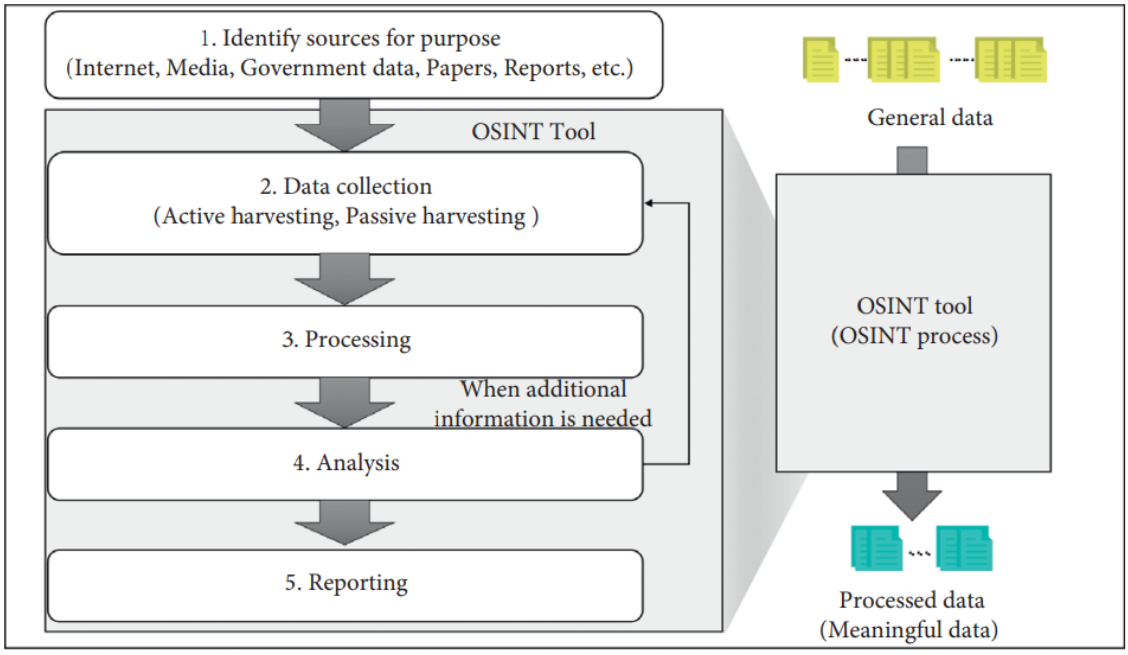



**Ilustración 1. Estructura de OSINT [2]**

Puesto que OSINT abarca una amplia gama de fuentes abiertas de información, se pueden encontrar muchas disciplinas derivadas como

SOCMINT (Social Media Intelligence), HUMINT (Human Intelligence), CYBINT (Cyber Intelligence), etc.

En concreto, para la realización de este trabajo, se da especial relevancia a la disciplina SOCMINT, la cual se enfoca en el análisis de la información obtenida a través de redes sociales, incluyendo conversaciones, mensajes, publicaciones, imágenes, videos y cualquier otro contenido generado por los usuarios en estas plataformas.

A modo de ejemplo, la información que un usuario publica en sus redes sociales puede ser considerada información pública. Por ello, cualquier usuario con una cuenta en esa misma red social podría acceder a dicha información con la intención de realizar un posterior análisis.

## 2.2. Metodología

Para la realización de este trabajo se ha elegido el ciclo de inteligencia propuesto por INCIBE [3] ya que permite definir de una manera clara y completa todas las fases presentes en el proceso de OSINT.

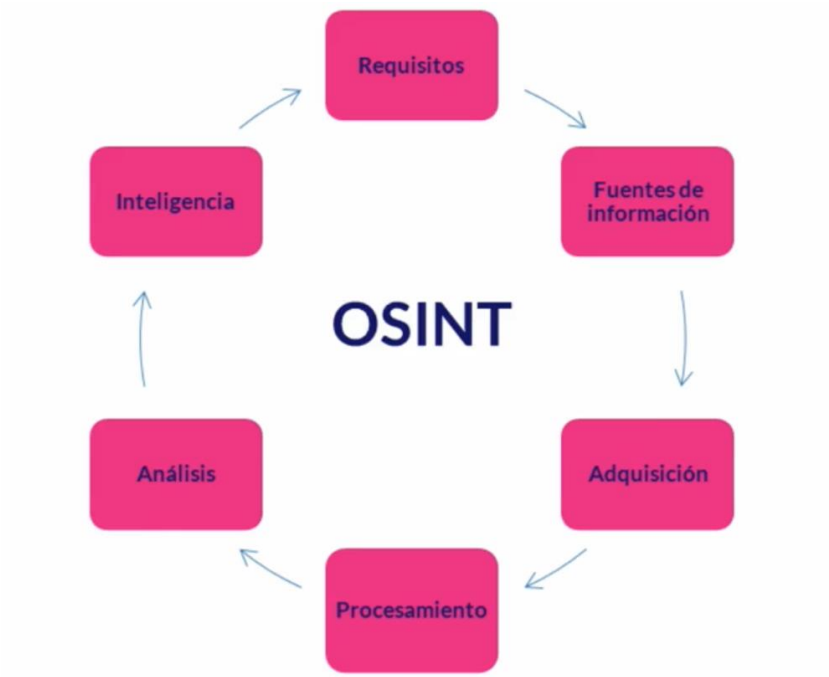



**Ilustración 2. Ciclo de Inteligencia INCIBE [4]**

En primer lugar, se encuentra la fase de requisitos en la cual se establecen los requerimientos que se quieren cumplir. En este caso, serían los requisitos definidos en los objetivos del trabajo y que se tienen en cuenta para las futuras fases. Concretamente, se parte de la base que se quiere obtener toda la información disponible que pueda ser de utilidad acerca de uno o varios objetivos determinados.

En la segunda etapa, se deben identificar las fuentes de información que se tendrán en cuenta para alcanzar los requisitos definidos en la fase previa. En este caso, el trabajo se centra en la obtención de información enfocada en el ámbito digital. No obstante, existe el problema de que en internet se puede encontrar una cantidad de información que resulte inabordable por lo que es necesario realizar un estudio para concretar dichas fuentes.

El periodo de adquisición se centra en recopilar la información a través de las fuentes especificadas. Por ello, se llevará a cabo un análisis exhaustivo de los motores de búsqueda y les herramientas OSINT disponibles, junto con el estudio de la distribución.

La fase de procesamiento tiene como principal fin establecer un determinado formato sobre toda información obtenida para facilitar su posterior análisis. Por consiguiente, se priorizarán los métodos y herramientas que favorezcan la recolección de datos en archivos de texto de fácil procesamiento.

La penúltima etapa puede ser considerada la más importante. En este punto de análisis se trata de generar inteligencia a través de los datos recopilados y procesados. Es decir, se trata la información conseguida de las diferentes fuentes con el objetivo de descubrir patrones que puedan aportar alguna conclusión significativa. En esta fase es donde se centraliza la información obtenida a través de los métodos y herramientas definidos previamente.

Por último, se muestra la información que se ha considerado útil. Todo ello con el objetivo de disponer de un informe que permita interpretar y explotar de manera eficaz las conclusiones descubiertas. Por tanto, se estudiarán las diferentes alternativas disponibles con la intención de mostrar la información de forma clara y concisa.

## 2.3. Motores de Búsqueda en la Web

Los motores de búsqueda son herramientas en línea que permiten a los usuarios buscar y acceder a información en la web. Para ello, utilizan complejos algoritmos y procesos automatizados a través de los cuales rastrean e indexan todo el contenido web, lo que permite ofrecer resultados relevantes y útiles a las consultas realizadas por los usuarios.

Debido a su capacidad para ayudar a encontrar información sobre cualquier tema, los motores de búsqueda se han convertido en unas herramientas imprescindibles en la actualidad. Algunos de los motores de búsqueda más populares incluyen Google, Bing, Yahoo! y DuckDuckGo.


**Ilustración 3. Principales motores de búsqueda web**

Sin duda alguna, los motores de búsqueda son de vital importancia en la obtención de información de fuentes abiertas ya que permiten acceder a una amplia variedad de información pública como noticias, sitios webs, foros, publicaciones en redes sociales y muchos otros tipos de contenido. Además, ofrecen opciones avanzadas de búsqueda mediante el uso de operadores específicos, los cuales permiten optimizar el resultado de las consultas.

Por otra parte, para los motores mencionados existen una gran variedad de Dorks. Es decir, consultas de búsqueda avanzada que usan operadores específicos para obtener resultados más precisos y detallados, lo que facilita su posterior análisis o filtrado de información relevante.

En definitiva, la importancia de los motores de búsqueda para OSINT radica en su capacidad para facilitar la recopilación y análisis de una gran cantidad de información en un tiempo eficiente. No obstante, existe una gran variedad de motores web por lo que se centrará el estudio, principalmente, sobre Google y Bing. Las razones de dicha elección se deben a que:

- ✓ Son los dos motores de búsqueda más utilizados en la actualidad. Por este motivo, la gente está familiarizada con su uso.
- ✓ Indexan mayor cantidad de datos. Concretamente, Google suele ofrecer muchos más resultados que otras alternativas.
- ✓ Hay disponibles una gran cantidad de listas de Dorks preparados. De esta forma, se facilita la obtención de un contenido específico.
- ✓ Google y Bing ofrecen algunos tipos de operadores particulares de cada motor para realizar búsquedas avanzadas. Por ello, los resultados pueden complementarse mediante búsquedas que ofrezcan distintos contenidos.

### 2.3.2. Búsqueda en la Deep Web

Como se puede ver en la siguiente imagen, la web está dividida en tres partes.


**Ilustración 4. Divisiones de la web**

La primera corresponde con la web superficial, la cual representa la parte accesible por los motores de búsqueda convencionales como Google o Bing. Y, en ella, se incluye todo el contenido que es indexado por estos motores de búsqueda por lo que es accesible para cualquier persona que tenga una conexión a Internet.

La segunda hace referencia a la Deep web, es decir, la parte de internet que no es indexada por los motores de búsqueda convencionales. Entre otras cosas, incluye contenido que está protegido por contraseña, contenido de bases de datos, contenido encriptado y contenido detrás de firewalls corporativos.

Por último, se encuentra la Dark web, también conocida como la web oscura. Es considerada la parte de la Deep web dedicada para fines ilegales por lo que se accede a ella a través de software específico.

Resumiendo lo planteado, la parte de la web que es indexada por motores de búsqueda como Google representa solo la punta del iceberg. Además, aunque parte de la Deep web se utiliza para fines ilegales (Dark web), gran parte de la web profunda contiene contenido legal y legítimo. Por consiguiente, esta parte representa una fuente rica y valiosa de información que es de gran utilidad para investigaciones OSINT por lo que no se debe pasar por alto.

Entre los motores de búsqueda que hay disponibles para navegar en la Deep web se encuentran algunos como Torch, Grams, DuckDuckGo Onion o Ahmia.

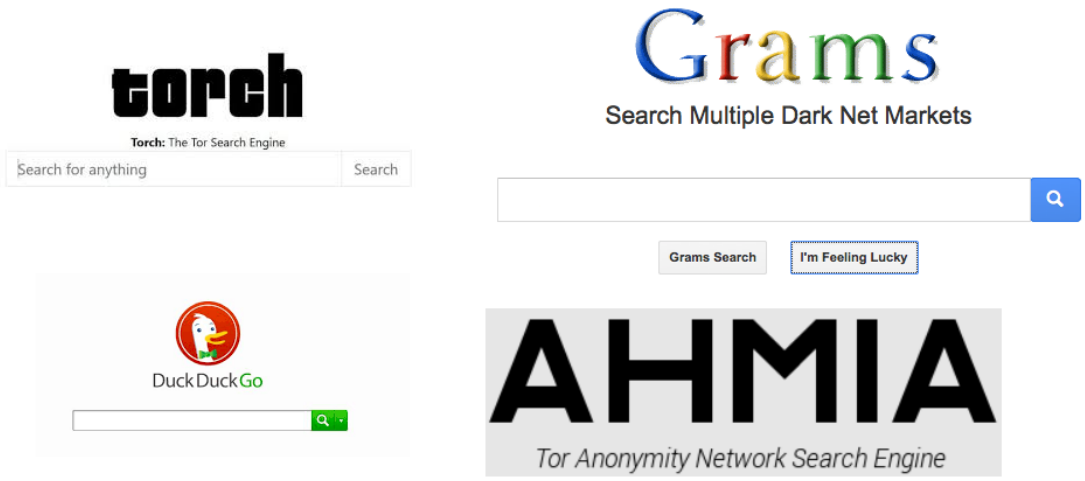



**Ilustración 5. Principales motores de búsqueda Deep Web**

En resumen, la Deep Web es una fuente valiosa de información que puede ayudar a los investigadores a obtener pruebas importantes en investigaciones de OSINT. Si bien puede ser más difícil acceder a la información, el esfuerzo puede valer la pena por la cantidad de información valiosa que se puede encontrar allí. Por este motivo, durante la realización de este trabajo se tratará de hacer uso de este tipo de motores de búsqueda para complementar los resultados obtenidos con los motores convencionales. Además, de esta forma se pretende obtener resultados más completos y fiables.

## 2.4. Herramientas para OSINT

Actualmente, existe una gran variedad de herramientas disponibles que se encargan de recopilar, analizar y presentar información pública disponible en internet y otras fuentes accesibles al público. No obstante, ante tanta variedad de opciones, surge la cuestión de cuál o cuáles elegir. Por ello, en este apartado se investigan las opciones que se han considerado más completas y que mejor pueden adaptarse a los objetivos definidos en el trabajo.

Por otra parte, aunque la información se puede extraer manualmente, es tal la cantidad que puede ser recolectada que prácticamente en la totalidad de los casos se vuelve necesario utilizar un software de apoyo. Además, el uso de este tipo de software no solo permite procesar toda la información obtenida sino que, en la mayoría de casos, ahorra un tiempo y esfuerzo significativo. En definitiva, debido a la gran cantidad de información que se encuentra disponible a día de hoy, se vuelve más que necesario el uso de herramientas que permitan agilizar dicho proceso.

En la selección de herramientas se ha decidido dividir su elección por la categoría en la que se engloban. De esta forma, se pretende disponer de un repertorio de herramientas en el cual se priorice que los resultados obtenidos puedan complementarse en lugar solaparse. Sin embargo, se tratará de complementar los resultados obtenidos en la medida de lo posible con el objetivo de reforzar la credibilidad y fiabilidad de la información recopilada.

Además, esta división por categorías facilita tanto la elección de herramientas como la definición de la información objetivo que se pretende obtener con cada una de estas.

En lo relativo a la obtención de información a partir de correos electrónicos se ha elegido **theHarvester** [5], la cual se trata de una herramienta de código abierto que se ejecuta en línea de comandos. Esta herramienta es bastante completa ya que permite obtener información desde otro tipo de fuentes como nombres, dominios y subdominios o direcciones IP entre otras. Para obtener dicha información hace uso de múltiples motores de búsqueda como Google, Bing, DuckDuckGo, LinkedIn o Twitter. También, permite realizar escaneos activos y pasivos, cuyos resultados se pueden guardar en un archivo XML o JSON.



**Ilustración 6. Herramienta theHarvester**

Más enfocada en la obtención de direcciones IP y dominios, se ha seleccionado a **DMitry** [6]. Se trata de una aplicación de código abierto y en línea de comandos que es capaz de recopilar información relevante sobre un host determinado. Esta herramienta puede realizar desde simples búsquedas whois hasta escaneos de puertos TCP o informes de tiempo de actividad. También, permite agregar nuevas funcionalidades. Por ello, este software representa una opción ideal para tratar de obtener toda la información acerca de un host.

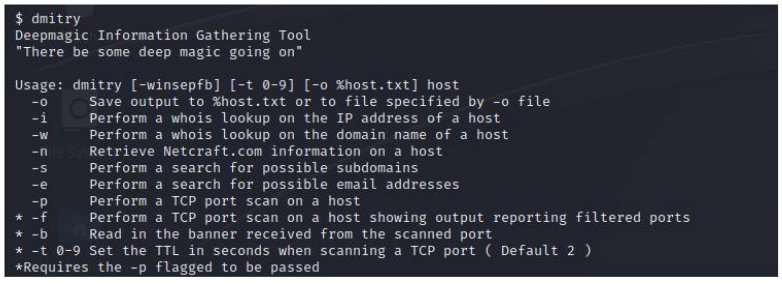



**Ilustración 7. Herramienta DMitry**

En cuanto a la obtención de metadatos se han seleccionado **ExifTool** [7] y **FOCA** [8]. La primera, se trata de un programa de código abierto que se ejecuta en línea de comandos. Esta herramienta, muestra información muy útil como el tipo de dispositivo con el que se generó el recurso multimedia, la fecha de creación, las coordenadas de localización, etc. Además, no solo permite obtener los metadatos de un archivo, sino que ofrece la posibilidad de manipularlos.


**Ilustración 8. Herramienta ExifTool**

De igual manera que **ExifTool**, **FOCA** es capaz de encontrar metadatos e información oculta en los documentos que analiza. Sin embargo, también permite realizar búsquedas de todo tipo de archivos que puedan contener metadatos sobre un determinado dominio. Para ello, hace uso de motores de búsqueda como Google, Bing o Exalead. Además, cuenta con una interfaz gráfica bastante intuitiva.

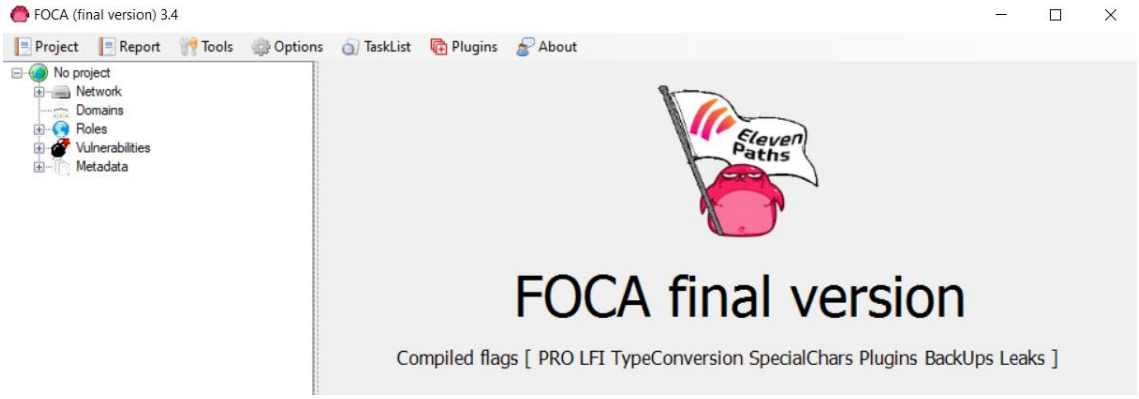



**Ilustración 9. Herramienta FOCA**

Otro software bastante interesante que se ha seleccionado es **OSINTGRAM** [9]. Esta herramienta ofrece, a través de línea de comandos, la posibilidad de obtener multitud de información de una determinada cuenta de Instagram. Entre dicha información se encuentran los seguidores, las publicaciones que le han gustado a un usuario, etc. De esta forma, se facilita enormemente el proceso de recolección y análisis de información de esta red social.

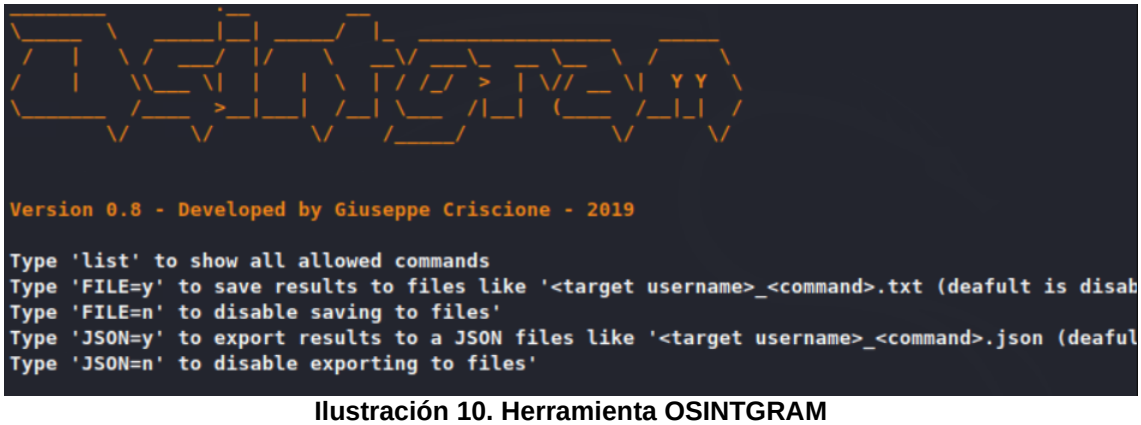



**Ilustración 10. Herramienta OSINTGRAM**

Por último, se han elegido las siguientes herramientas de tipo generalista. Es decir, herramientas las cuales ofrecen una gran cantidad de opciones y, por ello, es difícil agruparlas en una categoría concreta. Por tanto, la finalidad de estas es la de ayudar a obtener una visión global del objetivo u objetivos seleccionados complementando sus resultados con los obtenidos con los programas anteriores.

Una de ellas es **OSRFramework** [10]. Este software de código abierto agrupa un conjunto de bibliotecas, las cuales permiten automatizar tareas para la recolección de información de fuentes abiertas. Ofrece funcionalidades muy interesantes como la validación de nombres de usuario, búsqueda en la web profunda, extracción de expresiones regulares, generación de alias, búsquedas de correos y muchas otras. Además, los resultados se almacenan en archivos de texto o CSV.


**Ilustración 11. Herramienta OSRFramework**

Otra herramienta generalista es **Spiderfoot** [11], la cual permite automatizar búsquedas OSINT. Es de código abierto y cuenta con más de 200 módulos que permiten realizar búsquedas a través de muchos parámetros como nombre de dominio, dirección IP, número de teléfono o email entre otros. Además, ofrece una interfaz bastante clara e intuitiva a través de una interfaz web donde se pueden visualizar los resultados obtenidos.

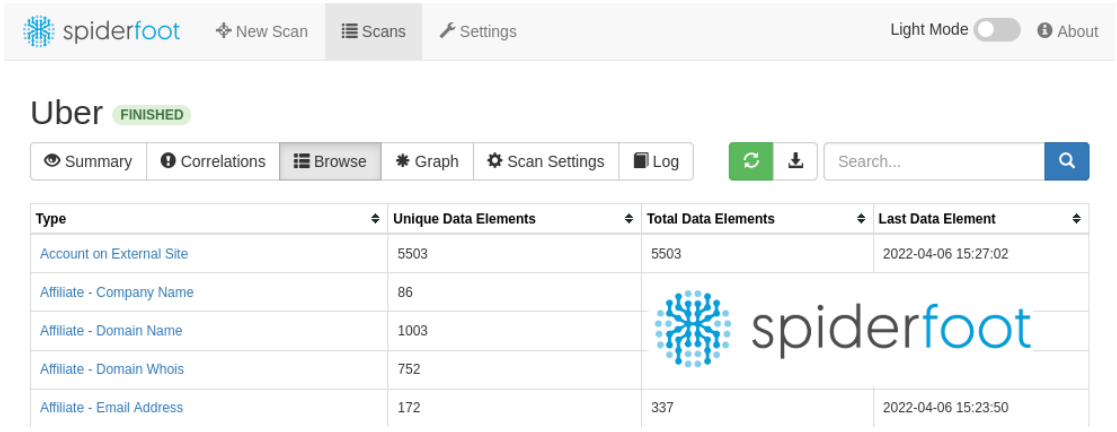



**Ilustración 12. Herramienta Spiderfoot**

Llegado este punto, se han definido las herramientas, las cuales se baraja su incorporación para el desarrollo de este trabajo de fin de master. A continuación, procedo a justificar el porqué de la elección de estas herramientas y no otras.

- ✓ Se tratan de herramientas de código abierto. De esta forma se permite disponer de todas las funcionalidades del software sin la necesidad de tener que pagar licencias o acogerse a versiones de prueba temporales. Por este motivo, no se han seleccionado otras alternativas como Maltego [12].

- ✓ La mayoría se encuentran disponibles en las principales distribuciones enfocadas a OSINT o resulta simple su instalación y configuración.

- ✓ Al ser herramientas bastante populares, se dispone de gran información sobre su uso en la red.

## 2.5. Distribuciones para OSINT

Tal y como se ha visto en el apartado anterior, existen muchas herramientas enfocadas en OSINT. Si a esto se le añade que dichas herramientas pueden funcionar en varias distribuciones, se complica el proceso, ya no solo de qué programas elegir, sino qué distribución usar.

Una posible opción es la de instalar un sistema operativo concreto, normalmente un sistema Linux. No obstante, puede que esta opción no sea la óptima puesto que ciertas herramientas conllevan un complicado o tedioso proceso de instalación. Además, siempre pueden surgir ciertos problemas, por ejemplo, con la instalación de dependencias o el uso de versiones concretas.

Por otra parte, han surgido distribuciones enfocadas en OSINT [13]. Es decir, sistemas operativos los cuales traen preinstaladas ciertas herramientas. Estos sistemas aportan una serie de importantes ventajas como es el ahorro de tiempo y esfuerzo en la búsqueda, selección e instalación de las herramientas.

Por consiguiente, usar distribuciones preparadas puede ser la mejor opción tanto para ahorrar tiempo y esfuerzo como para disponer de un mayor grado de flexibilidad ya que se dispone inicialmente de un gran repertorio de herramientas listas para su uso.

Entre las distribuciones más usadas para OSINT [14] se encuentran las siguientes:

- **Kali Linux** [15], es una distribución basada en Debian GNU/Linux. A pesar de que este sistema está enfocado principalmente para tareas de auditorías, es uno de las más utilizados para OSINT. Esto se debe a que es un sistema muy popular el cual incorpora una gran cantidad de herramientas para este uso como Maltego, theHarvester, DMitry, etc.


**Ilustración 13. Distribución Kali Linux**

- **Buscador**, se trata de una máquina virtual basada en Ubuntu. Se encuentra preconfigurada para realizar investigaciones en línea y cuenta con una gran cantidad de herramientas y extensiones configuradas en los navegadores. No obstante, tal y como se indica en su página oficial [16], lleva sin actualizarse desde enero de 2019 por lo muchas funciones se encuentran obsoletas.


**Ilustración 14. Distribución Buscador**

- **Huron** [17], es un sistema operativo de 64 bits basado en Debian. Ofrece una gran cantidad de herramientas enfocadas en OSINT como OSRFramework, Trape, Knock o theHarvester entre otras.


**Ilustración 15. Distribución Huron**

- **Osintux** [18], distribución basada en Ubuntu LTS y Debian. Es muy parecida a “Buscador” con la diferencia de que esta se enfoca más en investigaciones de organizaciones y personas. Además, cuenta con herramientas como OSRFramework, OSINTFramework, PIPL o Recon-NG entre muchas otras.

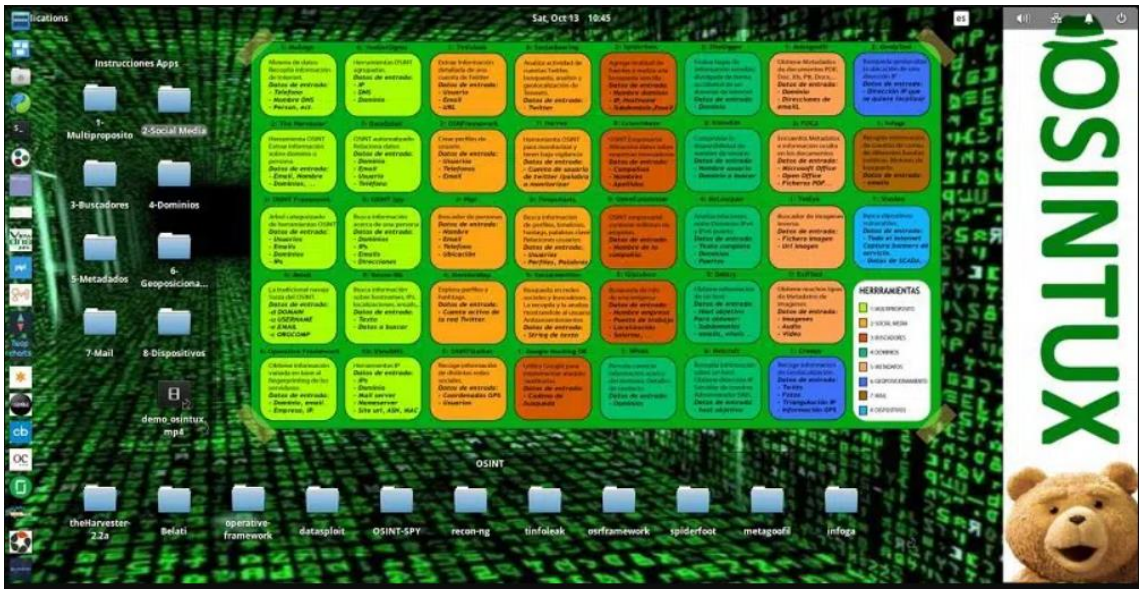



**Ilustración 16. Distribución Osintux**

Adicionalmente, existen muchas otras distribuciones las cuales no se han mencionado, pero sí que se han tenido en cuenta. Algunas de estas son **Dora OSINT VM** [19], **Tsurugi Linux** [20], **CSI Linux** [21] o **TRACE LABS OSINT VM** [22].

En conclusión, elegir una distribución preconfigurada permite ahorrar mucho tiempo y esfuerzo. Por este motivo, para la realización de este trabajo, se usará una de estas distribuciones mencionadas, concretamente Kali Linux. Los motivos de dicha elección son los siguientes:

- ✓ Se trata de una de las distribuciones más populares y conocidas en la actualidad. Por ello, cuenta con una amplia comunidad de usuarios junto con una gran cantidad de documentación y tutoriales disponibles, lo cual hace que este sistema sea fácil de aprender y utilizar.
- ✓ El mantenimiento del sistema se encuentra activo actualmente por lo que suelen lanzar nuevas versiones con bastante frecuencia. Por ello, esta distribución ofrece un entorno actualizado bastante seguro y que incorpora las últimas versiones de las herramientas que ofrece.

## 2.6. Análisis y Visualización de datos

A diferencia de lo que se puede pensar, esta fase es crucial es el proceso de OSINT. Esto se debe a que la efectividad de las conclusiones obtenidas está muy influenciada de si se realiza o no una correcta interpretación de los datos descubiertos en fases anteriores. Además, este proceso adquiere mayor relevancia si se tiene en cuenta que la inteligencia obtenida tras el análisis de los datos es utilizada para la toma de decisiones. Por consiguiente, es más que necesario disponer de un sistema que facilite el análisis e interpretación de los datos descubiertos.

Si bien es cierto que hoy en día la mayoría de las herramientas OSINT permiten recopilar una gran cantidad de información gracias a todas las fuentes y bases de datos que analizan. Sin embargo, muchas veces esta información se almacena en ficheros de logs difíciles de analizar e interpretar lo que dificulta en gran medida el paso más importante de las investigaciones OSINT, obtener conclusiones efectivas a través de los datos recogidos.

En definitiva, en este apartado se pretende facilitar, en la medida de lo posible, la correcta interpretación y obtención de conclusiones a partir de la información recopilada.

### 2.6.1. Elastic Stack

Puesto que la información que se quiere mostrar proviene de diferentes herramientas las cuales tienen sus propios formatos, es necesario buscar alguna alternativa que permita estandarizar dichos datos convirtiéndolos según se precise. Y, es en este punto, donde la pila Elastic se convierte en una de las mejores soluciones tanto para la ingesta de datos a medida como para disponer de un panel central de visualización personalizado y fácil de interpretar. En definitiva, Elastic Stack es un grupo de productos de código abierto diseñado para ayudar a los usuarios a tomar datos de cualquier tipo de fuente y formato, y buscar, analizar y visualizar esos datos en tiempo real.

En primer lugar, dentro de la pila Elastic se deben identificar cuatro componentes que se encuentran claramente diferenciados, los cuales ofrecen funciones concretas.

- **Logstash** [23], se trata de una herramienta que permite centralizar la recogida información. Además, ofrece multitud de opciones adiciones relacionadas con el procesamiento de los datos como la normalización y redistribución de la información, análisis completos, monitorización, alertas, etc.

- **Beats** [24], son agentes de datos ligeros que se utilizan para recopilar cierto tipo de información. Por consiguiente, realizan una función mucho más específica y limitada que Logstash. Entre los Beats oficiales disponibles se encuentran **Filebeat** [25] destinado a la recopilación de logs y otros datos, **Metricbeat** [26] para datos de métricas, **Packetbeat** [27] para datos de red, **Winlogbeat** [28] para logs de eventos de Windows, **Auditbeat** [29] para información de auditoría y **Heartbeat** [30] para monitoreo de tiempo de actividad. Adicionalmente a los Beats oficiales, existen multitud de Beats desarrollados por la comunidad [31].

- **ElasticSearch** [32], motor de búsqueda basado en la biblioteca Lucene. Este software permite almacenar de forma centralizada todos los datos para ofrecer resultados de búsqueda muy rápidos. Además, soporta una gran variedad de tipos de datos ya sean datos numéricos, de texto, geográficos, estructurados o no estructurados, etc.

- **Kibana** [33], se trata de una herramienta que permite la búsqueda y visualización de los datos indexados en ElasticSearch. Permite un alto grado de personalización, pudiendo crear paneles a medida a través de resultados de búsquedas específicas.

Una vez identificados los componentes, se muestran las posibles alternativas por las que se puede optar con la pila Elastic.

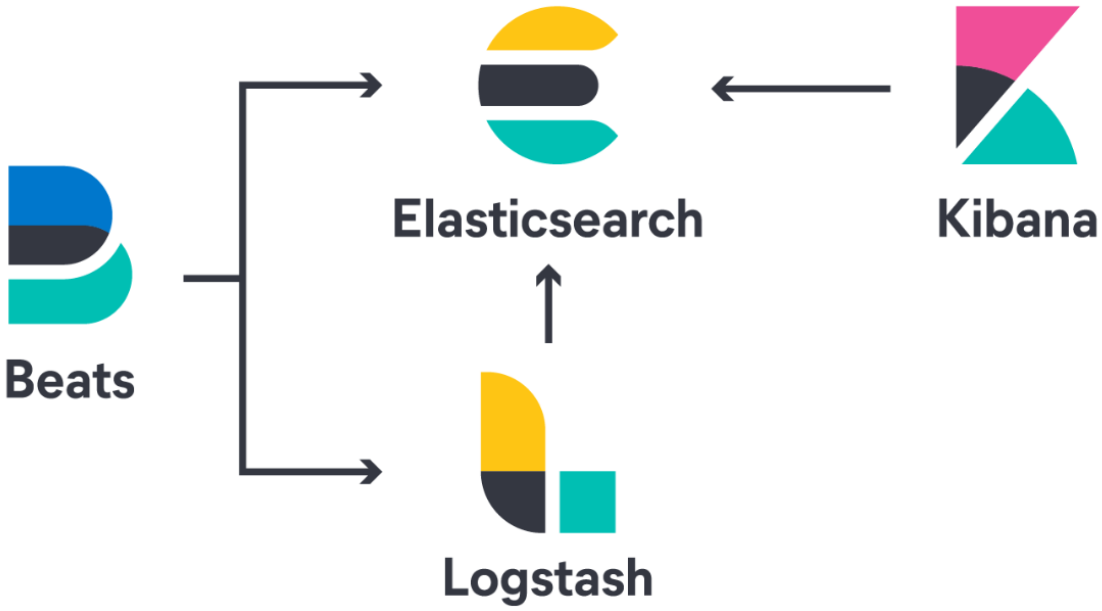



**Ilustración 17. Alternativas Pila Elastic**

En concreto, para la realización de este trabajo se han barajado estos dos posibles entornos.

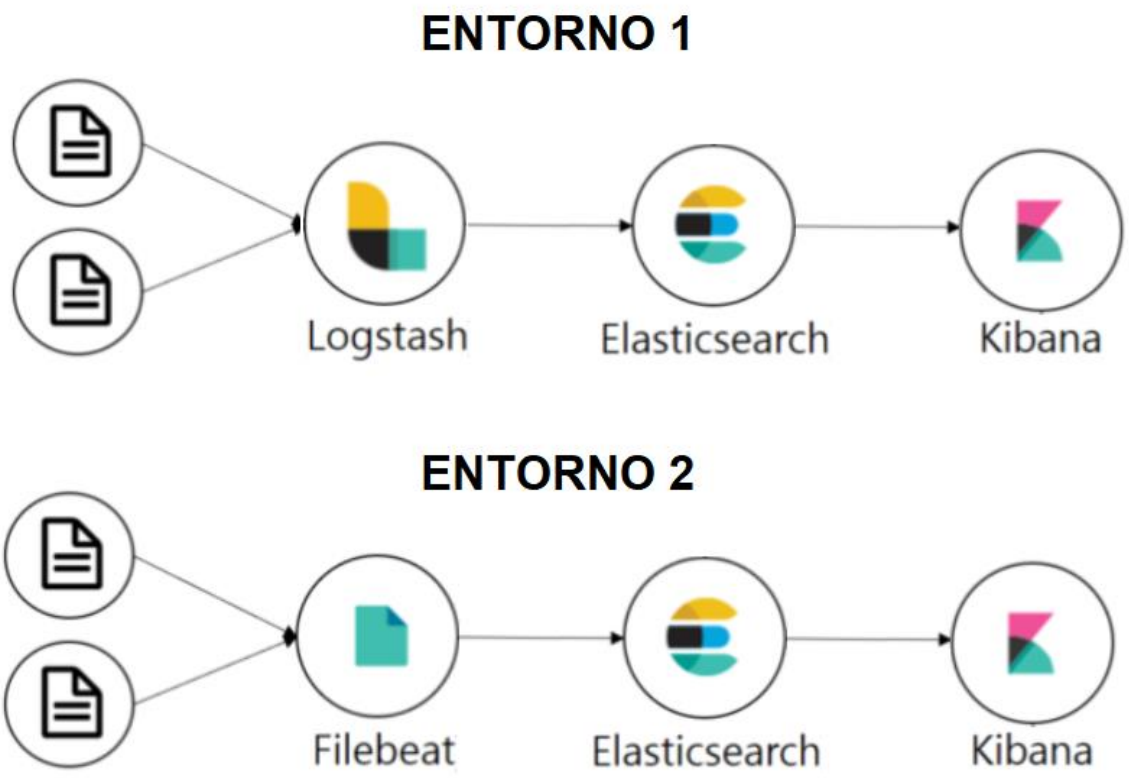



**Ilustración 18. Posibles entornos Elastic Stack**

En ambos entornos los datos son almacenados y procesados por ElasticSearch y visualizados por Kibana. No obstante, difieren en el método de recopilación de los registros. Mientras que en el primero se obtienen a través Logstash, en el segundo se utiliza Filebeat.

A continuación, se procede a analizar una serie de ventajas y desventajas de cada entorno.

- Puesto que Logstash en un software completo de recolección, puede ofrecer una mayor cantidad de opciones como es la salida de los datos a diferentes puntos o la indexación o enriquecimiento avanzado de los datos. No obstante, esta herramienta exige un alto consumo de recursos lo que puede ralentizar en gran medida el entorno.

- Filebeat, al tratarse de un agente ligero, puede correr sin consumir muchos recursos lo que puede ser crucial ya que al mismo tiempo pueden estar corriendo varias herramientas OSINT que necesiten la mayor parte de los recursos del sistema. Sin embargo, puede darse el caso que se necesite realizar operaciones de conversión o enriquecimiento de datos que no estén disponibles en este software, el cual se encuentra limitado a los módulos que trae incorporados.

En definitiva, cada escenario cuenta con una serie de ventajas y desventajas. En este caso, se ha considerado mejor el segundo escenario puesto que se quiere evitar una alta demanda de recursos, de tal forma que sea posible correr la distribución sin la necesidad de disponer de un sistema demasiado potente.

No obstante, aún falta por determinar si los módulos que trae incorporados Filebeat son suficientes para cumplir con los objetivos del trabajo. Además, se quiere analizar hasta qué punto el sistema puede verse ralentizado por el uso de Logstash. De esta forma, se probarán ambos entornos antes de realizar la elección definitiva.

# 3. Resultados

## 3.1. Implantación de la distribución

En este apartado se detalla el proceso de instalación y configuración del sistema Kali Linux. En concreto, se ha seleccionado la última versión (2023.1), la cual se puede descargar a través del siguiente enlace:

https://cdimage.kali.org/kali-2023.1/kali-linux-2023.1-installer-amd64.iso

Específicamente, se trabajará sobre un entorno virtualizado a través del software de código abierto y multiplataforma Oracle VM VirtualBox [34]. Con esto se consigue que, al utilizar un entorno virtualizado, la distribución Kali pueda ejecutarse sobre sistemas como Windows, macOS o Ubuntu entre otros. De esta forma, se facilita la instalación de este sistema ya que no es necesario disponer de un equipo dedicado para su uso.

Por otra parte, la página oficial ofrece imágenes del sistema preparadas [35] para importarse sobre VirtualBox. No obstante, se partirá de la imagen ISO, la cual ofrece un mayor grado de personalización del sistema.

En cuanto a la configuración del entorno sobre VirtualBox, se ha definido la máquina de tipo Linux-Debian 64 Bits con 6GB de RAM, 6 CPU y el adaptador de red en tipo NAT.

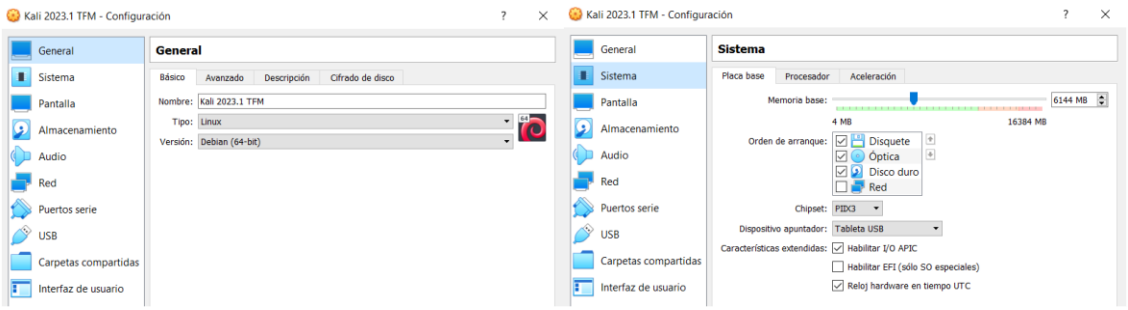



**Ilustración 19. Configuración VirtualBox Kali 1**

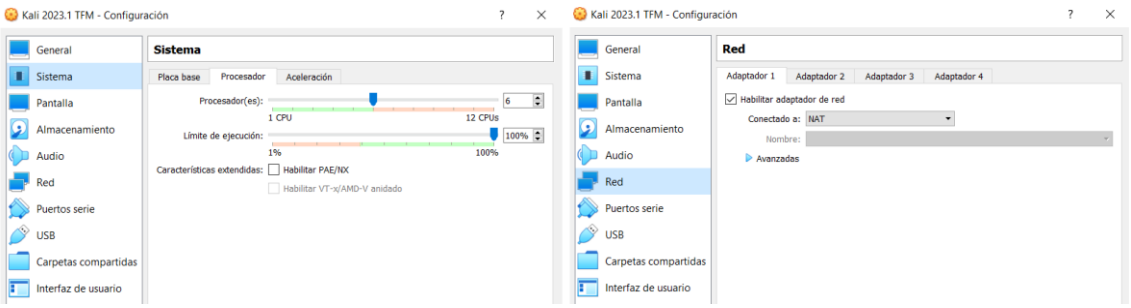



**Ilustración 20. Configuración VirtualBox Kali 2**

### 3.1.1. Actualizar el sistema operativo

Tras haber instalado la distribución, la siguiente tarea consiste en actualizar el sistema operativo para contar con las últimas versiones de los paquetes. Para ello se usan las siguientes instrucciones.

```
# sudo apt-get update
# sudo apt-get upgrade
```

## 3.2. Implantación de las herramientas

### 3.2.1. TheHarvester

Esta herramienta se encuentra preinstalada en el sistema Kali en su versión 4.2.0. Por este motivo, no es necesario llevar a cabo ningún proceso adicional de instalación.

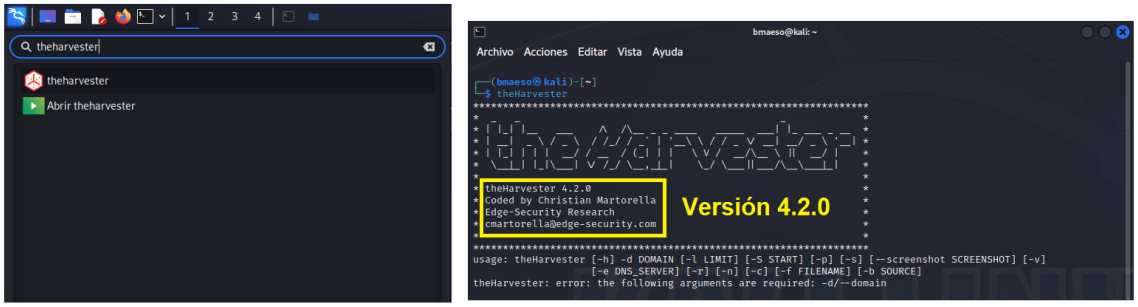



**Ilustración 21. Herramienta theHarvester disponible en Kali**

### 3.2.2. DMitry

DMitry tampoco necesita ningún paso en su instalación ya que se encuentra preinstalado en Kali.

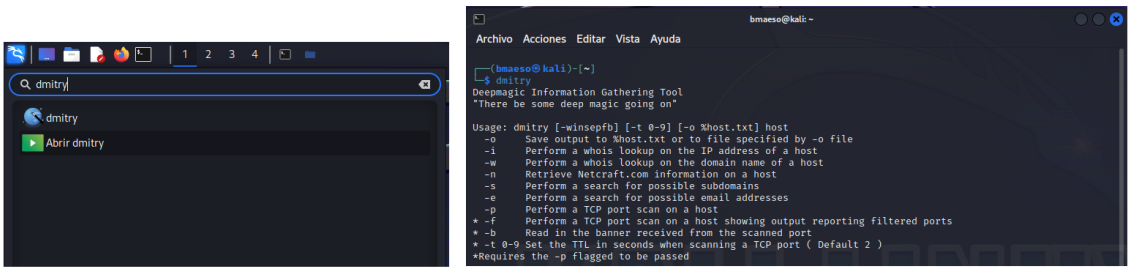



**Ilustración 22. Herramienta DMitry disponible en Kali**

### 3.2.3. FOCA

Este software no se encuentra disponible para esta distribución puesto que, al analizar su página oficial [8], se confirma que dicha herramienta solo se encuentra disponible para entornos Windows. Por ello, se desestima el uso de esta alternativa.

### 3.2.4. ExifTool

ExifTool está disponible de manera predeterminada en su versión 12.57.

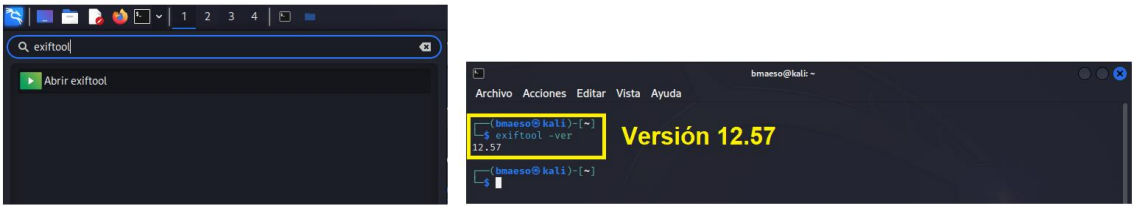



**Ilustración 23. Herramienta ExifTool disponible en Kali**

### 3.2.5. OSINTGRAM

Este software no se encuentra preinstalado por lo que se procede a descargar de su repositorio oficial de GitHub [9].

```
# git clone https://github.com/Datalux/Osintgram.git
```

Puesto que esta herramienta usa Python3 y necesita ciertos requerimientos, se creará un entorno virtual de Python para evitar la posibilidad de que las versiones requeridas de los módulos entren en conflicto con las actuales del sistema. Para ello, se debe instalar el paquete que permite operar con entornos virtuales.

```
# sudo apt install python3.11-venv
```

Después, se procede a crear el entorno virtual e instalar los requerimientos.

```
# cd Osintgram
# python3 -m venv venv
# pip install -r requirements.txt
```

Finalmente, se debe acceder dentro del directorio al archivo "*config/credentials.ini*" e introducir credenciales válidas de una cuenta de Instagram. Esto es necesario puesto que la herramienta necesita interactuar con la API de la red social para la cual es necesario acceder a través de una cuenta registrada.


**Ilustración 24. Configuración archivo de credenciales en OSINTGRAM**

En este punto, la herramienta se encuentra disponible y configurada para ser utilizada.

### 3.2.6. OSRFramework

OSRFramework no se encuentra instalado en el sistema, pero sí está disponible en los repositorios oficiales de Kali por lo que se puede instalar fácilmente a través de la siguiente instrucción.

```
# sudo apt install osrframework
```

Tras la ejecución del comando, se puede verificar que el software se encuentra instalado.


**Ilustración 25. Herramienta OSRFramework disponible en Kali**

### 3.2.7. Spiderfoot

Spiderfoot está disponible en el sistema en su versión 4.0.0.

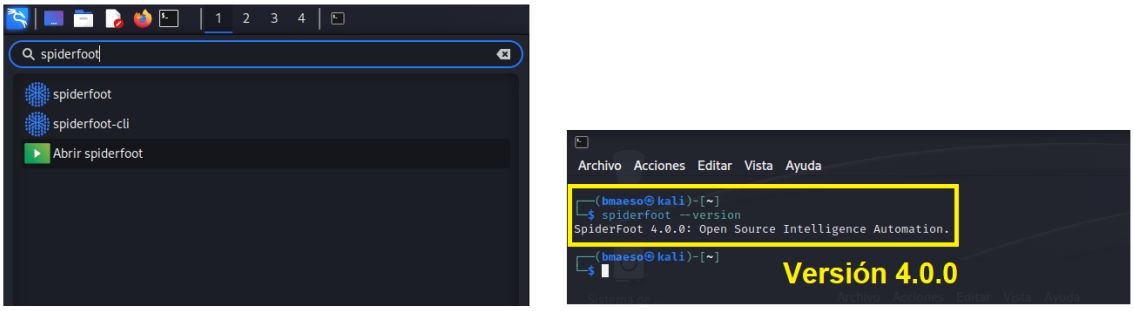



**Ilustración 26. Herramienta Spiderfoot disponible en Kali**

## 3.3. Funcionamiento de las herramientas

En este apartado se analiza el funcionamiento y resultados que ofrece cada herramienta. En general, se estudian los tipos de consultas disponibles para tratar de obtener información de utilidad. Además, se da prioridad al hecho de poder exportar los resultados en documentos de texto para poder ser procesados en fases posteriores.

En cuanto a los objetivos seleccionados, se han escogido aleatoriamente de tal forma que se permite mostrar cómo podrían integrarse resultados provenientes de diferentes herramientas y objetivos. Con ello se consigue trabajar sobre un escenario más complejo en el que resulta aún más necesario entender en su totalidad tanto el funcionamiento como resultados que ofrece cada herramienta.

### 3.3.1. TheHarvester

Concretamente se va a usar esta herramienta para obtener toda la información disponible acerca de un dominio determinado. En este caso, se analizará toda la información disponible de la plataforma *YouTube*. Para ello se hará uso de la siguiente instrucción en la cual se indica con la opción "*-d"* el dominio (YouTube), "*-l"* el límite de consultas (500), "*-b*" las fuentes sobre las que realizar las búsquedas (all) y "*-f*" para guardar la salida en un archivo (*theHarvester*).

```
# theHarvester -d youtube.com -l 500 -b all -f theHarvester
```

En definitiva, la consulta anterior realiza una búsqueda de toda la información disponible para el dominio "*youtube.com*", se queda con las 500 primeras entradas encontradas mediante la investigación en todas las plataformas registradas en su configuración y exporta el resultado a un archivo de texto.

Puesto que theHarvester trae incorporadas muchas plataformas de búsqueda, la información obtenida puede ser bastante extensa. Además, la herramienta ofrece estos resultados en un archivo JSON y XML.

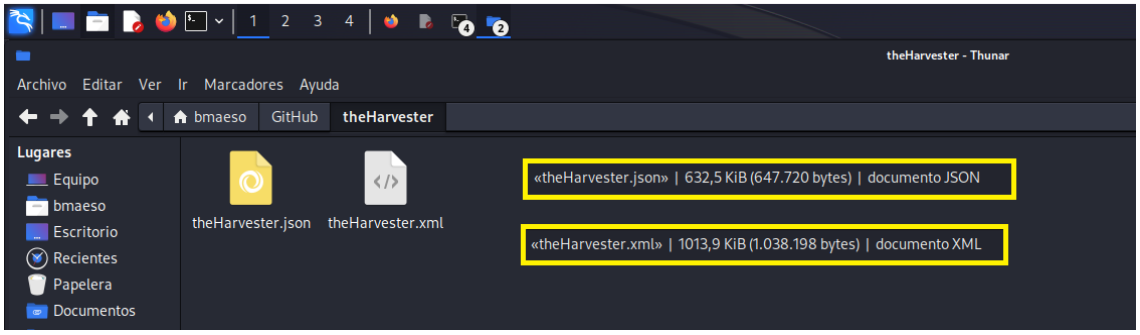



**Ilustración 27. Archivo JSON y XML generado por theHarvester**

Sin embargo, a pesar de que se ha encontrado bastante información, esta se encuentra en un formato muy poco visual lo que dificulta enormemente su fácil interpretación y explotación de los datos obtenidos. Por este motivo, será necesario procesar la información obtenida en fases posteriores para desechar la que no sea de utilidad y mostrar de manera más visual aquella información que se considere relevante.


**Ilustración 28. Información mostrada por theHarvester**

### 3.3.2. DMitry

Este software permite obtener una gran cantidad de información acerca de un host. En concreto, se utiliza dicha herramienta para mostrar todos los correos electrónicos disponibles de un dominio. Para ello, se usará la siguiente instrucción la cual trata de obtener todos los correos disponibles asociados al dominio "*Gmail.com*" y los guarda en un archivo de texto, "*dmitry_gmail.txt*".

```
# dmitry -e gmail.com -o dmitry_gmail
```

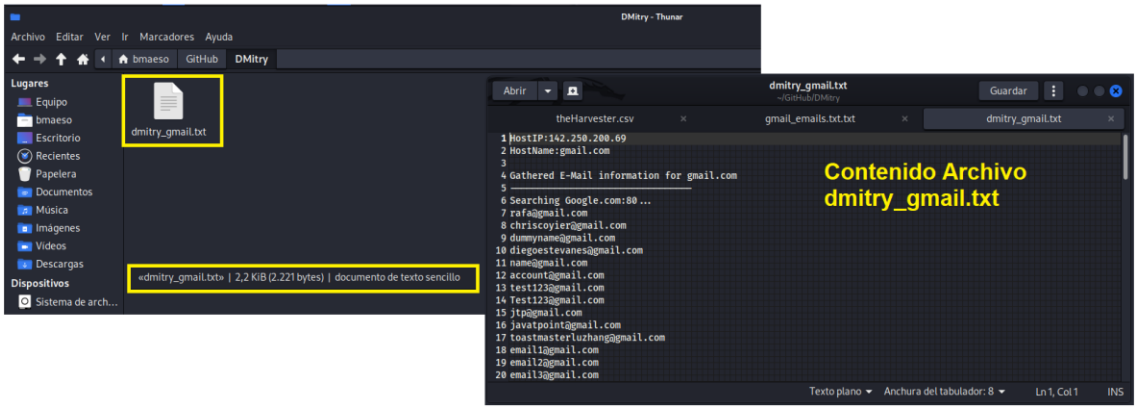



**Ilustración 29. Contenido del archivo generado por DMitry**

Sin embargo, el archivo no muestra solo las líneas con correos electrónicos, sino que contiene líneas que informan del proceso y que no son de utilidad para ser almacenadas. Por ello, se hace necesario llevar a cabo un proceso de normalización en el que se transforme dicho fichero en otro que solo contenga las líneas que muestran correos electrónicos.

### 3.3.3. ExifTool

ExifTool es una excelente herramienta para la extracción de datos de diversa cantidad de archivos. Por ello, en el desarrollo de este trabajo, se usa con la finalidad de extraer toda la información disponible sobre 15 imágenes de prueba descargadas de internet.


**Ilustración 30. Imágenes de prueba descargas de internet para ExifTool**

Para extraer todos los metadatos disponibles en las imágenes, se hace uso del siguiente comando, el cual permite guardar los resultados en un archivo de tipo CSV llamado "*exiftool_perros.csv*".

```
# exiftool -csv * > exiftool_perros.csv
```


**Ilustración 31. Contenido del archivo generado por ExifTool**

Sin embargo, el resultado obtenido es muy difícil de interpretar a simple vista por lo que resulta necesario procesarlo en apartados posteriores.

### 3.3.4. OSINTGRAM

OSINTGRAM se utiliza con la finalidad de obtener una gran cantidad de datos interesantes sobre la red social Instagram. Por ello, se hace uso de esta herramienta con la intención de poder elaborar un perfil detallado sobre un usuario.

No obstante, al utilizar la herramienta para tratar de obtener información acerca de un usuario, esta muestra un error de tipo timeout que se ha producido cuando se trata de acceder a la API de Instagram.

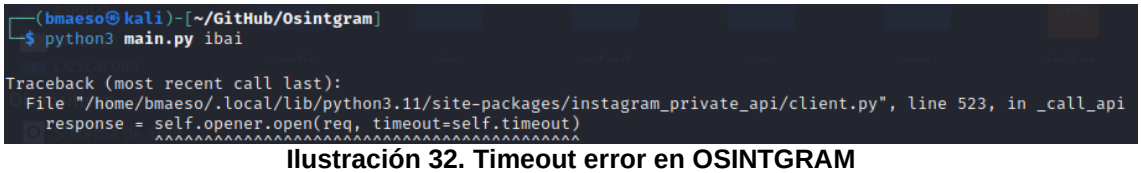



**Ilustración 32. Timeout error en OSINTGRAM**

Por otra parte, al acceder a la propia información del usuario que se ha creado sí que deja acceder. Por este motivo, se llega a la conclusión de que la API de esta red social está bloqueando las peticiones de la herramienta, es decir, la herramienta no se encuentra funcional a día de hoy. Además, tras haber hecho las llamadas a la API, Instagram ha suspendido la cuenta de pruebas que se ha creado.

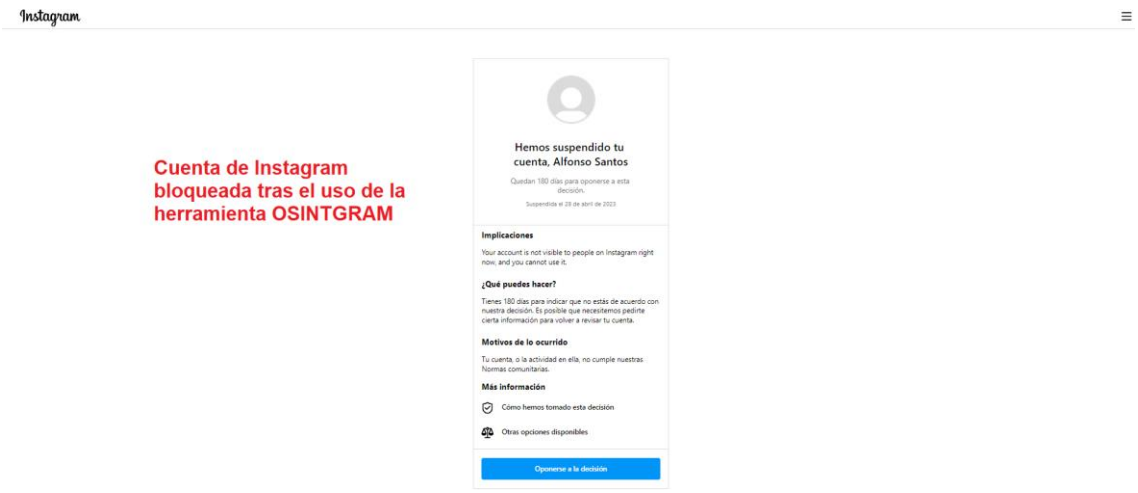



**Ilustración 33. OSINTGRAM, Mensaje de cuenta suspendida en Instagram**

En definitiva, a día de hoy se han tomado medidas en la API de Instagram para bloquear el uso de esta herramienta. Por consiguiente, se ha desestimado continuar utilizando este software.

### 3.3.5. OSRFramework

OSRFramework ofrece una gran cantidad de funcionalidades relacionadas con OSINT. Por este motivo, dicha herramienta se divide en varios módulos, cada uno con una función muy específica.

- **alias_generator**: Crea una lista de posibles alias en base a datos introducidos sobre una persona.
- **checkfy**: Encuentra posibles direcciones de correo electrónico en función de una lista de alias conocidos.
- **domainfy**: Comprueba la existencia de dominios, es decir, detecta dominios registrados.
- **mailfy**: Comprueba si existe un determinado correo.
- **phonefy**: Busca información sobre un determinado número de teléfono. Principalmente, trata de averiguar si dicho número está asociado a actividades maliciosas como, por ejemplo, spam.
- **searchfy**: Trata de encontrar cuentas a partir de palabras clave.
- **usufy**: Busca la existencia de un determinado usuario en cientos de plataformas.

Todos estos módulos son muy similares en cuanto a la salida que ofrecen. Además, dicha salida se corresponde con un archivo en formato CSV separado por comas. Por este motivo y porque la herramienta ofrece un uso muy genérico, se ha decidido centrar su utilización sobre el módulo “usufy”. Es decir, se usa OSRFramework con la intención de obtener todas las plataformas en las que está definido un usuario concreto. Para ello, se ha definido el siguiente comando que realiza la búsqueda del usuario “*ibaillanos*” en todas las plataformas.

```
# usufy.py -p all -n ibaillanos
```

El resultado de dicha instrucción genera un archivo llamado "*profiles.csv*".

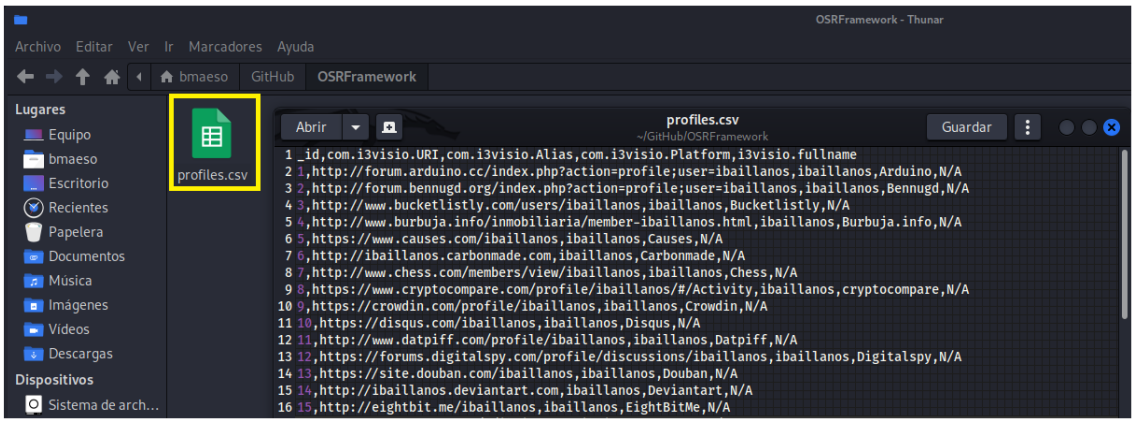



**Ilustración 34. Contenido del archivo generado por OSRFramework**

Este fichero contiene una línea por cada plataforma en la que se ha encontrado la existencia del nombre de usuario especificado, "*ibaillanos*". No obstante, este documento no es nada visual por lo que es necesario procesarlo más adelante.

### 3.3.6. Spiderfoot

Spiderfoot es una herramienta la cual permite recolectar una gran cantidad de información de diversas fuentes. Además, puesto que se trata de una herramienta muy completa, facilita búsquedas en función de la fase de investigación, de la información que se quiera obtener o de los módulos que se deseen utilizar.

En primer lugar, es necesario levantar la herramienta para lo que se usa el siguiente comando en el que se especifica que se va a utilizar de manera local a través del puerto 80.

```
# sudo spiderfoot -l 127.0.0.1:80
```

Después, se accede al enlace "*http://localhost/*" donde se ha definido un escaneo de tipo *Footprint*. Este escaneo permite obtener toda la información que el dominio objetivo definido, "*Google.com*", tiene expuesto en internet.

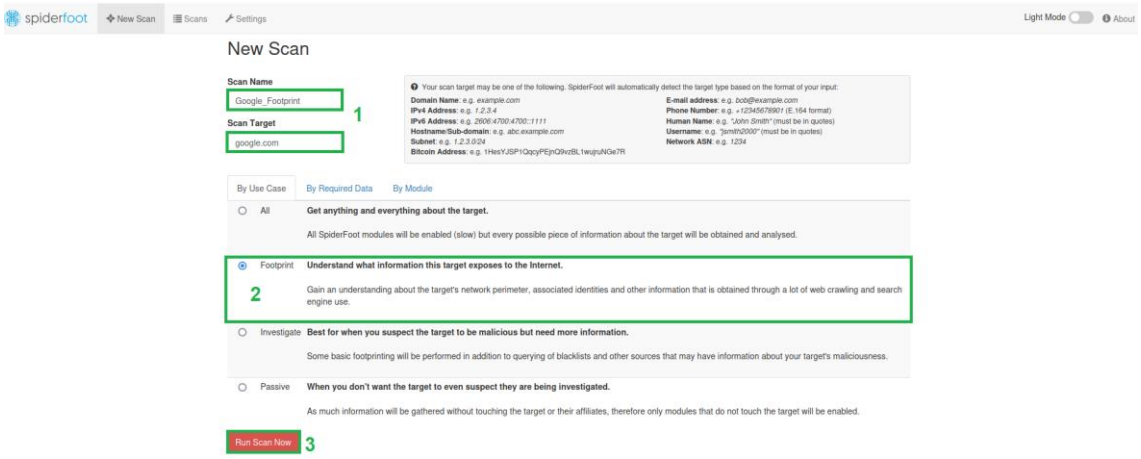



**Ilustración 35. Spiderfoot, Escaneo Footprint del dominio Google.com**

A continuación, se muestra el resultado obtenido tras el escaneo.

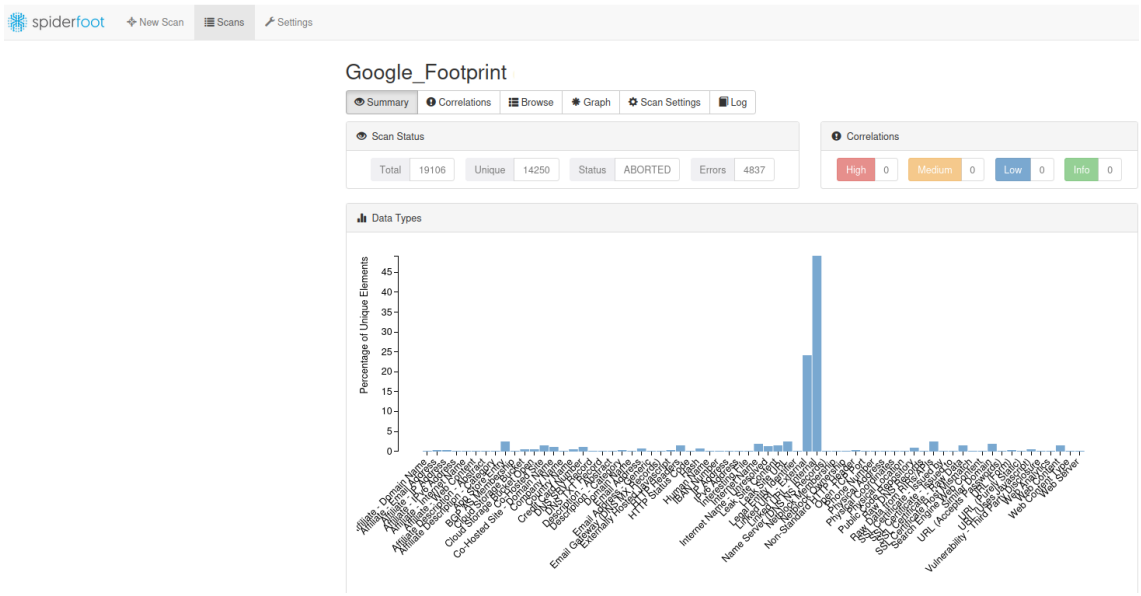
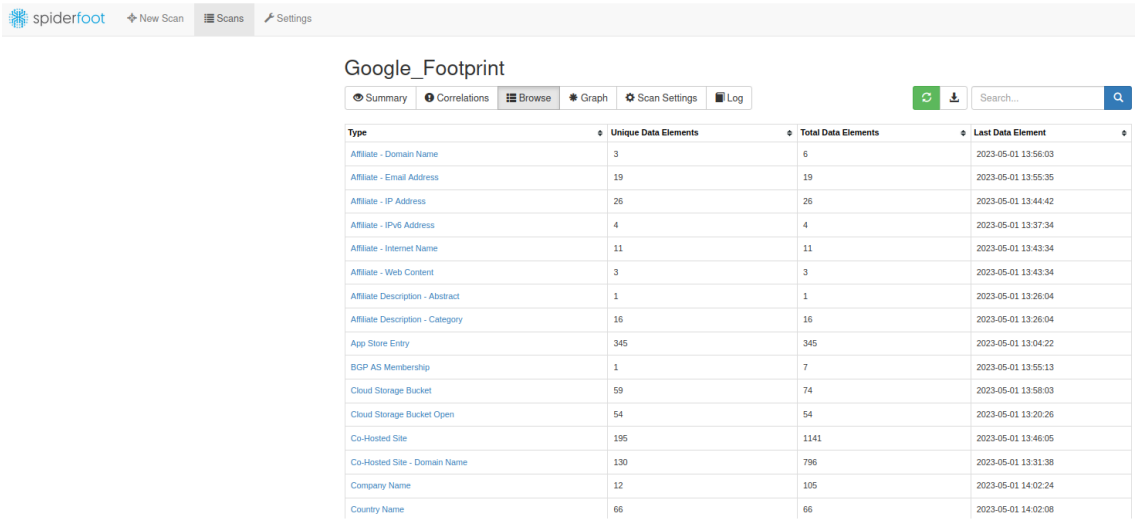



**Ilustración 36. Spiderfoot, Gráfico del escaneo a Google.com**

Como se puede observar, los resultados obtenidos por Spiderfoot se muestran de manera bastante visual. Sin embargo, puede resultar un poco complicada la interpretación de los datos en el gráfico debido a la gran cantidad de información que se ha recopilado. Por ello, la herramienta también ofrece una visualización en modo tabla.




| Type | Unique Data Elements | Total Data Elements | Last Data Element |
|---|---|---|---|
| Affiliate - Domain Name | 3 | 6 | 2023-05-01 13:56:03 |
| Affiliate - Email Address | 19 | 19 | 2023-05-01 13:55:35 |
| Affiliate - IP Address | 26 | 26 | 2023-05-01 13:44:42 |
| Affiliate - IPv6 Address | 4 | 4 | 2023-05-01 13:37:34 |
| Affiliate - Internet Name | 11 | 11 | 2023-05-01 13:43:34 |
| Affiliate - Web Content | 3 | 3 | 2023-05-01 13:43:34 |
| Affiliate Description - Abstract | 1 | 1 | 2023-05-01 13:26:04 |
| Affiliate Description - Category | 16 | 16 | 2023-05-01 13:26:04 |
| App Store Entry | 345 | 345 | 2023-05-01 13:04:22 |
| BGP AS Membership | 1 | 7 | 2023-05-01 13:55:13 |
| Cloud Storage Bucket | 59 | 74 | 2023-05-01 13:58:03 |
| Cloud Storage Bucket Open | 54 | 54 | 2023-05-01 13:20:26 |
| Co-Hosted Site | 195 | 1141 | 2023-05-01 13:46:05 |
| Co-Hosted Site - Domain Name | 130 | 796 | 2023-05-01 13:31:38 |
| Company Name | 12 | 105 | 2023-05-01 14:02:24 |
| Country Name | 66 | 66 | 2023-05-01 14:02:08 |

**Ilustración 37. Spiderfoot, Tabla del escaneo a Google.com**

En definitiva, Spiderfoot no solo permite facilitar en gran medida las búsquedas OSINT, sino que, además, cuenta con una interfaz bastante clara e intuitiva la cual facilita enormemente la interpretación de los resultados obtenidos.

No obstante, se pretende disponer de un entorno centralizado donde visualizar todos los datos obtenidos por las herramientas. Por este motivo se exportará, por ejemplo, la información relacionada con las URL enlazas al dominio, "*Linked URL - Internal*".

Este proceso es bastante sencillo, puesto que el software ofrece una opción para descargar dichos datos en formato CSV. Dicha descarga genera de forma predeterminada un archivo llamado "*Spiderfoot.csv*".

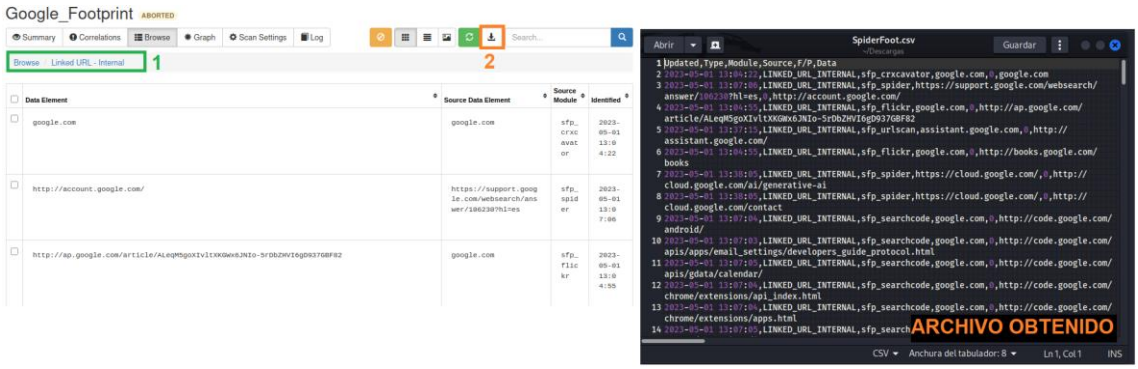



**Ilustración 38. Spiderfoot, Exportación de datos Linked URL – Internal**

Posteriormente, estos datos van a ser procesados para visualizarlos en un entorno centralizado.

## 3.4. Automatización del entorno

Una vez se ha analizado el funcionamiento de las herramientas y se ha obtenido la información a partir de estas, se pretende automatizar el proceso de

obtención de dicha información. Con tal fin, se ha desarrollado el siguiente script.

```
#!/bin/bash

##### VARIABLES GLOBALES #####
#Ruta de creacion del entorno de trabajo
ruta='/home/bmaeso/GitHub/'
#Ruta de imagenes de prueba
imagenes='/home/bmaeso/Descargas/imagenes/'
#Archivo exportacion Spiderfoot
spidercsv='/home/bmaeso/Descargas/spiderfoot_exp/SpiderFoot.csv'

##### PERSONALIZAR BUSQUEDAS #####
#THEHARVESTER
th_dom='youtube.com'
#DMITRY
dm_dom='gmail.com'
#OSRFRAMEWORK
osr_user='ibaillanos'

##### ACTIVAR/DESACTIVAR HERRAMIENTAS #####
theharvester=false
dmitry=false
exiftool=true
osrframework=false
spiderfoot=false

#CREACION DEL ENTORNO DE TRABAJO
mkdir $ruta
cd $ruta

if $theharvester ; then
  mkdir theHarvester
fi

if $dmitry ; then
  mkdir DMitry
fi

if $exiftool ; then
  mkdir ExifTool
fi

if $osrframework ; then
  mkdir OSRFramework
fi

if $spiderfoot ; then
```

```
  mkdir Spiderfoot
fi

##### THEHARVESTER #####
if $theharvester ; then
  cd "${ruta}theHarvester"
  theHarvester -d "${th_dom}" -l 500 -b all -f theHarvester
  #Normalizacion a CSV
  curl -X POST "https://www.convertcsv.io/api/v1/json2csv?" -H
"Authorization: Token 6755b2e17b014838762fa9d2249d9dcd6a80e2cc" -F
"infile=@theHarvester.json" --output theHarvester.csv
fi

##### DMITRY #####
if $dmitry ; then
  cd "${ruta}DMitry"
  dmitry -e "${dm_dom}" -o dmitry_gmail
  #Normalizacion a CSV
  strings dmitry_gmail.txt | grep -Eio '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-
]+\.[A-Za-z]{2,6}$' > dmitry_gmail.csv
fi

##### EXIFTOOL #####
if $exiftool ; then
  cd "${ruta}ExifTool"
  cp -r "${imagenes}" "${ruta}ExifTool"
  exiftool -csv * > exiftool_perros.csv
  #Normalizacion a CSV
  sed -n '2,$p' exiftool_perros.csv > exiftool_perros_parse.csv
fi

##### OSRFRAMEWORK #####
if $osrframework ; then
  cd "${ruta}OSRFramework"
  usufy.py -p all -n "${osr_user}"
  #Normalizacion a CSV
  sed -n '2,$p' profiles.csv > osrframework_ibaillanos.csv
fi

##### SPIDERFOOT #####
if $spiderfoot ; then
  cd "${ruta}Spiderfoot"
  cp "$spidercsv" "${ruta}Spiderfoot"
  #Normalizacion a CSV
  sed -n '2,$p' SpiderFoot.csv > spiderfoot_google.csv
fi

##### MENSAJE DE FINALIZACION #####
echo "[+] EL SCRIPT HA FINALIZADO CORRECTAMENTE"
```

El código anterior se encuentra dividido en varias secciones. Las primeras líneas se enfocan en la definición de las variables globales entre las cuales se encuentra la ruta donde se pretende crear el entorno de trabajo, la carpeta en la que están almacenadas las imágenes que se analizarán con ExifTool y la ruta del archivo exportado por Spiderfoot.

Después, aparece una sección dedicada a personalizar las búsquedas de las herramientas. Dentro de esta se ofrece la posibilidad de definir el dominio concreto que analiza cada herramienta de las presentes. Además, en las siguientes líneas se ha añadido la funcionalidad de poder activar o desactivar la ejecución de las herramientas. De esta forma se permite personalizar aún más las búsquedas en función de las necesidades del usuario.

En la siguiente fase se crean los directorios del entorno de trabajo y se va ejecutando cada herramienta para obtener los resultados que posteriormente se van a procesar. Es importante mencionar que se ejecuta cada herramienta siguiendo el orden de aparición propuesto en la documentación. Además, cada programa cuenta con una sección de normalización a CSV, el cual se va a explicar en el siguiente apartado relacionado con la visualización de los datos. No obstante, este proceso de normalización se lleva a cabo para facilitar la posterior subida de los datos a través del agente Filebeat.

En definitiva, el script anterior se encarga no solo de automatizar el proceso, sino que, también, permite personalizar las herramientas que se ejecutan y sus consultas normalizando los resultados obtenidos. Con ello, se consigue automatizar totalmente el proceso permitiendo subir directamente y en tiempo real los resultados generados a través de Filebeat.

## 3.5. Visualización de los datos

### 3.5.1. Implantación del entorno

Para lograr el mayor grado de compatibilidad entre los diferentes elementos, se ha seleccionado la versión 8.6.2 de todos ellos. Dicha elección se debe a que se trata de una las últimas versiones estables disponible a la fecha de comenzar la investigación por lo que se garantiza que traen incorporadas tanto las ultimas características de funcionalidad como de seguridad.

#### 3.5.1.1. ElasticSearch

Este software se ha descargado desde su página oficial a través del siguiente enlace.

```
https://www.elastic.co/downloads/elasticsearch
```

Puesto que se trata de un paquete portable que se autoconfigura, sólo es necesario ejecutar el siguiente comando para arrancar el servicio de ElasticSearch.

```
# ./elasticsearch-8.6.2/bin/elasticsearch
```

Por defecto, ElasticSearch corre en el puerto local 9200.

```
https://127.0.0.1:9200/
```

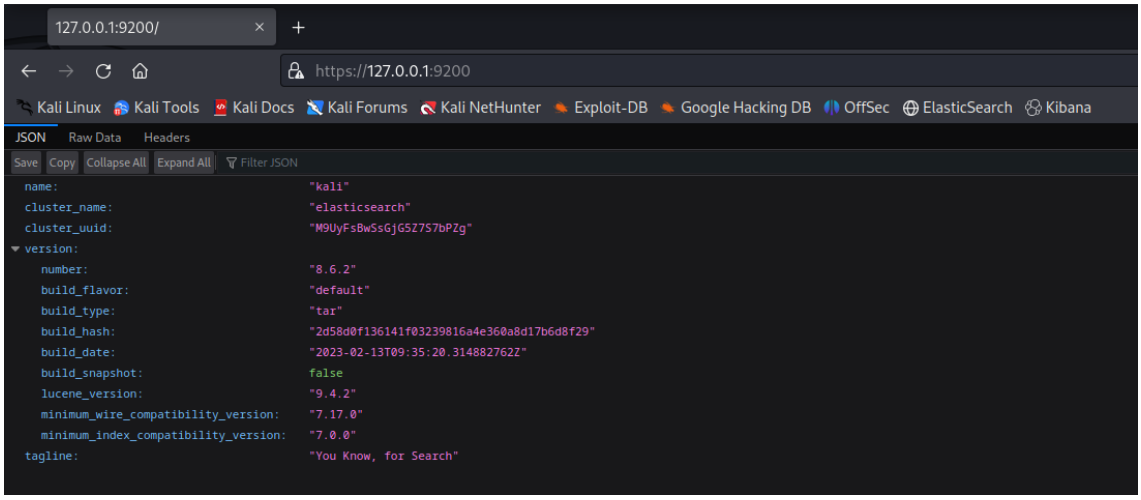



**Ilustración 39. Servicio ElasticSearch ejecutándose correctamente**

Además, puesto que se trata de una de las últimas versiones, utiliza de manera predetermina el protocolo HTTPS, el cual ofrece una capa adicional de seguridad.

### 3.5.1.2. Kibana

El enlace de descarga de Kibana es el siguiente.

```
https://www.elastic.co/downloads/kibana
```

Para levantar el servicio se hace uso de esta instrucción.

```
# ./kibana-8.6.2/bin/kibana
```

Este servicio corre de manera predeterminada en el puerto local 5601.

```
http://localhost:5601
```

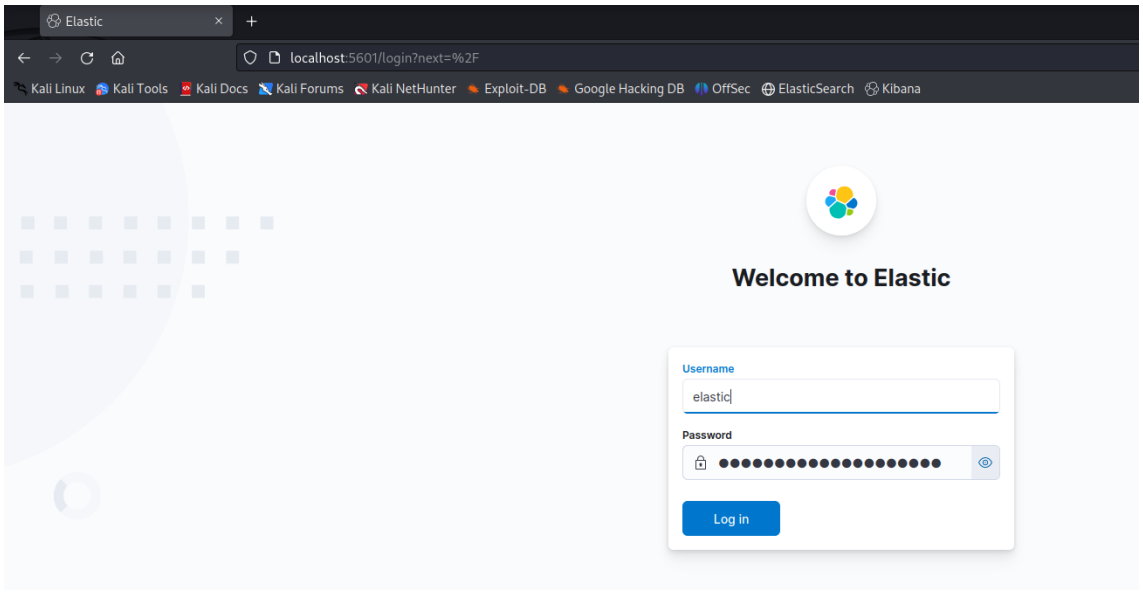



**Ilustración 40. Servicio Kibana ejecutándose correctamente**

### 3.5.1.3. Logstash

El software de Logstash está disponible a través de este enlace.

```
https://www.elastic.co/downloads/logstash
```

Para levantar el servicio se hace uso de este comando, el cual necesita que se haga referencia a un archivo de configuración en el que se especifica qué datos se van a mandar a ElasticSearch.

```
# ./logstash-8.6.2/bin/logstash -f ARCHIVO_CONFIGIRACION
```

En este punto es necesario comentar que al probar este servicio la máquina se colapsa debido a la gran cantidad de recursos que requiere este servicio. Por este motivo, se ha desestimado el uso de este software para el trabajo propuesto y, en su lugar, se utiliza Filebeat.

### 3.5.1.4. Filebeat

Este agente ligero se puede descargar de la página oficial.

```
https://www.elastic.co/downloads/beats/filebeat
```

El servicio se arranca a través del siguiente comando.

```
# sudo ./filebeat -e -c filebeat.yml
```

La opción "*-e*" deshabilita la salida de registros en un archivo y "*-c*" define el archivo de configuración que, de manera predeterminada, viene incorporado en el paquete con el nombre "*filebeat.yml*".

### 3.5.2. Preparación del entorno de trabajo

Para llevar a cabo este proyecto se ha creado un entorno de trabajo alojado en la siguiente ruta.

```
/home/bmaeso/GitHub
```

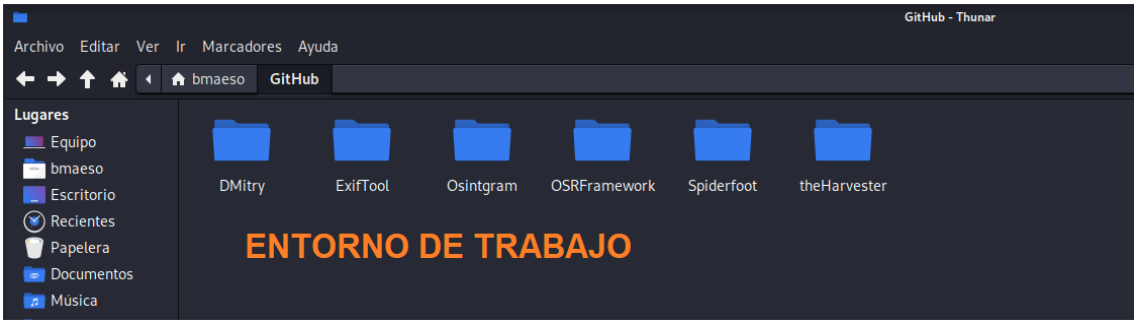



**Ilustración 41. Entorno de trabajo en Kali Linux**

En dicha ruta se ha creado una carpeta con el nombre de cada herramienta donde se almacenan los resultados obtenidos por cada una de ellas. De esta forma se permite disponer de los resultados de una manera centralizada y ordenada.

Por otra parte, en la metodología se adopta la estrategia de normalizar los datos a un formato concreto con la intención de homogenizar los resultados y favorecer tanto su análisis como comprensión. En concreto, se ha elegido el formato CSV separado por comas ya que se ha considerado un formato bastante conocido y que, a la vez, permite lidiar de la mejor manera contra las limitaciones de funcionalidad que ofrece el agente ligero Filebeat.

A continuación, se muestra un resumen de la estructura global que va a implementar.

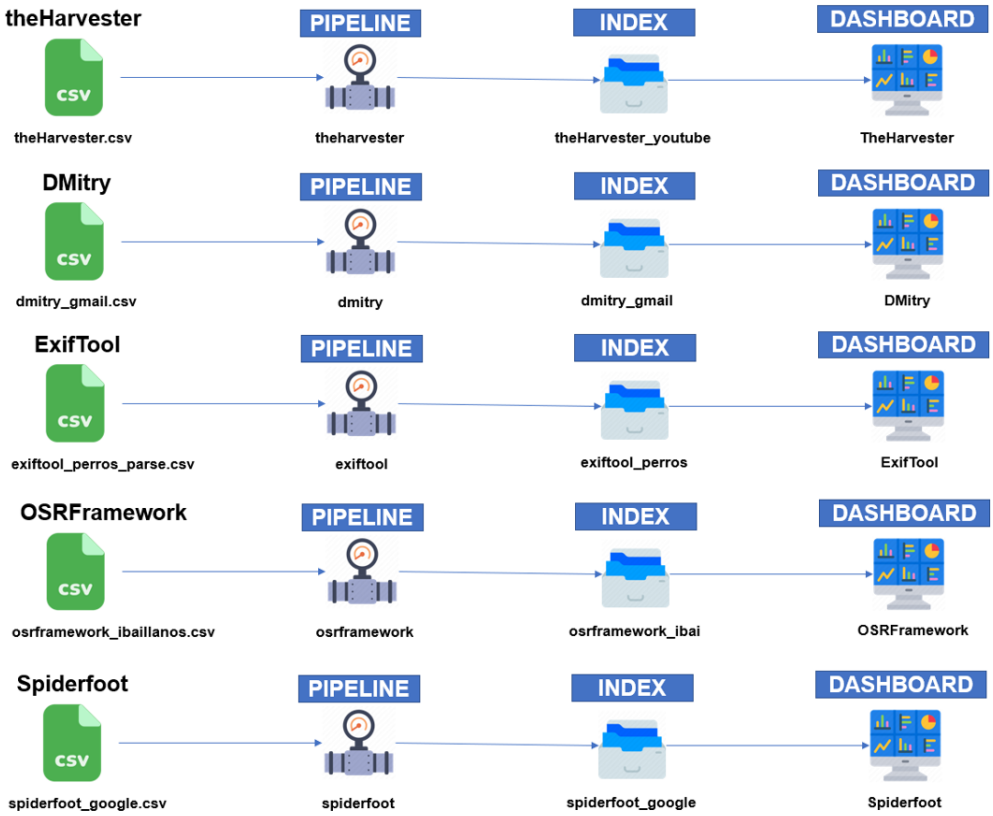



**Ilustración 42. Resumen del ciclo de los datos**

Como se puede observar, el ciclo de los datos consta de 4 fases. La primera se centra en normalizar el archivo a formato CSV separado por comas de tal forma que se puedan homogeneizar los resultados obtenidos de todas las herramientas.

Es importante mencionar que existen dos alternativas posibles para el proceso de filtrado e interpretación de la información obtenida, las cuales se detallan a continuación:

- Establecer el filtrado directamente en el archivo de configuración de Filebeat.
- Indicar en el archivo de configuración de Filebeat un pipeline específico definido en ElasticSearch.

Entre ambas alternativas se ha definido en la que interfieren los pipelines ya que ofrecen un mayor potencial y flexibilidad. Esta opción es considerada la mejor puesto que permite disponer de un archivo de configuración mucho más simple al modularizar los procesos entre varias herramientas y no en una sola. Además, al configurar el procesado directamente en ElasticSearch, no es necesario modificar el archivo de configuración de Filebeat en caso de que se quisiera interpretar los datos de una nueva forma.

Por consiguiente, la segunda fase hace referencia a la definición del pipeline. En este paso, se procesa la información de acuerdo a unas directrices concretas. Tras este proceso se ha eliminado la información inútil y se dispone

únicamente del contenido que se desea almacenar. De esta forma, da comienzo el tercer paso en el que se indexa dicho contenido en ElasticSearch.

Para finalizar, se crea una Dashboard para visualizar los datos en tiempo real y, con ello, facilitar la interpretación y comprensión del contenido de cara a obtener conclusiones eficaces.

### 3.5.3. TheHarvester

Para llevar a cabo el procesado y visualización de la información obtenida por esta herramienta, se parte del archivo JSON generado, "*theHarvester.json*"


**Ilustración 43. Contenido del archivo theHarvester.json**

En primer paso es identificar el patrón que sigue la información mostrada en este documento, la cual se puede visualizar a través de alguna herramienta o plataforma online. En este caso, la estructura se encuentra dividida en 5 secciones: *asns* (Sistemas Autónomos), *hosts*, *interesting_urls*, *ips* y *shodan*.


**Ilustración 44. Estructura del archivo theHarvester.json**

Puesto que uno de los objetivos del trabajo es agrupar los resultados de las herramientas, se debe normalizar la información. Específicamente, se pretende transformar el resultado obtenido a formato CSV delimitado por una coma (","). Para ello, se hace uso del convertidor online de JSON a CSV.

https://www.convertcsv.com/json-to-csv.htm

Este convertidor ofrece una API a través de la que se pueden transformar archivos con el comando CURL.

```
# curl -X POST "https://www.convertcsv.io/api/v1/json2csv?" -H
"Authorization: Token 6755b2e17b014838762fa9d2249d9dcd6a80e2cc" -F
"infile=@theHarvester.json" --output theHarvester.csv
```

La instrucción anterior permite obtener el archivo “*theHarvester.csv*” a partir del archivo inicial “*theHarvester.json*” utilizando un token de la API a través de una consulta de tipo POST.

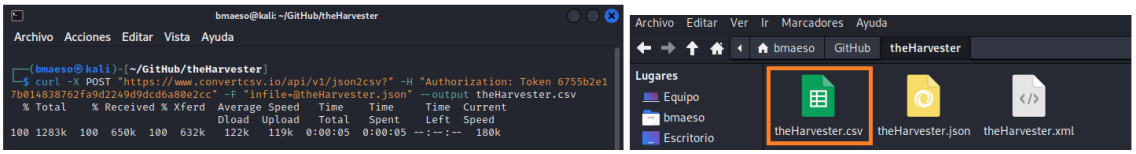



**Ilustración 45. Generación del archivo theHarvester.csv**

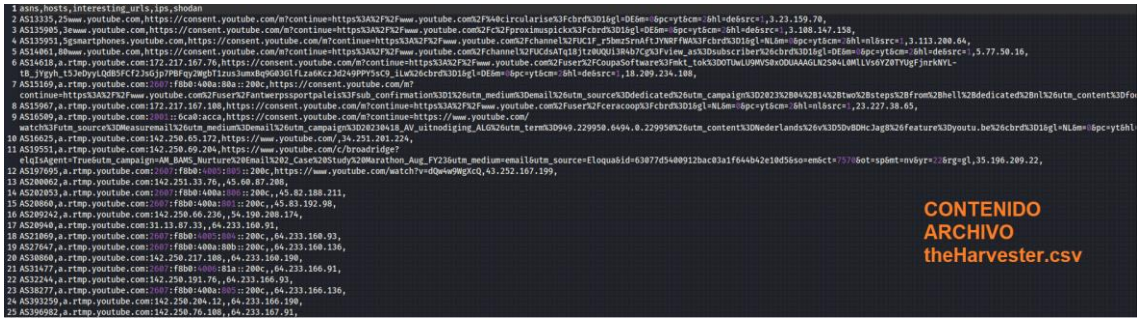



**Ilustración 46. Contenido archivo theHarvester.csv**

Después, se ha añadido a la configuración general de Filebeat el nuevo documento generado incorporando estas nuevas líneas en la sección “*filebeat.inputs*”.

```
- type: log
  id: theHarvester_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/theHarvester/theHarvester.csv
  index: theHarvester_youtube
  pipeline: theharvester
```

En las líneas anteriores, se ha definido una nueva ruta de entrada la cual se corresponde con el archivo “*theHarvester.csv*” que generará un nuevo índice “*theHarvester_youtube*” tras ser procesado por un pipeline llamado “*theharvester*”.

El resultado del pipeline encargado de procesar los datos es el siguiente.

```
PUT _ingest/pipeline/theharvester
{
  "description": "Parse theHarvester data",
  "processors": [
    {
      "csv": {
```

```
        "field": "message",
        "target_fields": [
          "asns",
          "host",
          "url",
          "ip",
          "shodan"
        ],
        "separator": ",",
        "trim": true
      }
    },
    {
      "remove": {
        "field": [
          "@timestamp",
          "message"
        ]
      }
    },
    {
      "convert": {
        "field": "ip",
        "type": "ip"
      }
    }
  ],
  "on_failure": [
    {
      "remove": {
        "field": [
          "asns",
          "host",
          "url",
          "ip",
          "shodan"
        ]
      }
    }
  ]
}
```

El código definido en el pipeline divide la entrada en los campos correspondientes: *asns*, *host*, *url*, *ip* y *shodan*. Además, convierte el campo *ip* en un dato de tipo IP y, en caso de fallo, elimina los datos de esa entrada lo cual permite descartar líneas inútiles.

El siguiente paso es ejecutar la nueva configuración de Filebeat.

```
# sudo ./filebeat -e -c filebeat.yml
```

Tras su ejecución, se puede comprobar que en la sección "*Stack Management > Index Management*" se ha creado el nuevo índice, "*theharvester_youtube*".

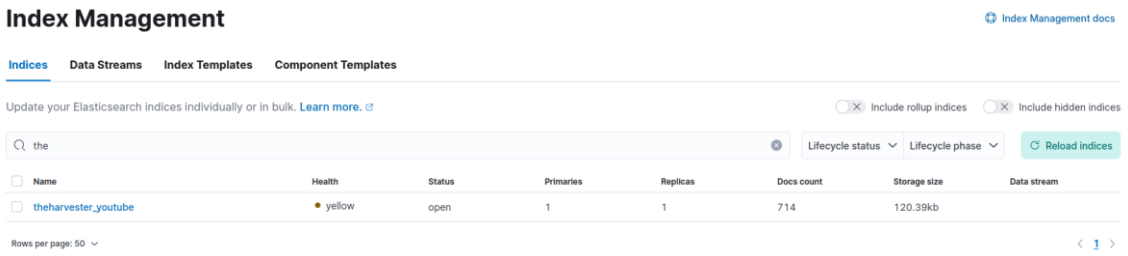



**Ilustración 47. Creación del índice theharvester_youtube**

A continuación, se muestra una parte del resultado sobre los datos procesados que se han obtenido en la sección "*Dev Tools > Console*" a través de la instrucción.

```
GET theharvester_youtube/_search
```

La siguiente captura demuestra que los datos se han procesado correctamente y que las líneas que pueden contener datos inútiles han sido descartadas. En este caso, la única línea que interesa descartar es la primera ya que solo contiene el nombre de las columnas posteriores.

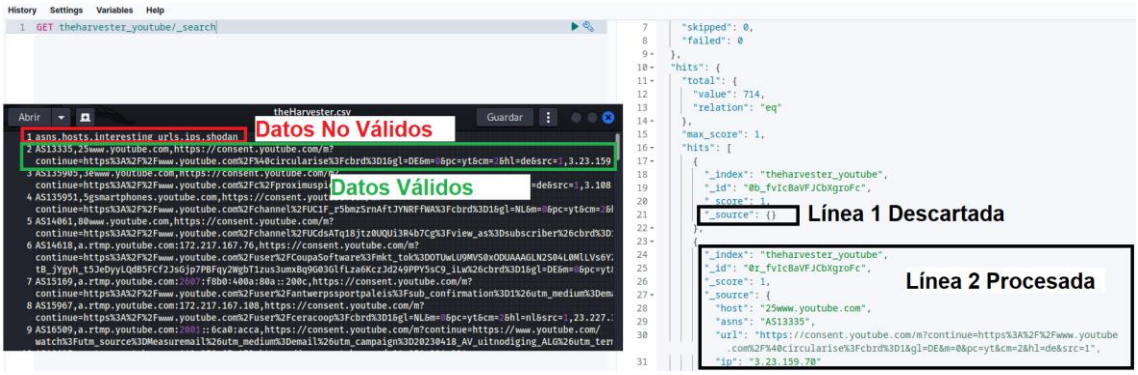



**Ilustración 48. Procesamiento del Pipeline theharvester**

Para finalizar, se ha creado un nuevo perfil para la vista de datos sobre el índice "*theHarvester_youtube*" sobre el cual se han definido una serie de estructuras que permiten visualizar los datos en tiempo real de manera ordenada y clara.


**Ilustración 49. Dashboard de datos obtenidos con theHarvester**

### 3.5.4. DMitry

El primer paso es normalizar el archivo generado previamente. Para ello, se ha hecho uso del siguiente comando, el cual permite filtrar solo los correos mediante una expresión regular y almacena la salida en un archivo llamado "*dmitry_gmail.csv*".

```
# strings dmitry_gmail.txt | grep -Eio '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}$' > dmitry_gmail.csv
```

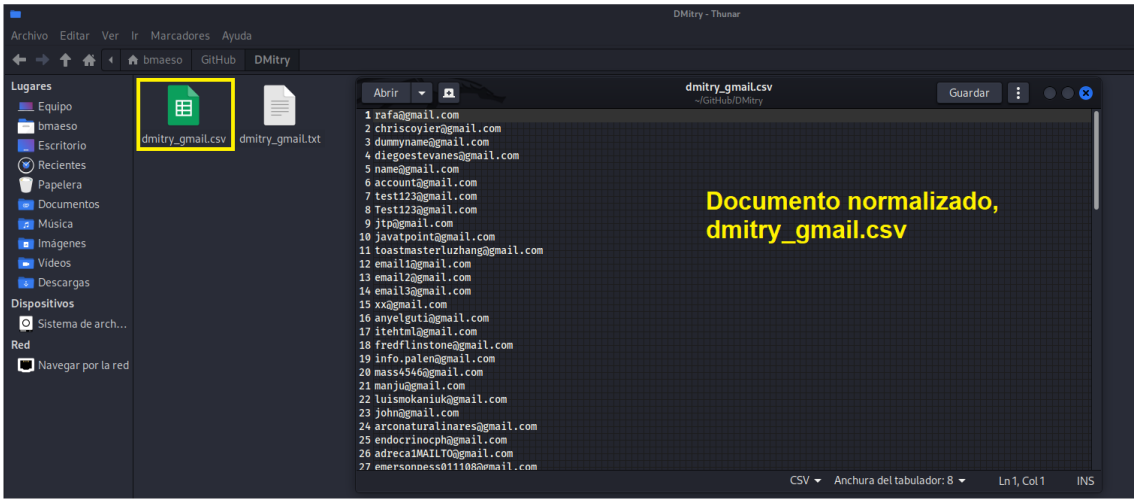



**Ilustración 50. DMitry, Documento normalizado dmitry_gmail.csv**

Como se puede apreciar, el documento ha sido normalizado mostrando únicamente las líneas que contienen correos electrónicos.

El siguiente paso es configurar una nueva entrada en la configuración de Filebeat y crear un pipeline que procese dicha entrada.

A continuación, se muestra la nueva entrada de Filebeat, en la que se indica que se use el archivo "*dmitry_gmail.csv*" creando un nuevo índice llamado "*dmitry_gmail*" a través de del pipeline "*dmitry*".

```
- type: log
  id: dmitry_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/DMitry/dmitry_gmail.csv
  index: dmitry_gmail
  pipeline: dmitry
```

En el contenido del pipeline "*dmitry*" se especifica que el campo entrante se llame "*email*" y que se eliminen las entradas por defecto "*message*" y "@timestamp" dejando de esta forma solo la información de los correos.

```
PUT _ingest/pipeline/dmitry
{
  "description": "Parse DMitry data",
  "processors": [
    {
```

```
      "csv": {
        "field": "message",
        "target_fields": [
          "email"
        ],
        "trim": true
      }
    },
    {
      "remove": {
        "field": [
          "message",
          "@timestamp"
        ]
      }
    }
  ]
}
```


**Ilustración 51. Procesamiento del Pipeline dmitry**

Como último paso, se ha creado un Dashboard que muestre la información almacenada en el índice "*dmitry_gmail*".

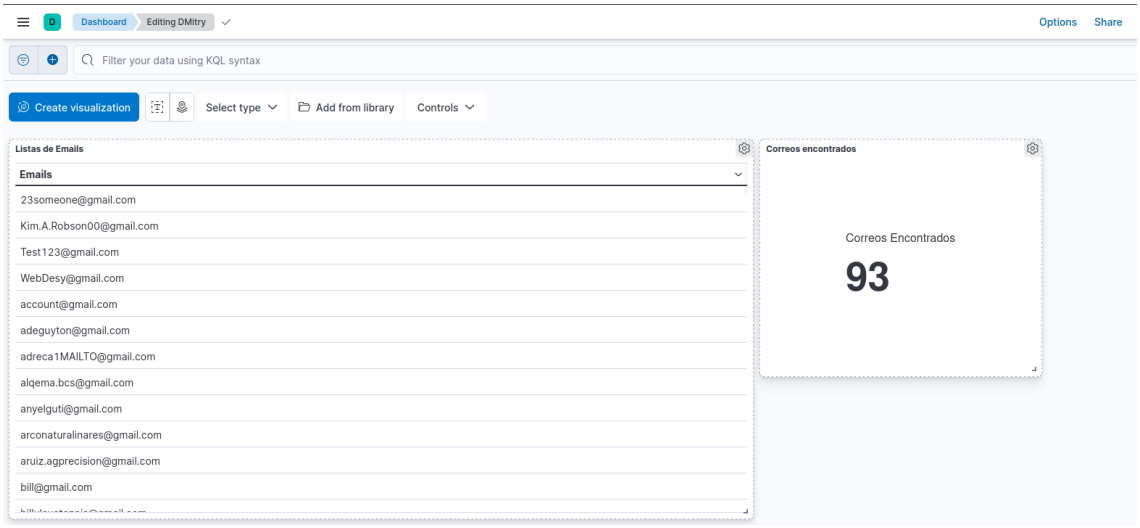



**Ilustración 52. Dashboard de datos obtenidos con DMitry**

En la visualización anterior se muestra que hay 93 correos encontrados y se permite buscar y operar sobre cada entrada de manera sencilla y clara.

### 3.5.5. ExifTool

Puesto que el archivo obtenido previamente se encuentra en formato CSV separado por comas, el proceso de normalización no es necesario. No obstante, se hace uso de la siguiente instrucción, la cual permite eliminar la primera línea del archivo con la intención de quitar información inútil.

```
# sed -n '2,$p' exiftool_perros.csv > exiftool_perros_parse.csv
```

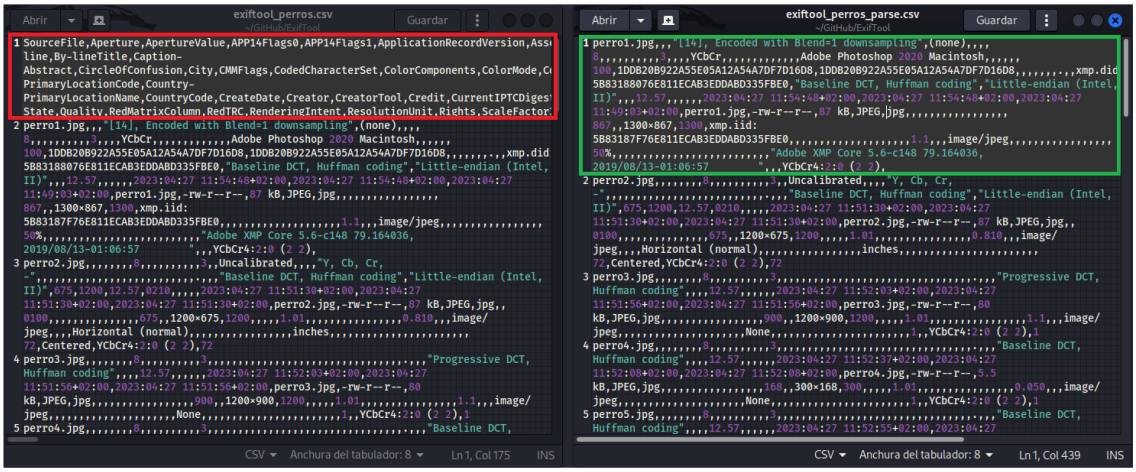



**Ilustración 53. ExifTool, Eliminación de información inútil**

El resultado del comando genera un nuevo archivo llamado “exiftool_perros_parse.csv” en el que la primera línea ya contiene información de utilidad. A diferencia, el archivo original, contiene en su primera línea el nombre de los campos a interpretar lo cual es interesante para la creación del pipeline, pero no para ser guardado en ElasticSearch.

Una vez creado el archivo a procesar, se crea añade la nueva configuración a Filebeat.

```
- type: log
  id: exiftool_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/ExifTool/exiftool_perros_parse.csv
  index: exiftool_perros
  pipeline: exiftool
```

Además, se crea un nuevo pipeline llamado “*exiftool*” para procesar cada campo usando la información especificada en la primera línea del archivo generado inicialmente, “exiftool_perros.csv”.

```
PUT _ingest/pipeline/exiftool
{
  "description": "Parse ExifTool data",
```

```
  "processors": [
    {
      "csv": {
        "field": "message",
        "target_fields": [
          "SourceFile","Aperture","ApertureValue","APP14Flags0",
          "APP14Flags1","ApplicationRecordVersion","AssetID",
          "AuthorsPosition","BitsPerSample","BlueMatrixColumn","BlueTRC",
          "By-line","By-lineTitle","Caption-Abstract","CircleOfConfusion",
          "City","CMMFlags","CodedCharacterSet","ColorComponents",
          "ColorMode","ColorSpace","ColorSpaceData","ColorTransform",
          "Comment","ComponentsConfiguration","ConnectionSpaceIlluminant",
          "Copyright","CopyrightFlag","CopyrightNotice","Country",
          "Country-PrimaryLocationCode","Country-PrimaryLocationName",
          "CountryCode","CreateDate","Creator","CreatorTool","Credit",
          "CurrentIPTCDigest","DateCreated","DateTimeCreated",
          "DateTimeOriginal","DCTEncodeVersion","DerivedFromDocumentID",
          "DerivedFromInstanceID","Description","DeviceAttributes",
          "DeviceManufacturer","DeviceMfgDesc","DeviceModel",
          "DeviceModelDesc","Directory","Dlref","DocumentID",
          "EncodingProcess","ExifByteOrder","ExifImageHeight",
          "ExifImageWidth","ExifToolVersion","ExifVersion",
          "ExposureCompensation","ExposureMode","ExposureProgram",
          "ExposureTime","FileAccessDate","FileInodeChangeDate",
          "FileModifyDate","FileName","FilePermissions","FileSize",
          "FileType","FileTypeExtension","Flash","FlashpixVersion",
          "FNumber","FocalLength","FocalLength35efl",
          "FocalPlaneResolutionUnit","FocalPlaneXResolution",
          "FocalPlaneYResolution","Format","FOV","GreenMatrixColumn",
          "GreenTRC","Headline","HyperfocalDistance","ICCProfileName",
          "ImageDescription","ImageHeight","ImageRank","ImageSize",
          "ImageWidth","InstanceID","Instructions","IPTCDigest","ISO",
          "JFIFVersion","Keywords","LegacyIPTCDigest","LicensorURL",
          "LicsensorURL","LightValue","Luminance","Make","MaxApertureValue",
          "MeasurementBacking","MeasurementFlare","MeasurementGeometry",
          "MeasurementIlluminant","MeasurementObserver","MediaBlackPoint",
          "MediaWhitePoint","Megapixels","MetadataDate","MeteringMode",
          "MIMEType","Model","ModifyDate","ObjectName","Orientation",
          "PrimaryPlatform","ProfileClass","ProfileCMMType",
          "ProfileConnectionSpace","ProfileCopyright","ProfileCreator",
          "ProfileDateTime","ProfileDescription","ProfileFileSignature",
          "ProfileID","ProfileVersion","Province-State","Quality",
          "RedMatrixColumn","RedTRC","RenderingIntent","ResolutionUnit",
          "Rights","ScaleFactor35efl","SceneCaptureType","ShutterSpeed",
          "ShutterSpeedValue","Software","Source","SpecialInstructions",
          "State","Subject","Technology","TimeCreated","Title","Urgency",
          "Url","ViewingCondDesc","ViewingCondIlluminant",
          "ViewingCondIlluminantType","ViewingCondSurround","WebStatement",
          "WhiteBalance","XMPToolkit","XResolution","YCbCrPositioning",
          "YCbCrSubSampling","YResolution"
        ],
        "separator": ",",
        "trim": true
      }
    },
    {
      "remove": {
        "field": [
          "message",
```

```
            "@timestamp"
          ]
        }
      }
    ]
}
```

A continuación, se muestra una parte del resultado del procesado de los datos.

```
GET exiftool_perros/_search
```

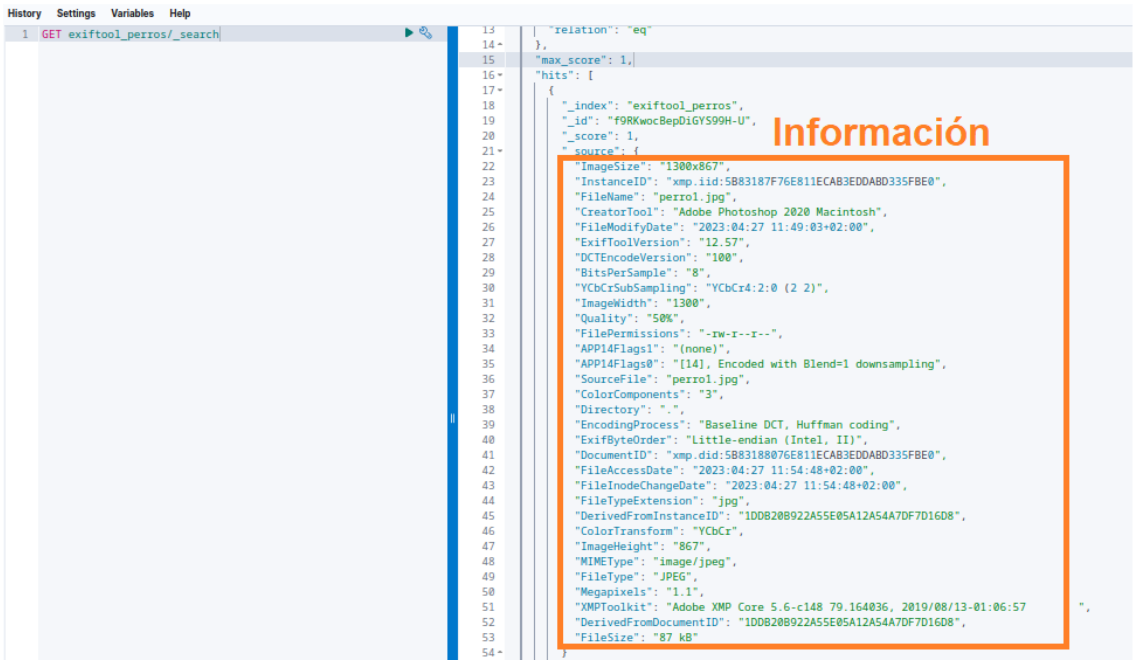



**Ilustración 54. Procesamiento del Pipeline exiftool**

Como se puede apreciar en la imagen anterior, sólo se almacena la información de los campos que no están en blanco. De esta forma el proceso de almacenamiento e interpretación es mucho más eficaz.

Para finalizar, se muestra un Dashboard que se ha creado sobre el índice almacenado.

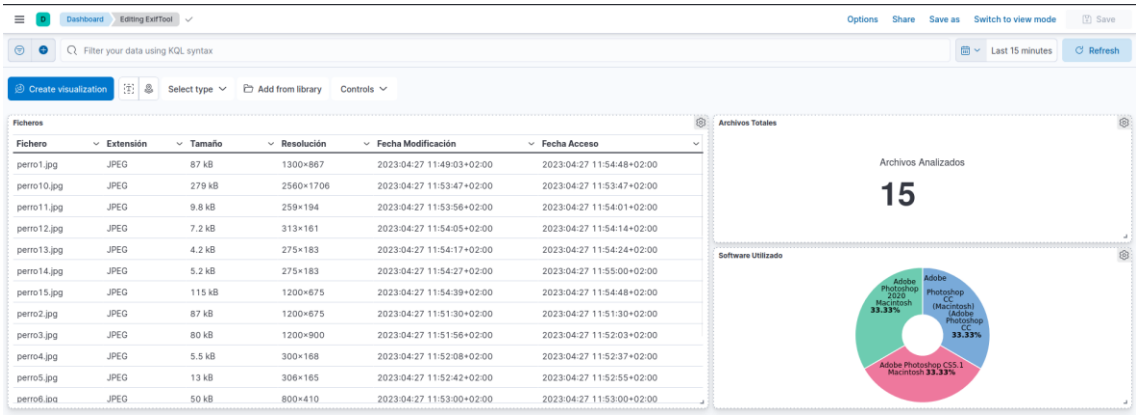



**Ilustración 55. Dashboard de datos obtenidos con ExifTool**

En la visualización se muestran detalles muy interesantes sobre los archivos analizados como puede ser su tamaño, extensión o fecha de creación entre otros. Además, ofrece otro panel en el que indica el número de archivos analizados, en este caso 15, y un diagrama circular que especifica las estadísticas del software con el que se han creado los archivos procesados.

### 3.5.6. OSRFramework

Debido a que el archivo generado se encuentra en formato CSV tampoco es necesario llevar a cabo ningún proceso de normalización. Aun así, se procesará dicho archivo con la intención de eliminar la primera línea, la cual muestra información sobre el nombre de los campos. Para ello, se ha hecho uso del siguiente comando que genera un nuevo archivo, "*osrframework_ibaillanos.csv*".

```
# sed -n '2,$p' profiles.csv > osrframework_ibaillanos.csv
```

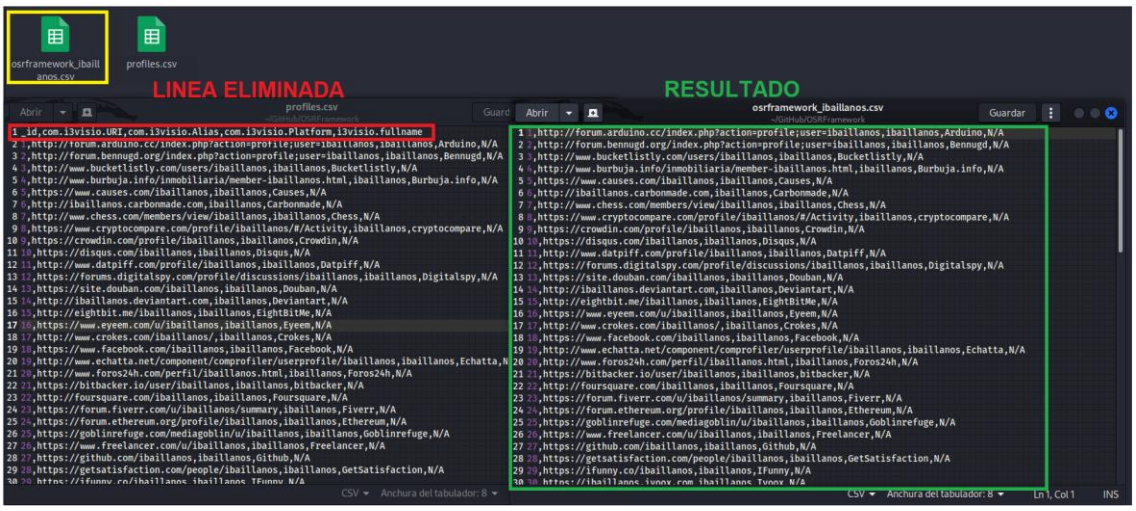



**Ilustración 56. OSRFramework, Creación del archivo normalizado**

Los siguientes pasos son crear la entrada en la configuración de Filebeat y el pipeline que procese dicho archivo. Primero, se muestra la configuración de Filebeat.

```
- type: log
  id: osrframework_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/OSRFramework/osrframework_ibaillanos.csv
  index: osrframework_ibai
  pipeline: osrframework
```

La configuración del pipeline es el siguiente.

```
PUT _ingest/pipeline/osrframework
{
  "description": "Parse OSRFramework data",
  "processors": [
```

```
    {
      "csv": {
        "field": "message",
        "target_fields": [
          "id",
          "uri",
          "alias",
          "platform",
          "fullname"
        ],
        "separator": ",",
        "trim": true
      }
    },
    {
      "remove": {
        "field": [
          "@timestamp",
          "message"
        ]
      }
    }
  ]
}
```

Por otra parte, se verifica que resultado almacenado en ElasticSearch es el esperado a través del comando mostrado.

```
GET osrframework_ibai/_search
```

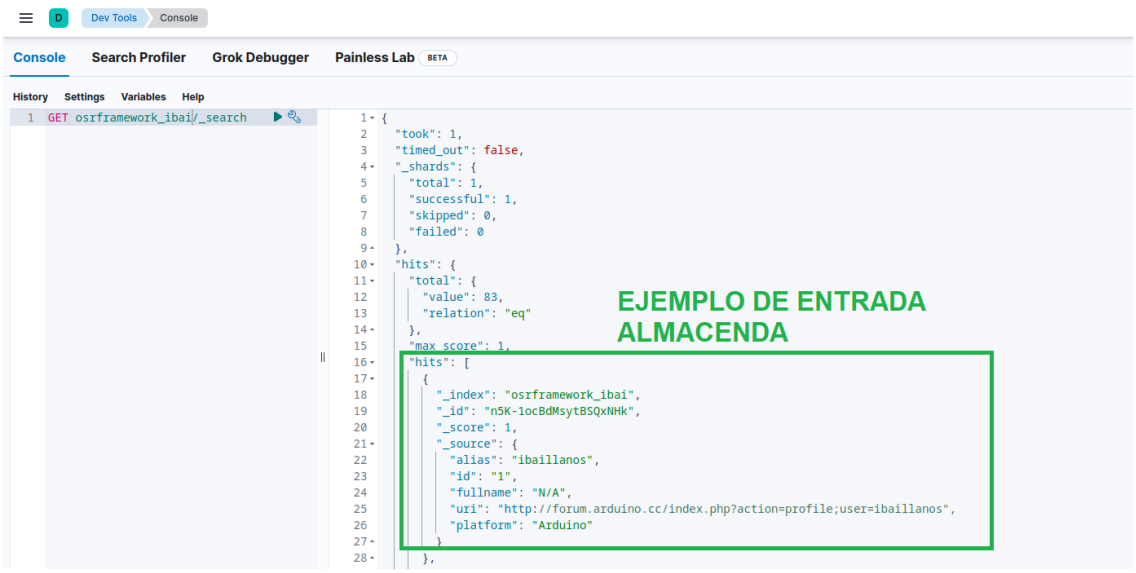



**Ilustración 57. Procesamiento del Pipeline osrframework**

Para finalizar se ha creado un Dashboard en el que se visualiza de manera bastante clara e intuitiva los datos recogidos por esta herramienta.

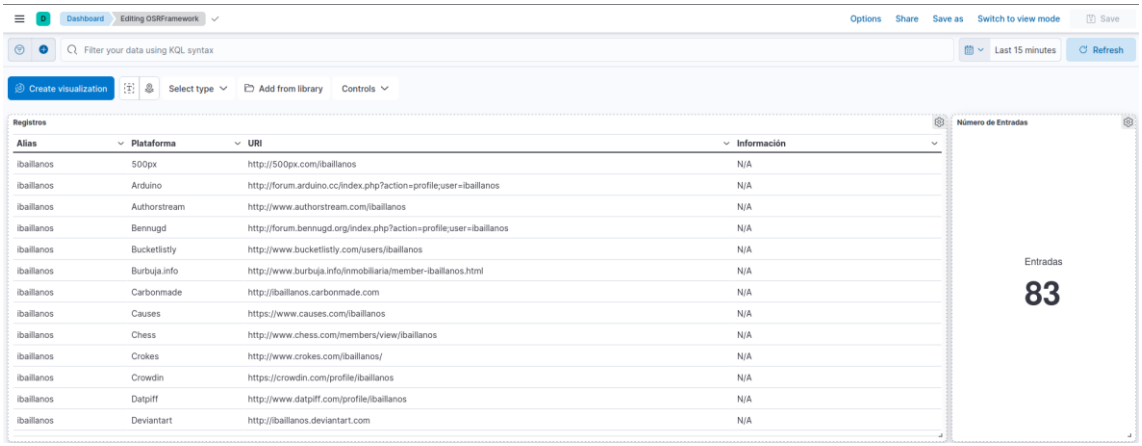



**Ilustración 58. Dashboard de datos obtenidos con OSRFramework**

### 3.5.7. Spiderfoot

Siguiendo la estrategia de apartados anteriores, se va a procesar el archivo obtenido para lo cual primero es necesario eliminar la primera línea.

```
# sed -n '2,$p' SpiderFoot.csv > spiderfoot_google.csv
```

El comando anterior elimina la primera línea presente en el archivo "*SpiderFoot.csv*" guardando el resultado en un nuevo archivo llamado "*spiderfoot_google.csv*".

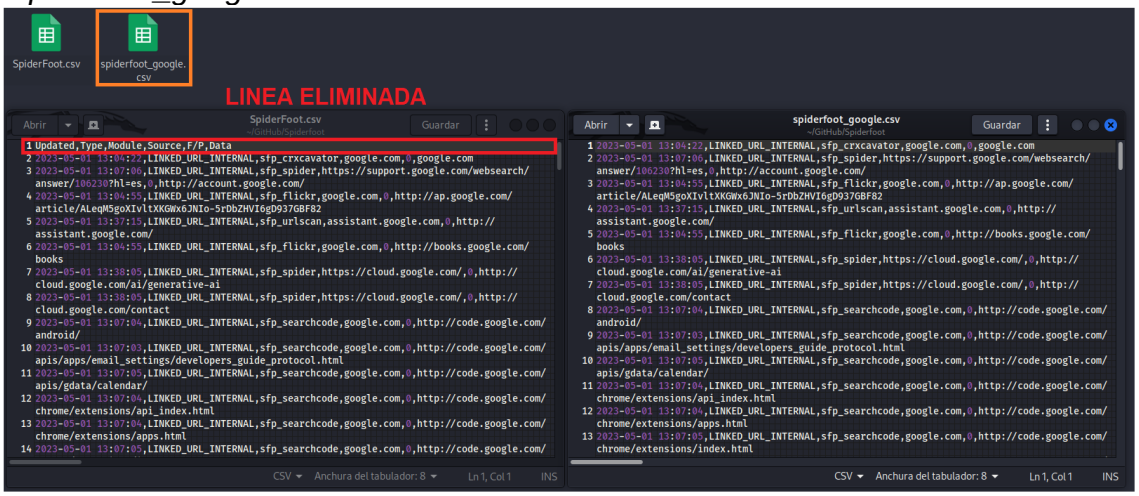



**Ilustración 59. Spiderfoot, Generación del archivo spiderfoot_google.csv**

Los siguientes pasos son la generación de una nueva entrada en el archivo de configuración de Filebeat y la creación de un nuevo pipeline que procese adecuadamente la información que se quiere incorporar.

La nueva entrada de Filebeat queda configurada de la siguiente forma.

```
- type: log
  id: spiderfoot_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/Spiderfoot/spiderfoot_google.csv
```

```
  index: spiderfoot_google
  pipeline: spiderfoot
```

La configuración del pipeline “spiderfoot” se muestra a continuación.

```
PUT _ingest/pipeline/spiderfoot
{
  "description": "Parse Spiderfoot data",
  "processors": [
    {
      "csv": {
        "field": "message",
        "target_fields": [
          "updated",
          "type",
          "module",
          "first_link",
          "info",
          "second_link"
        ],
        "separator": ",",
        "trim": true
      }
    },
    {
      "remove": {
        "field": [
          "@timestamp",
          "message"
        ]
      }
    }
  ]
}
```

Tras lanzar Filebeat, se obtiene el nuevo índice “*spiderfoot_google*” con el siguiente contenido.

```
GET spiderfoot_google/_search
```


**Ilustración 60. Procesamiento del Pipeline spiderfoot**

Para finalizar, se ha creado un nuevo Dashboard para visualizar los datos almacenados.

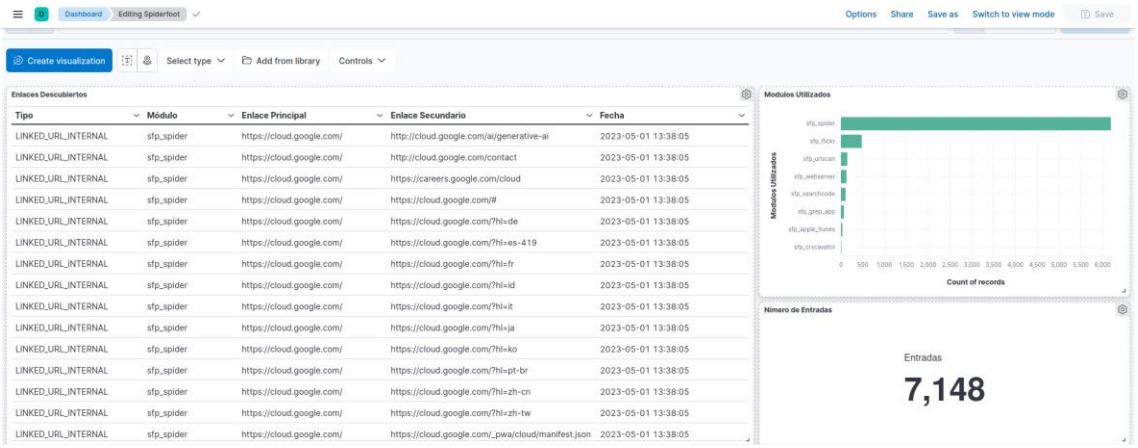



**Ilustración 61. Dashboard de datos obtenidos con Spiderfoot**

En la visualización anterior se pueden ver datos muy interesantes como el número de entradas obtenidas, los porcentajes de uso de los módulos utilizados y toda la información relativa a las entradas recopiladas.

# 4. Conclusiones y Trabajos Futuros

En un mundo cada vez más globalizado, en el que el desarrollo tecnológico está creciendo exponencialmente, conlleva que cada vez adquiera mayor relevancia la información que se encuentra publicada en la red. Además, con la aparición de nuevas herramientas y nuevos términos como Big Data, IoT o AI entre muchos otros, esta necesidad por decidir qué datos publicar se ve notablemente acentuada. Por este motivo es que se ha realizado este proyecto.

A continuación, se mencionan las conclusiones finales del trabajo.

- ✓ Existen multitud de herramientas y distribuciones enfocadas para la recolección de información de fuentes abiertas. No obstante, estos programas ofrecen sus salidas sin seguir un estándar global lo que dificulta en gran medida la homogeneización de los resultados.

- ✓ La distribución Kali Linux incorpora o, con pequeñas modificaciones, puede incorporar la mayoría de herramientas OSINT disponibles.

- ✓ Durante el desarrollo del trabajo ha quedado clara la necesidad e importancia de procesar y unificar la información adquirida a través de los programas seleccionados.

- ✓ Se ha podido constatar que las herramientas OSINT escogidas aportan una gran cantidad de información acorde a los objetivos del trabajo.

- ✓ La Pila Elastic ofrece funcionalidades muy potentes tanto para gestionar el almacenamiento e indexación (ElasticSearch) de la información obtenida, como para crear visualizaciones específicas (Kibana). Además, se ha determinado que el software Logstash puede saturar distribuciones que cuentan con recursos limitados por lo que, en muchos casos, es mejor utilizar agentes más ligeros como Filebeat.

- ✓ Un sistema como el desarrollado puede ser utilizado y adaptado a las necesidades particulares de un usuario que no sea experto en la materia.

- ✓ Siguiendo la metodología propuesta e incluyendo el proceso de normalización, se puede integrar fácilmente el resultado de cualquier otra herramienta en el entorno.

En definitiva, durante este trabajo se ha desarrollado un entorno OSINT que permite automatizar todos los procesos haciendo posible homogeneizar los resultados de varias herramientas pensadas para funcionar de forma aislada y gestionando toda la información de manera centralizada a través de Dashboards personalizados. Además, de que el entorno es fácil de usar y se puede adaptar a las necesidades concretas de cada usuario. Por consiguiente, se considera que se ha logrado el resultado esperado que persigue el trabajo.

## 4.1. Seguimiento de la planificación establecida

En este punto se realiza una revisión sobre la planificación definida al comienzo del trabajo.

En general, se ha seguido correctamente los plazos definidos en la planificación propuesta. Si bien es verdad que se han producido ciertos retrasos puntuales. No obstante, aunque estos se habían previsto dentro de los posibles riesgos, no se podía determinar exactamente el retraso que podrían conllevar sobre la planificación. Concretamente, estos atrasos se han debido principalmente a problemas durante la fase de implementación.

También, se debe tener en cuenta que en muy poco tiempo se han tenido que implementar distintas herramientas (Pila Elastic, TheHarvester, DMitry, ExifTool…) sobre un sistema específico. Si a lo anterior se le añade que se ha realizado la automatización de todos los procesos y la homogenización de los resultados, implica una considerable carga de trabajo tanto en la parte de investigación como en la de implementación. Además, se deben mencionar el tiempo dedicado a solucionar pequeños problemas no previstos en un inicio.

### 4.1.1. Problemas encontrados en la implementación del trabajo

A continuación, se describen los principales problemas encontrados en la fase de implementación, los cuales han producido un cierto retraso en la planificación del proyecto.

#### 4.1.1.1. Estudio del diseño final de la Pila ELK

Se han estudiado diferentes tipos de escenarios para determinar cuál se adapta mejor a las necesidades del proyecto. En primer lugar, se pensó en un sistema formado por Logstash, ElasticSearch y Kibana. No obstante, el rendimiento del sistema se veía demasiado afectado por el funcionamiento de Logstash por lo que se tuvo que descartar este escenario y buscar otra alternativa más eficaz.

#### 4.1.1.2. Estudio de la viabilidad de Filebeat

Puesto que Filebeat se trata de un agente ligero, ofrece funcionalidades mucho más limitadas que Logstash. Por este motivo, se ha tenido que examinar en profundidad cada una de las opciones específicas que ofrece cada uno de los módulos de los que dispone. Todo ello con el objetivo de determinar si este software podría reemplazar a Logstash en este escenario específico. Finalmente, para optimizar el funcionamiento del software lo máximo posible, se decidió mantener activo un único módulo descartando los demás.

#### 4.1.1.3. Estudio del diseño final de Filebeat

Ha sido necesario destinar más de un par de días para la realización de pruebas asociadas con la conexión con ElasticSearch. Además, se han tenido

que barajar diferentes tipos de estrategias vinculadas al modo de gestionar el archivo de configuración de la herramienta y con el uso de pipelines compatibles con los datos enviados.

#### 4.1.1.4. Normalización de los datos

Si al problema de que cada herramienta OSINT de las seleccionadas ofrece una salida particular se le añade que en el entorno escogido se dispone de un agente ligero con ciertas limitaciones para procesar los datos, el problema se ve acentuado. Por consiguiente, se decidió abandonar la idea de procesar de manera independiente cada archivo con su estructura concreta y adoptar la estrategia de unificar la estructuración de los archivos. En definitiva, se tuvieron que destinar alrededor de tres días para buscar posibles alternativas que permitieran normalizar los archivos a través de comandos disponibles en el sistema, el uso de servicios online mediante una API o herramientas específicas.

#### 4.1.1.5. Problemas con la herramienta ExifTool

Puesto que dicha herramienta no trae incorporada la funcionalidad de extraer de forma automática archivos relevantes de un dominio, se han buscado posibles alternativas para maximizar su potencial. En primer lugar, se optó por el uso complementario de la herramienta Metagoofil [36], la cual permite automatizar la extracción de documentos sobre un dominio mediante consultas. No obstante, el resultado no fue el esperado puesto que dicho software ofrece escasos resultados y, si se aumenta el número de consultas, se obtiene un bloqueo de estas por el motor de búsqueda de Google.

En segundo lugar, se utilizó como alternativa el paquete wget [37]. Dicho software también permite recuperar archivos a través de los protocolos más utilizados como son HTTP, HTTPS, FTP o FTPS. Sin embargo, la única opción que ofrece de recuperar estos archivos es mediante el uso de búsquedas recursivas por lo que la delimitación del número total de archivos que se desea obtener se ve muy limitada. Esto supone que, si se analiza un dominio con una gran cantidad de archivos almacenados, la ejecución del software pueda demorar un tiempo demasiado elevado.

En definitiva, tras analizar las opciones anteriores, ambos métodos fueron descartados y se eligió determinar una ubicación sobre la cual operar. De esta forma se deja libertad a cómo obtener los datos relevantes del dominio y se centra el proceso en automatizar la extracción de los datos de interés de estos archivos ubicados en una ruta concreta.

#### 4.1.1.6. Tareas pendientes en la implementación

Si bien es cierto que se ha obtenido un sistema que permite automatizar y centralizar los resultados obtenidos a través de herramientas OSINT. No obstante, ha faltado tiempo para probar un mayor número de funcionalidades que ofrecen dichas herramientas. De esta forma, se podrían haber probado más opciones e incorporado un mayor número de información en los archivos procesados lo que desemboca en informes más completos.

## 4.2. Evaluación de los objetivos alcanzados

En este apartado se analiza si se han cumplido los objetivos establecidos al comienzo del proyecto.

**Objetivos de investigación:**

- ✓ **Investigación de las técnicas y herramientas OSINT disponibles para la obtención de datos personales, información de redes sociales, etc.**

Se han podido estudiar algunas de las principales técnicas y herramientas utilizadas para recabar este tipo de información. No obstante, puede resultar complicado llevar a cabo la elección de las herramientas ya que existen muchas alternativas que ofrecen resultados similares. Además, dicho programas cuentan con el inconveniente de que suelen estar diseñados para funcionar de manera aislada lo que complica la integración de varios resultados.

- ✓ **Investigación sobre las posibilidades que ofrecen los principales motores de búsqueda (Google, Bing…).**

Se considera que estos motores de búsqueda ofrecen un excelente servicio el cual no debe pasarse por alto en una investigación OSINT. También, se han investigado los motores de búsqueda no convencionales los cuales pueden realizar búsquedas en la web profunda por lo que permiten acceder a información adicional.

- ✓ **Investigación de plataformas enfocadas en OSINT.**

Se han estudiado las principales plataformas que permiten realizar OSINT junto con las herramientas y funciones que ofrece cada distribución. Además, se han tenido en cuenta factores adicionales como la documentación, la frecuencia de lanzamiento de nuevas versiones o la comunidad activa de usuarios.

- ✓ **Investigación de alternativas para la visualización de los datos.**

Se han estudiado las diferentes alternativas que ofrecen las herramientas de la Pila Elastic en lo relacionado con el tratamiento de ficheros de registro (Logstash y Filebeat). Además, se ha investigado los métodos de indexación (ElasticSearch) y diseño de visualizaciones personalizadas (Kibana).

**Objetivos de implantación:**

- ✓ **Instalación y configuración de la distribución seleccionada y de las herramientas y dependencias necesarias.**

Este objetivo se considera cumplido ya que se ha implementado la distribución Kali Linux y se ha logrado ejecutar sobre esta las herramientas seleccionadas. Si bien es verdad, una de las herramientas no se ha podido ejecutar en el sistema seleccionado, pero no por problemas de dependencias o configuración, sino debido a problemas de compatibilidad.

- ✓ **Aprender a utilizar las herramientas OSINT escogidas.**

Se ha aprendido a utilizar las herramientas seleccionadas por lo que se considera alcanzado este objetivo. No obstante, por falta de tiempo no se ha llegado a realizar un mayor número de consultas o el uso de funciones de búsqueda más avanzadas sobre dichas herramientas.

- ✓ **Automatizar consultas e integrar las herramientas.**

Se ha logrado automatizar por completo las consultas y normalizar sus resultados. Por ello, ha sido posible unificar toda la información obtenida en un sistema centralizado. Por tanto, este objetivo se considera alcanzado.

- ✓ **Procesar y visualizar los resultados obtenidos a partir de las herramientas OSINT.**

Tras haber conseguido normalizar toda la información obtenida a través de las herramientas, ha sido posible definir una estrategia de procesamiento efectiva que permite almacenar la información de forma centralizada y visualizar todos los datos en visualizaciones personalizadas. Por consiguiente, se ha logrado la consecución de este objetivo.

- ✓ **Priorizar la eficiencia de los procesos involucrados y la eficacia de la información procesada y de los resultados obtenidos.**

Durante el proceso se ha probado la incorporación de diferentes entornos con la finalidad de elegir la alternativa que mejor se adapte a las necesidades del proyecto. Además, se ha barajado la incorporación de diferentes herramientas, priorizando aquellas que se ejecutan en modo terminal y ofrecen resultados rápidos requiriendo pocos recursos para funcionar.
Por otra parte, se ha determinado una estrategia de procesamiento de la información que garantice que únicamente se almacene en el sistema aquella información considerada de utilidad. Por todo ello, este objetivo se considera alcanzado.

## 4.3. Trabajo futuro

A continuación, se enumeran varias líneas de trabajo futuro con las que mejorar el trabajo.

- Analizar en mayor profundidad las herramientas incorporadas de tal forma que se pueda agregar un mayor número de funcionalidades que estas ofrecen. Un ejemplo concreto de ello es el tratado con el software OSRFramework, el cual ofrece diferentes módulos adicionales que su incorporación puede ser de gran utilidad.

- Agregar nuevas herramientas para la recolección de información. Debido a que el tiempo dedicado a este proyecto es reducido, se ha tenido que limitar el número de herramientas a utilizar. No obstante, siguiendo la estrategia definida, se pueden añadir nuevas herramientas en el proceso de una manera bastante metódica y trivial.

- Automatizar la creación pipelines. En este proyecto se ha utilizado directamente la línea de comandos que ofrece la herramienta Kibana para interactuar con ElasticSearch, la cual es bastante fácil de utilizar.

Además, se dispone de un asistente para la creación de pipelines por lo que, tras aplicar dicha conversión, se genera de manera automática el índice. No obstante, sería muy interesante automatizar también este proceso.

- Perfeccionamiento y automatización de los Dashboards. Si bien es verdad que las visualizaciones creadas se adaptan bien a los resultados obtenidos mostrando toda aquella información que se ha considerado más relevante. Sin embargo, en este escenario se ha optado por crear una visualización por cada herramienta. Por tanto, como posible idea de mejora, se podrían crear plantillas para que después se puedan exportar y reutilizar en diferentes herramientas con pocas modificaciones.

# 5. Glosario

**CYBINT:** Combinación de las palabras "Ciber" e "Inteligencia" (Cyber Intelligence). Se refiere al proceso de recopilación, análisis y utilización de información cibernética con fines de inteligencia. Puede considerarse como un subconjunto de OSINT.

**Dork:** Término empleado en el ámbito de la seguridad informática para referirse a una cadena de búsqueda específica utilizada en motores de búsqueda para encontrar información sensible o vulnerable en sitios web.

**Footprint:** También conocido como "huella" en español, se refiere a la información rastreable y dejada por una entidad (Como una persona, una organización o un sistema) en el entorno digital. Puede incluir información como direcciones IP, registros de actividad, datos de navegación, entre otros.

**FTP:** El Protocolo de Transferencia de Archivos (File Transfer Protocol) es un protocolo estándar utilizado para transferir archivos a través de una red. Permite la transferencia de archivos entre un cliente y un servidor.

**FTPS:** El Protocolo de Transferencia de Archivos Seguro (File Transfer Protocol Secure) es una versión segura del protocolo FTP que utiliza una capa de seguridad adicional a través de SSL/TLS (Secure Sockets Layer/Transport Layer Security) para cifrar la comunicación entre el cliente y el servidor.

**HTTP:** El Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es el protocolo utilizado para el intercambio de información en la World Wide Web. Define la forma en que los mensajes son formateados y transmitidos, permitiendo la comunicación entre un cliente y un servidor web.

**HTTPS:** El Protocolo de Transferencia de Hipertexto Seguro (Hypertext Transfer Protocol Secure en inglés) es una versión segura del protocolo HTTP que utiliza una capa de seguridad adicional a través de SSL/TLS para cifrar la comunicación entre el cliente y el servidor web.

**HUMINT:** Abreviatura de "Human Intelligence" (Inteligencia Humana en español). Hace referencia a la información de inteligencia obtenida a través de fuentes humanas, como espías, informantes o agentes encubiertos. Puede considerarse como un subconjunto de OSINT.

**IP:** Siglas de "Internet Protocol" (Protocolo de Internet en español). Es un protocolo de red que permite la comunicación y el intercambio de datos entre dispositivos conectados a una red de computadoras.

**Oracle VM VirtualBox:** Es una aplicación de virtualización que permite la creación y ejecución de máquinas virtuales en un entorno de software. Es decir, permite ejecutar múltiples sistemas operativos simultáneamente en un solo equipo físico.

**Pipeline:** En el ámbito de la programación y la ingeniería de software, se refiere a una secuencia de pasos o procesos interconectados que se ejecutan de manera secuencial para realizar una tarea o procesar datos.

**POST:** Es un método utilizado en el protocolo HTTP para enviar datos desde un cliente a un servidor. Los datos se incluyen en el cuerpo de la solicitud y son enviados para su procesamiento en el servidor.

**Timeout:** Se refiere a un período de tiempo establecido para una operación o proceso. Si una operación no se completa dentro de ese tiempo establecido, se produce un "timeout".

**Token:** Elemento de autenticación utilizado para verificar la identidad de un usuario. Puede ser una contraseña, un código numérico o un objeto físico, como una tarjeta inteligente.

**API:** Siglas de Interfaz de Programación de Aplicaciones (Application Programming Interface). Es un conjunto de reglas y protocolos que permiten que diferentes aplicaciones se comuniquen entre sí y compartan información y funcionalidades.

**CURL:** Herramienta de línea de comandos utilizada para realizar transferencias de datos en diversas redes. Es capaz de trabajar con una variedad de protocolos, como HTTP, HTTPS, FTP, entre otros.

**OSINT:** Siglas de "Open Source Intelligence" (Inteligencia de Fuentes Abiertas en español). Se refiere a la recopilación y análisis de información obtenida de fuentes de acceso público, como sitios web, redes sociales y bases de datos públicas, con el fin de obtener inteligencia y conocimiento.

**SOCMINT:** Siglas de "Social Media Intelligence" (Inteligencia de Medios Sociales en español). Se refiere a la recolección y análisis de información generada en plataformas de redes sociales con el fin de obtener inteligencia sobre individuos, grupos, tendencias o eventos.

# 6. Bibliografía

[1] U. E. Internet, «Los riesgos para la intimidad de los “Me gusta” de Facebook». https://www.elmundo.es/elmundo/2013/03/11/ciencia/1363027585.html (accedido 1 de marzo de 2023).

[2] M. Ganguly, «THE FUTURE OF INVESTIGATIVE JOURNALISM IN THE AGE OF AUTOMATION, OPEN-SOURCE INTELLIGENCE (OSINT) AND ARTIFICIAL INTELLIGENCE (AI)».

[3] «INCIBE | INCIBE». https://www.incibe.es/ (accedido 15 de marzo de 2023).

[4] «Osint La Informacion Es Poder | INCIBE-CERT | INCIBE». https://www.incibe.es/incibe-cert/blog/osint-la-informacion-es-poder (accedido 16 de marzo de 2023).

[5] C. Martorella, «theHarvester». (accedido 30 de marzo de 2023) [En línea]. Disponible en: https://github.com/laramies/theHarvester

[6] J. Greig, «Dmitry» (accedido 30 de marzo de 2023) [En línea]. Disponible en: https://github.com/jaygreig86/dmitry

[7] «exiftool». ExifTool by Phil Harvey, (accedido 31 de marzo de 2023) [En línea]. Disponible en: https://github.com/exiftool/exiftool

[8] «FOCA (Fingerprinting Organizations with Collected Archives)». ElevenPaths, (accedido 31 de marzo de 2023) [En línea]. Disponible en: https://github.com/ElevenPaths/FOCA

[9] G. Criscione, «Osintgram». (accedido 3 de abril de 2023) [En línea]. Disponible en: https://github.com/Datalux/Osintgram

[10] i3visio, «OSRFramework». (accedido 3 de abril de 2023) [En línea]. Disponible en: https://github.com/i3visio/osrframework

[11] S. Micallef, «spiderfoot». (accedido 4 de abril de 2023) [En línea]. Disponible en: https://github.com/smicallef/spiderfoot

[12] «Maltego Homepage». https://www.maltego.com/ (accedido 4 de abril de 2023).

[13] «List of Operating Systems for OSINT (Open-Source Intelligence)», *PenTestIT*, 10 de octubre de 2018. https://pentestit.com/operating-systems-open-source-intelligence-osint-list/ (accedido 5 de abril de 2023).

[14] «Sistemas operativos para investigaciones OSINT + Comparativa», 21 de abril de 2020. https://ciberpatrulla.com/sistemas-operativos-osint/ (accedido 5 de abril de 2023).

[15] «Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution», *Kali Linux*, 29 de marzo de 2023. https://www.kali.org/ (accedido 5 de abril de 2023).

[16] «IntelTechniques by Michael Bazzell». https://inteltechniques.com/ (accedido 5 de abril de 2023).

[17] HuronOsint, «HURON». 17 de abril de 2023. (accedido 5 de abril de 2023) [En línea]. Disponible en: https://github.com/HuronOsint/OsintDistro

[18] «Osintux | Distribución Linux inteligencia en fuentes abiertas OSINT», *Osintux*. https://www.osintux.org/ (accedido 5 de abril de 2023).

[19] A. Shear, «DORA OSINT VM». 17 de abril de 2023. (accedido 5 de abril de 2023) [En línea]. Disponible en: https://github.com/axlshear/dora-osint-vm

[20] «Tsurugi Linux | Digital Forensics, Osint and malware analysis Linux Distribution». https://tsurugi-linux.org/ (accedido 5 de abril de 2023).

[21] «CSI Linux - Download today!» https://csilinux.com/ (accedido 5 de abril de 2023).

[22] «Trace Labs». https://www.tracelabs.org/ (accedido 5 de abril de 2023).

[23] «Logstash: Recopila, parsea y transforma logs», *Elastic*. https://www.elastic.co/es/logstash (accedido 7 de abril de 2023).

[24] «Beats: Agentes de datos para Elasticsearch | Elastic». https://www.elastic.co/es/beats/ (accedido 7 de abril de 2023).

[25] «Filebeat: Análisis de logs ligero y Elasticsearch», *Elastic*. https://www.elastic.co/es/beats/filebeat (accedido 7 de abril de 2023).

[26] «Metricbeat: Agente ligero para métricas», *Elastic*. https://www.elastic.co/es/beats/metricbeat (accedido 7 de abril de 2023).

[27] «Packetbeat: Analíticas de red con Elasticsearch», *Elastic*. https://www.elastic.co/es/beats/packetbeat (accedido 7 de abril de 2023).

[28] «Winlogbeat: Analizar logs de eventos de Windows», *Elastic*. https://www.elastic.co/es/beats/winlogbeat (accedido 7 de abril de 2023).

[29] «Auditbeat: Agente ligero para información de auditoría», *Elastic*. https://www.elastic.co/es/beats/auditbeat (accedido 7 de abril de 2023).

[30] «Inteligencia artificial para operaciones (AIOps) con Elastic Observability», *Elastic*. https://www.elastic.co/es/observability/aiops (accedido 16 de marzo de 2023).

[31] «Community Beats | Beats Platform Reference [8.8] | Elastic». https://www.elastic.co/guide/en/beats/libbeat/current/community-beats.html (accedido 7 de abril de 2023).

[32] «Elasticsearch: Motor de búsqueda y analítica distribuido oficial», *Elastic*. https://www.elastic.co/es/elasticsearch (accedido 7 de abril de 2023).

[33] «Kibana: Explora, visualiza y descubre datos», *Elastic*. https://www.elastic.co/es/kibana (accedido 7 de abril de 2023).

[34] «Oracle VM VirtualBox». https://www.virtualbox.org/ (accedido 12 de abril de 2023).

[35] «Get Kali», *Kali Linux*. https://www.kali.org/get-kali/ (accedido 12 de abril de 2023).

[36] «metagoofil | Kali Linux Tools», *Kali Linux*. https://www.kali.org/tools/metagoofil/ (accedido 26 de abril de 2023).

[37] «Wget - GNU Project - Free Software Foundation». https://www.gnu.org/software/wget/ (accedido 26 de abril de 2023).

# 7. Anexo A: Configuración de Filebeat

A continuación, se muestra el contenido del archivo de configuración de Filebeat donde se subrayan en amarillo las líneas que han sido añadidas o modificadas.

```
###################### Filebeat Configuration Example
#########################

# This file is an example configuration file highlighting only the most
common
# options. The filebeat.reference.yml file from the same directory contains
all the
# supported options with more comments. You can use it as a reference.
#
# You can find the full configuration reference here:
# https://www.elastic.co/guide/en/beats/filebeat/index.html

# For more available modules and options, please see the
filebeat.reference.yml sample
# configuration file.

# ============================== Filebeat inputs
===============================

filebeat.inputs:

# CONFIG THEHARVESTER
- type: log
  id: theHarvester_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/theHarvester/theHarvester.csv
  index: theHarvester_youtube
  pipeline: theharvester

# CONFIG DMITRY
- type: log
  id: dmitry_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/DMitry/dmitry_gmail.csv
  index: dmitry_gmail
  pipeline: dmitry

  # CONFIG EXIFTOOL
- type: log
```

```
  id: exiftool_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/ExifTool/exiftool_perros_parse.csv
  index: exiftool_perros
  pipeline: exiftool

  # CONFIG OSRFRAMEWORK
- type: log
  id: osrframework_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/OSRFramework/osrframework_ibaillanos.csv
  index: osrframework_ibai
  pipeline: osrframework

# CONFIG SPIDERFOOT
- type: log
  id: spiderfoot_csv
  enabled: true
  paths:
    - /home/bmaeso/GitHub/Spiderfoot/spiderfoot_google.csv
  index: spiderfoot_google
  pipeline: spiderfoot

# ============================== Filebeat modules
==============================

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  reload.enabled: false

  # Period on which files under path should be checked for changes
  #reload.period: 10s

# ======================= Elasticsearch template setting
=======================

setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false

# ================================== General
==================================
```

```
# The name of the shipper that publishes the network data. It can be used to
group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging

# ============================== Dashboards
==============================
# These settings control loading the sample dashboards to the Kibana index.
Loading
# the dashboards is disabled by default and can be enabled either by setting
the
# options here or by using the `setup` command.
#setup.dashboards.enabled: false

# The URL from where to download the dashboards archive. By default this URL
# has a value which is computed based on the Beat name and version. For
released
# versions, this URL points to the dashboard archive on the
artifacts.elastic.co
# website.
#setup.dashboards.url:

# ================================== Kibana
==================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana
API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and
5601)
  # In case you specify and additional path, the scheme is required:
http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "localhost:5601"

  # Kibana Space ID
```

```
  # ID of the Kibana Space into which the dashboards should be loaded. By
default,
  # the Default Space will be used.
  #space.id:

# ============================== Elastic Cloud
==============================

# These settings simplify using Filebeat with the Elastic Cloud
(https://cloud.elastic.co/).

# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:

# The cloud.auth setting overwrites the `output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth:

# ================================== Outputs
==================================

# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ------------------------
---
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "syfchRNdPqpZMjZ*t9rc"

  ssl:
    enabled: true
    ca_trusted_fingerprint:
"2B3FC23161E4F2880ED9A0676189BDFC38F68BD9CDBA2F15A6E4606A426F4D5C"



# ---------------------------- Console Output ------------------------------
---
#output.console.pretty: true
```

```
# ------------------------------ Logstash Output -------------------------------
---
#output.logstash:
  # The Logstash hosts
  #hosts: ["localhost:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

# ================================= Processors
=================================
processors:
  - drop_fields:
      fields: ["agent", "log", "input", "host", "ecs" ]
  #- add_host_metadata:
  #    when.not.contains.tags: forwarded
  #- add_cloud_metadata: ~
  #- add_docker_metadata: ~
  #- add_kubernetes_metadata: ~

# ================================== Logging
==================================

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
#logging.level: debug

# At debug level, you can selectively enable logging only for some
components.
# To enable all selectors use ["*"]. Examples of other selectors are "beat",
# "publisher", "service".
#logging.selectors: ["*"]

# ============================= X-Pack Monitoring
==============================
# Filebeat can export internal metrics to a central Elasticsearch monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.
 The
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
#monitoring.enabled: false
```

```
# Sets the UUID of the Elasticsearch cluster under which monitoring data for
this
# Filebeat instance will appear in the Stack Monitoring UI. If
output.elasticsearch
# is enabled, the UUID is derived from the Elasticsearch cluster referenced
by output.elasticsearch.
#monitoring.cluster_uuid:

# Uncomment to send the metrics to Elasticsearch. Most settings from the
# Elasticsearch output are accepted here as well.
# Note that the settings should point to your Elasticsearch *monitoring*
cluster.
# Any setting that is not set is automatically inherited from the
Elasticsearch
# output configuration, so if you have the Elasticsearch output configured
such
# that it is pointing to your Elasticsearch monitoring cluster, you can
simply
# uncomment the following line.
#monitoring.elasticsearch:

# ============================== Instrumentation
===============================

# Instrumentation support for the filebeat.
#instrumentation:
    # Set to true to enable instrumentation of filebeat.
    #enabled: false

    # Environment in which filebeat is running on (eg: staging, production,
etc.)
    #environment: ""

    # APM Server hosts to report instrumentation results to.
    #hosts:
    #   - http://localhost:8200

    # API Key for the APM Server(s).
    # If api_key is set then secret_token will be ignored.
    #api_key:

    # Secret token for the APM Server(s).
    #secret_token:


# ================================= Migration
==================================
```

```
# This allows to enable 6.7 migration aliases
#migration.6_to_7.enabled: true
```

En la sección "*filebeat.inputs*" están definidas las configuraciones de las herramientas seleccionadas en el trabajo. Todas ellas siguen la misma estructura por lo que se ha cogido el primer ejemplo para detallar qué es lo que hace exactamente cada línea.

```
1º - type: log
2º   id: theHarvester_csv
3º   enabled: true
4º   paths:
5º     - /home/bmaeso/GitHub/theHarvester/theHarvester.csv
6º   index: theHarvester_youtube
7º   pipeline: theharvester
```

1. Define los datos de entrada de tipo registro.
2. Identificador utilizado para la entrada.
3. Indica que la entrada está habilitada.
4. Sección en la que se definen la ruta o rutas de los archivos de registro.
5. Ruta del archivo de registro que se va a procesar.
6. Índice de ElasticSearch en el que se guarda el contenido procesado.
7. Pipeline registrada en ElasticSearch, la cual procesa la información recogida.

Después, es necesario realizar modificaciones en el apartado "*output.elasticsearch*" para establecer la conexión entre el agente Filebeat y ElasticSearch.

```
  hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "syfchRNdPqpZMjZ*t9rc"

  ssl:
    enabled: true
    ca_trusted_fingerprint:
"2B3FC23161E4F2880ED9A0676189BDFC38F68BD9CDBA2F15A6E4606A426F4D5C"
```

Es necesario definir la dirección IP y el puerto, el protocolo, las credenciales y la huella digital del certificado. Además, es importante mencionar que el certificado ha sido generado automáticamente por ElasticSearch al iniciarlo por primera vez.

La última configuración realizada se encuentra en la sección “`processors`”.

```
processors:
  - drop_fields:
      fields: ["agent", "log", "input", "host", "ecs" ]
```

Puesto que Filebeat agrega ciertos campos por cada línea que procesa, con la instrucción `drop_fields` se indica que no se quieren almacenar esos campos adicionales. De esta forma se consigue optimizar al máximo el espacio de almacenamiento en ElasticSearch. No obstante, Filebeat agrega, al menos, el campo ***@timestamp***, el cual no puede ser eliminado directamente mediante el proceso mencionado por lo que se vuelve necesario removerlo especificándolo en la configuración del pipeline.