UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Ingeniería Informática en Ingeniería del Software

Trabajo Fin de Grado

Planteamiento de un CTF para practicar hacking ético e informática forense

UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Ingeniería Informática en Ingeniería del Software

Trabajo Fin de Grado

Planteamiento de un CTF para practicar hacking ético e informática forense

Autor: Mª Teresa Conejo Sequedo

Tutor: Andrés Caro Lindo

Co-Tutor/es: José Carlos Sancho Núñez

ÍNDICE GENERAL DE CONTENIDOS

ÍNDICE DE FIGURAS

RESUMEN

El hacking ético y la informática forense son áreas que están atrayendo una gran atención en los últimos años. La formación en estos temas está cada vez más demandada, existiendo la posibilidad de combinar metodologías tradicionales con otras alternativas como pueden ser la solución de retos o problemas relacionados con la temática.

El presente Trabajo de Fin de Grado gira precisamente en torno a la construcción de retos de hacking ético y de informática forense, del estilo de Configuración de CTF (*Capture the Flag*), consistente en la elaboración de entornos donde practicar dichos retos, permitiendo a los participantes desarrollar distintas habilidades según el reto que estén resolviendo en ese momento. Inclusive en el mismo podemos englobar el entendimiento de nuevas herramientas, un aumento en el conocimiento de la ciberseguridad, proporcionar una forma de entrenamiento...

En primer lugar, se definen los CTF como un estilo de competición, generalmente en equipo, que se basa en diversas pruebas, donde hay que conseguir *banderas*. Posteriormente, el proyecto se enfoca en 4 de las categorías en las que se dividen las pruebas, que son: esteganografía (ocultar mensajes u objetos, dentro de otros mensajes u objetos de forma que su existencia no pueda ser detectada), informática forense (adquirir, preservar, obtener y presentar datos que han sido procesados electrónicamente y guardados en soportes informáticos), reconocimiento (técnica que requiere de un amplio conjunto de herramientas, ya que sus objetivos pueden abarcar cualquier punto dentro de la red) y programación (resolver alguna tarea compleja con ayuda de un código o script que facilite este cometido).

Atendiendo a estas categorías se desarrollan un total de 8 retos, que se especifican a continuación junto con lo que nos aportan:

1. Una imagen vale más que mil palabras: reto de esteganografía para conocer de primera mano lo que puede ocultar una imagen, así como todo lo que los metadatos de un archivo nos pueden ofrecer y la posibilidad de modificarlos.
2. El mapa de… ¿la bandera?: desafío de esteganografía ampliado a otros conceptos como el tratamiento de un fichero pcap u ocultar información en archivos de audio.

3. El mundo multimedia: también se trata de un reto de esteganografía, ampliando la manipulación de archivos multimedia, al último de los elementos que faltaba, los vídeos. Con el descubrimiento de la escasez de herramientas para el tratamiento de estos.
4. Podría ser algo más: mediante este reto, se abarca una parte de la informática forense, centrándose en el uso de imágenes de sistemas, en este caso de un servidor. Se tratan aspectos éticos como el uso de fuerza bruta, con la intención de crackear para descubrir contraseñas ajenas.
5. Tráfico de armas: a través del siguiente desafío se adquirieren nuevo conocimientos relacionados con la preservación de evidencias, la recuperación de elementos, descubrir extensiones de archivos, crackear contraseñas de documentos, nuevas formas de cifrado...
6. Un reto explosivo: se abandona el clásico reto forense donde nos limitamos a analizar una imagen, ya que en esta ocasión se parte de un fichero que registra un tráfico de red. Se trabaja con la unión de varios archivos, localización de pistas ocultas…
7. Lo que ofreció: recorrido por distintas herramienta open source, análisis de la falta de seguridad en estructuras como las redes sociales…
8. Una lección de matemáticas: práctica del principal conocimiento adquirido en la carrera. Se parte de un problema, se analiza y se encuentra la forma de solventarlo mediante un código.

PALABRAS CLAVES: CTF, análisis, herramientas y bandera.

1. INTRODUCCIÓN

Este trabajo sentará las bases de planificación para la preparación, puesta en marcha y finalización de un proyecto sobre la implementación de algunos retos de hacking ético, del estilo de configuración de CTF (Capture The Flag). El proyecto abarca el segundo semestre y la investigación para su puesta en marcha a partir de todos los medios y recursos disponibles es el motor impulsor del desarrollo del mismo.

El proyecto consiste en la elaboración de entornos donde practicar dichos retos, permitiendo a los participantes desarrollar distintas habilidades según el reto que estén resolviendo en ese momento. Para comprender el enfoque de este trabajo, primero debemos tener claro qué son los retos Captura la bandera o *Capture the flag* (CTF). De éstos estaremos hablando durante toda la implementación, puesta en marcha y finalización del proyecto.

Los CTF son un estilo de competición, generalmente en equipo, que se basa en diversas pruebas. Cada una de ellas nos da una serie de puntos en función de la dificultad (o de otros criterios). Las pruebas se basan en conseguir banderas. Estas banderas son una secuencia de caracteres que suele empezar siempre de la misma forma (por ejemplo, precedida de la palabra FLAG). En cada competición los organizadores informan de cómo van a comenzar las suyas.

Las pruebas suelen ser de criptografía, ingeniería inversa, explotación web, forense, etc. El límite está en la imaginación del organizador.

Los retos se pueden dividir en dos modalidades, por una parte, **jeopardy**; y por otra, **ataque y defensa**. La primera modalidad se suele usar en concursos con mucha gente, ya que es más sencillo de gestionar, y se intenta reunir el mayor número de puntos. En cambio, la segunda se usa donde hay pocos participantes: los equipos tienen las banderas y tratan de protegerlas mientras que atacan al enemigo para robar las suyas. O en otros casos, evitan que los contrincantes puedan obtener alguna.

Para poder resolver algunos de estos retos, lo primero es tener conocimientos en la rama de la seguridad informática e informática forense, así como utilizar diversas herramientas.

No obstante, durante este documento no tratamos de explicar cómo se resuelven estos retos de los que hablamos, sino cómo se desarrollan algunos de ellos, para que otras personas puedan solventarlos y obtener sus banderas. Se detallarán las herramientas que han sido necesarias para construirlos, así como las fuentes y diversas informaciones, que han permitido darle forma y completarlos.

Los retos fabricados se basan en la primera modalidad (jeopardy), se resolverán de forma individual o grupal, con la intención de ir recuperando las banderas ocultas en los mismos, y reuniendo los puntos adjudicados a cada uno según unos criterios u otros factores establecidos. Todos estos detalles, serán expuestos en puntos posteriores.

2. OBJETIVOS

Al igual que en años anteriores, este 2019, se celebran las Jornadas Nacionales de Investigación en Ciberseguridad (JNIC). Se trata de un congreso científico que promueve el contacto, intercambio y discusión de ideas, conocimientos y experiencias entre la red académica y de investigación, por una parte, y profesionales y empresas por otra.

Dentro de estas jornadas, como novedad en 2019, se presenta una competición de CTF. Los participantes deben resolver un conjunto de retos, formados por una serie de tareas, constituyendo éstas un paso que conduce a alcanzar la siguiente, y finalmente, la bandera. El objetivo de este proyecto es el diseño y construcción de un conjunto de retos permitiendo así dar soporte a eventos como este. Del objetivo principal pueden obtenerse otros sub-objetivos expuestos a continuación:

- Descubrir el nivel de participación, actividad y ritmo de los participantes a partir del desenlace, dibujado por los mismos en los desafíos abarcados durante la competición, y las ideas empleadas para llegar a él.
- Conocer nuevas herramientas tanto por parte del desarrollador del certamen como de la persona que participa en él. Determinando el uso que se hace de dichas herramientas y cómo se enmarcan dentro de su estrategia.
- Investigar nuevas posibles formas de planteamiento y desarrollo de los CTF, promoviendo un aumento en el conocimiento de la Ciberseguridad. Esto nos permite alcanzar un mayor grado de robustez, en lo que respecta a evitar la

exposición de la información que pueda perjudicar tanto a personas como sistemas. Mejorando de esta manera la capacidad para actuar antes, durante y después, ya que con estos métodos se amplía en muchos casos el conocimiento para romperla.

- En relación con la anterior, descubrir el pensamiento de un delincuente, en cuanto a diversas formas de ocultar la información comprometedora o dificultar la obtención de la misma.
- Construir una forma de entrenamiento, en la que se prepara a la persona para situaciones en las que normalmente no están. Así mismo, puede ser utilizado como un conjunto de procesos o pruebas de selección en empresas, para formar sus equipos de Ciberseguridad.
- Concretar un modelo de evaluación idóneo para un trabajo centrado en la Seguridad Informática, así como en las derivaciones de la misma, permitiéndonos cercar un área tan amplia como es esta, a través de una serie de pasos englobados en un único reto.
- Comparar la adecuación de unos retos frente a otros, en función de diversos factores como la complejidad, el ingenio, la creatividad…así como otras muchas cualidades que pueden definir tanto al que lo creó como al que aportó su solución.

3. ANTECEDENTES / ESTADO DEL ARTE

El mundo de la Informática ha avanzado a pasos agigantados en los últimos veinte años. Con la llegada de Internet, la Informática y las comunicaciones van de la mano, y con ellas ha nacido la posibilidad de realizar ciberdelitos. No sólo porque ahora es un medio ideal para la realización de delitos, sino porque la propia Informática puede ser el objeto del mismo.

Nos referimos, concretamente, a delitos contra la propiedad intelectual, la propiedad industrial, el derecho a la intimidad (interceptación de comunicaciones), el patrimonio (estafas, apropiación indebida y fraudes), la libertad y amenazas, el honor (calumnias e injurias), el mercado y los consumidores (revelación de secretos, publicidad engañosa y falsedades documentales), la libertad sexual y prostitución. Asimismo, delitos de sabotaje informático, convencionales (espionaje, espionaje

industrial y terrorismo informático), del mal uso de la red (cybertorts, usos comerciales no éticos, actos parasitarios y obscenidades). Y, por último, delitos tradicionalmente denominados "informáticos" (acceso no autorizado, destrucción de datos, Hacking y ciberterrorismo, infracción de los derechos de autor, infracción del copyright de bases de datos, interceptación de e-mail, estafas electrónicas, transferencia de fondos y phising).

Hace 30 años nadie se preocupaba de si alguien podía acceder a su sistema informático de manera ilegítima y causar daños y, en general, de los ciberdelitos. Ahora, el desarrollo acelerado de la Informática se ha encargado de hacer patente dicha preocupación, y es por ello que es un tema que debe tomarse a conciencia por profesionales de la Ley, a la par que por profesionales de la Informática, y buscar una solución.

Por la importancia de este tema es por lo que hemos decidido elaborar un Proyecto Final de Carrera, que describa esta problemática desde el punto de vista del ciberdelincuente, abarcando distintas vías y formas con las que puede delinquir.

En el presente documento nos centraremos en los delitos de Hacking, entendiendo por este, el conjunto de técnicas para acceder a un sistema informático sin autorización, haciendo uso de medios como Sniffers o escaneadores de puertos, (programas que buscan claves, passwords y puertos abiertos), y que actúan conjuntamente con otras aplicaciones. O lo que es lo mismo, como el medio para vulnerar el sistema con la intención de adquirir información ajena.

Es aquí donde surge la figura del hacker, al cual se le define como un entusiasta de la informática con un gran interés en aprender acerca de los sistemas y de cómo usarlos de formas innovadoras. Esto les lleva a penetrar en estos, normalmente a través de Internet, en busca de esa información que los lleve a encontrar más conocimientos. Una vez dentro, se limitan a dejar su marca, sin estropear los datos de los demás. Disfrutan del reto intelectual de superar o rodear las limitaciones de forma creativa.

El desarrollo de este proyecto, nos ha permitido transformarnos en este personaje a medida que desarrollábamos los distintos desafíos, en los que buscábamos en todo momento el amplio abanico de posibilidades que nos ofrecen algunas de las herramientas, para poder quebrantar la seguridad presente en archivos o sistemas. Sin

embargo, esta imagen que adoptamos la hacemos desde un marco ético y legal. Es decir, se tratan de retos de hacking ético. Nos convertimos en hacker éticos o de sombrero blanco, que son aquellos que apuntan a mejorar la seguridad, encontrar agujeros en ella y notificar a la víctima para que tenga la oportunidad de arreglarlos antes de que un hacker menos escrupuloso (ciberdelincuente) los explote.

Por tanto, el hacking ético no es otra cosa que analizar el sistema de seguridad TI y simular ataques propios de la ciberdelincuencia para evaluar la seguridad del mismo. Además, este se encuentra en una evolución que le permitirá dejar de ser una herramienta usada en procesos de investigación de delitos informáticos, a ser una de las profesiones del futuro. Es por ello, que a partir de la labor que llevan a cabo los desarrolladores de los CTF, podemos simular este tipo de situaciones y permitir acercarnos más al entendimiento de este nuevo oficio. Facilitando su práctica, de una forma meramente académica y concienciando que su uso se puede hacer con la intención de ayudar y sin perder la opción de ampliar nuestros conocimientos.

Existen diferentes organismos dedicados a este sector. En España existe el Instituto Nacional de Ciberseguridad en España (INCIBE), que se trata de una entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos. Es una red académica y de investigación, profesionales, empresas y especialmente para sectores estratégicos. Con una actividad basada en la investigación, la prestación de servicios y la coordinación con agentes con competencias en la materia.

Por otra parte, destinados al entrenamiento en seguridad informática, mediante la elaboración de CTF's, existen varios sitios. Algunos solo aportan un calendario de futuros eventos (CTF-Time, CTF Calendar), otros están destinados a ofrecer información sobre sitios web activos, estadísticas, últimas noticias… (Capture the flag – captf.com, We Chall). Y finalmente, podemos encontrar aquellos dedicados a publicar retos (Hack Players, Security Art Work, CTF365, Root me…), también para los más pequeños (rootz, picoctf) y páginas de competiciones internacionales (RuCTF, Nuit du Hack, DEFCON).

En último lugar, es importante comentar que, de cara a la confección de los desafíos, la resolución de muchos de los retos proporcionados por las páginas anteriores, pueden servir de inspiración para los mismos, y para la construcción de

algunas de sus partes, debido a que en muchas ocasiones es la misma. No obstante, también existen algunas webs que ofrecen varias ideas, centrándose en herramientas, como Mellivora (con una base de datos básica y un motor escrito en PHP). Pero no existe ninguna línea de investigación donde se enmarque este cometido y que nos proporcione una explicación más general. La única forma de aprendizaje para poder desarrollar estos retos, proviene de la perseverancia y la práctica.

4. METODOLOGÍA

Los CTF se dividen en dos o incluso tres modalidades como ya se comentó en secciones anteriores (jeopardy, ataque y defensa, mixto de los dos anteriores). Y además los retos se suelen distribuir, generalmente, en las siguientes categorías:

- Análisis Forense (*Forensics*): son los retos más comunes; en ellos se hace uso de imágenes de memoria, de discos duros o capturas de red, las cuales almacenan diferentes tipos de información.
- Criptografía (*Crypfto*): se trata de textos cifrados mediante un criptosistema determinado (uso de algoritmos DES, AES…).
- Esteganografía (*Stego*): imágenes, sonidos o vídeos que ocultan información en su interior.
- Explotación (*Pwn*): este tipo de retos está más relacionado con el concepto que conocemos de Hacking, pues se basan en el descubrimiento de vulnerabilidades en un servidor.
- Ingeniería Inversa (*Reversing*): consisten en inferir en el funcionamiento de un software. Lo más común, son binarios de Windows y Linux.
- Programación (*PPC*): También conocidos como *Professional Programming & Coding*, y son desafíos en los que se requiere desarrollar un programa o script que realice una determinada tarea.
- Web: muy similar a los explotación, salvo porque ahora se lleva a cabo el descubrimiento de vulnerabilidades en una aplicación Web.
- Reconocimiento (*Recon*): consisten en la búsqueda de la bandera en distintos sitios de Internet. Para resolverlos se ofrecen pistas, tal como el nombre de una persona o se proporciona algún archivo que conduzca a estas.

- Trivial (*Trivia*): diferentes preguntas relacionadas con la seguridad informática.
- Misceláneo (*Misc*): retos aleatorios que pueden pertenecer a distintas categorías sin especificar.

El trabajo de fin de carrera realizado, se trataba de un proyecto coordinado con otro compañero, por lo que ambos llevaríamos a cabo el desarrollo de diferentes retos de algunas de las categorías que acabamos de comentar. Una vez identificados los distintos tipos existentes, hacia los que poder enfocar nuestro cometido, acordamos junto con el profesor llevar a cabo una división. De esta forma, cada uno de nosotros se centraría en la preparación de 8 ó 9 retos, desarrollados sobre 3 de esas categorías.

Finalmente, en coordinación con mi compañero, las categorías plasmadas en el proyecto son Esteganografía, Forense y Reconocimiento. Además, también se incluyó Programación, dado que de Reconocimiento no existe mucha información. Por lo que, las categorías que se irán explicando a continuación más detalladamente junto con sus retos pertinentes, son esteganografía, forense, reconocimiento y programación.

5. IMPLEMENTACIÓN Y DESARROLLO

Durante el siguiente apartado, realizaremos un recorrido que nos permitirá conocer de primera mano el desarrollo de los diferentes retos, profundizando en las categorías en las que han sido enmarcados, en qué consiste cada reto, cómo han sido construidos, que herramientas y fuentes de datos se han utilizado para elaborarlos, qué elementos externos han permitido complementarlos…

5.1. ESTEGANOGRAFÍA

La Esteganografía es el área que trata el estudio y aplicación de técnicas que permiten ocultar mensajes u objetos, dentro de otros mensajes u objetos (portadores), de forma que su existencia no pueda ser detectada. Esteganografía proviene del griego “steganos” que significa cubierto u oculto, y “graphos” que significa escritura.

La esteganografía se confunde erróneamente con la criptografía, lo cual es lógico ya que ambas son técnicas que protegen la confidencialidad de la información,

pero ambas se diferencian entre sí por la forma en que logran su objetivo. La esteganografía tiene como principal diferencia la intención de que ni siquiera se perciba la información oculta, para que de este modo permanezca segura. Aunque puede complementarse con la criptografía, dando un nivel de seguridad extra, es decir, es muy común que el mensaje a ocultar sea previamente cifrado, de tal modo que a un eventual intruso no sólo le costará advertir la presencia del mismo, sino que, si llegara a obtener la mensajería oculta, la encontraría cifrada.

Aunque habitualmente solemos considerar la esteganografía como una técnica para ocultar información dentro de imágenes, lo cierto es que existen muchos otros tipos:

- Esteganografía pura: no existe clave, y por tanto, la persona que quiere capturar la información no es capaz de reconocerla mediante una técnica estego de un mensaje normal.
- Esteganografía de clave secreta: en este caso, se depende de una clave que deben conocer tanto el emisor como el receptor.

Además, dentro de estas dos categorías, y atendiendo ya a los canales utilizados, podemos hablar de:

- Esteganografía en texto: ofuscar la información en un texto, de manera que algunas (o todas) las palabras del mismo, sirvan para descifrar el mensaje.
- Esteganografía en sistemas operativos y ficheros: es similar al anterior, salvo porque en este caso nos apoyamos en las restricciones propias del canal utilizado.
- Esteganografía en formato de ficheros: ocultamos información en las limitaciones del fichero, o de los elementos de control de los sistemas encargados de leer el fichero.
- Esteganografía hardware: nuevamente, aprovechamos las limitaciones (o fallos de seguridad) de un elemento físico para ofuscar información dentro de él.
- Esteganografía en tecnologías web: ocultar información entre lenguajes de maquetado como son el HTML.

- Esteganografía en protocolos de comunicación: ocultar información aprovechando el formato de los protocolos.
- Esteganografía en contenido multimedia: tanto en imágenes como en sonidos o vídeos. Esta técnica es la que principalmente hemos aplicado en los retos que se explicarán a continuación.

5.1.1. Una imagen vale más que mil palabras

En el siguiente reto, se entrega un documento con una imagen en su interior. Sin embargo, esta imagen no proporciona ningún tipo de información, solo está ahí con la intención de despistar. El verdadero meollo del reto lo encontraremos dentro del docx, dado que estos presentan una estructura xml y al igual que cualquier archivo zip o rar, se pueden comprimir y descomprimir.

En el interior del documento, existirán dos imágenes. Una de ellas, se trata de una pista para poder avanzar en el reto. La otra imagen, tiene oculta una dirección url (entre sus colores), que nos conducirá hacia la última pista. Finalmente, la dirección nos llevará a una imagen (de Madrid). Las imágenes y videos, pueden almacenar coordenadas en su interior, y está también albergará una ubicación, que se corresponderá con el Museo del Prado de Madrid (su bandera).

La ficha técnica correspondiente a este reto es:

Nombre: Una imagen vale más que mil palabras.

Historia:

Desde pequeños siempre que nos han contado algo, no nos hemos conformado con la imagen que creaba nuestra mente sobre aquello que nos contaban, hemos querido más. Es una realidad, el ser humano es curioso por naturaleza y siempre quiere ver con sus propios ojos, para poder asimilar los conceptos que se le presentan. Esto es así, porque una imagen vale más que mil palabras.

Pero ¿y si ocurriera al contrario? ¿y si tuvieses que descubrir la información que se oculta tras una imagen?

Categoría: Esteganografía.

Nivel: Fácil.

Descripción:

Se proporcionará un documento Word, que contendrá una imagen. Todo lo que necesitamos lo tenemos en dicho documento.

Bandera: Prado.

Pistas:

- Sería bonito que existiese algo que nos hiciese a todos felices.
- Una imagen guarda recuerdos y los mantiene vivos en el paso del tiempo.

En lo que sigue, mostraremos cómo se ha construido el reto. En primer lugar, tuvimos que preparar el documento Word, que contendría una imagen (la cual realmente estaba ahí para despistar). Estos primeros pasos son sencillos, creamos un documento Word y le insertamos una imagen cualquiera.

El siguiente paso fue modificar los metadatos de este documento, para que no mostrase el autor del mismo. Esto no es algo necesario como tal, pero de esta manera evitábamos que se pudiese difundir información personal. Para hacer esto, utilizamos el programa ExifTool. Este programa permite consultar los metadatos.

Conjuntamente con el anterior, necesitábamos el programa MetadataTouch, que permite modificarlos como tal.

Una vez, que ya tuvimos todo listo, procedimos a modificar los metadatos de nuestro documento. Estos son los datos que inicialmente mostraba (arrastrando el Word sobre el ejecutable de exiftool):

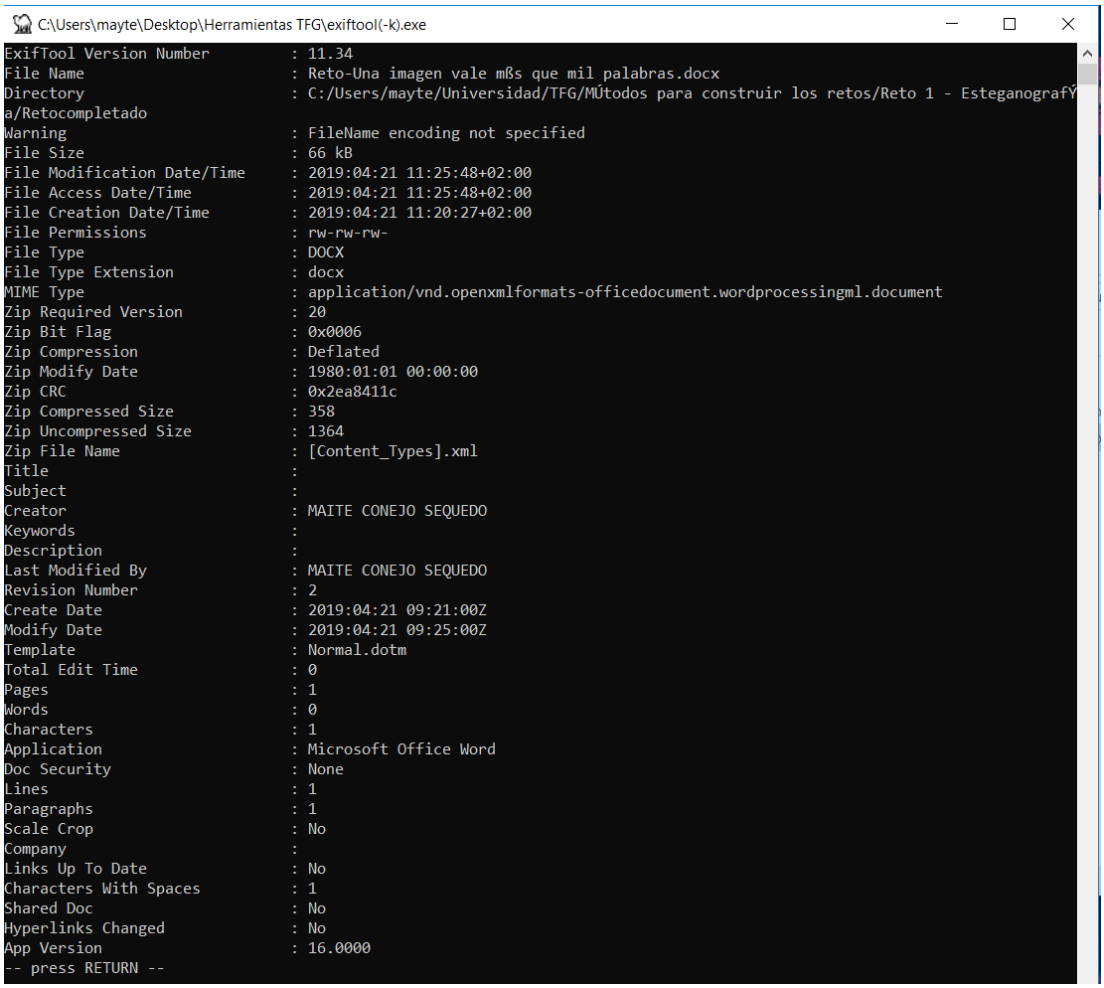


Figura 1. Datos mostrados por exiftool antes de modificarlos.

Aparecía el nombre del creador y del último que había modificado el documento. MetadataTouch nos permitió modificar estos datos de la siguiente forma:

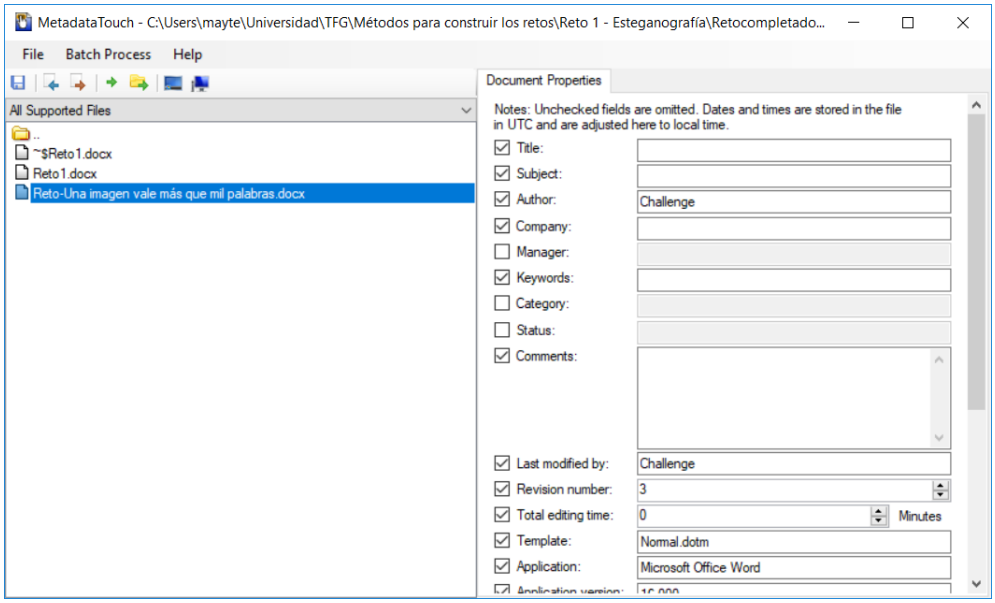


Figura 2. Modificación con MetadataTouch.

Obteniendo finalmente esto:

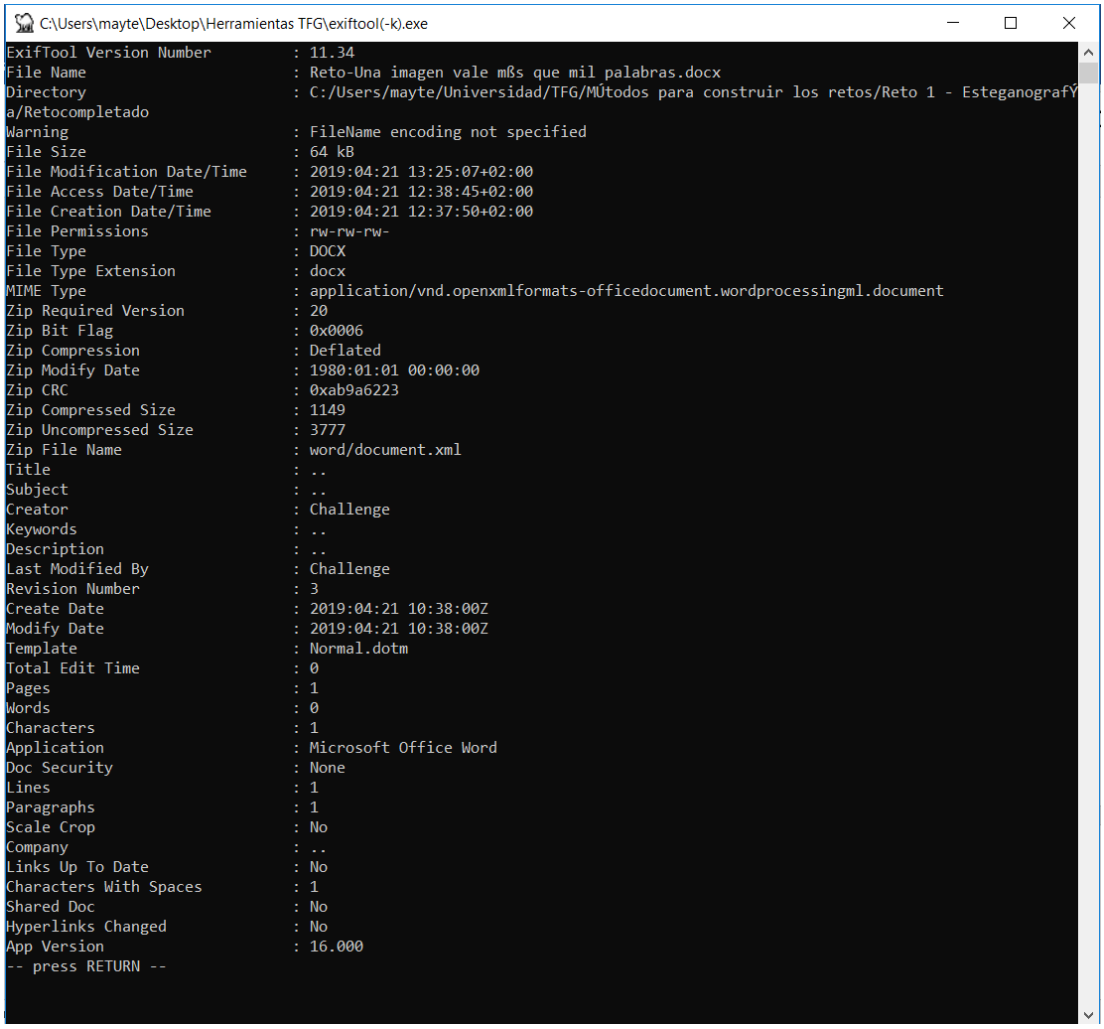


Figura 3. Datos mostrados por exiftool después de modificarlos.

Una vez hecho esto, continuamos con la elaboración del reto.

Nuevamente, y volvemos a repetir, es importante explicar que los documentos Word y Excel realmente se tratan de estructuras xml, y funcionan como un archivo zip o rar. Por lo que podemos almacenar información en su interior y posteriormente recuperarla.

Cuando insertamos datos en un documento, automáticamente se crean las carpetas necesarias para almacenarlos. De igual forma, al contener cualquier imagen habrá una carpeta media en el mismo. Fue en dicha carpeta donde almacenamos la imagen que sí contenían la siguiente acción (y que fue previamente preparada).

Para hacer esto, modificamos la extensión de nuestro archivo a .zip. Una vez hecho, se pudieron extraer las carpetas correspondientes.

Buscamos la carpeta media y ahí pusimos la imagen modificada con edición y la que serviría de pista para la misma (todas las imágenes de esta carpeta debían tener la misma extensión, de lo contrario, el Word no abriría).

Después, volvimos a meter las carpetas en su interior y cambiamos su extensión a .docx.

❖ Imagen con la url, oculta dentro del Word:

Necesitábamos preparar la imagen con la url, la cual solo se podría visualizar modificando las curvas de color de la misma.

Con ayuda de paint net, capturamos el color de la imagen (no exactamente el mismo, sino que intentamos desplazarlo un poco en la paleta para que no fuese muy parecido, ya que si no, no se vería) y escribimos un texto (la url, previamente preparada, y que se trataba de la dirección a la que habíamos subido la imagen que contenía las coordenadas).

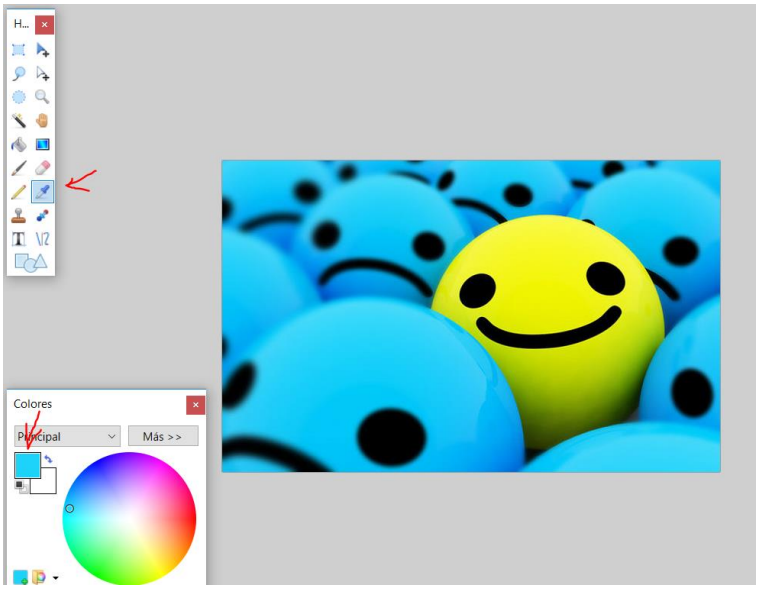


Figura 4. Incluir texto oculto entre los colores de una imagen.

❖ Imagen con coordenadas:

Queríamos añadir unas coordenadas a nuestra imagen, aquellas que nos proporcionarían el lugar cuyo nombre sería la bandera de nuestro reto. Para obtener las coordenadas a incluir, buscamos en google maps el elemento en cuestión, nos situamos sobre él y pulsamos clic derecho, le dimos a la opción ¿Qué hay aquí?, y esto nos devolvió sus coordenadas.

Como el reto estaba tratando sobre todo lo que podía ocultar una imagen, elegimos un lugar repleto de imágenes, un museo. Más concretamente, el museo del Prado de Madrid. Esa sería nuestra bandera: **PRADO**.

Pues bien, buscamos el museo en Google Maps:

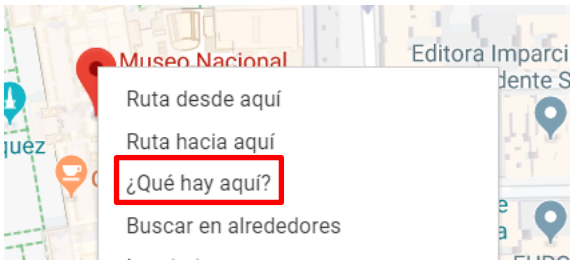


Figura 5. Consultar coordenadas en Google Maps.

Eso nos devolvió sus coordenadas. Guardamos dichas coordenadas: 40.414051, -3.692116.

Para añadir a nuestra imagen esas coordenadas, utilizamos la web Geolmgr, ahí subimos la imagen y colocamos la latitud y longitud según correspondía.

La imagen elegida en esta ocasión, fue una de Madrid, con la intención de proporcionar una pista a los participantes. Introdujimos las coordenadas obtenidas:

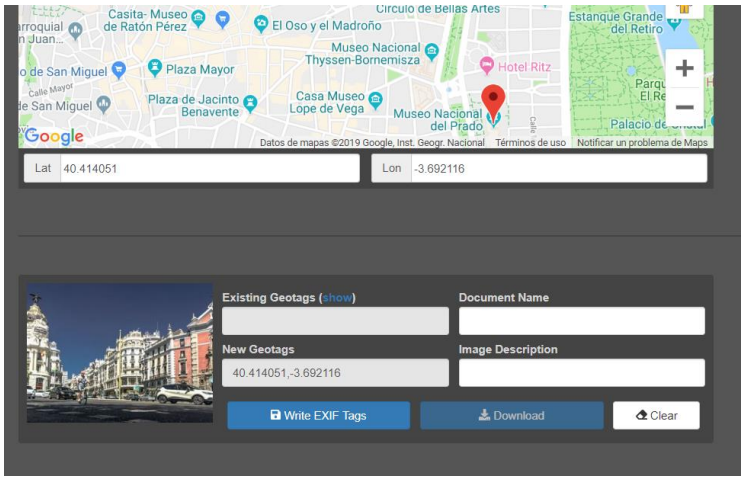


Figura 6. Incluir coordenadas en una imagen.

Por último, se subió la imagen a la página imgbb, y al hacerlo, nos proporcionó la url dónde había sido publicada: https://ibb.co/hcm8SG9. Esta fue la que se camufló entre los colores de la imagen inicial.

5.1.2. El mapa de… ¿la bandera?

Para el siguiente, se proporciona un archivo pcap y dentro de él se debe localizar una imagen y un audio. Al extraer esta imagen del archivo, podemos obtener una dirección. Esta dirección, nos facilita la descarga de un fichero zip, el cual contiene dos imágenes almacenadas en su interior. Una de las imágenes es una pista para saber qué hacer con la otra, y la segunda imagen presenta un nombre extraño y además no se puede abrir. Se tratará de un mensaje cifrado que nos permitirá obtener el fichero final con la bandera, oculto dentro de la misma.

El audio presente en el fichero anterior, nos indicará la forma en la que podemos obtener a partir de la imagen el fichero con la bandera.

La ficha técnica correspondiente a este reto es:

Nombre: El mapa de… ¿la bandera?

Historia:

Felipe estaba en skype cuando empezó a recibir varios mensajes sospechosos, intrigado por lo que ocultaban los mismos, decide investigar por su cuenta. Durante ese proceso descarga varios archivos, tal vez alguno de ellos oculte algo interesante. ¿Podrías ayudarle a descubrir qué es?

Categoría: Esteganografía.

Nivel: Difícil.

Descripción:

Proporcionamos un archivo pcap de wireshark, que ocultará el intercambio de mensajes y de varios archivos. Si recuperamos el adecuado, nos debería reconducir a la siguiente pista.

Bandera: pasión.

Pistas:

- Todas las pistas son proporcionadas, solo hay que saber buscarlas.

Necesitábamos proporcionar un archivo wireshark con el intercambio de varios mensajes, y entre ellos, el de una imagen y un audio. La intención era simular que el archivo había sido recogido de un lugar con bastante tráfico. Aunque, se descubrió

que existía la posibilidad de descargar estos ficheros ya preparados desde página como Netresec, y modificarlos para que contuviesen lo que quisiésemos. En este caso, nos decantamos por esta opción.

Todos los ficheros en esa página son públicos y se pueden descargar. Tenemos varias categorías: Ciberdefensa, tráfico malware… Se descargó unos de los paquetes sin categorizar, una captura simple, de un tráfico por skype. Pero este fichero, necesitaba ser modificado, ¿y cómo haríamos esto?

Antes de comenzar, es necesario hacer un inciso y comentar que, para la elaboración de este reto, se utilizó la versión 1.12.8 de wireshark, ya que es la única que incluye una opción para modificar paquetes. Toda la información necesaria para conocer cómo utilizar esta funcionalidad, está disponible en Network Computing. Esta opción nos proporciona la posibilidad de eliminar paquetes, transformarlos de un formato a otro, cambiar direcciones IP...

También, existen otras funcionalidades que proporciona wireshark, entre ellas, destacamos editcap y mergecap, a continuación, explicaremos cómo funcionan estas de forma más detallada.

La captura elegida contaba con más de 2000 paquetes. Como el archivo utilizado era muy grande, usamos la herramienta editcap incluida en wireshark, siguiendo una de sus opciones. Nos quedamos con una captura de solo 57 paquetes (fuimos eliminando los paquetes poco a poco), el comando utilizado para ello fue el siguiente (aquí se muestra una captura con los 100 primeros):

```
C:\Program Files\Wireshark>editcap -r C:\Users\mayte\Desktop\SkypeIRC.pcap C:\Users\mayte\Desktop\Skype.pcap 1-100

C:\Program Files\Wireshark>
```

Figura 7. Seleccionar paquetes con Wireshark (funcionalidad editcap)

Una vez que tuvimos la captura deseada, era cuando tocaba editar el fichero. Para hacerlo, utilizamos el propio wireshark ya que este nos ofrecía dicha posibilidad. Esto se conseguía haciendo clic derecho sobre el paquete que queríamos editar y opción editpacket. La intención de esta modificación, era la de incluir entre los mensajes del chat, los que se veían, varios mensajes que nos encaminasen a la pista.

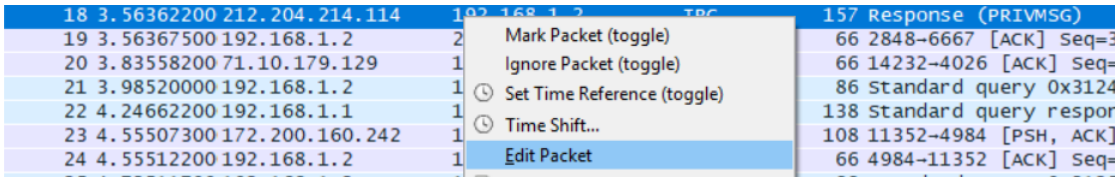


Figura 8. Editar paquetes con Wireshark.

A continuación, mostramos un ejemplo con el cambio correspondiente al mensaje "El nombre es una pista". En la parte de Internet Relay, pulsamos sobre el mensaje:

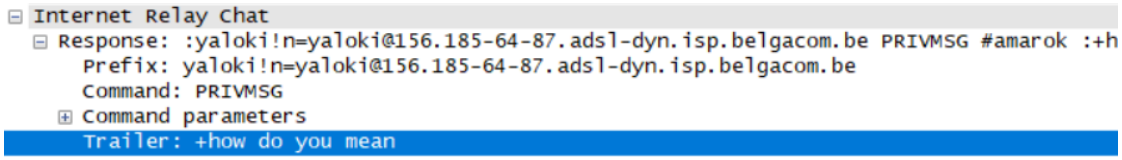


Figura 9. Ejemplo de modificación de un paquete en Wireshark 1.

Y en la nueva ventana, escribimos el mensaje:

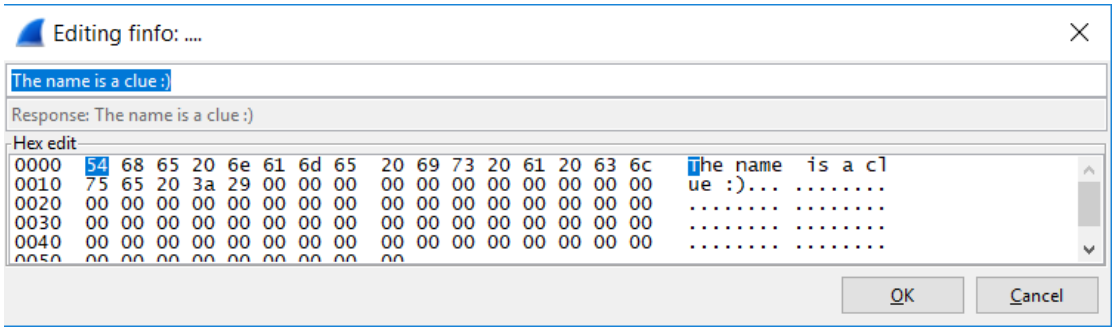


Figura 10. Ejemplo de modificación de un paquete en Wireshark 2.

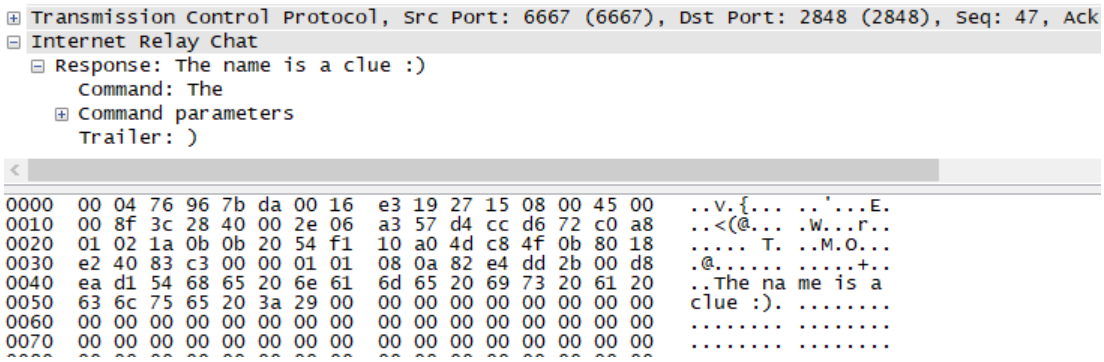


Figura 11. Ejemplo de modificación de un paquete en Wireshark 3.

Y esto mismo, fue lo que hicimos para modificar el resto de mensajes presentes en la captura.

Ahora vamos a mostrar cómo se incluyó la recuperación de la imagen y el audio con una captura hecha con nuestro wireshark, y su posterior unión con este archivo editado.

Lo primero que hicimos fue ver la dirección IP que se utilizó en el archivo y la fecha en la que tuvo lugar, ya que estos elementos debían ser los mismos.

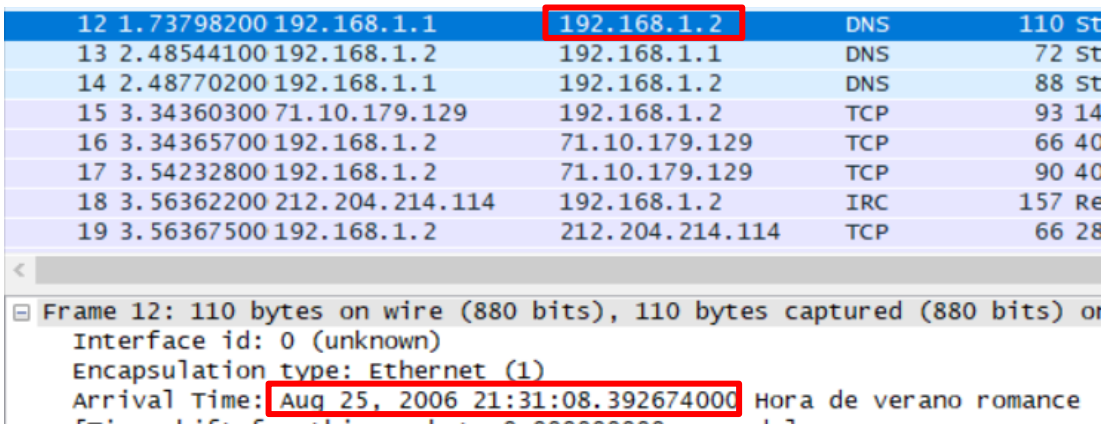


Figura 12. Consulta de la IP y fecha de un paquete en Wireshark.

La dirección IP era la 192.168.1.2 y la fecha databa en el 25 de agosto de 2006, a las 9 y media de la noche. Siendo la del último paquete guardado, esta misma. Por lo que establecimos nuestro ordenador en función de estos dos valores.

Para poder conseguir la hora y fecha que queríamos:

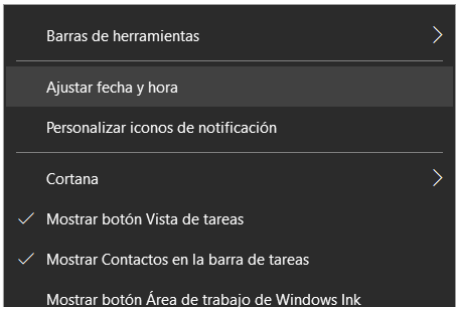


Figura 13. Modificar la fecha del equipo 1.

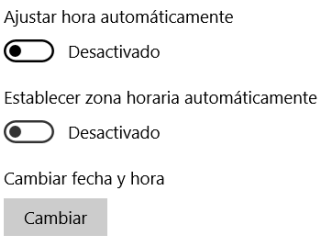


Figura 14. Modificar la fecha del equipo 2.

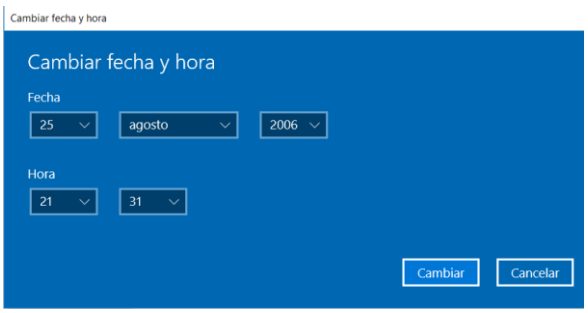


Figura 15. Modificar la fecha del equipo 3.

Para modificar la IP, nos situamos sobre el icono de la wifi, hicimos clic derecho y entramos en abrir configuración de red e internet.

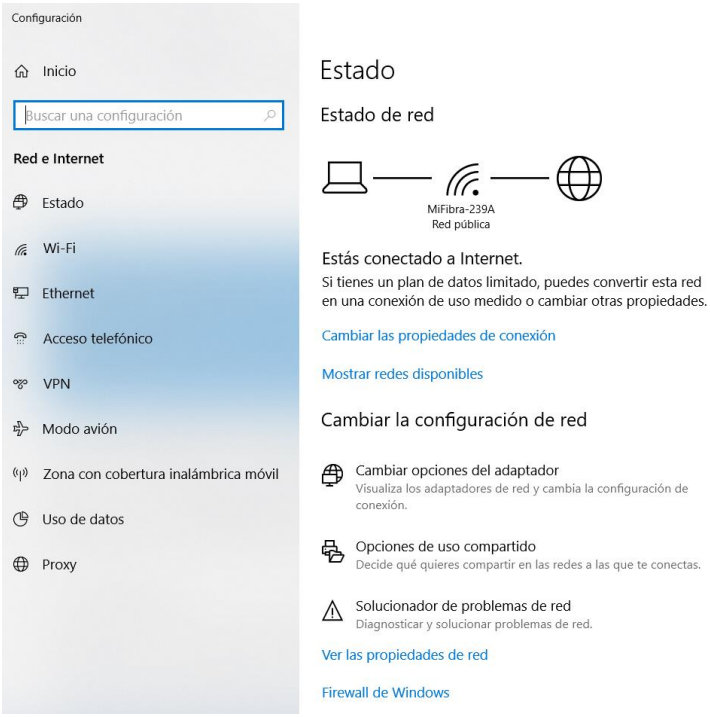


Figura 16. Modificar la IP del equipo 1.

Una vez dentro, pulsamos en cambiar opciones del adaptador. Esto nos abrió una nueva ventana, y sobre esta hicimos clic derecho en wifi y propiedades.

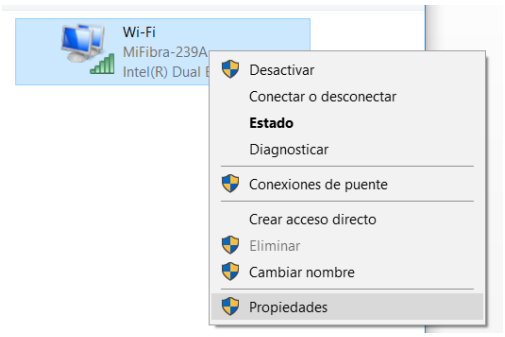


Figura 17. Modificar la IP del equipo 2.

Nos situamos sobre protocolo de internet versión 4 y propiedades.

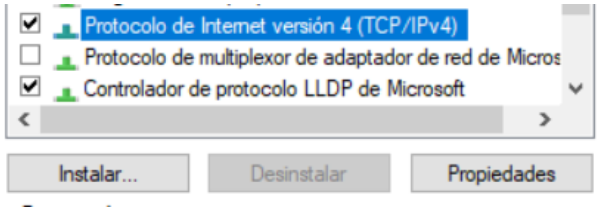


Figura 18. Modificar la IP del equipo 3.

Se nos abrió una nueva ventana, es en esta donde configuramos la dirección IP (solo cambiamos la IP, manteniendo la puerta de enlace y el servidor DNS porque si no, podríamos tener problemas para conectarnos a internet).

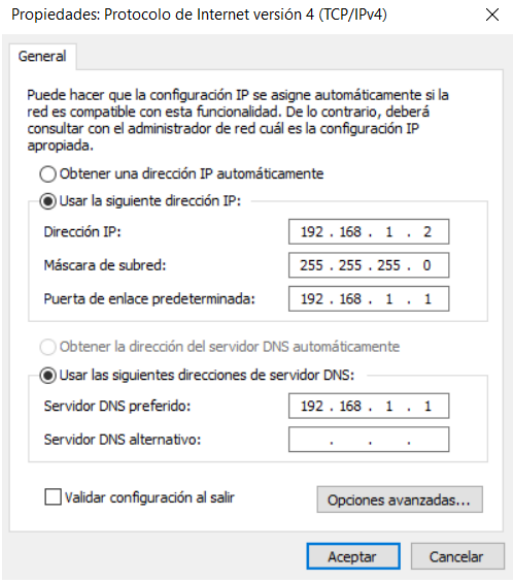


Figura 19. Modificar la IP del equipo 4.

Para poder conseguir obtener la imagen y el audio desde internet, entramos en la página Ge.tt (y aquí subimos ambos). Pusimos al wireshark a capturar solo los paquetes http, después, descargamos el elemento correspondiente.

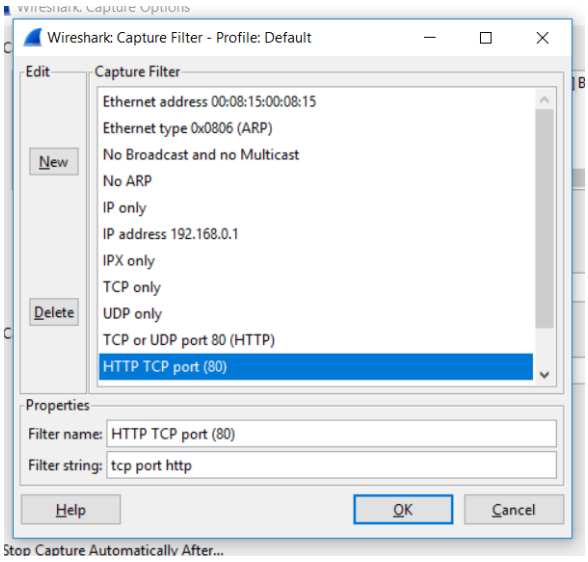


Figura 20. Filtro de capturas en Wireshark.

Este proceso lo realizamos de forma independiente, se crearon 2 capturas, una con la imagen y otra con el audio.

Después, obtuvimos los paquetes que nos interesaban, como habíamos hecho con el primer pcap para quedarnos con los 57. Tanto para uno, como para otro, debíamos tener especial cuidado de no quitar paquetes de más, ya que, aunque no apareciese explícitamente, la imagen o el audio se seguía intercambiando por ellos. Fuimos quitando poco a poco para evitar que esto pasará. Mostramos una captura de cómo se realizó este paso. En este caso con la captura de la imagen, pero con la del audio se realizó exactamente igual:

```
C:\Program Files\Wireshark>editcap C:\Users\mayte\Desktop\imagen.pcap C:\Users\mayte\Desktop\imagen1.pcap 1-32

C:\Program Files\Wireshark>editcap C:\Users\mayte\Desktop\imagen1.pcap C:\Users\mayte\Desktop\imagen2.pcap 1-14

C:\Program Files\Wireshark>editcap C:\Users\mayte\Desktop\imagen2.pcap C:\Users\mayte\Desktop\imagen3.pcap 1-17

C:\Program Files\Wireshark>editcap C:\Users\mayte\Desktop\imagen3.pcap C:\Users\mayte\Desktop\imagen4.pcap 109-131
```

Figura 21. Eliminar paquetes con Wireshark (funcionalidad editcap)

Son las versiones que se crearon, controlando en todo momento que la trama que contenía la imagen o el audio, permaneciese intacta, una trama como la que se muestra a continuación:

```
165 6.63607100 212.97.130.93    192.168.1.2     HTTP    1044 HTTP/1.1 200 OK  (JPEG JFIF image)
```

Figura 22. Formato de un paquete con una imagen en Wireshark.

Por último, para unir nuestros ficheros, utilizamos la opción mergecap, también mencionada antes. Primero, creamos una carpeta con todos los archivos, y utilizamos el siguiente comando:

```
C:\Program Files\Wireshark>mergecap C:\Users\mayte\Desktop\union\*.pcap -w C:\Users\mayte\Desktop\union\ficherocompleto.pcap

C:\Program Files\Wireshark>
```

Figura 23. Unir fichero pcap con Wireshark (funcionalidad mergecap)

- Audio capturado con la pista:

Queríamos incluir una pista para indicar que el extraño nombre de la imagen era una pista para poder recuperar el fichero con la bandera. Y para incluir un mensaje en el audio utilizamos el programa DeepSound.

Como audio elegimos la canción del abecedario, ya que la intención era explicar que el extraño nombre de la imagen se debía a utilizar el abecedario como una técnica de cifrado, dividiéndolo a la mitad, como si estuviese frente a un espejo. Con la intención de que no se identificase demasiado rápido que la canción era la del abecedario, utilizamos una en alemán.

En primer lugar, abrimos nuestro programa y cargamos el audio (con la opción de Open carrier files):

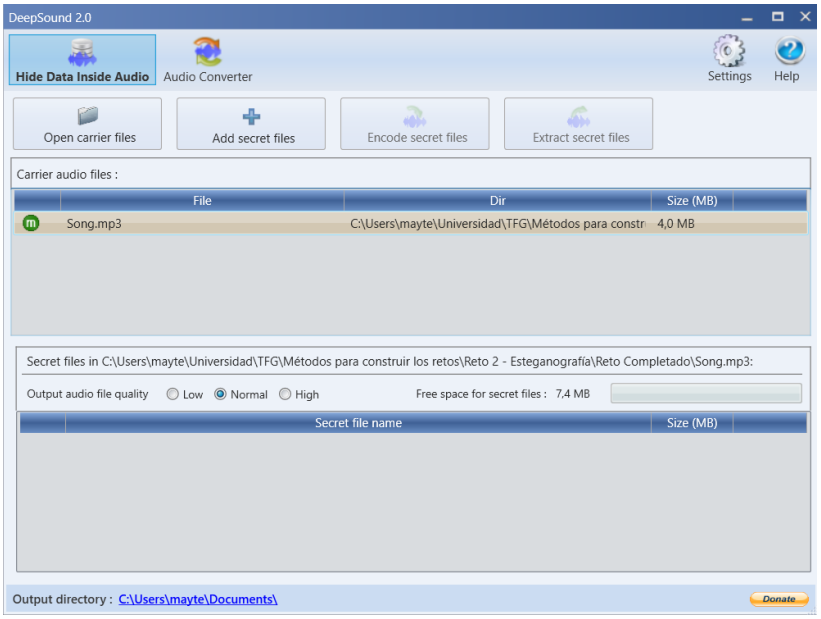


Figura 24. Ocultar información en archivo de audio con DeepSound.

Hicimos clic en "Add secret files" para añadir los archivos secretos. En este caso, el archivo añadido fue un txt con el siguiente acertijo: “Por mi núcleo me has de doblar, para encontrar una nueva forma de encriptar”.

Pulsamos sobre la opción encode secret files y el archivo fue codificado. Nuestra canción fue creada y esta vez, con extensión .wav.

- Imagen capturada:

La imagen que encontraríamos entre el tráfico del archivo pcap, incluiría la url que nos conduciría a la siguiente parte.

Para incluir la url como un mensaje oculto, utilizamos la consola del sistema, de una forma muy sencilla, esta se colocaría al final de su código hexadecimal. Primero, creamos un fichero txt con el enlace que queríamos que guardase, y después cambiamos su extensión a .jpg. Lo siguiente fue utilizar el comando: copy /b imagen.jpg + fichero.jpg. De esta forma se copiaría el contenido del fichero a la imagen.

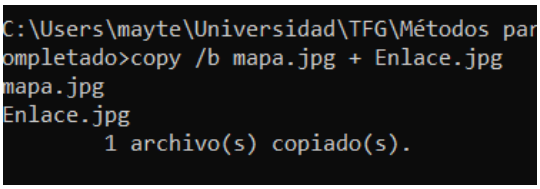


Figura 25. Ocultar mensaje en el código hexadecimal de una imagen.

- Imagen que contiene el texto cifrado y el fichero final:

Teníamos que preparar la imagen que contendría el texto y el fichero. Para preparar dicha imagen utilizamos un editor cualquiera, como paint, ya que bastaba con una imagen sencilla.

Abrimos una ventana de PAINT, e insertamos el texto cifrado. Este texto sería en realidad la contraseña que nos permitiría extraer el fichero de su interior. El tipo de cifrado elegido fue una encriptación en base 64 mediante la web superpatanegra.

El resultado de cifrar la cadena "CyLdxuyfPV0o5pH" fue "Q31MZHh1EWZQVjBvNXBI".

Una vez que teníamos lista la imagen, el siguiente paso fue ocultar el fichero dentro de la misma. Para ocultar el fichero solución (un fichero txt con la bandera escrita en su interior), utilizamos la herramienta openstego. La cual configuramos de esta forma:

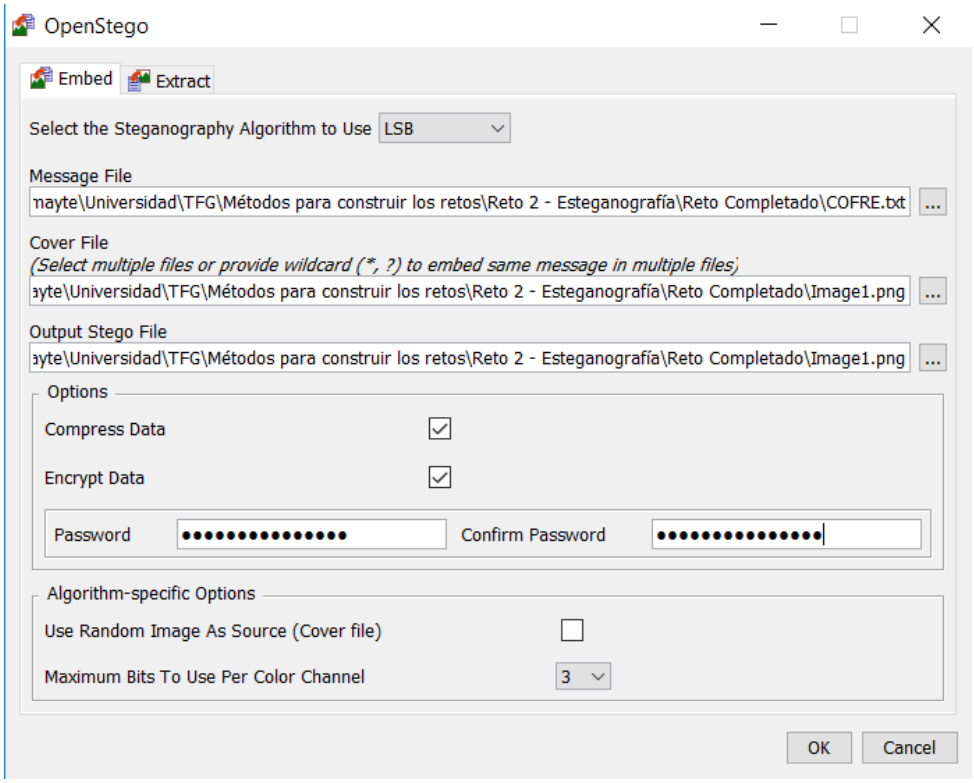


Figura 26. Ocultar información en una imagen con Openstego.

El algoritmo utilizado fue LSB, indicamos el fichero en la opción de Message File, y, por último, la imagen tanto en el segundo como en el último campo. Además, marcamos Encrypt data para poder establecer la clave.

Esta misma imagen no tendría la cabecera PNG, que sería la razón por la que no se podría abrir. Simplemente abrimos nuestra imagen con un lector hexadecimal, y le borramos dicha cabecera. De la siguiente manera:

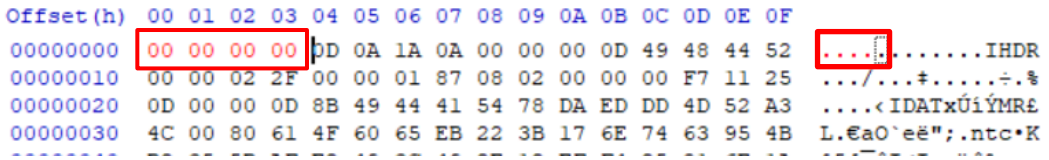


Figura 27. Modificar cabecera hexadecimal de una imagen.

Además, el propio nombre de la imagen también conformaba una pista para los participantes. Ya que sería con el que indicaríamos que lo que muestra la imagen es la contraseña para obtener la bandera final. Sin embargo, el nombre, no sería legible fácilmente sino que también estaría encriptado, algo que indicábamos a través del mensaje oculto en el audio.

Utilizamos el método reflejado, que consistía en dividir a la mitad el abecedario, de la siguiente forma:

a b c d e f g h i j k l m

n o p q r s t u v w x y z

Y crear las palabras cifradas, haciendo que cada letra se cambie por otra en base a su reflejo. En este caso, elegimos como nombre la palabra password. Por lo que el nombre final fue cnffjbeq.

Por último, pero no menos importante, subimos a internet un zip con dicha imagen y otra que solo estaba como pista (o más bien para despistar). La dirección a la que subimos dicho archivo, fue la que colocamos en la imagen recuperada del archivo wireshark, y que ya hemos explicado antes.

Utilizamos drive, para subir el archivo. Y el enlace obtenido fue https://drive.google.com/open?id=1J1RD7JcFPL_2VV9mvA_LFC5VSDAtlpkS. No obstante, este enlace, nos redireccionaba a la página de google drive, y en realidad, queríamos que fuese un enlace de descarga directa, que nada más ponerlo en el navegador nos descargase el .zip.

Por lo que hicimos lo siguiente. El primer paso, fue utilizar el comienzo de la url: https://docs.google.com/uc?export=download&id=, y a este le añadimos el resto del enlace de drive, desde el igual: 1J1RD7JcFPL_2VV9mvA_LFC5VSDAtlpkS.

Finalmente, el enlace fue: https://docs.google.com/uc?export=download&id=1J1RD7JcFPL_2VV9mvA_LFC5VSDAtlpkS. O su versión abreviada, http://cort.as/-K6Z1. La que realmente se utilizó para incluir en la primera imagen.

5.1.3. El mundo multimedia

Se proporciona una imagen, que oculta un enlace de descarga. Dicho enlace al ponerlo en un navegador nos descargará un zip, con un video y una imagen. Tanto el video como el audio de este, ocultan información. En el primero, se ocultará un mensaje que nos proporcionará una contraseña, así como una pista que nos dirá que la imagen que se entregó antes junto al video, tiene un audio secreto. El segundo, el audio solo será un aviso, para que miremos con especial atención al primero.

La clave del vídeo nos ayudará a conseguir el audio. Por otro lado, el audio tendrá una imagen. Para poder obtener dicha imagen necesitamos una clave que se podrá escuchar en el audio. La imagen extraída se tratará de un QR, que finalmente nos proporcionará la bandera.

La ficha de descripción para el mismo es esta:

Nombre: El mundo multimedia.

Historia:

Érase una vez, un mundo multimedia, lleno de imágenes, audios y videos. Cada uno de estos, se agrupaban independientes de los otros, pues todos se consideraban diferentes entre sí. Mientras uno mostraba escenarios estáticos, otro lo hacía de forma dinámica, y el tercero no podía mostrar un paisaje, pero te podía hacer visualizarlo mediante la sonoridad del mismo.

Por fin, un día todos los archivos se unieron, pues había una cosa que todos compartían, los tres podían esconder secretos…

Categoría: Esteganografía.

Nivel: Difícil.

Descripción:

Se proporciona una imagen, la cual nos indicará la siguiente acción a realizar. Es importante ser meticulosos, pues todo lo que necesitamos, lo obtendremos a través de ella.

Bandera: ooSZbSvcjkFCiPZ.

Pistas:

- Hay que hacer una radiografía de todo lo que se pueda, fijémonos bien en sus huesos.
- El maquillaje no lo oculta todo, siempre queda alguna impureza.
- Debemos vaciar los bolsillos.

Se proporcionó una imagen inicial que tendría un enlace de descarga de un archivo zip, con un video y una imagen. Esta imagen inicial la formamos con el programa imagehide.

Primero, preparamos el archivo zip, con el video y con la imagen (estos fueron construidos previamente y serán explicado a continuación). Una vez hecho esto, lo subimos a drive y obtuvimos el enlace para compartir (este fue preparado del mismo modo que en el reto anterior).

Lo siguiente fue insertar dicho enlace dentro de la imagen. Para hacer esto, simplemente abrimos el programa e importamos la imagen en la que queríamos almacenar el enlace. Posteriormente, indicamos el enlace en el cuadro de texto, como se muestra en la siguiente captura:

Figura 28. Ocultar mensaje en una imagen con Imagehide.

Pulsamos Write Data, y ya podíamos guardar la imagen. La mejor opción para guardarla era con extensión .png.

- El video con mensajes ocultos:

Dentro del video que proporcionamos, ocultamos 2 mensajes, uno dentro del propio video, el cual hicimos desde la consola y el otro mediante, edición (este será una pista).

Además, el audio del vídeo ocultaría un mensaje en su espectrograma. Lo primero que hicimos fue crear un mensaje (bmp) que ocultar en dicho espectrograma.

A continuación, abrimos el programa de Coagula Light e importamos la imagen, una vez hecho pulsamos sobre los botones de color verde y rojo, para que se transformase en sonido, como se muestra en la siguiente captura:

Figura 29. Convertir texto en sonido con Coagula.

Al guardarlo, el resultado fue un archivo de audio con extensión .wav.

Como comentamos más arriba, el vídeo contendría una pista, la cual estaría oculta entre sus fotogramas. Editamos dicho vídeo y unimos los audios creados, con filmora9. Incluimos la pista "la imagen alberga sonoridad", y lo hicimos así. Primero importamos todos los elementos, el video y los audios, y después, situamos el video en la línea de tiempo, le quitamos el sonido (el cual cambiamos por los creados) e incluimos la pista:

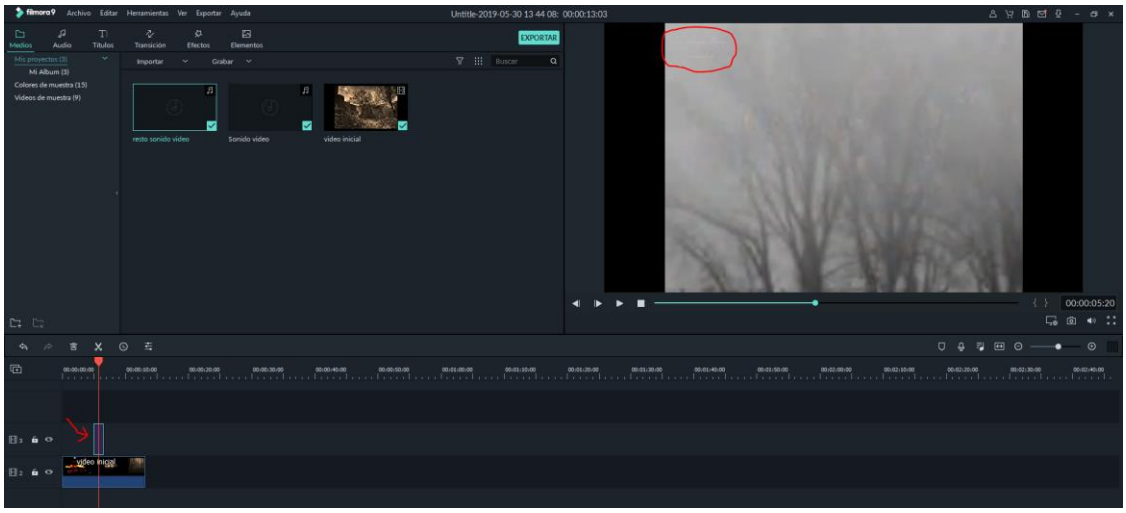


Figura 30. Ocultar texto entre los fotogramas de un video.

Al terminar, pudimos guardar el video, desde la opción de exportar.

El mensaje dentro del video se correspondía con la contraseña para poder obtener el audio oculto en la imagen. Para hacer este paso, utilizamos la consola del sistema, igual que lo hicimos en el reto anterior para ocultar el enlace.

Creamos un fichero de texto con dicha contraseña (audiosecreto), y lo guardamos con la misma extensión que el video que habíamos renderizado, es decir, con extensión .mp4.

```
C:\Users\mayte\Universidad\TFG\Métodos para construir los retos\
fía\Reto Completo>copy /b Video.mp4 + Contraseñaaudio.mp4
Video.mp4
Contraseñaaudio.mp4
        1 archivo(s) copiado(s).

C:\Users\mayte\Universidad\TFG\Métodos para construir los retos\
fía\Reto Completo>
```

Figura 31. Ocultar mensaje en el código hexadecimal de un vídeo.

- Imagen con el audio:

Junto al video guardábamos una imagen de una nota musical, que nos serviría como pista para comprender que el audio junto al video ocultaba algo, así como para resguardar el audio final a analizar. Para almacenar dicho audio dentro de la imagen nuevamente utilizamos la herramienta openstego y dentro de esta, el algoritmo LSB.

- Audio oculto en la imagen, con QR:

En este audio, también se ocultarían 2 mensajes, uno a través de la música y otro en su interior (la bandera).

Para la parte del mensaje subliminal, utilizamos audacity. Para así fusionar una canción real con el mensaje secreto (el cual sería en código morse, y la clave para sacar el QR oculto). Dicha clave, fue la palabra “cima”, y para obtener este código morse hicimos uso de la página Ea8br.

Una vez que lo tuvimos, abrimos audacity, e importamos la canción a la que queríamos añadir dicho código. En lugar de una canción, utilizamos sonidos (para que el archivo final tuviese un peso reducido), extraídos de la web elongsound. Estas eran ambas pistas mezcladas:

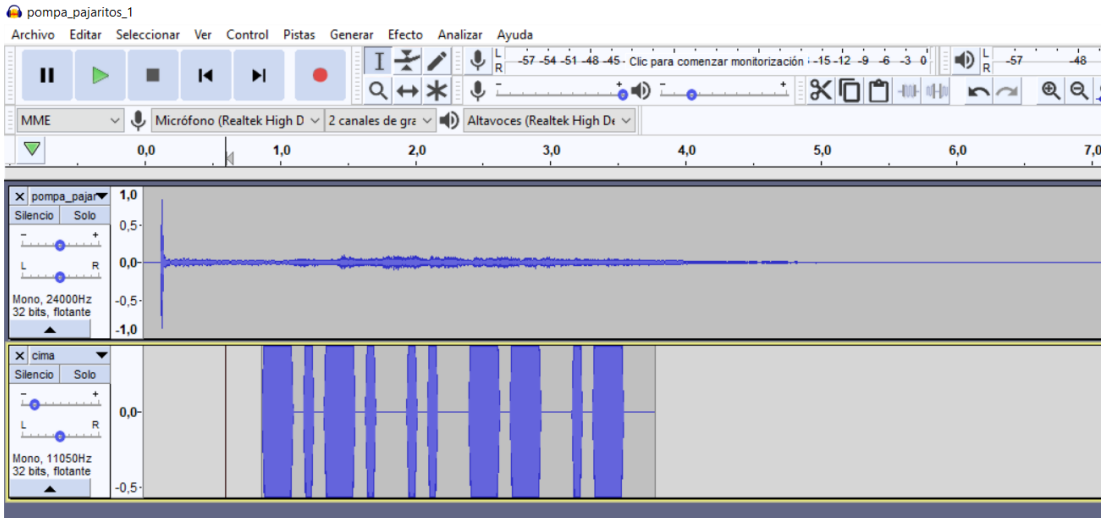


Figura 32. Mezcla y edición de dos pistas de audio con Audacity.

En cuanto a la parte de guardar información dentro del audio, lo hicimos con DeepSound. Abrimos el programa e incluimos la pista de audio que habíamos creado

(con el código morse) y el código a ocultar en ella. Y se estableció como contraseña cima.

- ❖ QR con la bandera:

Con la web QR Code Generator generamos un QR de la bandera, la cual fue una cadena (ooSZbSvcjkFCiPZ) generada de forma aleatoria.

Por último, abrimos con Paint dicho código, ahí lo ajustamos quitándole los trozos blancos, para posteriormente, invertir sus colores.

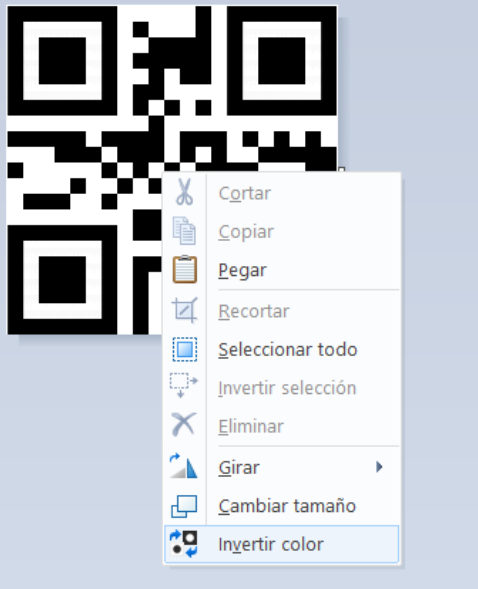


Figura 34. Invertir colores de un código QR con Paint 1.

Figura 33. Invertir colores de un código QR con Paint 2.

5.2. FORENSE

Definimos la informática forense como la ciencia de adquirir, preservar, obtener y presentar datos que han sido procesados electrónicamente y guardados en soportes informáticos.

En una investigación forense es importante tener conocimiento de las normativas existentes para la obtención de las pruebas o resultados, garantizar el cumplimiento de una serie de condiciones sobre la evidencia (admisible, auténtica, completa, confiable y creíble), y cuál es el procedimiento para llevar a cabo una investigación, cuándo realizarla y las cuestiones legales que se deben tener en cuenta. Por tanto, se debe efectuar un trabajo que cumpla los requerimientos exigidos por la ley para no vulnerar los derechos de terceros, y para que las evidencias obtenidas puedan utilizarse como prueba ante los tribunales.

Se pueden distinguir cuatro tipos principales de operaciones en el campo forense:

- Análisis forense de sistemas: Windows, Linux / *NIX, etcétera
- Análisis forense de dispositivos móviles: de un teléfono móvil se puede recuperar todo tipo de información como registros de llamadas y mensajería, material multimedia (con datos que incluso nos proporcionan coordenadas, debido al GPS), los trayectos realizados por la persona poseedora del dispositivo (las conexiones establecidas por el terminal, el GPS, los archivos multimedia,…), reconstrucción de elementos que se mostraron en pantalla en último lugar, información que se ha estado escribiendo,…
- Análisis forense de redes: implica el monitoreo y el análisis del tráfico informático con fines de detección de intrusiones, pruebas legales o recolección de información. Se realizan capturas de todo el tráfico, que plasman información como la conexión con redes o el intercambio de archivos.
- Análisis forense cloud o de internet: es importante comentar que la visión tradicional de un caso forense es muy complicada en un entorno 'Cloud'. En el caso de que los datos se encuentren en la nube y tengamos que analizar un caso de fraude nos vamos a encontrar con la siguiente problemática: no existe control de datos, no hay acceso a la infraestructura física, se deben contemplar aspectos legales de la jurisdicción entre países y multipropiedad, existe una falta de herramientas para ampliar la escala, los sistemas son distribuidos y virtuales, no hay interfaces estándar…

Además, la ciencia forense digital debe enfrentarse a lo que los criminales contraponen para evitar ser descubiertos o detenidos. Cabe destacar acciones como las de sobre escritura de datos (wiping), destrucción del disco completo, ocultación u ofuscación de datos, aprovechamiento de fallos en herramientas forenses… En el caso de los retos desarrollados, se debería haber aplicado algunas de estas técnicas anti forenses para darle realismo a la situación creada en cada uno. Sin embargo, la intención no es la de complicar al máximo la labor de la persona que lo está resolviendo, sino que pueda aprender a través del mismo, y estar preparado para un caso de la misma índole.

Nuevamente a través de la informática forense podemos garantizar factores como la prevención, poder detectar posibles vulnerabilidades de seguridad que pueda haber en los equipos informáticos con el objetivo de corregirlas en caso de ser detectadas, y en caso de que se produzca una brecha, facilitar la recopilación de toda la información y evidencias necesarias para poder averiguar cuál es el origen del ataque, o qué ha podido pasar para que se haya producido este suceso. Este último objetivo mencionado, es principalmente el que ha sido plasmado a través de los retos que iremos exponiendo seguidamente.

5.2.1. Podría ser algo más

El siguiente sería un caso propuesto a un perito con la intención de localizar a un usuario malicioso que ha entrado en un sistema y ha robado información sensible, así como también ha eliminado algunos documentos. Se proporciona una imagen clonada, que analizar con herramientas como Autopsy.

En el disco se almacenarán los logs que pueden proporcionar información acerca de los usuarios que entran en el sistema, así como de los posibles archivos que hayan podido ser descargados. En el archivo log de accesos se almacenarán los accesos al sistema. Entre los elementos eliminados veremos una imagen de un recibo y un documento, que no se podrá abrir, porque su extensión no será correcta, y habrá que averiguar cuál es, así como lo que hay dentro.

En el interior del documento, existirán tres imágenes, una de ellas tan solo será una pista. Y también un archivo zip que en su interior ocultará un documento con la bandera.

La ficha de descripción para este reto fue:

Nombre: Podría ser algo más…

Historia:

Un usuario malicioso ha entrado al sistema de la empresa y ha robado información sensible, así como también ha eliminado algunos documentos. Se te ha propuesto este caso, con la intención de que descubras quién ha podido acceder al sistema y porque ha borrado algunos archivos, tal vez se esconda algo más detrás de esto, quizás podría tener relación con la desaparición de hace unos meses, de hasta unos 200000 euros, ¿Serás capaz de averiguarlo?

Categoría: Forense.

Nivel: Medio.

Descripción:

Se proporcionará una imagen clonada del servidor de la empresa, que analizar con herramientas como Autopsy. Esta nos proporcionará todo lo necesario.

Bandera: ética.

Pistas:

- Los registros son hasta el día del suceso.
- No todo es lo que parece y a veces lo que parece, lo es todo.

En primer lugar, creamos un servidor web para simular lo que sería el servidor de la empresa al que se conectan los empleados, y por el que descargan cosas, y de esta manera poder simular todo el proceso.

Lo que inicialmente se pensó en hacer, fue registrar un dominio, mediante la página noip. No obstante, al final se decidió instalar el servidor en un usb, ya que posteriormente se tenía que hacer una imagen del mismo, para que los usuarios pudiesen analizar los movimientos que habían tenido lugar en él. Es por ello, que se hizo en local, y sin utilizar el dominio.

Primero preparamos la memoria usb formateándola completamente. Para esto, se utilizó la herramienta Eraser. Abrimos el programa, creamos una nueva tarea, indicando la memoria que se quería borrar.

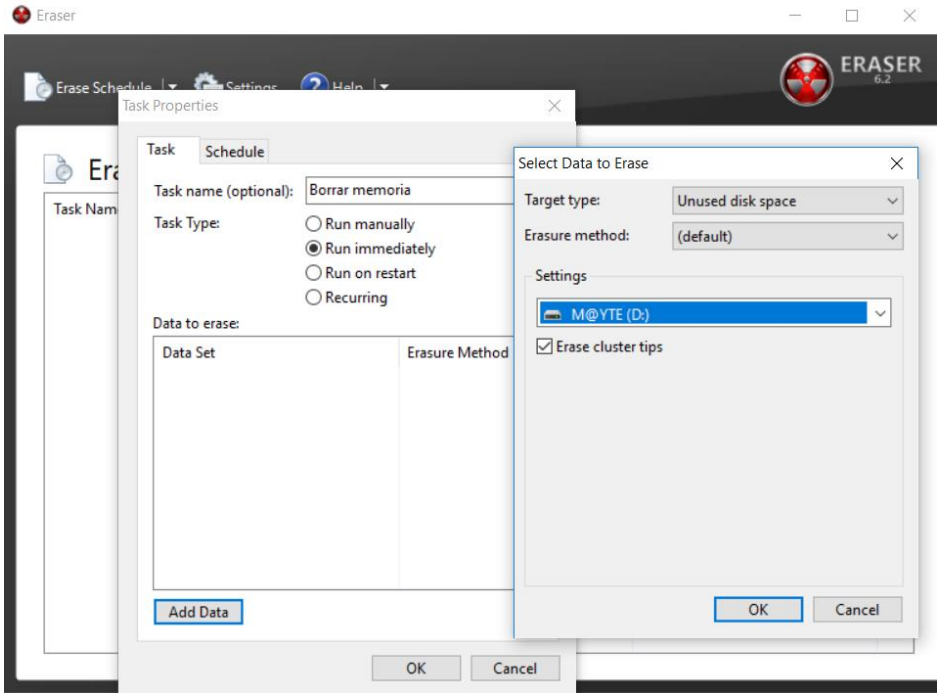


Figura 35. Reseteo completo de una memoria con Eraser.

El siguiente paso, consistía en preparar el servidor en el pen drive. Para la parte del servidor, se descargó XAMPP, que se trata del servidor de apache.

Al ejecutarlo, nos creó una carpeta llamada Xampp. Dentro de esta, encontramos un fichero ejecutable, setup_xampp.bat, sobre él hicimos doble click y esperamos a que se auto configurara, y nos devolviese un mensaje: «Presione cualquier tecla para continuar». Fue algo que se realizó la primera vez, después ya no fue necesario. Posteriormente, se ejecutó:

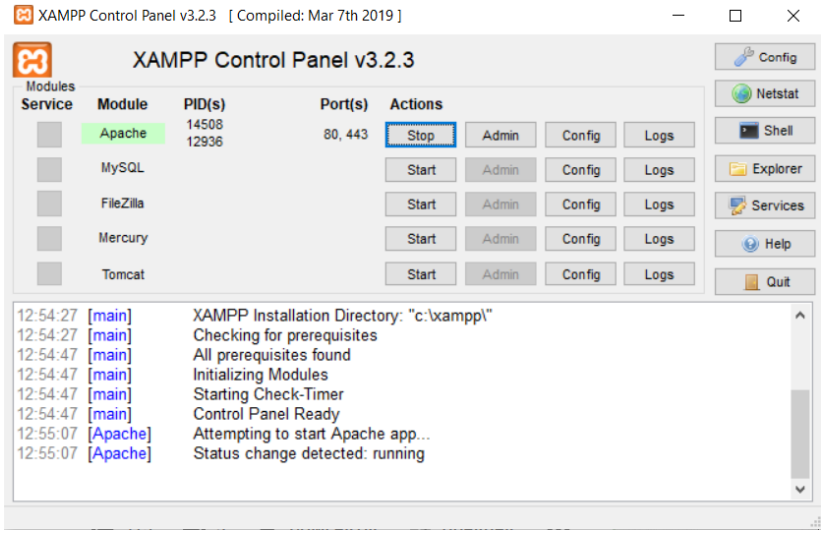


Figura 36. Ejecución del Servidor XAMPP.

De esta forma, ya podíamos visualizarlo en nuestro navegador:

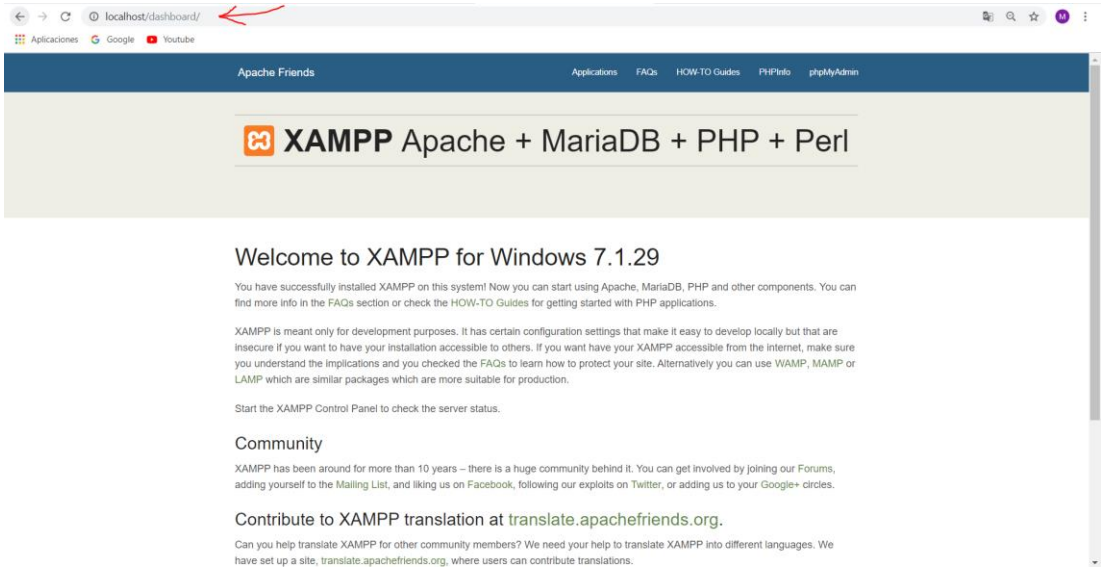


Figura 37. Servidor XAMPP en el navegador.

Hasta aquí, fue todo lo del servidor. Después, necesitábamos preparar nuestra página para que pudiésemos subir documentos a ella y descargarlos.

Creamos nuestra base de datos, accediendo a la pestaña de xampp con phpmyAdmin.

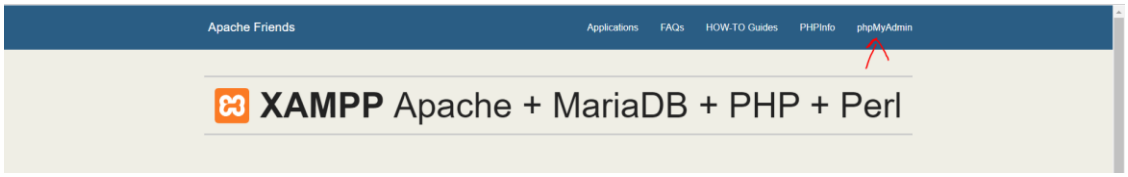


Figura 38. Acceso a phpmyAdmin desde XAMPP.

Esto último en un principio dio error, y para solucionarlo, desde la ventana de la aplicación accedimos al botón de config en el Apache (httpd-xampp.conf) y buscamos las líneas correspondientes al phpmyAdmin. Ahí cambiamos los comandos por estos:

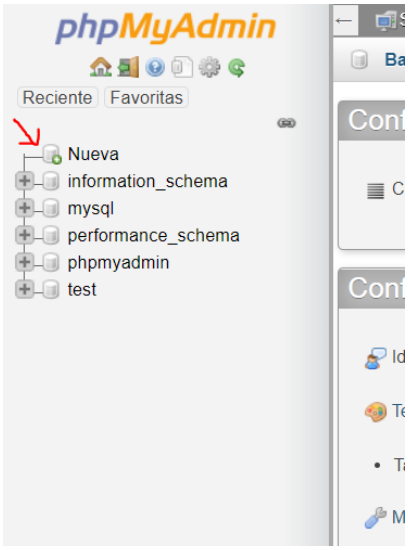
AllowOverride AuthConfig

Order deny,allow

Allow from all

Require all granted

Una vez hecho volvimos a relanzar el servidor y también el de mysql, y ya pudimos acceder a la pestaña anterior. Creamos la base de datos:



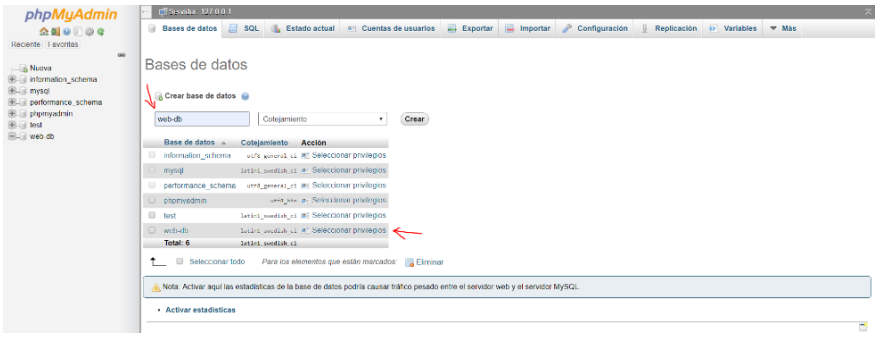


Figura 39. Creación de base de datos en phpmyAdmin 2.

Figura 40. Creación de base de datos en phpmyAdmin 1.

En este caso, el CMS utilizado es Joomla. Al descargar Joomla, obtuvimos un zip con un montón de carpetas y archivos que almacenamos en una carpeta única. Esta la guardamos en la carpeta htdocs de Xampp, y ya podíamos acceder al instalador (localhost/joomla).

En la primera ventana que se nos mostró, configuramos aspectos como el idioma, el nombre del sitio (KUBOK, el supuesto nombre de la empresa), y creamos el nombre del usuario (administrador) así como su contraseña.

Figura 41. Instalación del CMS Joomla en el Servidor 1.

Preparamos la base de datos:

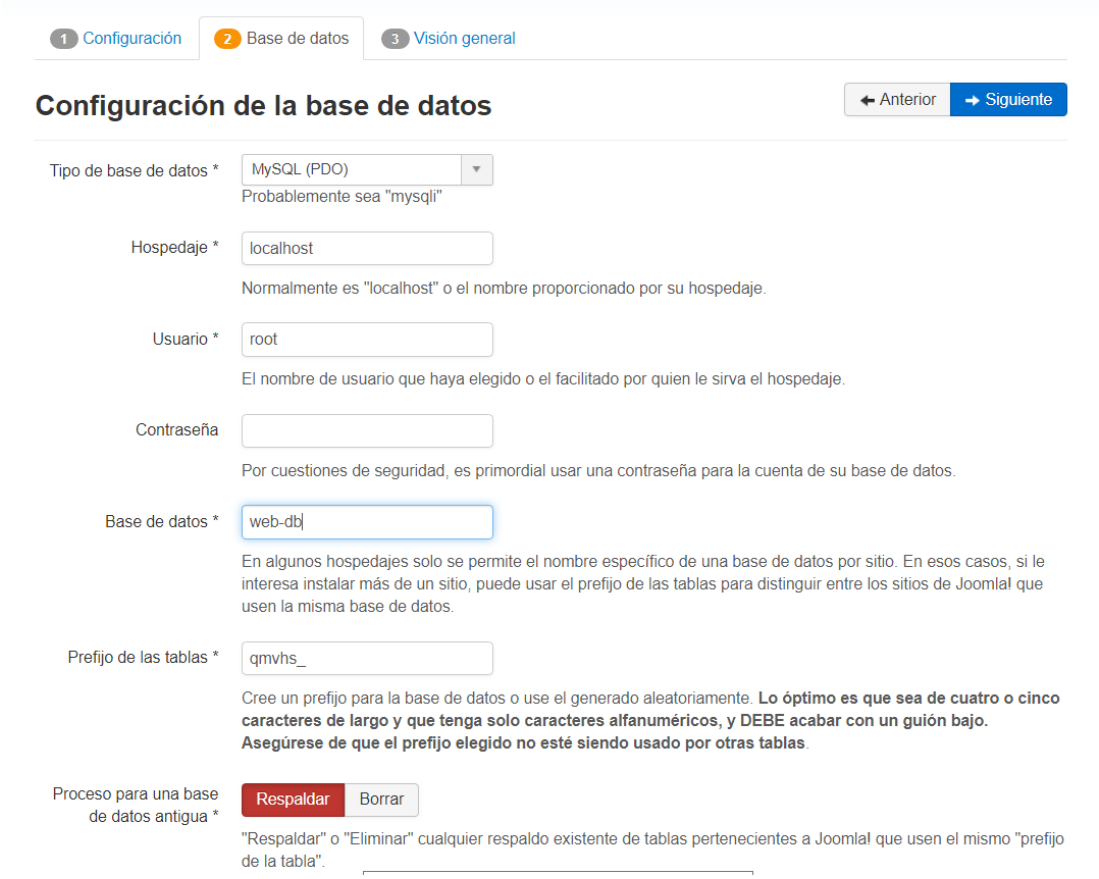


Figura 42. Instalación del CMS Joomla en el Servidor 2.

Y se nos mostraron todos los aspectos de la configuración:

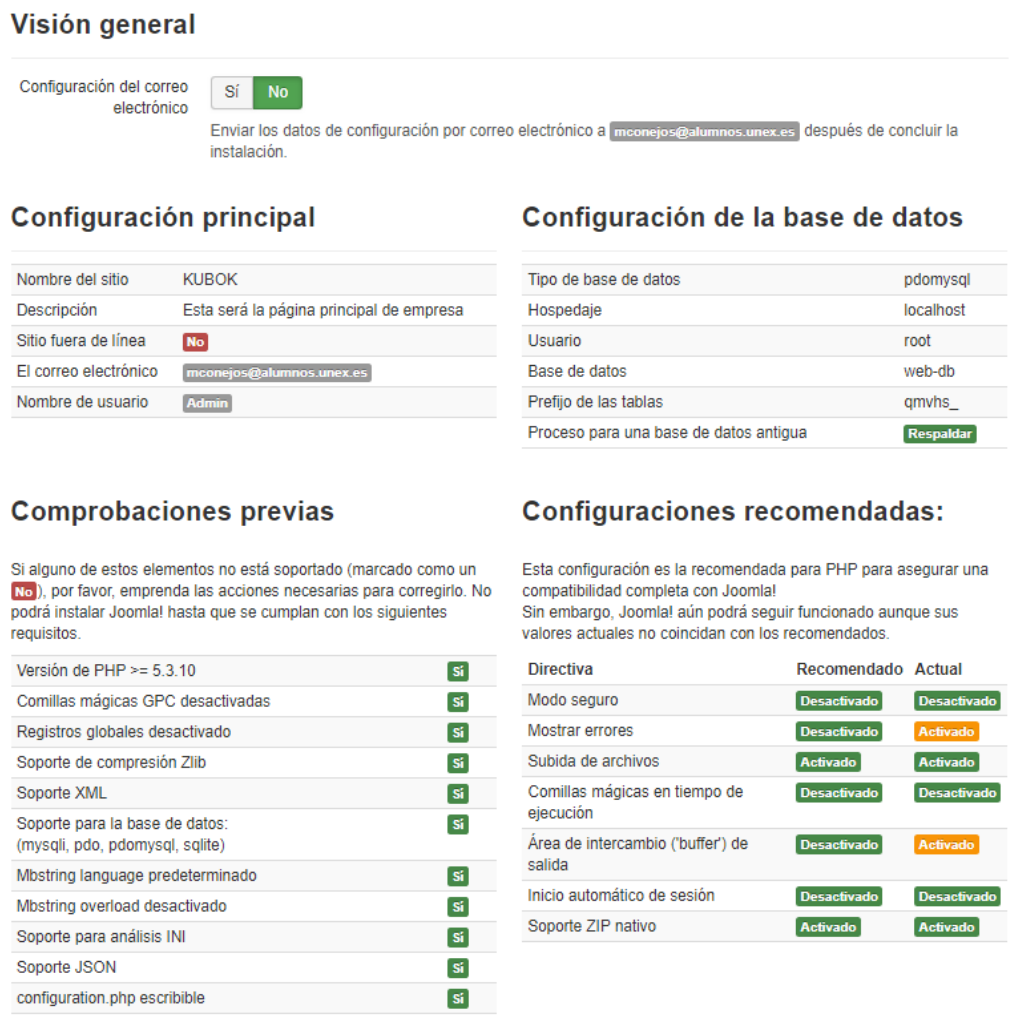


Figura 43. Instalación del CMS Joomla en el Servidor 3.

En esta vista no se cambió nada, se mantuvo todo tal y como se muestra en la captura. Después, ya pudimos instalar llegando a la siguiente ventana:

Figura 44. Instalación del CMS Joomla en el Servidor 4.

En ella, borramos la carpeta de instalación.

Figura 45. Instalación del CMS Joomla en el Servidor 5.

El segundo paso, fue modificar la página para que se pareciese un poco más a la de una empresa. Accedimos a la ventana de administración:

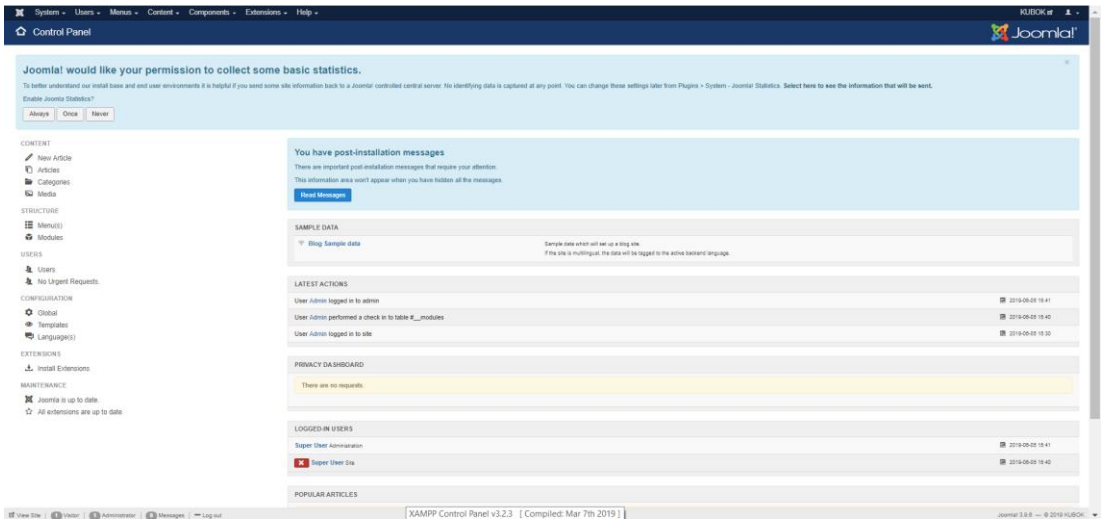


Figura 46. Vista de administración en Joomla.

Es aquí donde realizamos algunos cambios, para dar forma a nuestra web, que se viera un poco mejor y pudiésemos subir archivos entre otras cosas.

Añadimos varios usuarios, desde la vista de usuarios. Los usuarios añadidos fueron:

Jose7 → passJose

Luisa6 → passLuisa

Maria8 → passMaria

Mariano9 → passMariano

Utilizamos unas contraseñas sencillas, para que fueran fáciles de recordar, ya que el usuario tampoco tenía muchas funcionalidades sobre la web. La inserción fue más o menos así:

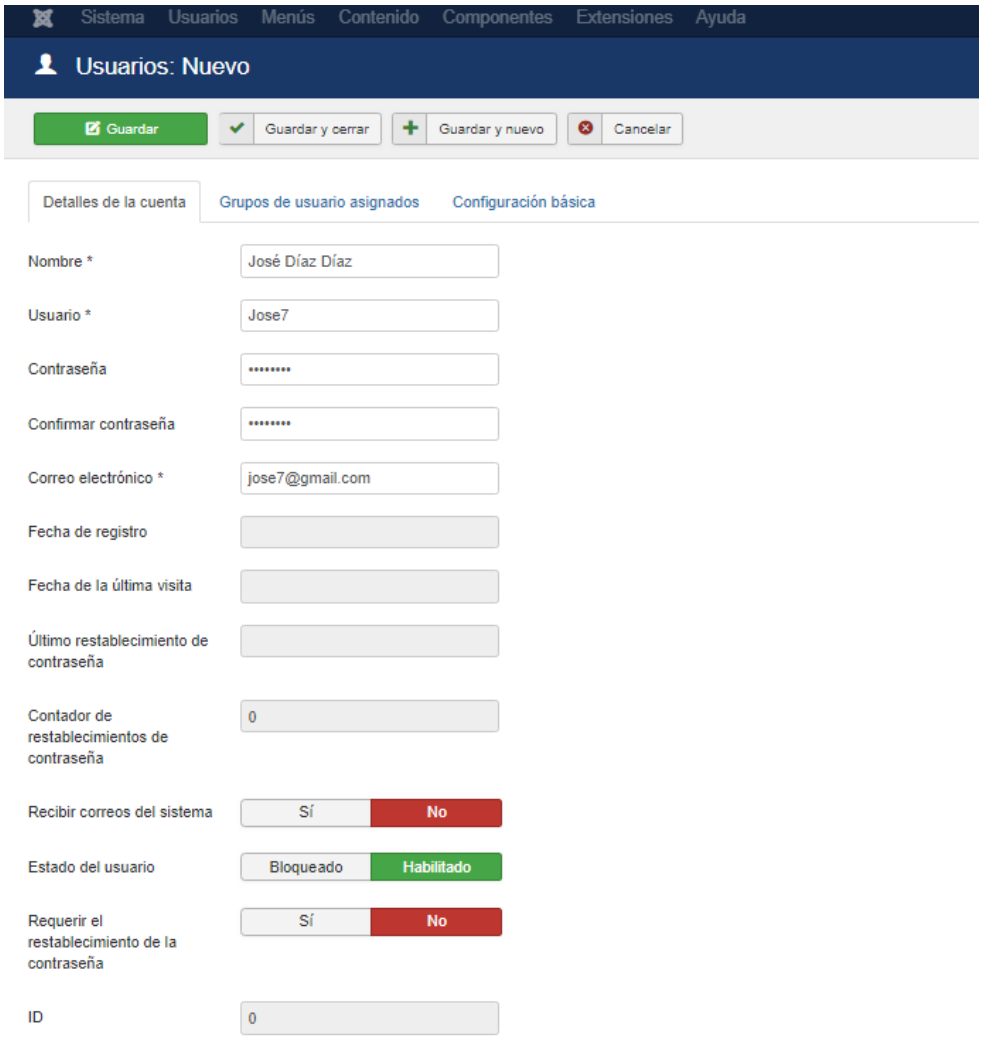


Figura 47. Inserción de un usuario en la base de datos desde Joomla.

Una vez hecho esto, cambiamos el idioma y el estilo de la web, y descargamos una plantilla (y sus dependencias). Para establecerlas, usamos la opción:

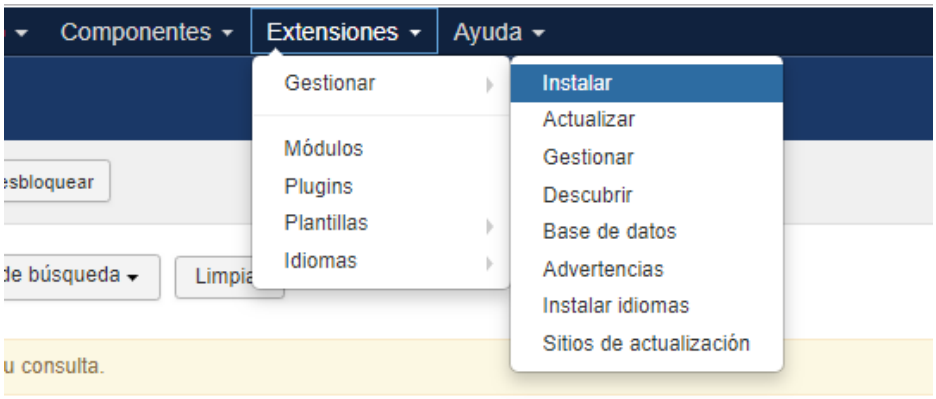


Figura 48. Instalar extensiones desde Joomla.

Sin embargo, el usuario seguía sin poder subir archivos. Para solventar esto, utilizamos phoca Download e instalamos un navegador de archivos (para poder gestionar mejor estos archivos desde la parte del administrador). La forma de instalarlos fue igual que con la plantilla.

Para darle forma al menú que se le mostraba al usuario, fuimos añadiendo distintos elementos y entre ellos las opciones de poder subir archivos y verlos gracias al componente phoca.

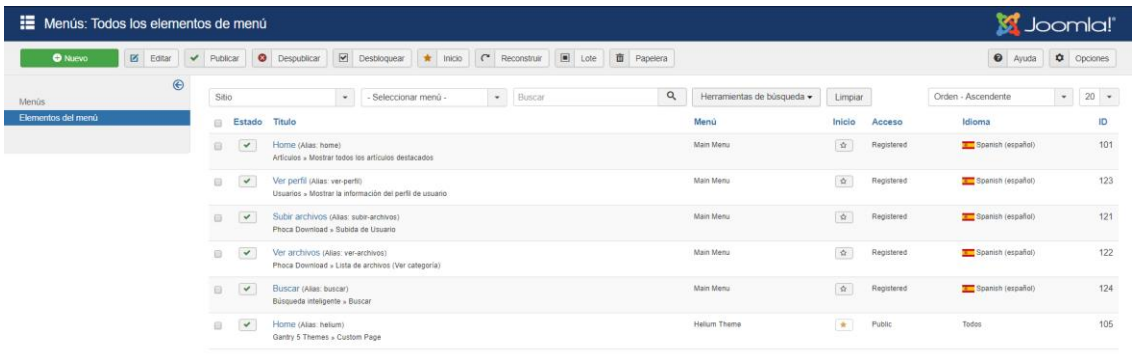


Figura 49. Creación de elementos del menú desde Joomla.

Con la intención de permitir que el usuario subiese ficheros, tuvimos que configurarlos de esta forma:

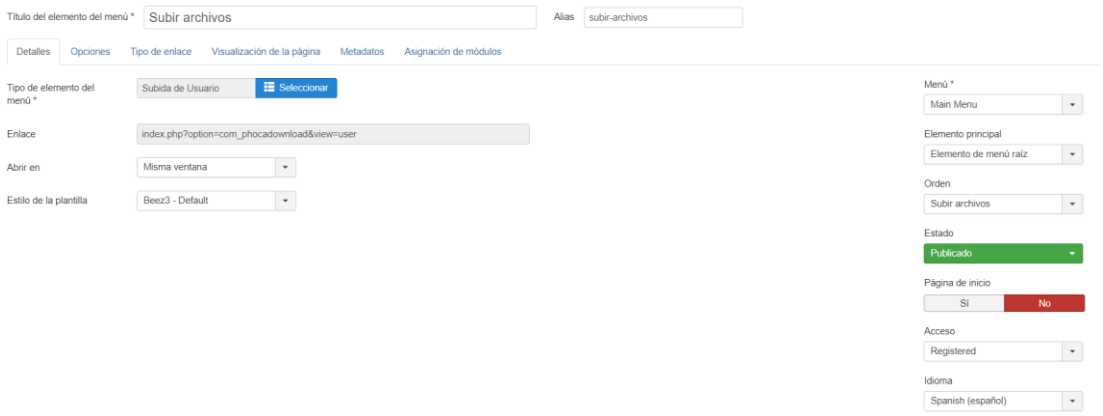


Figura 50. Opción de subir archivos desde Joomla con Phoca download.

La vista del usuario, una vez introducido sus credenciales, se veía algo así:

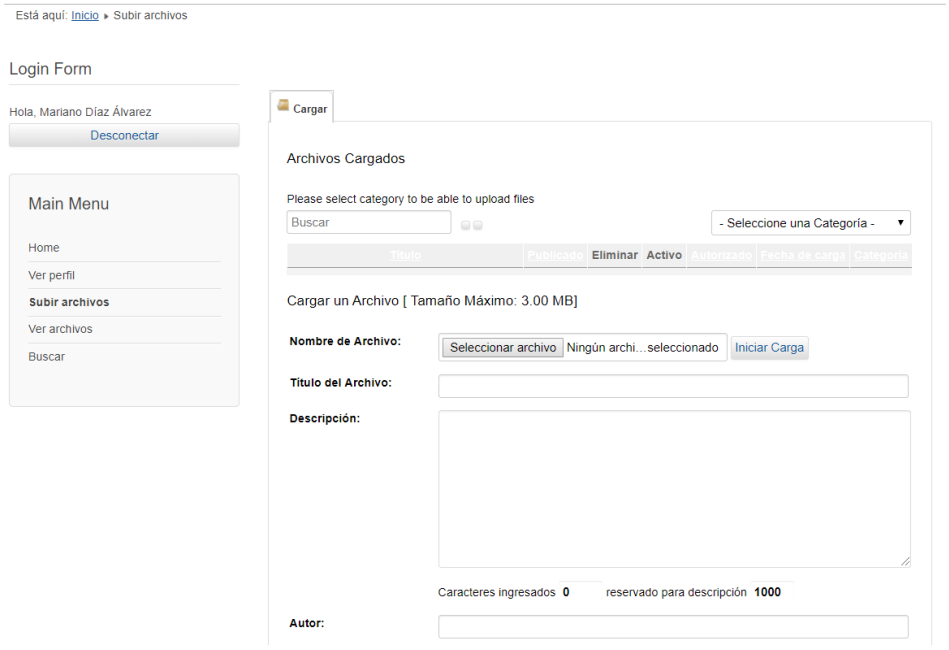


Figura 51. Vista del usuario en la página web.

Los archivos se establecieron para que solo se pudiesen borrar desde el administrador, y se supuso que el empleado, los había subido a la web por equivocación, ya que estos le podían involucrar. Por lo que se entraría en la cuenta del administrador para eliminar ambos.

Para entrar en dicha cuenta, se debía disponer de la contraseña, la cual adquiriría mediante fuerza bruta. La forma de simular este ataque, se hizo con ayuda de la herramienta Brutus, que es de Windows. Una vez descargado e instalado, lo configuramos indicando que el ataque era en local (127.0.0.1), el usuario (admin) y una lista de contraseñas.

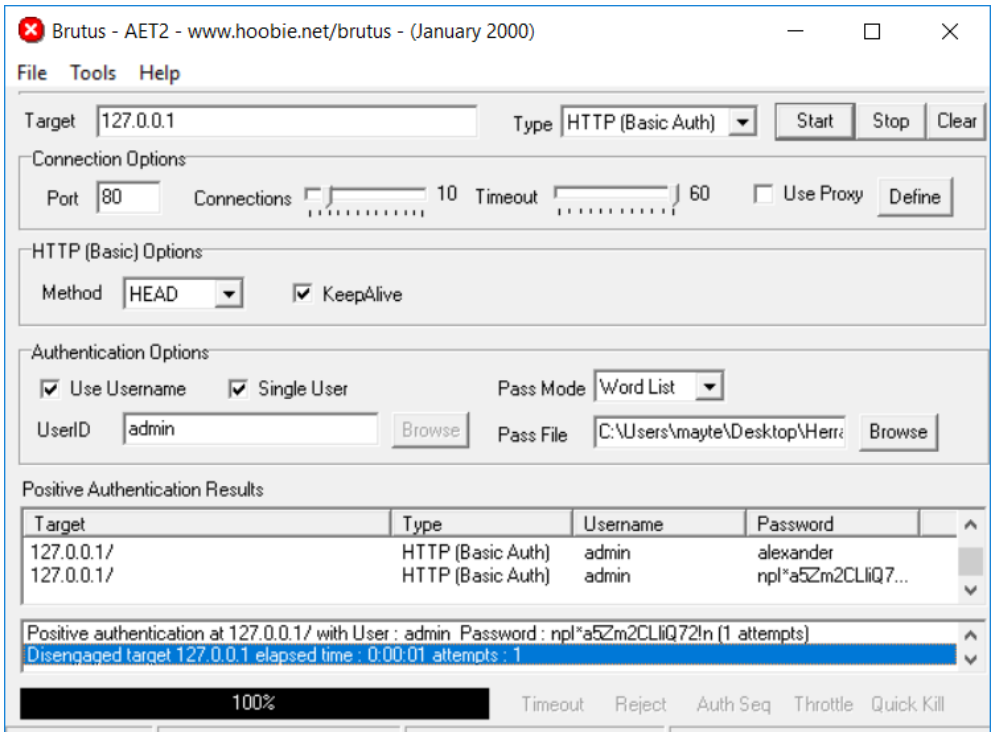


Figura 52. Ataque de fuerza bruta con Brutus.

Después de eso, entramos desde el administrador con la contraseña y borramos los archivos correspondientes (el documento y recibo previamente preparados).

Cuando ya hicimos todo esto, lo siguiente fue crear la imagen de dicho servidor. Para hacer esta imagen, utilizamos la herramienta FTK imager. Al abrir el programa, pulsamos en File y elegimos la opción de Create Disk Image. Esto nos mostró una ventana, como la que vemos en la siguiente captura:

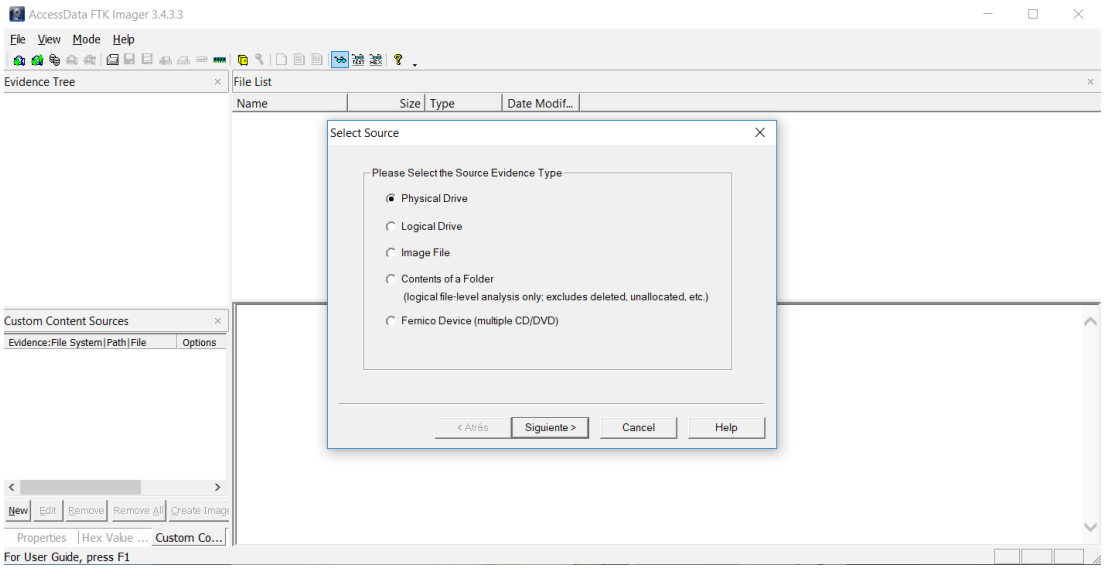


Figura 53. Creación de imagen dd del servidor con FTK Imager 1.

Aquí elegimos que haríamos la imagen a partir de una unidad física. Y pulsamos en siguiente, donde tuvimos que indicar que dicha unidad se trata del usb:

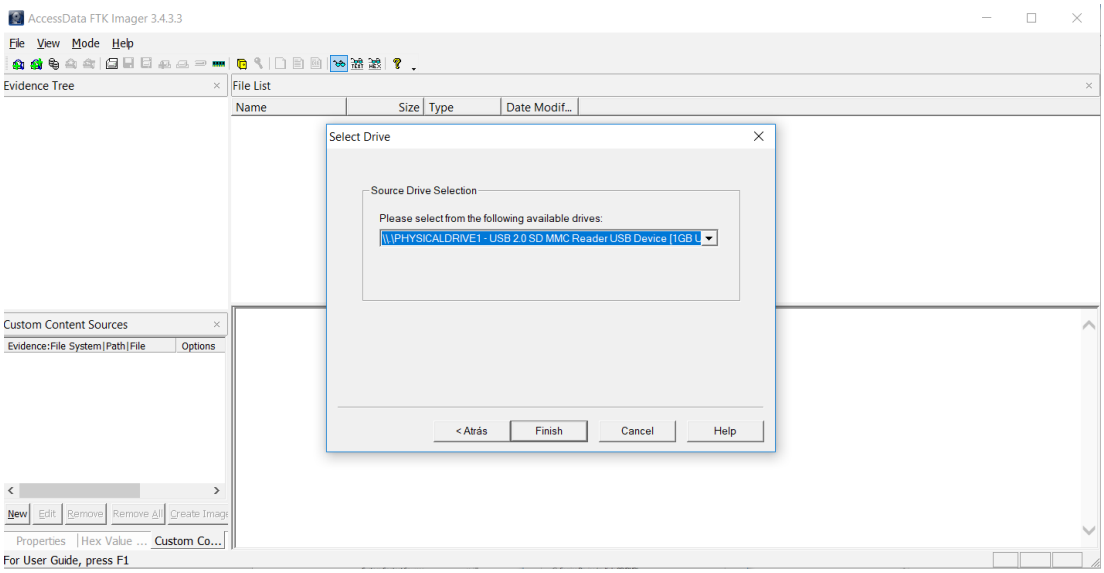


Figura 54. Creación de imagen dd del servidor con FTK Imager 2.

Después pulsamos sobre Finish, y nos apareció una nueva ventana. En esta clicamos sobre Add…

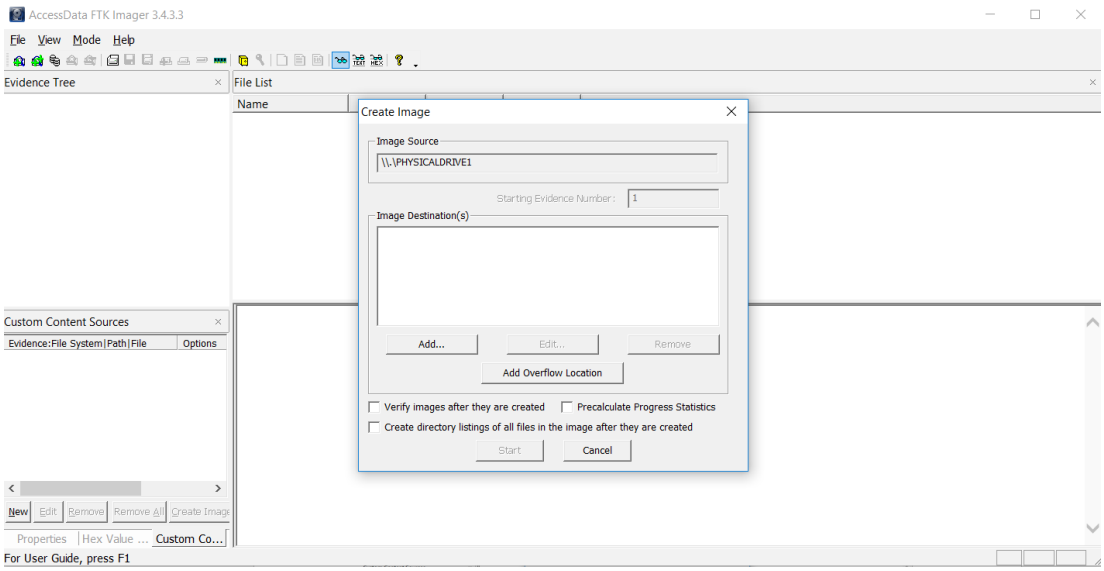


Figura 55. Creación de imagen dd del servidor con FTK Imager 3.

Esta opción, nos mostró una nueva ventana, donde fuimos eligiendo y rellenando una serie de datos. Lo primero que hicimos, fue marcar el tipo de imagen que queríamos, en este caso, raw image (con extensión dd):

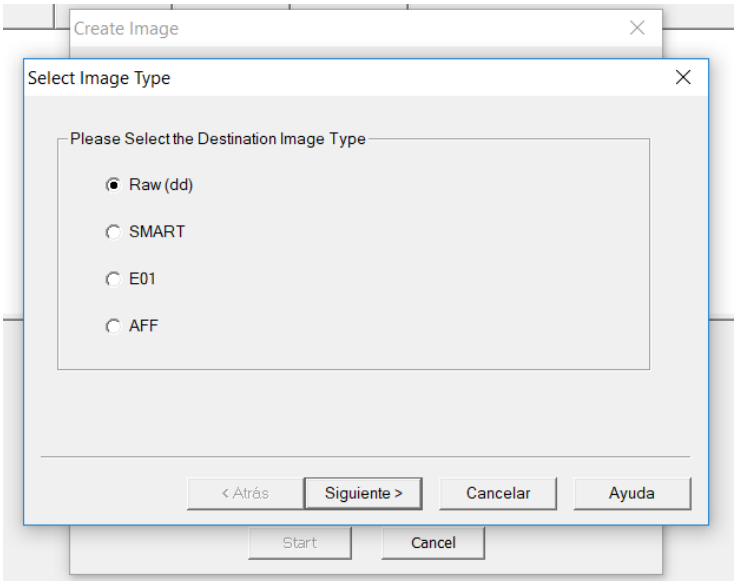


Figura 56. Creación de imagen dd del servidor con FTK Imager 4.

Después indicamos algunos datos correspondientes a la información de la imagen:

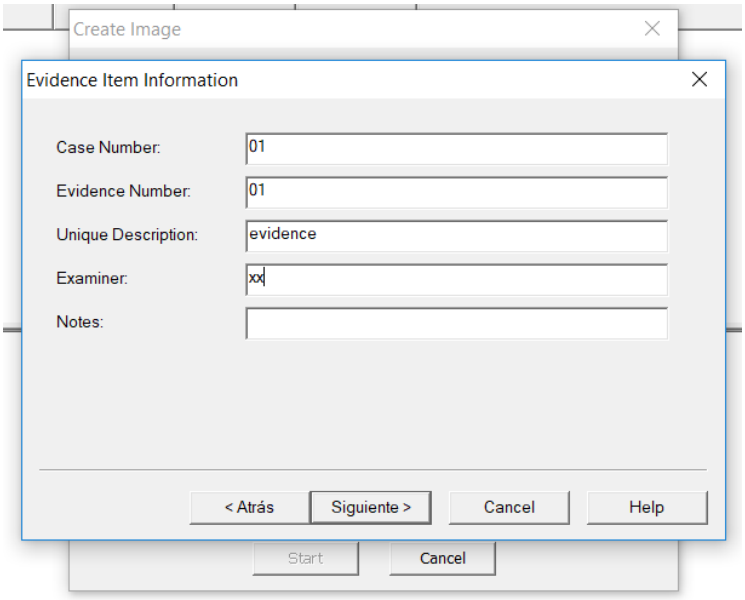


Figura 57. Creación de imagen dd del servidor con FTK Imager 5.

Finalmente elegimos la carpeta en la que queríamos guardar nuestra imagen, el nombre de la misma y si deseábamos que estuviese fragmentada (indicamos que no, mediante un 0 en Image Fragment Size):

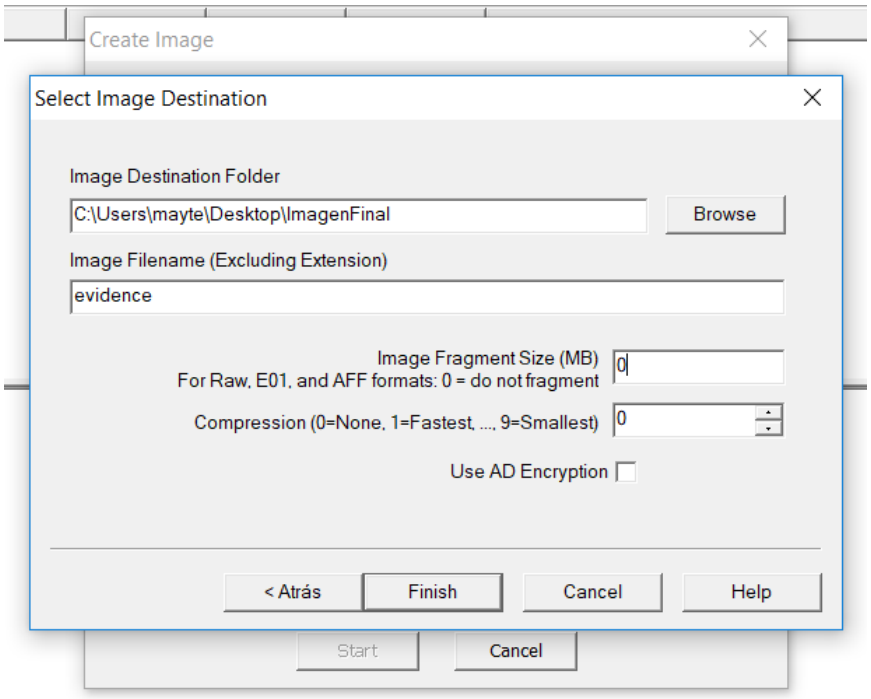


Figura 58. Creación de imagen dd del servidor con FTK Imager 6.

Después pulsamos sobre Finish y se creó nuestra imagen.

- Elaboración del recibo:

Para conseguir hacer un recibo que fuese más o menos real, buscamos una imagen en internet de algún recibo de una transferencia bancaria que pudiésemos modificar para ponerle los valores que necesitásemos (fecha, cantidad, entidad, beneficiario).

Como nombre de la empresa, se puso KUBOK. Para el iban ordenante se tomó un ejemplo de internet, y para el otro también.

- Preparación del archivo Excel:

Junto al recibo se habría producido la descarga de un archivo Word (docx), que en realidad se trataría de un archivo Excel.

En el archivo Excel pusimos unos gastos, pero los mínimos, como unos 750 euros del total. Donde había ido a parar el resto del dinero, se ocultaría dentro de dicho archivo. Básicamente, fuimos construyendo una tabla con los meses y posibles gastos, algunos bastante absurdos, las opciones de las tablas eran muy similares a las de Word.

Para meter lo que queríamos poner dentro hicimos igual que con el Word, extraíamos las carpetas y las volvimos a meter convirtiéndolo en .zip. Almacenamos una nueva carpeta previamente creada, con el nombre docs para que no llamase mucho la atención. Una vez hecho esto, cambiamos su extensión a .docx.

Se modificaron los metadatos para eliminar cualquier posible información que pudiese involucrarnos. Esto se hizo con la herramienta utilizada en el Reto 1.

- Creación de la carpeta docs:

Esta carpeta contendría una imagen de un correo, dónde se indicaría el personaje al que fue a parar el dinero del primer recibo. Para la creación de la carta, buscamos un formato de carta dentro de las plantillas disponible en Word y creamos una.

Además, incluimos en esta carpeta una pista sobre el diccionario rockyou, para indicar que era necesario el uso del mismo, y de esta manera poder descifrar el archivo zip con contraseña, el cual también estaría en esta carpeta.

Junto con las imágenes anteriores, introdujimos una imagen que sería una foto rápida de un recibo, la cual no se podría ver muy bien, ya que estaría borrosa.

Para hacer la parte de la imagen borrosa, utilizamos el programa Befunky. Es online y bastante sencillo, y en la propia página te explican cómo hacerlo.

- Parte final:

Creamos un documento Word, con contraseña. El documento, se trataba de un contrato de compraventa de una vivienda, y al final del mismo insertamos la bandera.

Posteriormente, procedimos a establecerle una contraseña. Básicamente, hicimos lo siguiente: entramos en el archivo y dimos clic en la pestaña de Archivo. Una vez dentro, elegimos la opción de proteger Documento. Ahí indicamos cifrar con contraseña, esto nos abrió una pequeña ventana, donde indicamos la contraseña que queríamos. Pusimos una bastante sencilla "leche".

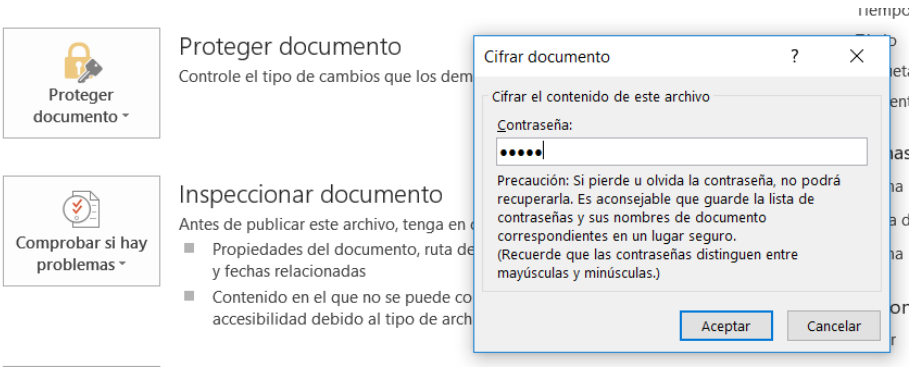


Figura 59. Establecer contraseña a un documento Word.

Sin embargo, seguía apareciendo información de la última persona que modificó el documento. Para evitar que se mostrase esto, hicimos una serie de pasos. En la opción de archivo, opciones, centro de confianza, configuración del centro de confianza, opciones de privacidad e inspector de documentos. Nos mostró un listado, donde debíamos dejar marcado propiedades del documento e información personal, y luego dar a inspeccionar.

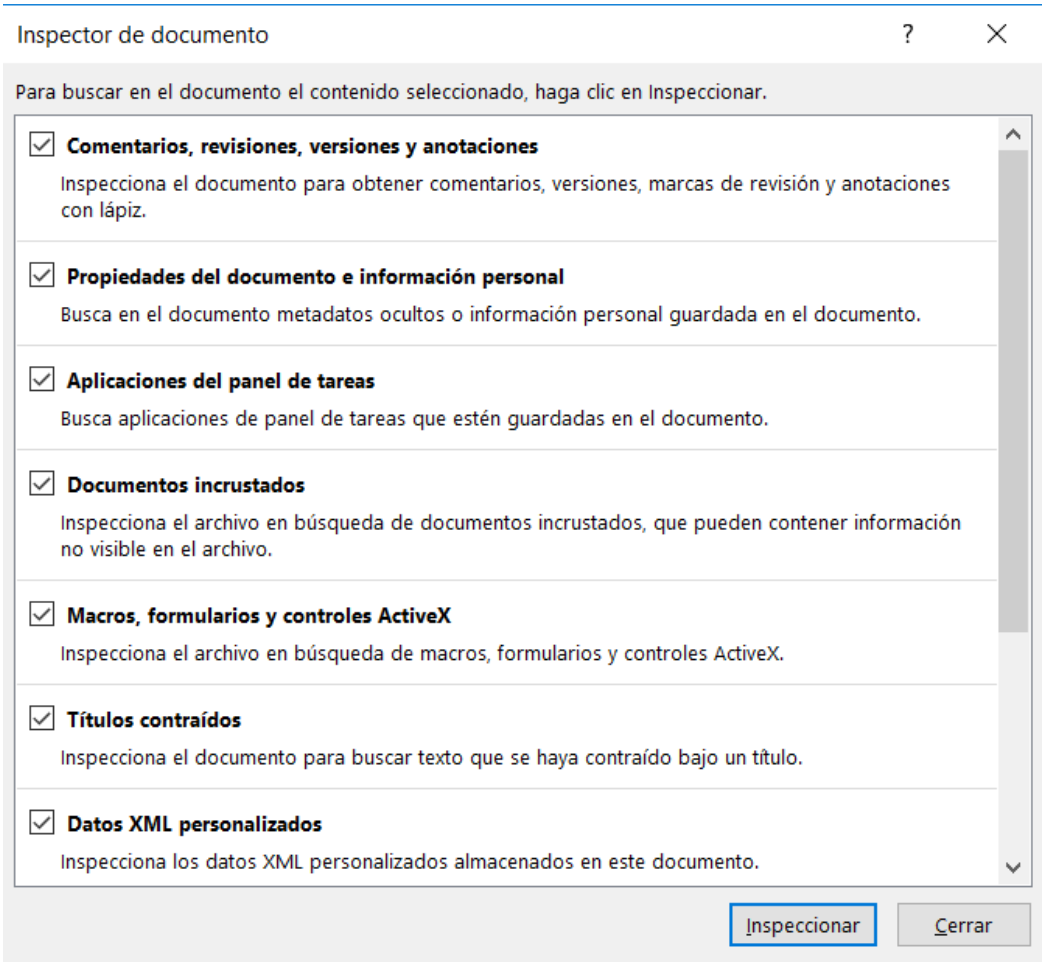


Figura 60. Inspector de documentos de un archivo docx.

Se nos mostró la opción de quitar todo, esto permitía eliminar cualquier posible información que hubiese quedado registrada de los autores.

Figura 61. Eliminar información personal de un documento docx.

Junto con el documento, se encontraría otra imagen que nos daría una pista sobre la contraseña del Word, además esta, también la tendría oculta en su código hexadecimal. Esta parte, se desarrolló igual que en los retos anteriores.

Finalmente, el archivo .zip que contendría a la imagen y al documento, presentaría contraseña, y esta tendríamos que obtenerla haciendo uso del famoso diccionario rockyou.txt.

Creamos nuestro archivo desde Kali Linux, desde la consola, comprimiendo ambos elementos. Establecimos la contraseña (chocolate) que queríamos, y utilizamos:

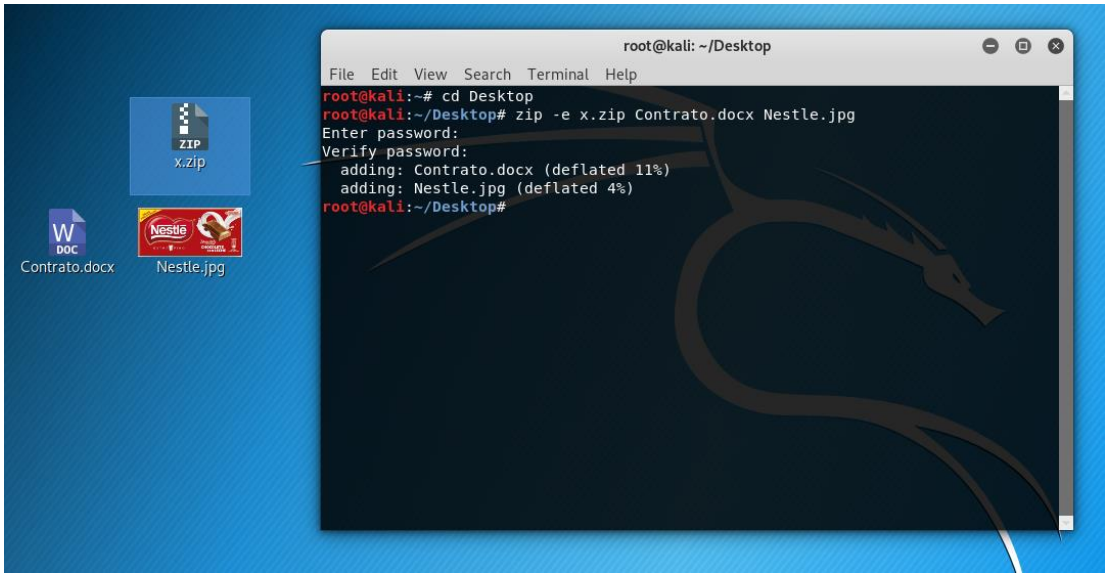


Figura 62. Establecer contraseña a un archivo zip desde Kali Linux.

5.2.2. Tráfico de armas

Partimos de un caso entregado a un perito, para conocer información acerca de una banda de traficantes de armas. Se proporciona una imagen de un pendrive, que uno de los policías infiltrados ha conseguido obtener.

Habrá varias preguntas que responder como quiénes son los proveedores (existirá un archivo de base de datos), la localización de los mismos (en el archivo anterior se incluye una forma de geolocalización), lista de compradores (archivo .csv con nombres y números), información útil sobre las armas (cuales son, cuantas tienen,... existirán varias imágenes y un registro), y la venta de licencias falsas (esta será una información a descubrir), se tratará de algún documento que lo acredite, con contraseña.

Exceptos las imágenes los otros elementos serán borrados del pen drive, por lo que será necesario recuperarlos.

La ficha de descripción del mismo es:

Nombre: Tráfico de armas.

Historia:

Uno de nuestros policías ha estado infiltrado con una de las mejores bandas de España implicada en el tráfico de armas. Ha necesitado varios meses, pero finalmente ha conseguido realizar una copia de uno de los pendrive donde se almacenaba información, cuando esta se cambiaba de un equipo a otro. Es necesario encontrar respuesta a varias preguntas como ¿quiénes son los proveedores?, sabemos que son americanos ¿pero de qué estados? (con la intención de poner sobre aviso a las autoridades pertinentes), y alguna información útil sobre las armas (¿qué tipo de armas son?, ¿cuántas tienen?). Tal vez, podamos encontrar algo más que nos pueda ser de interés.

Categoría: Forense.

Nivel: Medio.

Descripción:

Se proporciona una imagen de un pen drive, será necesario analizarla y recuperar alguna información que pudiese haber sido borrada. También es importante, poder responder a las dudas que subyacen a nuestros investigadores.

Bandera: todas las respuestas a las preguntas.

Pistas:

- Todo lo que necesitamos, está dentro de la imagen.

Preparamos los archivos necesarios, según han sido descritos en la descripción inicial sobre el reto.

- Preparación del archivo correspondiente a los proveedores:

En primer lugar, necesitábamos un listado de posibles nombres de proveedores. Para conseguir una buena cantidad de nombres y crear un documento medianamente completo se utilizaron páginas sobre nombres y apellidos estadounidenses. Entonces en un documento se fueron escribiendo posibles combinaciones.

También se incluyeron varios nombres españoles para gestionar aquellos individuos que eran mandados a los encargos. La idea era crear un archivo con formato de base de datos, que incluyera varias columnas: nombres, apellidos, números de teléfono, correos y algunos puntos de encuentro.

De algunos proveedores dispondríamos de números de teléfono porque serían más habituales y de otros solo de correos, o puede que ambas cosas. Para conseguir los números usamos una guía (abctelefonos). En el caso, de los españoles, los generamos de forma aleatoria. Para los correos electrónicos, utilizamos distintos dominios localizados por Internet, a parte de los básicos y comunes Gmail, Outlook y Hotmail. Lo siguiente, fue decidir de qué parte de los estados unidos iban a ser estos individuos, para ello utilizamos Wikipedia y así pudimos ver las diferentes ciudades dentro de los estados.

Por lo que en el archivo de texto anterior, indicamos para cada uno la ciudad a la que podrían pertenecer, y después buscamos los teléfonos correspondientes o un punto de encuentro, donde establecimos unas coordenadas. Utilizamos el api de google maps. Buscamos la localización que nos interesaba dentro de la ciudad, la marcábamos y obteníamos sus coordenadas.

Cuando ya tuvimos lista la tabla, la ordenamos en función de la columna apellidos, alfabéticamente. Para ello, en la pestaña de presentación correspondiente a la tabla, elegimos la opción Ordenar:

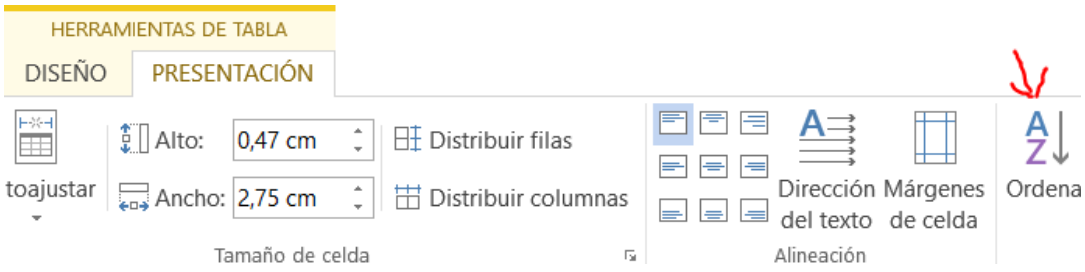


Figura 63. Ordenar una tabla alfabéticamente en Word.

Y configuramos la forma en que queríamos ordenar, en este caso fue por apellidos y de forma ascendente, para que fuese en orden alfabético:

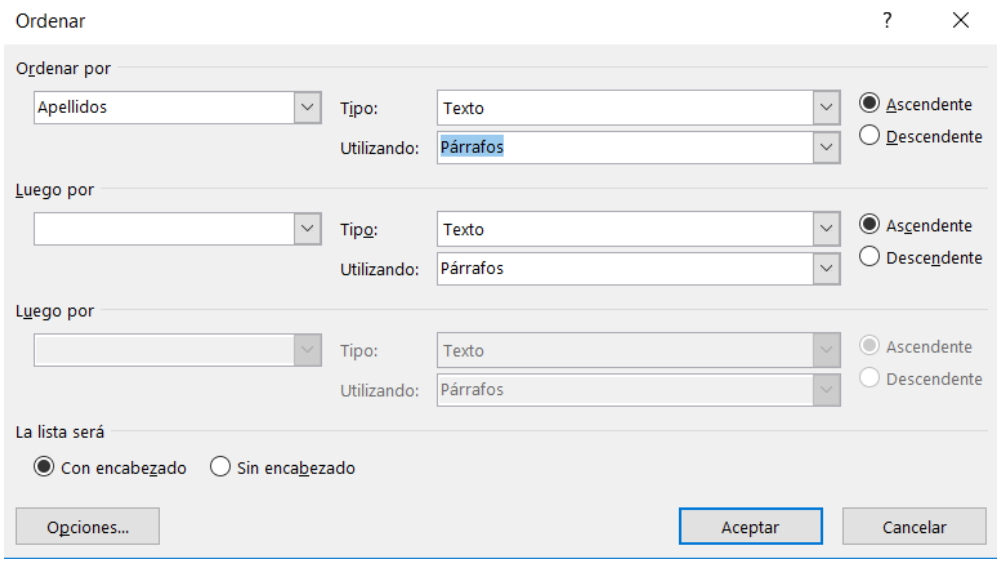


Figura 64. Configuración de orden alfabético en función de una celda de la tabla.

Después, creamos esta tabla en una base de datos. Para hacer esto, utilizamos Access, de Microsoft Office. Nos situamos sobre la tabla en una de sus columnas, fuimos a la pestaña de Presentación de la tabla y pulsamos el icono de convertir texto a:

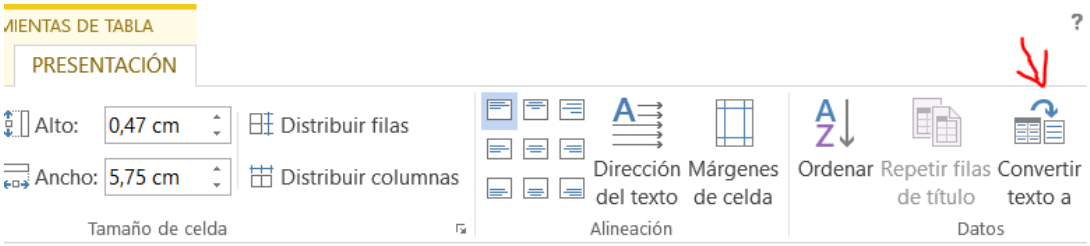


Figura 65. Convertir tabla de Word a texto.

Escogimos como carácter delimitador, el ;, y guardamos el archivo txt. Abrimos Microsoft Access, y creamos una base de datos nueva:

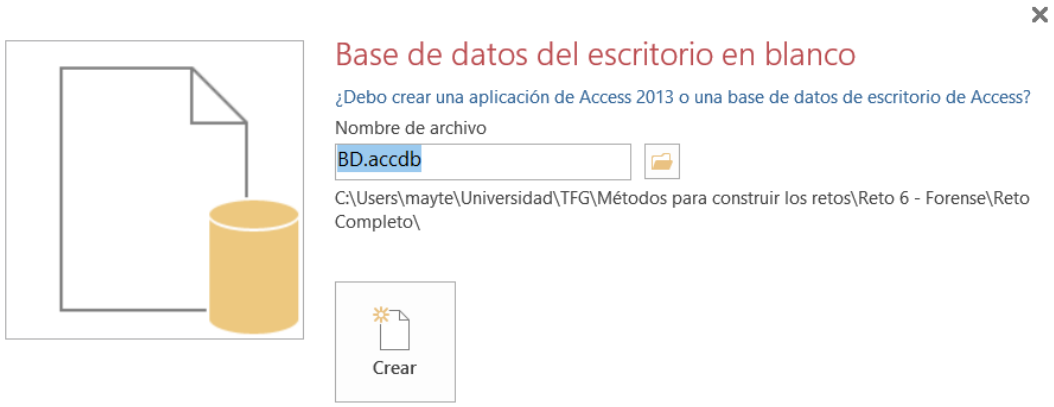


Figura 66. Creación de base de datos en Microsoft Access.

Una vez abierta la base de datos en cuestión, hicimos click en la ficha Datos externos y en el icono Archivo de texto, que encontramos en la cinta de opciones.

Figura 67. Importar tabla a Microsoft Access 1.

Utilizamos el botón Examinar para seleccionar el fichero de texto que habíamos creado en pasos anteriores y activamos también la primera de las casillas que aparecían disponibles. Luego pulsamos Aceptar.

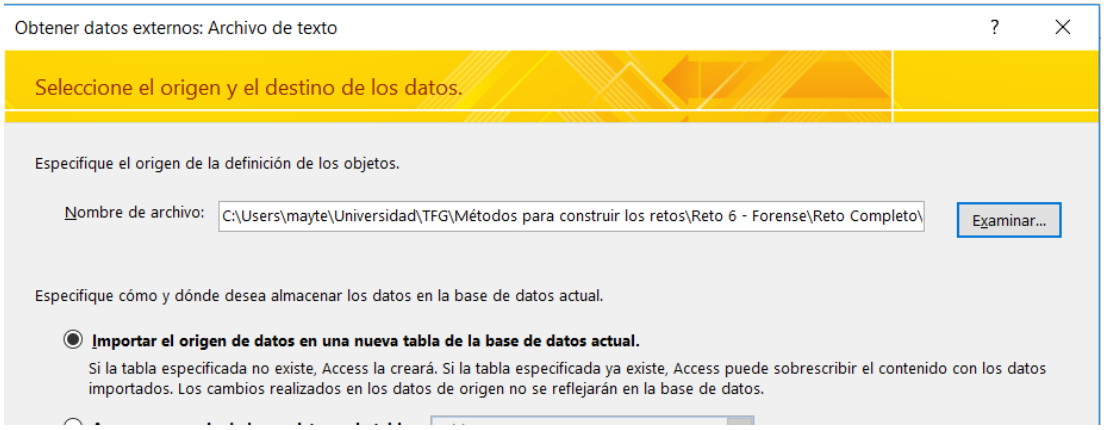


Figura 68. Importar tabla a Microsoft Access 2.

En la primera ventana del asistente de importación, activamos la casilla Delimitado y pulsamos Siguiente.

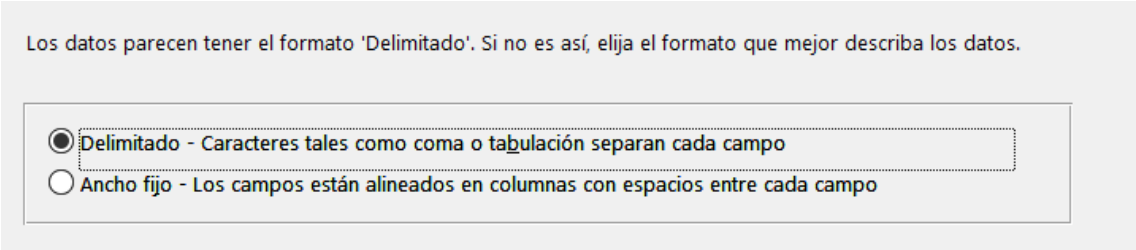


Figura 69. Importar tabla a Microsoft Access 3.

Luego pinchamos sobre Punto y coma y activamos también Primera fila contiene nombres de campos, ya que nuestra tabla de Word incorporaba una primera fila con los encabezados de cada campo (Nombre, Apellidos, Puntos de encuentro, Teléfonos y Correos). Pulsamos de nuevo Siguiente.

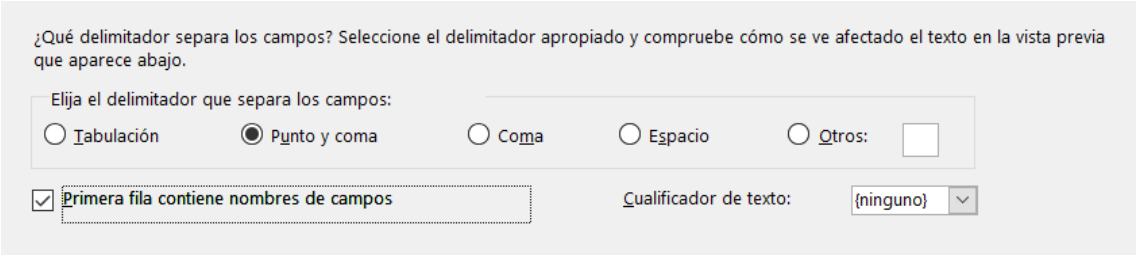


Figura 70. Importar tabla a Microsoft Access 4.

Al finalizar el proceso, la base de datos se quedó con una nueva tabla, propia de Access con los campos (columnas) y registros (filas) de la tabla de Microsoft Word.

Finalmente, importamos la tabla a una base de datos Oracle. Primero, debíamos importarlo a un documento Excel. Seleccionamos la tabla que queríamos exportar, botón derecho y exportar a Excel.

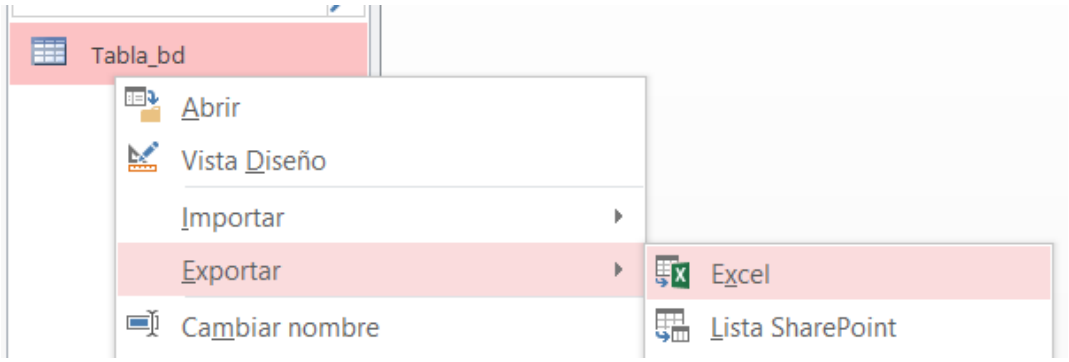


Figura 71. Importar tabla de Microsoft Access a una base de datos Oracle 1.

Elegimos donde queríamos que se guardase y pedimos que se conservase el formato. Pulsamos en Siguiente:

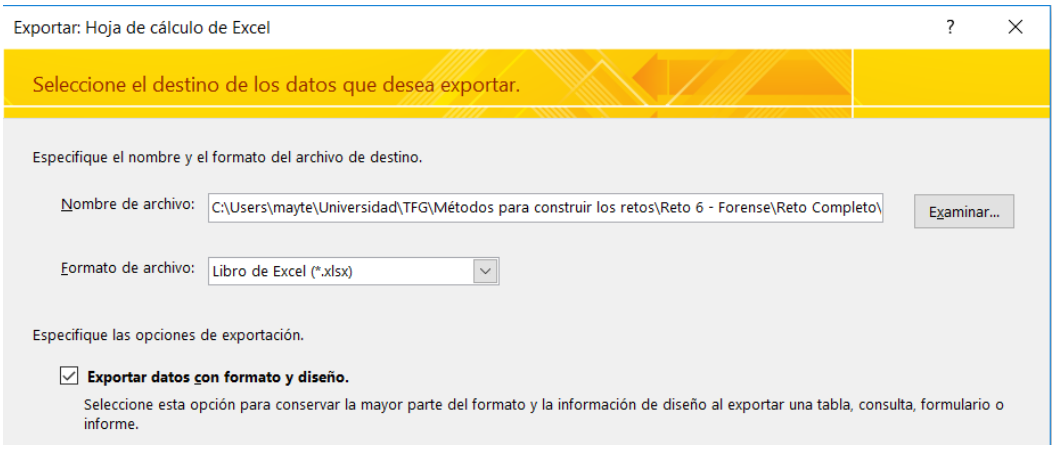


Figura 72. Importar tabla de Microsoft Access a una base de datos Oracle 2.

Una vez tuvimos el documento Excel accedimos a sql Developer, y nos conectamos a nuestra base de datos. Creamos una tabla:

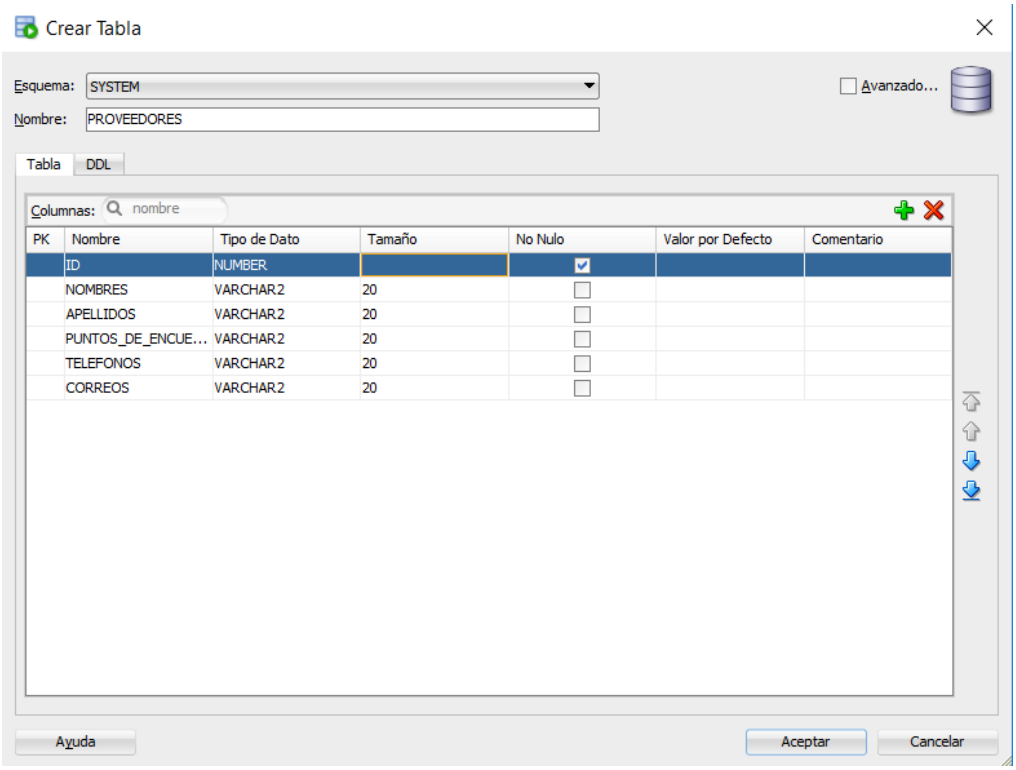


Figura 73. Importar tabla de Microsoft Access a una base de datos Oracle 3.

Una vez creada, importamos el archivo, haciendo clic derecho sobre ella:

Figura 74. Importar tabla de Microsoft Access a una base de datos Oracle 4.

Se nos abrió el asistente de importación, donde indicamos el archivo xlsx que acabábamos de crear:

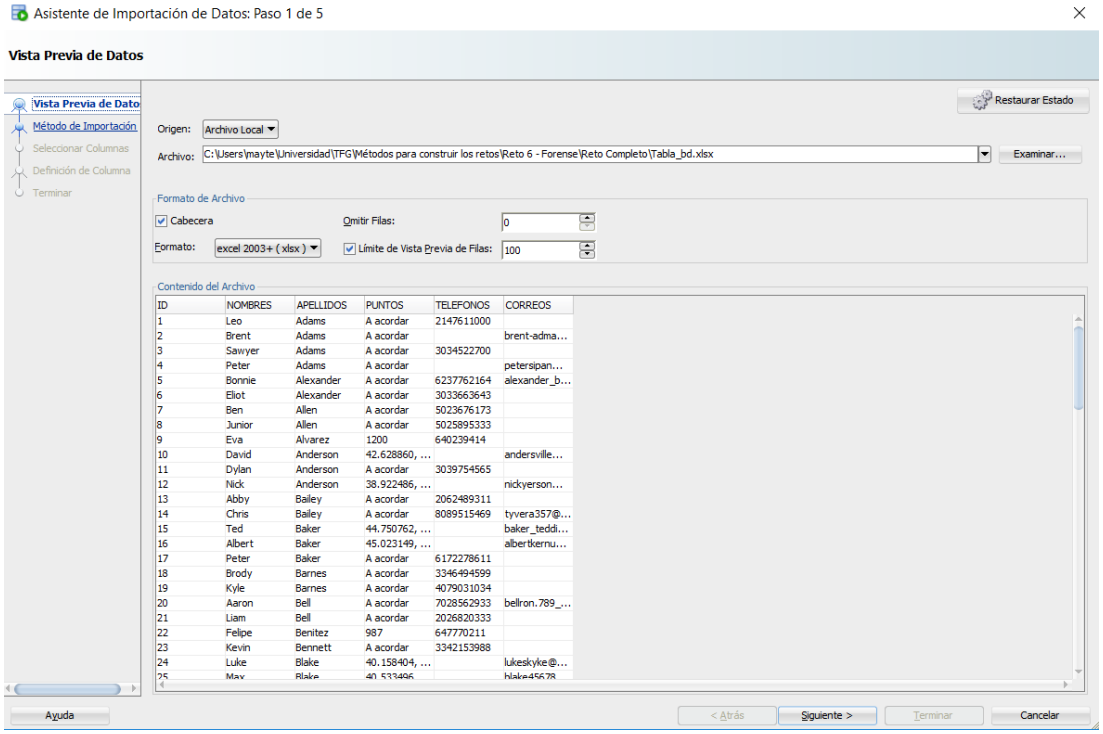


Figura 75. Importar tabla de Microsoft Access a una base de datos Oracle 5.

La primera pantalla nos mostró el contenido de nuestro archivo Excel. Como la primera línea del Excel, contenía los nombres de las columnas y queríamos mantenerlos, pulsamos el checkbox cabecera (header), de esta manera a la hora de relacionar un campo del Excel con un campo de nuestra tabla sería más intuitivo (la relación se realizará en pasos posteriores).

La siguiente ventana nos pidió elegir el método de importación y el límite de datos a importar, la cual no marcamos.

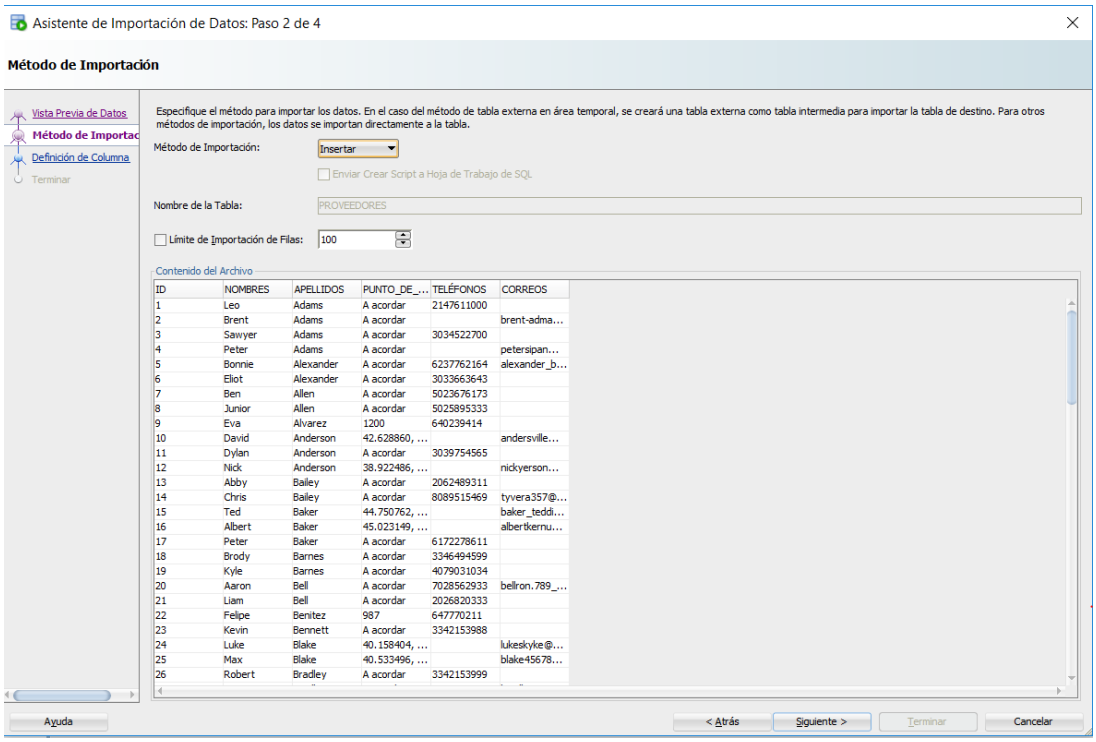


Figura 76. Importar tabla de Microsoft Access a una base de datos Oracle 6.

En la siguiente ventana escogimos los campos que nos interesaban insertar. En nuestro caso, importamos casi todas, por ello están en la ventana de la derecha.

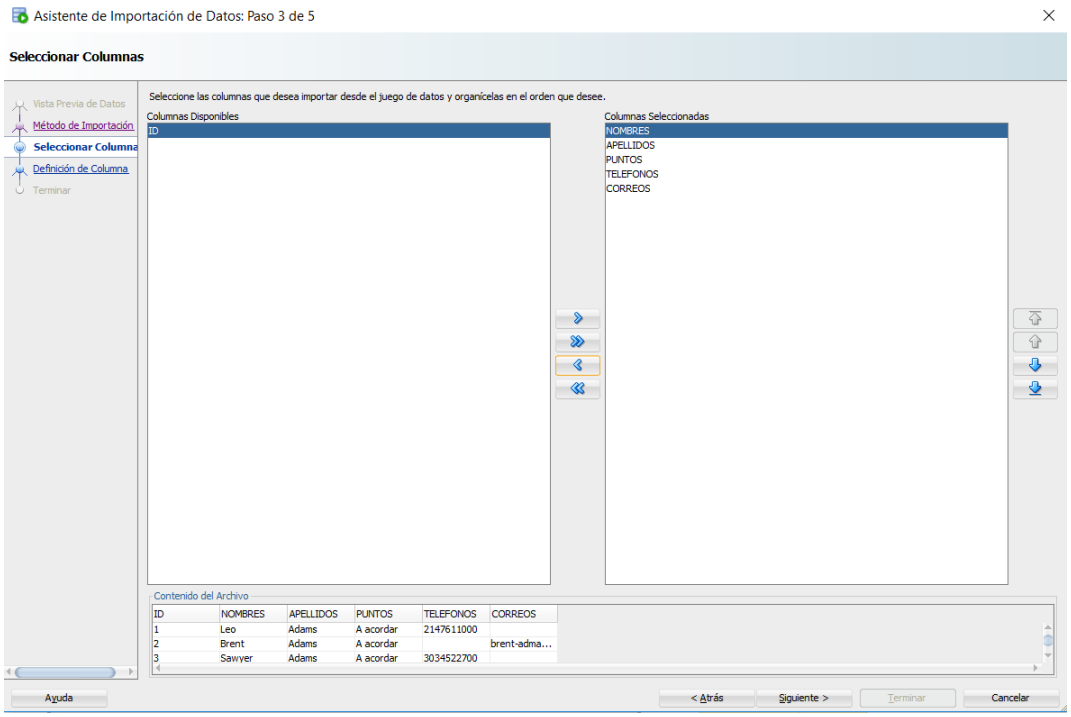


Figura 77. Importar tabla de Microsoft Access a una base de datos Oracle 7.

Por último, relacionamos la columna del Excel con la columna de la base de datos.

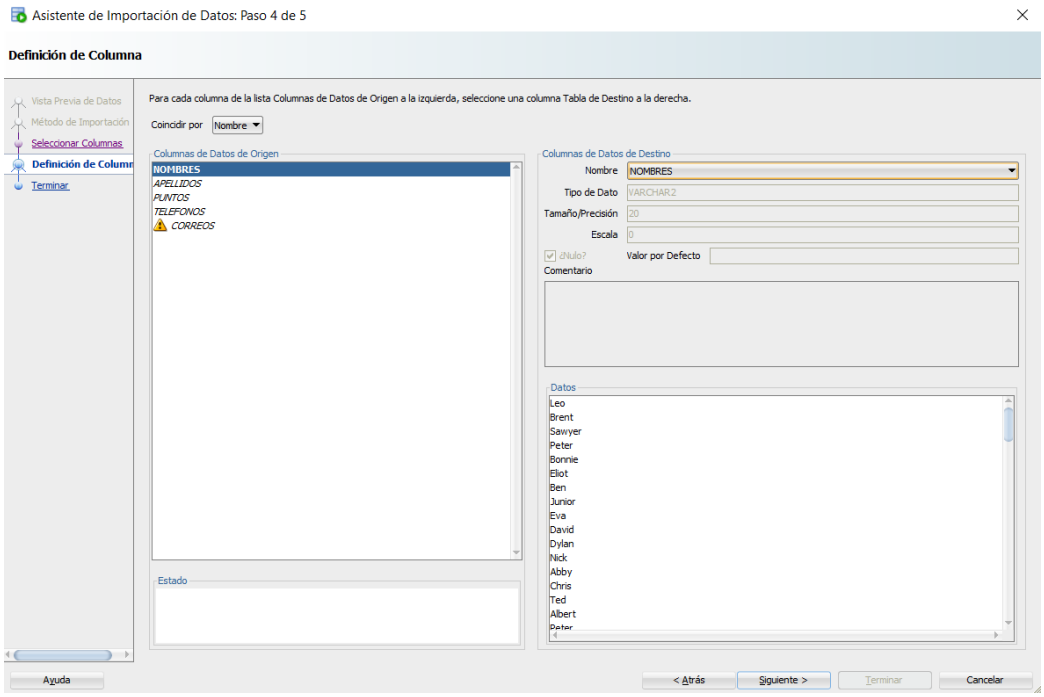


Figura 78. Importar tabla de Microsoft Access a una base de datos Oracle 8.

Finalmente, guardamos el archivo correspondiente a dicha tabla. Para ello, hicimos clic derecho sobre la tabla y elegimos la opción exportar:

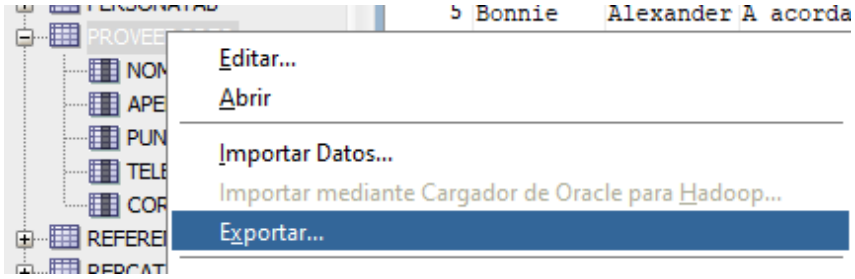


Figura 79. Creación del archivo sql con la tabla de proveedores 1.

Esto nos abrió una ventana, como la que se muestra a continuación. En la cual, indicamos que queríamos que se nos exporte como un archivo de formato insert y el nombre con el que queríamos guardarlo. El resto de datos los mantuvimos:

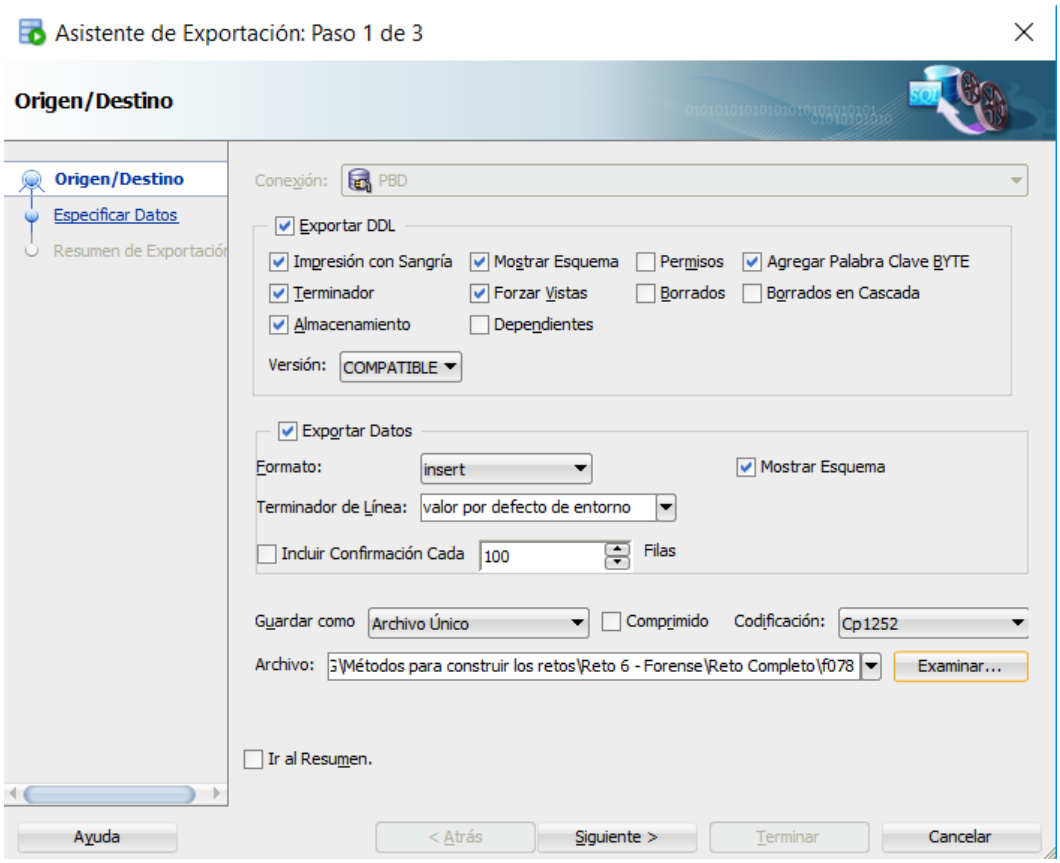


Figura 80. Creación del archivo sql con la tabla de proveedores 2.

Y ya teníamos el fichero, así que le eliminamos la extensión para que no se viese a simple vista que se trataba de eso.

❖ Preparación del archivo correspondiente a los compradores:

Al igual que con el anterior, necesitábamos un listado de posibles nombres de compradores. En este caso, estos podían ser particulares o incluso otras bandas. Solo almacenaríamos unos cuantos ya que no serían compradores ocasionales sino eventuales que se pondrían en contacto con ellos para comprarles las armas. Guardamos nombres y números de teléfonos (generados aleatoriamente). Creamos un archivo Excel con estos datos.

También incluimos algunos números, estos eran para hacer referencias a códigos de armas y cantidades. No obstante, aún no habíamos terminado con este fichero. Ciframos su contenido usando el cifrado de los illuminatis.

Para facilitar el trabajo a la hora de cifrar, descargamos la fuente con todos los símbolos. Una vez instalada, ya aparecía en nuestro Microsoft Office. Por lo tanto,

marcamos todo el texto escrito en el fichero y en la pestaña de la fuente elegimos este nuevo estilo, quedándonos algo así:

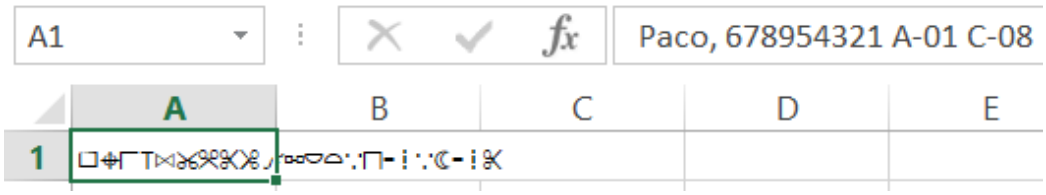


Figura 81. Proceso de cifrado haciendo uso de la fuente Illuminati Dirigens Berlin.

No obstante, si no teníamos esta fuente podíamos ver la información real, por lo que sustituimos el texto por una imagen del mismo. Y para que no se pudiese mover esa imagen, era necesario proteger la hoja. Por ello, fuimos a la opción de Formato y elegimos Proteger hoja:

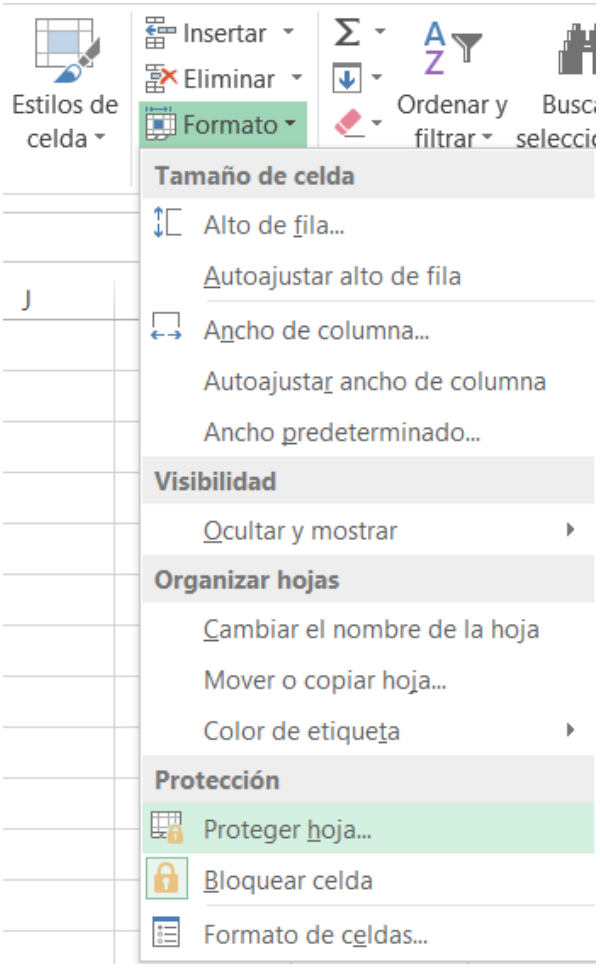


Figura 82. Proteger hoja de cálculo para evitar que se pueda modificar 1.

En la ventana que se nos abrió, establecimos una contraseña, **Cha6**.

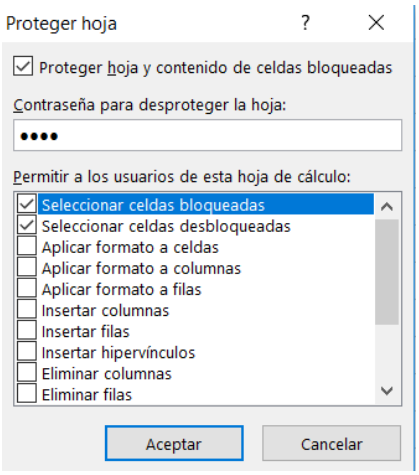


Figura 83. Proteger hoja de cálculo para evitar que se pueda modificar 2.

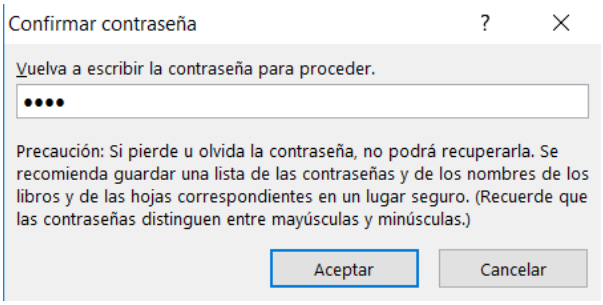


Figura 84. Proteger hoja de cálculo para evitar que se pueda modificar 3.

Ya no se podía modificar y mostraba un aviso al pulsar sobre alguna de las celdas:

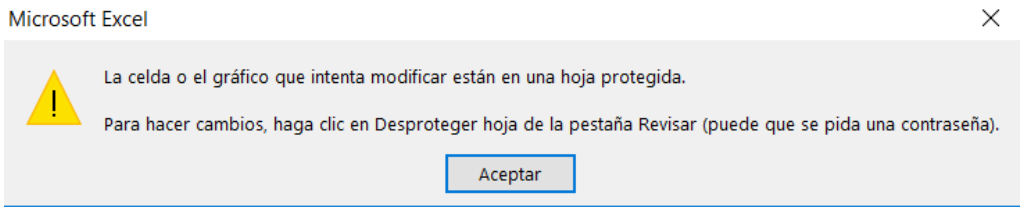


Figura 85. Mensaje mostrado cuando se intenta modificar una hoja protegida.

Para aquellos más despistados, hicimos un pequeño cambio de manera que al intentar abrir el archivo nos mostrase un aviso de error. Simplemente cambiamos la extensión del mismo a csv. Y al intentar abrirlo nos mostraría un mensaje así:

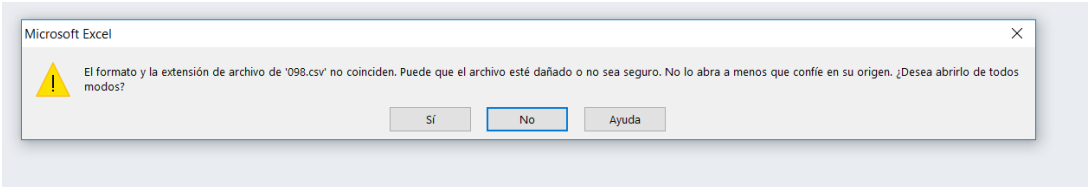


Figura 86. Mensaje mostrado cuando es necesario reparar un documento Office.

Una vez creado el documento, modificamos sus metadatos para que no apareciese ningún nombre como creador o última persona en modificarlo.

❖ Preparación de la información de las armas:

En el pen incluimos varias imágenes de armas, para que se pudiese proporcionar un poco de información sobre el tipo de armas con las que se traficaba,

después las nombramos con unos códigos que serían utilizados en el registro y que también guardarían relación con los puesto en el Excel.

Para conocer distintos tipos de armas y obtener algunas imágenes, utilizamos Wikipedia. Primero creamos un inventario de las armas, que fue la otra información que se proporcionó de las mismas. A la hora de construirlo, tuvimos en cuenta, lo que se estableció a los compradores en el archivo xltx creado más arriba. Para crear el inventario utilizamos la herramienta Spred32. Los valores indicados en el medio eran las cantidades que había de cada arma y a los lados de estos, estaban los que simbolizaban las ventas programadas. Los valores en la parte superior eran los números de serie que se asociaban con los que de los laterales, por ej: A523033

Por último, también se cambiaron los nombres de las imágenes para dar pistas, y poder entender el significado del inventario.

- Preparación de la licencia de armas:

Dentro del pen drive habría un documento con unos prototipos de licencias de armas así como de la guía de pertenencia de armas.

Guiándonos por la forma de estos carnets que encontramos entre las imágenes de google, construimos unos prototipos de los mismos. Para ello, primero necesitábamos el fondo de estos, obtuvimos una imagen de la licencia, de su parte trasera, con poca información.

Eliminamos todo el texto de la misma. Para llevar a cabo esto, utilizamos el programa Photo Stamp Remover, muy sencillo de utilizar. Con el pincel no situamos sobre aquello que queríamos eliminar, y lo fuimos marcando:

Figura 87. Eliminar elemento de una imagen con Photo Stamp Remover 1.

Después, simplemente pulsamos el botón Remover:

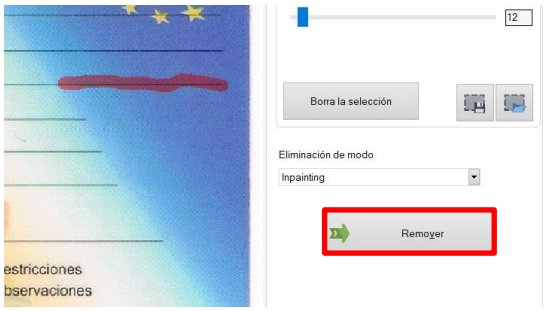


Figura 88. Eliminar elemento de una imagen con Photo Stamp Remover 2.

Y la línea desapareció, quedando únicamente el fondo que la rodea:

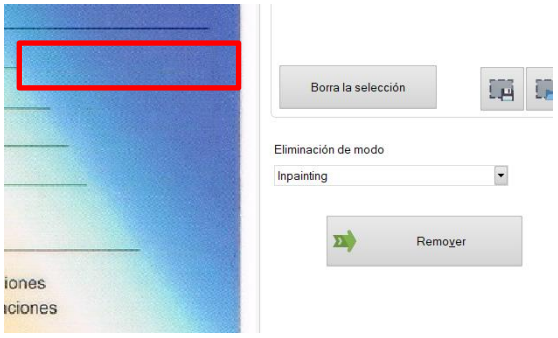


Figura 89. Eliminar elemento de una imagen con Photo Stamp Remover 3.

Finalmente conseguimos una imagen así:

Figura 90. Resultado obtenido con Photo Stamp Remover.

Por lo que procedimos a construir la licencia y guía de pertenencia. En primer lugar, respetando las medidas de un carnet de este tipo. Una vez que ya conocíamos esto, comenzamos a construir uno. Simplemente, abrimos un Word nuevo, y utilizamos la opción de cuadro de texto:

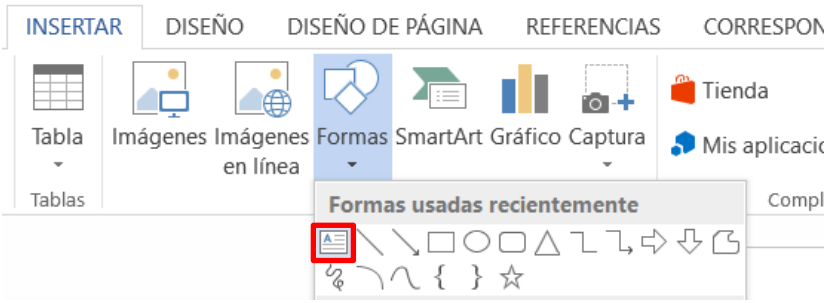


Figura 91. Crear un prototipo de carnet desde un documento Word.

Insertaremos la imagen que habíamos hecho y la adaptamos a la forma que habíamos creado más arriba. Le añadimos el contenido, tanto a la parte delantera como trasera. Hicimos igual con la guía.

Una vez creado el documento, se estableció una contraseña ("yoyoyo") que el usuario tendría que descifrar. Y nuevamente, para que no saliese información de autores, se repitió el proceso utilizado en retos anteriores.

❖ Preparación del pen drive:

Una vez que tuvimos listos todos los elementos del reto, procedimos a preparar la memoria en la que los incluimos. En primer lugar, se mantuvo la extensión de las imágenes, el docx y el csv. Y se borraron las correspondientes al fichero sql y a la imagen del inventario.

Utilizamos una memoria de 256 MB, con la intención de que la imagen final no fuese muy pesada y de que el proceso que viene a continuación no lo fuese tampoco. Formateamos la memoria haciendo uso de la herramienta Eraser. Abrimos la herramienta, y elegimos nueva tarea. Donde seleccionamos la memoria que queríamos formatear y run immediately (como ya hicimos en el reto anterior).

Incluimos los archivos en la memoria, y borramos todos salvo las imágenes (dos de ellas también se borraron).

A continuación, utilizamos otra memoria de 16 GB, en la cual, mediante la herramienta ImageUSB creamos el LiveUSB de la herramienta OSFClone, y con la misma, llevamos a cabo el clonado de la imagen de la memoria que contenía los archivos.

Al iniciar el programa, se realizó lo siguiente. En primer lugar, había que indicar el USB que deseábamos convertir en unidad de arranque. En el paso 2, se debía seleccionar la opción "write image to USB drive" y en el paso 3,

seleccionamos la versión de OSFClone (v1.2.1002). Por último, pulsamos la opción write.

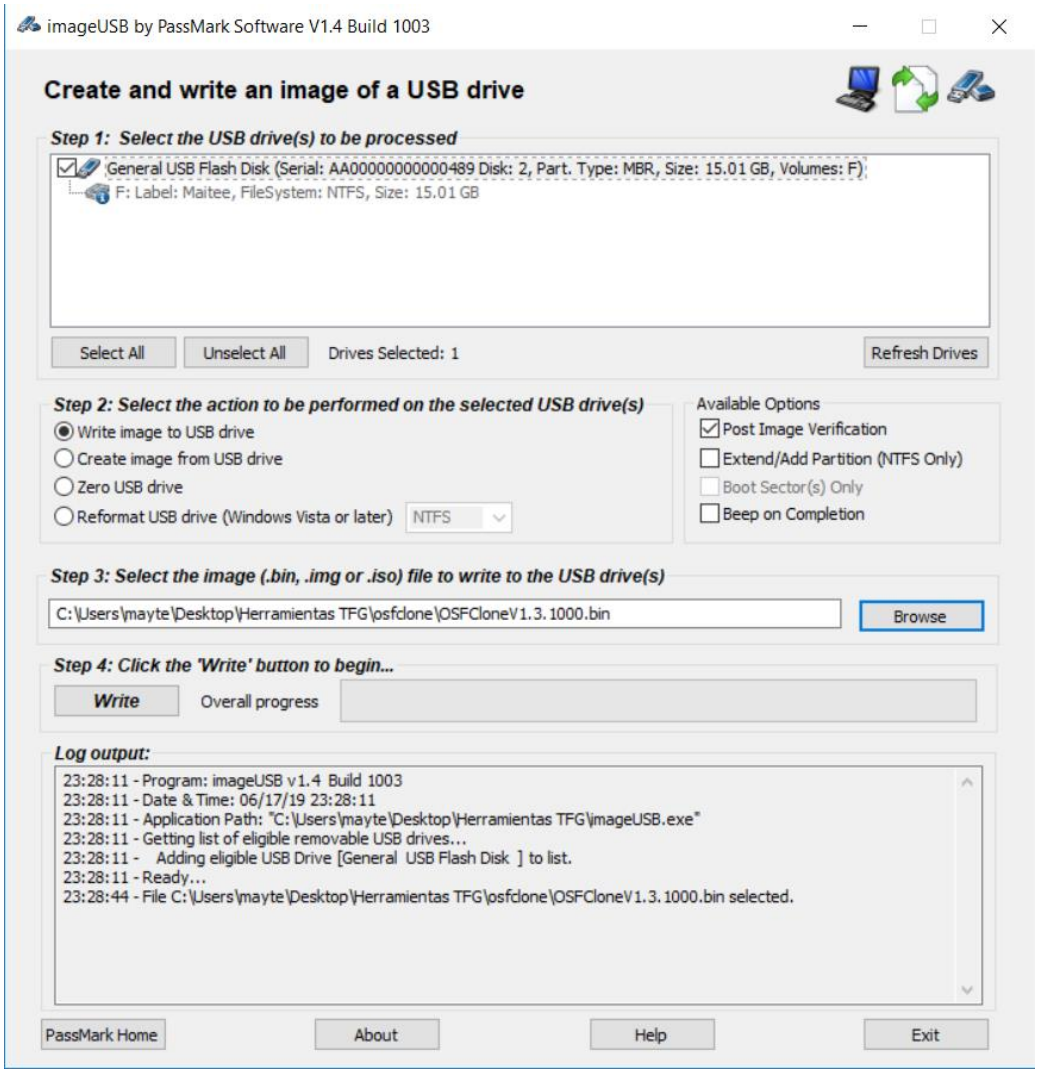


Figura 92. Preparación de una memoria como LiveUSB con ImageUSB 1.

Mientras se creaba, se mostraba el proceso mediante un porcentaje junto a la memoria USB que estábamos convirtiendo:

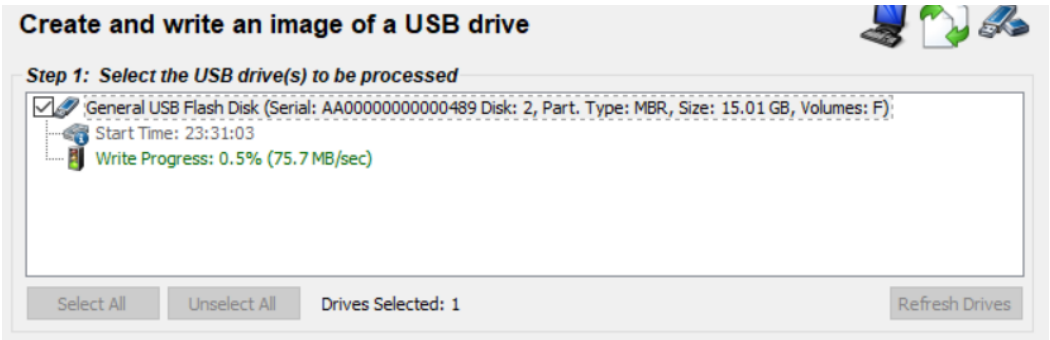


Figura 93. Preparación de una memoria como LiveUSB con ImageUSB 2.

Y cuando el proceso terminó se mostró de la siguiente manera:

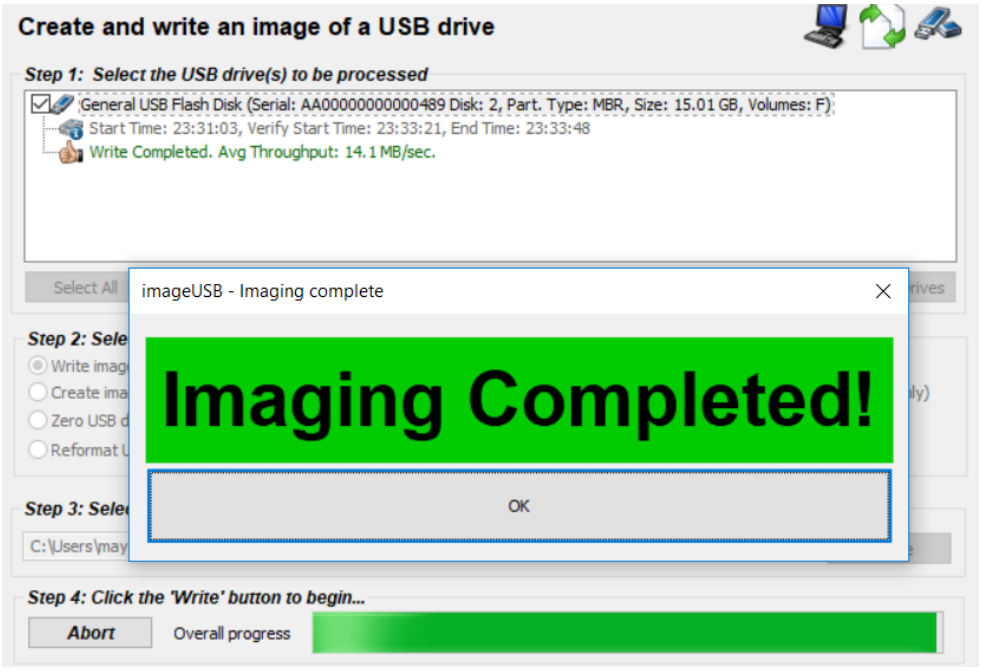


Figura 94. Preparación de una memoria como LiveUSB con ImageUSB 3.

Una vez preparado el pen, llevamos a cabo el clonado. Para realizar este proceso fue necesario entrar en la BIOS y cambiar la configuración de arranque, de esta forma, en lugar de hacerlo desde el disco duro se haría desde la memoria USB. Al reiniciar el ordenador, se nos mostró un menú como este (elegimos la opción 2):

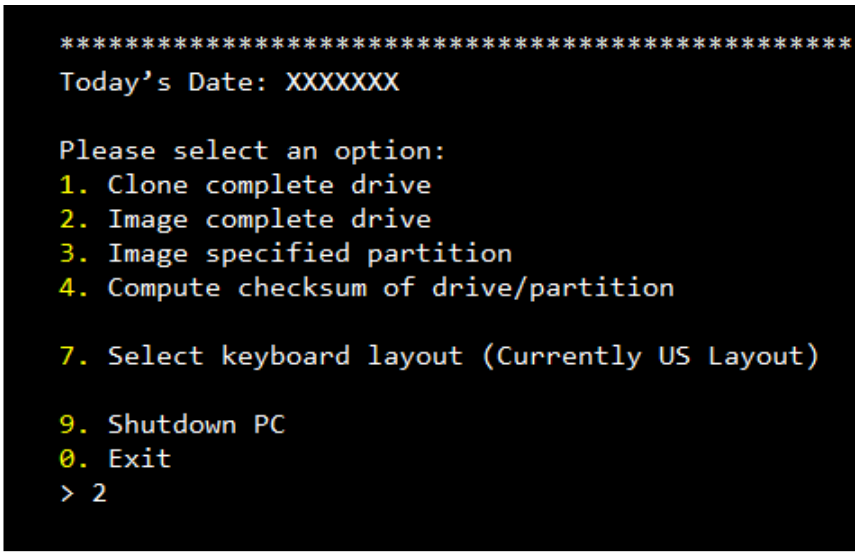


Figura 95. Pasos en el clonado de una memoria USB 1.

Esto nos mostró un nuevo menú. Aquí elegimos el formato de copia a usar, por tanto, escogimos la opción 1 (dd (via dc3dd)):

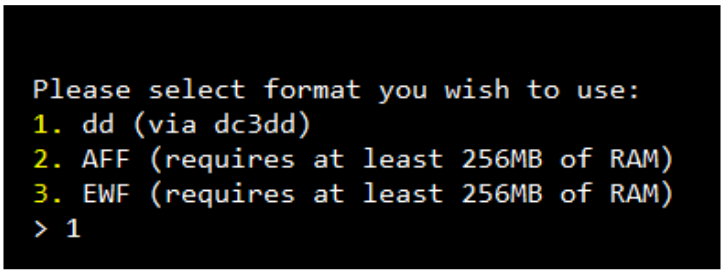


Figura 96. Pasos en el clonado de una memoria USB 2.

Entonces, se nos mostró una pantalla donde debíamos seleccionar la fuente que queríamos clonar, el lugar de destino y las opciones de copia. Para seleccionar la fuente (es decir, la memoria que contenía los archivos), marcamos la opción 1:

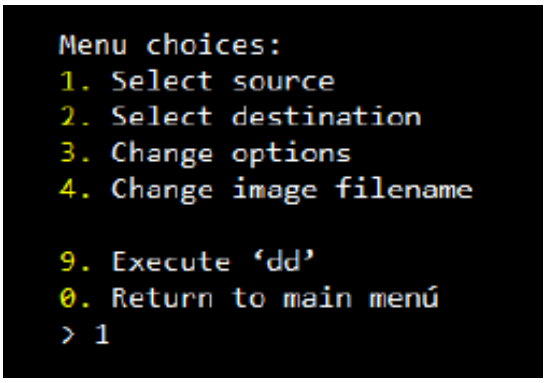


Figura 97. Pasos en el clonado de una memoria USB 3.

La pantalla que se nos mostró a continuación de selección, presentaba este orden:

- [0] -> Disco duro personal
- [1] -> Memoria que queremos clonar
- [2] -> Live flash USB

Por tanto, aquí de nuevo, elegimos la opción 1. Al volver al menú anterior, seleccionamos la opción 2 para elegir el destino en el que queríamos almacenar la imagen clonada.

Utilizamos para almacenar la imagen, nuestro propio disco duro, por lo que se marcó la opción 1:

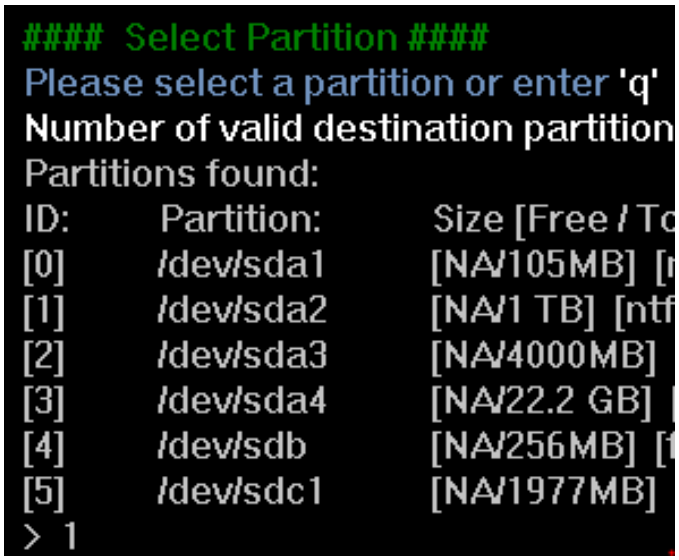


Figura 98. Pasos en el clonado de una memoria USB 4.

Por último, seleccionamos las opciones de copia mediante la alternativa 3. Aquí modificamos el método de Checksum, que viene por defecto como md5 por sha256. Para ello, indicamos 1.

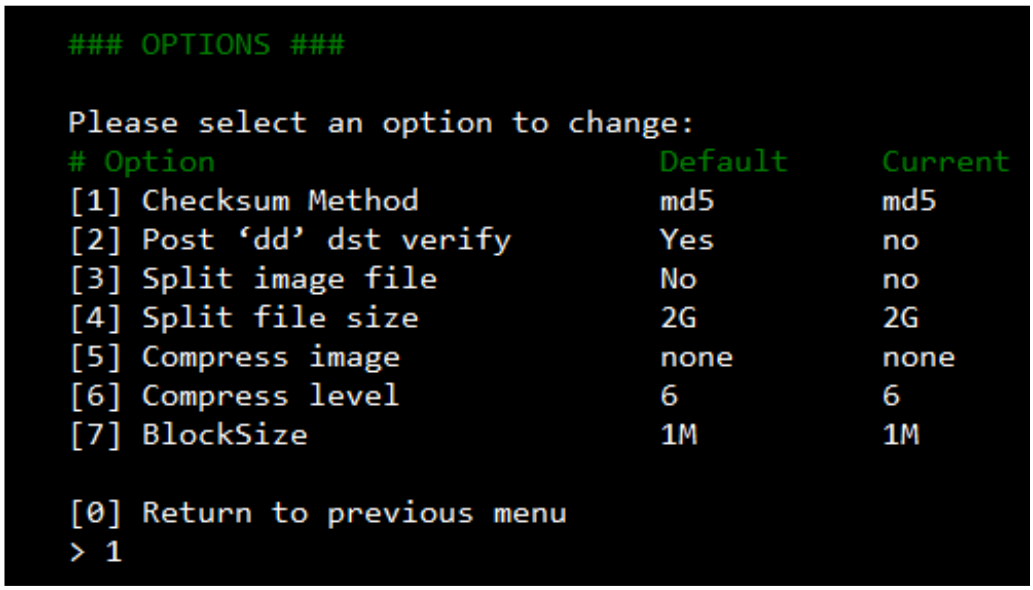


Figura 99. Pasos en el clonado de una memoria USB 5.

Y en el siguiente menú, seleccionamos 3:

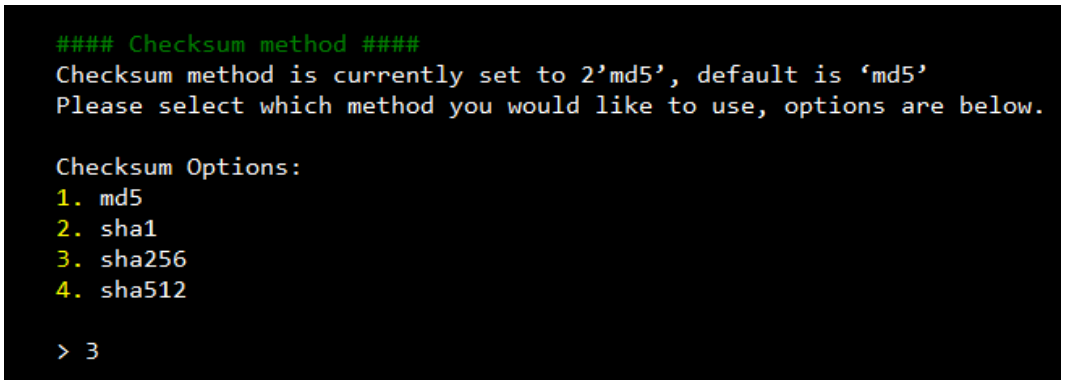


Figura 100. Pasos en el clonado de una memoria USB 6.

La razón de ello fue garantizar seguridad, dado que en los últimos años han quedado patentes las colisiones producidas por md5. Por último, al regresar al menú de nuevo, marcando 0, elegimos la opción 9, para que se realizase el clonado de la imagen.

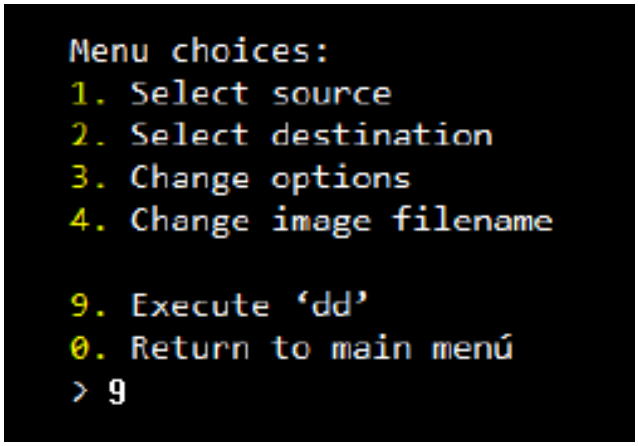


Figura 101. Pasos en el clonado de una memoria USB 7.

Al terminar el proceso, pudimos observar el hash de la imagen. Para finalizar, elegimos 9 en el menú inicial (el primero que nos apareció). Y finalmente tuvimos nuestra imagen:

Figura 102. Aspecto de una imagen de una memoria USB.

Una vez que habíamos terminado con este proceso de clonado. Tras haber creado el Live Flash USB la memoria quedó inutilizable, para recuperarla utilizamos el terminal del sistema, y pusimos DISKPART, que nos abrió un nuevo terminal:

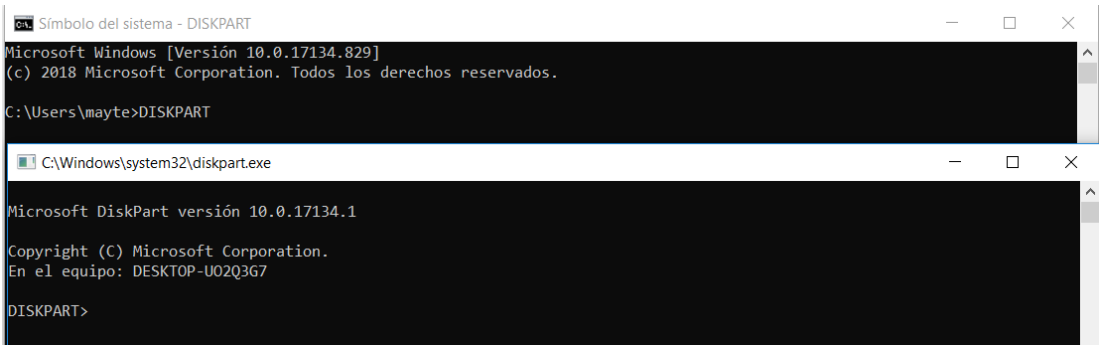


Figura 103. Recuperación de una memoria tras convertirla en un Live Flash USB 1.

Indicamos que nos mostrase la lista de discos:

```
DISKPART> LIST DISK

  Núm Disco  Estado      Tamaño   Disp     Din  Gpt
  ---------- ----------  -------  -------  ---  ---
  Disco 0    En línea        931 GB   851 MB        *
  Disco 1    En línea         15 GB    13 GB        *

DISKPART>
```

Figura 104. Recuperación de una memoria tras convertirla en un Live Flash USB 2.

La memoria que queríamos recuperar apareció como disco 1. Así que indicamos SELECT DISK 1:

```
DISKPART> SELECT DISK 1

El disco 1 es ahora el disco seleccionado.

DISKPART>
```

Figura 105. Recuperación de una memoria tras convertirla en un Live Flash USB 3.

Introdujimos clean:

```
DISKPART> CLEAN

DiskPart ha limpiado el disco satisfactoriamente.

DISKPART>
```

Figura 106. Recuperación de una memoria tras convertirla en un Live Flash USB 4.

Creamos partición:

```
DISKPART> CREATE PARTITION PRIMARY

DiskPart ha creado satisfactoriamente la partición especificada.

DISKPART>
```

Figura 107. Recuperación de una memoria tras convertirla en un Live Flash USB 5.

Y solicitamos formateo:

```
DISKPART> FORMAT FS=NTFS

    5 por ciento completado
```

Figura 108. Recuperación de una memoria tras convertirla en un Live Flash USB 6.

Finalmente:

```
DISKPART> FORMAT FS=NTFS

  100 por ciento completado

DiskPart formateó el volumen correctamente.

DISKPART>
```

Figura 109. Recuperación de una memoria tras convertirla en un Live Flash USB 7.

5.2.3. Un reto explosivo

Se entrega un archivo pcap, registrando el tráfico de red de un equipo. Este equipo pertenecerá a un laboratorio de explosivos, ya que uno de los químicos que trabaja en el laboratorio, ha estado diseñando un explosivo altamente peligroso. La idea sería descubrir, que ingredientes componen dicho explosivo (pues no queda ningún rastro de esta información de forma física, pero sí a través de la web), ya que ha subido este documento a la red, y a quién se lo quiere vender, empezando por localizar el lugar de quedada con el comprador.

En los mensajes de correo se dan pista para encontrar la localización de la reunión, y habrá un archivo que recuperar que nos proporcionará la parte correspondiente al tema de los componentes.

La ficha de descripción para este reto es:

Nombre: Un reto explosivo.

Historia:

Uno de los químicos del laboratorio de explosivos más grandes del mundo ha huido con la intención de comercializar un explosivo que ha diseñado. Este es muy peligroso, ni siquiera se conocen los componentes que va a utilizar en él así como el lugar donde se reunirá con el comprador.

Sería genial poder encontrar el listado de elementos encontrados y el punto de encuentro entre ambos.

Categoría: Forense.

Nivel: Medio.

Descripción:

Se proporcionará un archivo pcap con el tráfico de red registrado para el equipo del doctor, y de este podremos extraer todo lo que necesitamos.

Bandera: componentes del explosivo y punto de encuentro.

Pistas:

- Se debe revisar cada mínimo detalle.
- Lo mejor para encontrar lo que se está buscando, es probar todas las posibilidades que se nos ocurran.

El archivo pcap lo obtuvimos desde la página oficial de Wireshark, entre sus ejemplos. Nos decantamos por uno con intercambio de paquetes IMAP (correspondientes al correo), pero con una dirección IP que no sería válida si queríamos hacer la parte de subir el documento con los componentes. Entonces, modificamos la IP 131.151.32.21 por 192.168.1.6. Realizamos estos cambios, como ya habíamos hecho para el reto 2.

El siguiente paso, fue incluir entre los mensajes las pistas que nos conducirían a encontrar el fichero con el punto de encuentro, con el comprador. Las palabras claves serían: “festivus”, “mezcla”, “blog” y “google”, “pass” y “donde?”.

También eliminamos algunos paquetes haciendo uso de editcap.

El archivo había que subirlo con una página http nuevamente, igual que cuando descargábamos. Además, era necesario ver la dirección ip que se utilizaba en este archivo y la fecha en la que tuvo lugar, ya que estos elementos deberían ser los mismos.

La dirección IP era la 192.168.1.6 y la fecha databa en el 11 de Noviembre de 1999, a las 11 menos cinco de la noche. Siendo la del último paquete guardado, está

misma. Por lo que establecimos nuestro ordenador en función de estos dos valores, como se hizo en el reto 2.

Una vez realizados los cambios, subimos el documento. Para poder conseguir obtener el registro de que se había subido a internet, utilizamos la página ge.tt nuevamente, y pusimos al wireshark a capturar solo los paquetes http.

Obtuvimos los paquetes que nos interesaban como hicimos con los retos anteriores, teniendo cuidado de no quitar paquetes de más. Utilizamos la misma herramienta que antes, editcap, fuimos quitando los paquetes muy despacio, comprobando en todo momento, que la trama de la subida permanecía intacta.

Para unir nuestros ficheros, nuevamente, utilizamos la opción mergecap.

❖ Preparación del documento que contendría los componentes del explosivo:

Construimos un documento, que habría que recuperar del archivo pcap, no porque se lo hubiese descargado el usuario sino que había sido posteado en la red por él. Para complicar el tema de los ingredientes, se creó un documento de texto, con información oculta. Primero creamos un listado de explosivos que formarían la bomba diseñada. Para guiarnos en el conocimiento de los mismos, consultamos la información proporcionada por páginas como Wikipedia. El listado se sacó a partir de posibles combinaciones de explosivos, pero su mezcla no tiene ningún sentido, por lo tanto es totalmente ficticio. Después, procedimos a ocultar algunos de esos ítems, como ya habíamos mencionado más arriba. Para ello, marcamos las líneas que queríamos ocultar e hicimos clic derecho, y elegimos la opción Fuente:

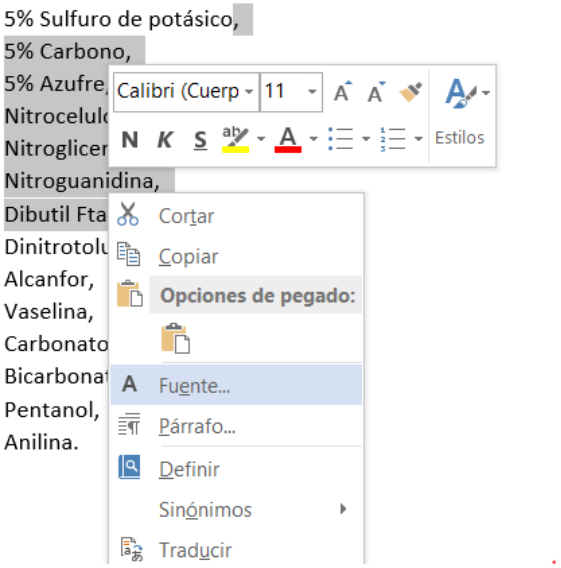


Figura 110. Ocultar texto en un documento docx 1.

Como podemos ver también marcamos la coma, para así dar una pequeña pista a aquellos que estarían resolviendo el reto. Al pulsar sobre la opción anterior, nos aparecía una nueva ventana en la que deberíamos marcar la casilla Oculto, y pulsar Aceptar para que se aplicasen los cambios:

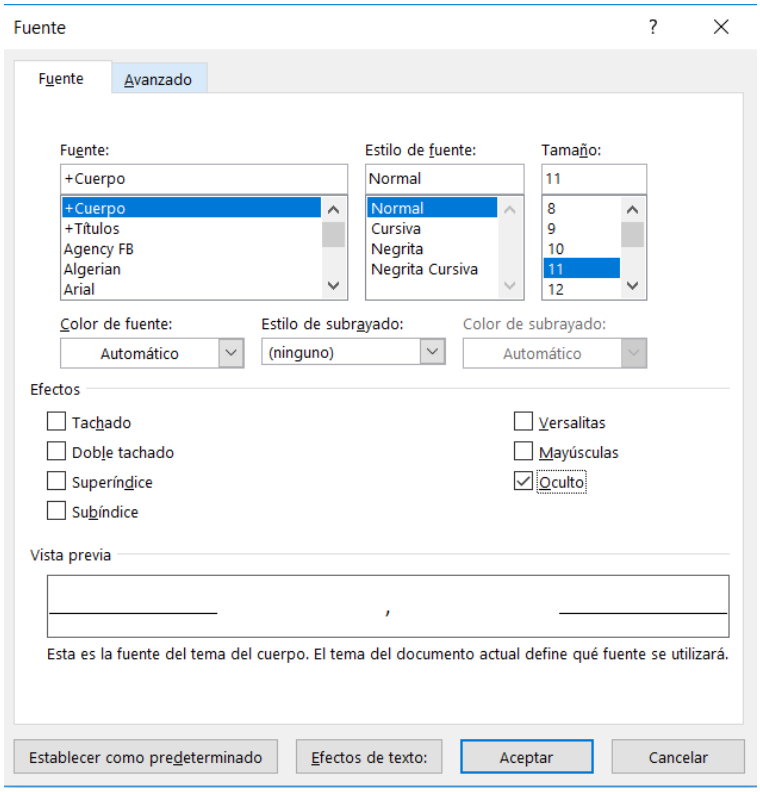


Figura 111. Ocultar texto en un documento docx 2.

Una vez hecho, el texto ya no se mostraba en el Word. Nuevamente modificamos los metadatos para evitar que se pudiese mostrar algún tipo de información. Además, con la intención de complicar un poco más el reto, se pensó en que el documento estuviese dañado (lo haríamos con ayuda de un archivo txt). El problema es que no se podía recuperar ya que una vez que se abre un Word con un fichero de texto y se modifica, entonces se elimina tanto el formato como el texto. Así que otra opción, era modificar la cabecera (la cambiamos por una de un archivo pdf así como el pie del documento). Borramos también la primera parte de la cabecera y donde aparecía algo de content-types.xml.

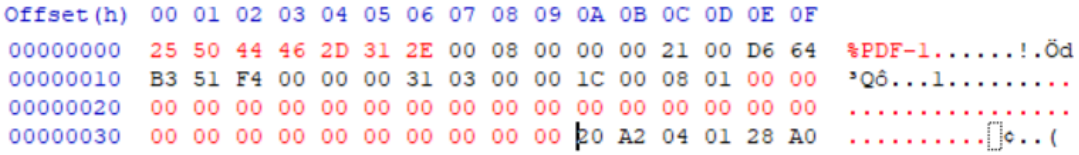


Figura 112. Convertir cabecera en una de un archivo PDF.

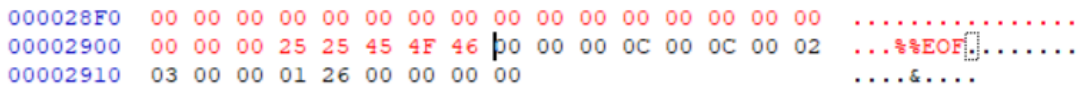


Figura 113. Convertir pie en uno de un archivo PDF.

Estos cambios, nos mostraban el aviso, y sería necesario reparar el documento para poder visualizarlo nuevamente. Aunque no se modificase la cabecera y pie, se podrá abrir el documento y ver el contenido, solo cambiando la extensión. Continuando con este paso, después, se cambió la extensión a pdf.

❖ Preparación de la pista que nos conduce al archivo del lugar de encuentro:

Durante los mensajes del correo registrados en wireshark, habíamos enviado varias palabras claves que nos conducían a esta parte. Estas eran “Festivus” y “blog”. Resulta que existen varias búsquedas en google que provocan respuestas de lo más curiosas, como por ejemplo, si se pone “la respuesta a la vida y al universo”, google te devuelve que es 42; y así muchas más. Entre todas estas respuestas interesantes está la de la búsqueda “Festivus”.

La idea, era que cuando el usuario realizase esta búsqueda pudiese ver la fecha del 23 de diciembre (que es cuando se celebra), y esto era necesario, porque creamos un blog (de ahí esa palabra en el correo) que se llamaba 23 de Diciembre. Y en este colocamos el enlace que nos permitiría descargar el archivo del punto de encuentro. Este enlace requeriría de una contraseña, la cual también se indicaba a través del correo, “dónde?”.

Para la creación de dicho blog utilizamos Blogger. Al entrar en la página, pulsamos sobre Crea tu blog, y lo creamos con este nombre y esta dirección:

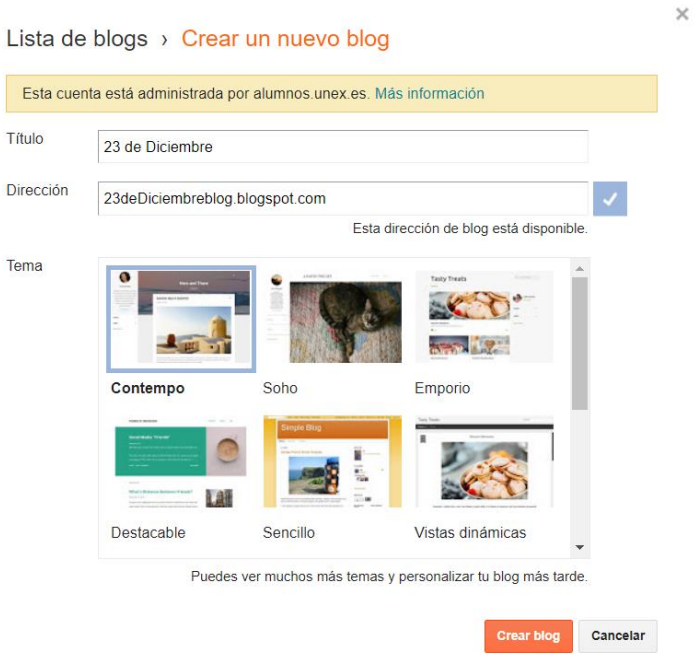


Figura 114. Creación de un blog con Blogger.

Cuando ya habíamos creado nuestro blog, creamos también las distintas páginas y elementos mediante el menú situado en el lateral. Así que creamos una página home y otra llamada Festivus. La página Festivus presentaría un mensaje sencillo y una forma más o menos así:

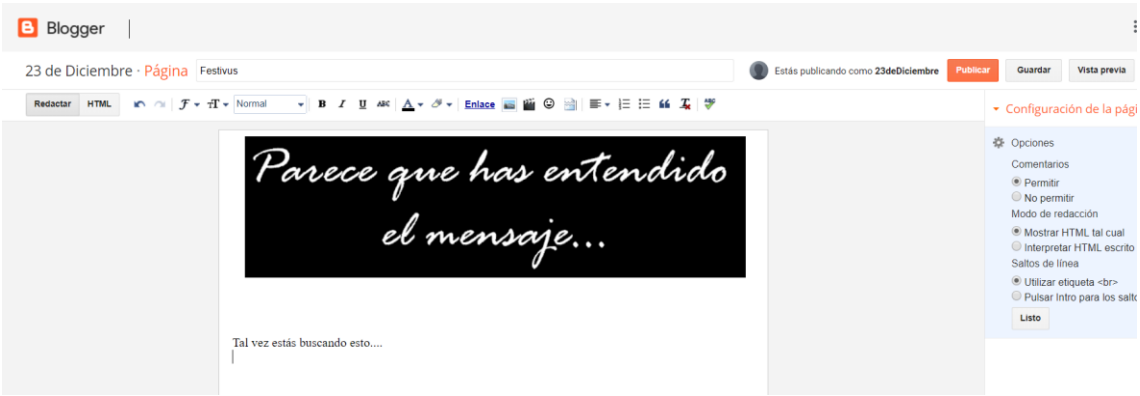


Figura 115. Forma de una página creada con Blogger.

Después, incluimos un enlace al archivo, pero este debía exigir una contraseña para poder descargar. Para hacer esta parte, creamos una entrada, con un código para gestionar esto, lo hicimos en la vista del html:

Figura 116. Código para establecer contraseña a una entrada.

Justo debajo del código, incluimos el texto que queríamos ocultar, es decir, el enlace: http://cort.as/-JvOf. Se comprueba la contraseña en if (getin!="donde?"), y en caso de fallo, nos redirecciona a otra entrada que tendrá una imagen de error. Si es correcta nos mostrará un mensaje indicándolo y se podrá ver el enlace.

Pero además, debíamos incluir esta entrada dentro de la página de Festivus. Esto simplemente lo hicimos con un enlace a la misma.

- Preparación del archivo con el punto de encuentro:

El enlace anterior nos descargaría un zip, que contendría la última parte del reto, la cual se describirá a continuación.

La ubicación real estaría en un archivo kml. Para construir este archivo seguimos los siguientes pasos. En primer lugar, buscamos algún descampado a las afuera de Madrid, por lo que finalmente nos decantamos por el Descampado de Valderribas.

El siguiente paso era crear el fichero kml. Por lo que, abrimos Google Earth (la versión para PC).

Después, buscamos el lugar que queríamos de encuentro. Simplemente, utilizamos el buscador situado a la izquierda:

Figura 117. Busqueda de un lugar con Google Earth.

Una vez localizado, creamos el archivo. Primero tendríamos que crear el punto que queríamos que este englobase. Creamos una ubicación con la herramienta situada en la parte superior:

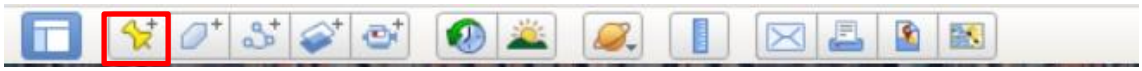


Figura 118. Tablas de herramientas de Google Earth.

Y marcamos el lugar, situando la chincheta sobre el mismo, al final obtuvimos algo así:

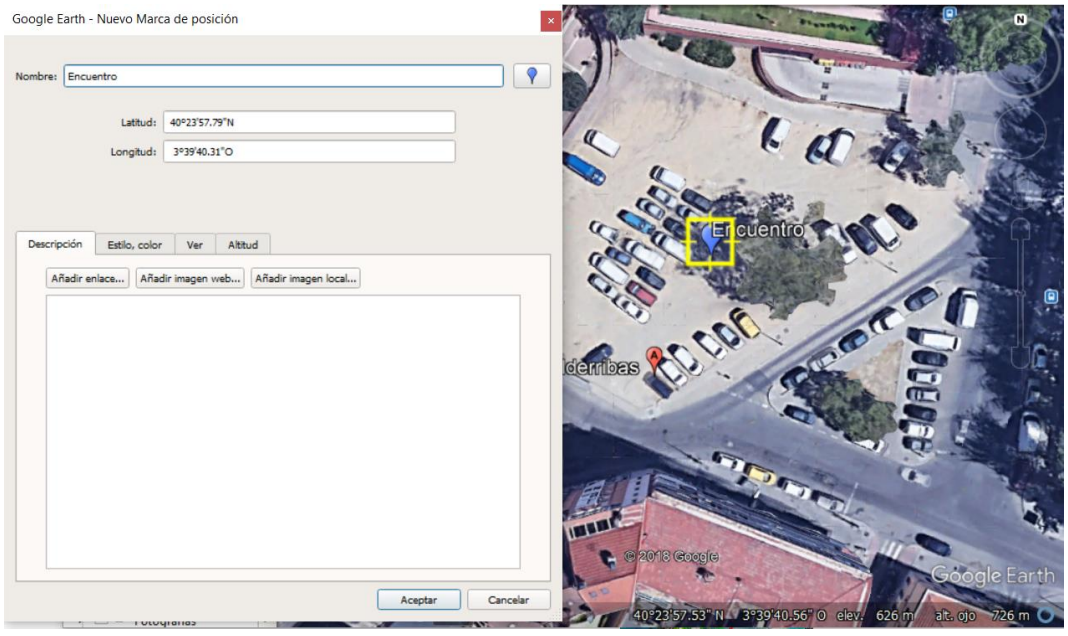


Figura 119. Marcar un lugar con Google Earth.

También, se incluyó una imagen para que fuese más intuitivo. Ese lugar creado se nos situó a la izquierda, en la jerarquía de sitios. Nos colocamos sobre él, dimos clic derecho y guardar sitio como:

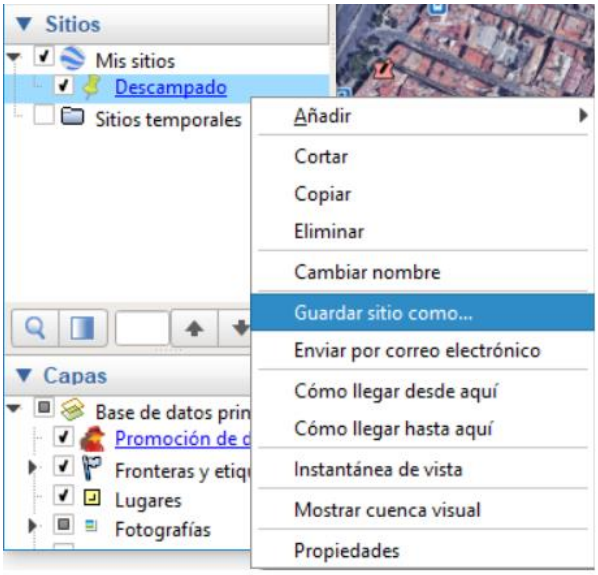


Figura 120. Guardar un lugar como archivo KML con Google Earth.

Y finalmente guardamos como un archivo kml. Cuando ya lo teníamos, le fuimos añadiendo algunos elementos como áreas, rutas y así aumentamos el tamaño del mismo. Ya que en un principio pesaba 2kb, conseguimos que aumentase hasta 8 kb, para poder fragmentarlo (y complicar un poco más todo esto). Fragmentamos este archivo en varias partes, las cuales tendría que volver a unir el participante para poder llegar hasta este punto establecido.

Necesitábamos realizar este paso con ayuda de un programa, en este caso escogimos HjSplit. Cuando ya teníamos nuestro programa abierto, se nos presentaron varias opciones.

Figura 121. Opciones proporcionadas por la herramienta HJ-Split.

Para dividir, elegimos la opción Split. Señalamos el archivo que queríamos dividir, el lugar donde queríamos que se guardasen las partes de la división y el tamaño de cada una de ellas (esto se tiene que hacer en función del peso que tenga el archivo y el número de partes que se quieran). Aquí se llevó a cabo una división de 8 partes de los 8kb, a 1kb:

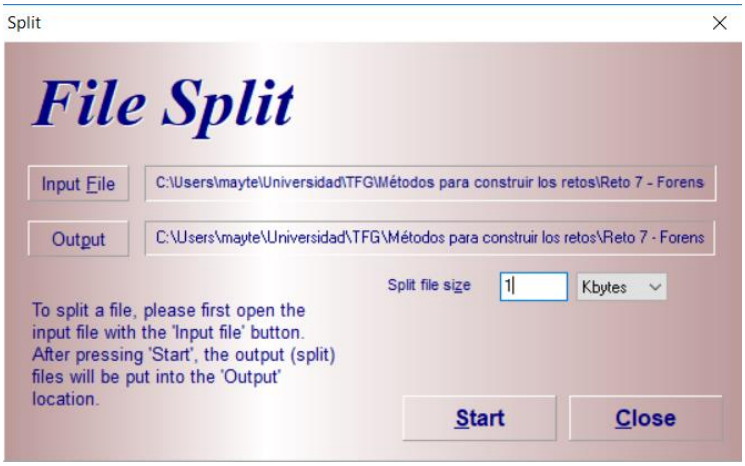


Figura 122. Dividir un archivo en varias partes con HJ-Split.

Pulsamos sobre ‘Start’ y esperamos a que el proceso se realizase completamente. Y ya teníamos las 8 partes (a las que posteriormente borramos su extensión).

Una vez que ya teníamos las partes, creamos un archivo zip que las contendría a todas. Este archivo lo subimos a drive y obtuvimos un enlace para él, de descarga directa (esto ya lo hicimos en retos anteriores).

5.3. RECONOCIMIENTO

En primer lugar, es importante comentar que nos encontramos ante una categoría de la que no existe mucha información. Pero podemos afirmar que entendemos el reconocimiento como la búsqueda de información sobre objetivos fuera del objetivo en sí. Aunque es muy útil en el mundo real, esta técnica generalmente no se usa mucho en CTF, pues es difícil de crear y limitar el tema del CTF.

Con la intención de aclarar un poco más este concepto, ya que al ser tan inusual puede ser un poco difícil de comprender. Podemos definirlo como una técnica que requiere de un amplio conjunto de herramientas, ya que el recorrido que se debe seguir para la reunión de información o para capturar la bandera en el caso de un reto, es muy extenso y puede abarcar cualquier punto dentro de la red.

En el caso, del desafío creado en el presente documento, se puede encontrar un conjunto de banderas repartidas en diversos puntos de Internet y se hace necesario el uso de distintas herramientas open source para poder ir recuperándolas y avanzar en él. Esto se detallará aún más en el apartado que viene a continuación.

5.3.1. Todo lo que ofreció

Se proporciona una archivo pcap. Al analizar el tráfico del mismo debemos descubrir un correo. El correo de esta persona nos conducirá a una cuenta en una red social (primera bandera). A través de dicha cuenta podremos recopilar información sobre esta persona, el lugar en el que reside o está de vacaciones (segunda bandera). En una de sus imágenes habrá un mensaje cifrado, este proporcionará una clave (tercera bandera). Esta clave, no permitirá, entrar en su cuenta de github, y a partir de ahí recuperaremos su número de teléfono (cuarta bandera).

Además, en esta cuenta habrá un documento, que no podremos abrir puesto que su formato no es correcto, en realidad se tratará de una imagen, que contendrá una dirección url oculta y que podremos obtener a través de un lector hexadecimal (quinta bandera). Al introducir esa url en un navegador nos descargará un pdf que contendrá toda la información sobre lo que el tipo ha estado filtrando (sexta bandera). Finalmente, habrá que localizar el nombre de la persona a la que se le ha estado proporcionando la información (séptima bandera).

La ficha de descripción:

Nombre: Todo lo que ofreció.

Historia:

Se sospecha que una persona ha estado filtrando información de una empresa, pero se desconoce de quién se trata ya que el correo que ha estado utilizando para ello es falso. Sin embargo, puede que no haya sido tan listo, y algo se le haya pasado por alto, ¿podrías encontrarlo?

Categoría: Reconocimiento.

Nivel: Medio.

Descripción:

Proporcionamos un archivo pcap de wireshark, que ocultará el intercambio de mensajes y algunos archivos. Existen varias banderas que localizar: algún nombre o nombre de usuario que identifique al sujeto, dónde se está alojando (ubicación), contraseña, número de teléfono, información que ha estado traspasando y nombre del receptor de la misma.

Bandera: @Miguel81335041, 47.388307, 8.513936, Xw8x0Xq2EL8gzpV, 654734040, http://cort.as/-IHEb, el documento y Antonio Ruíz Castro.

Pistas:

- Todas las pistas son proporcionadas, solo hay que saber buscarlas.

Necesitábamos proporcionar un archivo wireshark con el intercambio de varios mensajes, y entre ellos, el de un documento. Al igual que hicimos con el reto 2, de Esteganografía y el reto 3 de Forense. También descargamos un fichero ya preparado y lo modificamos para que contuviese lo que quisiésemos.

Descargamos uno de los ficheros que incluía mensajes del protocolo SMTP O IMAP, el protocolo utilizado en el correo electrónico, para que de esta manera nos recogiese varios mensajes. Al igual, que hicimos con el reto 2, utilizamos las funciones de wireshark para modificar el fichero y añadirle la información.

Después, incluimos la recuperación del documento inicial con una captura hecha con nuestro wireshark. Comprobando la IP y fecha utilizada como hicimos en retos anteriores. La dirección IP era 192.168.0.4 (pero nosotros pusimos 192.168.1.4, modificamos ese 0 para que se mantuviese nuestra puerta de enlace, ya que sino el servidor DNS no funcionaría) y la fecha era del 22 de agosto de 2013, a las 9 y

diecisiete de la noche. Por lo que establecimos nuestro ordenador en función de estos dos valores (exactamente igual que en retos anteriores).

Nuevamente subimos el documento a Ge.tt y lo descargamos de ahí, para que así quedase registrado en la captura. En esta, tratamos de reducir el número de paquetes, pero con cuidado de no quitar ninguno que afectase al documento.

Finalmente, unimos ambos ficheros mediante la función mergecap.

- Documento intercambiado, contenido:

Para crear un documento que tuviese sentido, hicimos lo siguiente. Ya que el usuario estaba proporcionando la información correspondiente de la empresa, y queríamos dar a entender que lo hacía a través del correo, simulamos que hasta el momento lo había estado haciendo con un correo falso y por ello no podían encontrarlo. Quedaría registrado el correo real en los metadatos de dicho documento.

Entonces en ese Word proporcionamos un poco de información de la empresa, sobre algún trabajo o proyecto pendiente (unos planos).

Modificamos los metadatos del documento con la herramienta utilizada en el primer reto, MetadataTouch. Entonces abrimos el programa y cargamos el documento al que queríamos modificar sus metadatos. Antes de nada, fue necesario crear la cuenta de correo para este personaje ficticio. Por ello, fuimos a Gmail, y creamos una nueva cuenta: migueloz.1231@gmail.com, esa es la dirección.

- Construcción de la primera bandera:

Hicimos una cuenta en twitter. El nombre de usuario sería la bandera, en este caso (**@Miguel81335041**).

Incluimos varios mensajes como que hemos tenido que volver a crear la cuenta, algunas fotos para la segunda bandera (el hotel, unos chocolates, un zoo), algún comentario sobre github y un código de barras para la tercera bandera. Se explicarán a continuación.

- Construcción de la segunda bandera:

La segunda bandera consistió en la localización donde se encontraba, en concreto, la geolocalización. Por ello, en esa cuenta creada, subimos algunas imágenes de un país (en esta caso, Suiza, de ahí la pista del chocolate), colocamos

alguna imagen de un hotel, de la que posteriormente podríamos obtener sus coordenadas en google maps o a través de sus metadatos.

Buscamos un hotel en Suiza, más concretamente en la ciudad de Zúrich, pero que no tuviese un nombre que le delatase como tal. Pillamos sus coordenadas desde google maps: **47.388307, 8.513936**, y se las incluimos a través de la página utilizada en el primer reto.

- Construcción de la tercera bandera:

La tercera bandera era la contraseña de la cuenta de github, la cual ya habíamos creado con anterioridad (se explicará a continuación). Ocultamos dicha contraseña en un código de barras. Pusimos **Xw8x0Xq2EL8gzpV,** que había sido generada de forma aleatoria. Para que tuviese sentido subir este código de barras, creamos un vale regalo, al que se lo incluimos.

- Construcción de la cuarta bandera:

Para esta cuarta bandera, creamos una cuenta en github con la intención de almacenar un número de teléfono en la misma (que sería la bandera en cuestión).

A la hora de crear la cuenta, generaremos la contraseña como una cadena aleatoria y la creamos con el correo anterior.

Para poder añadir el número de teléfono (originado de forma aleatoria) creamos un repositorio privado, ya que de esta forma solo podría verse dicho número si hubieses iniciado sesión o si tuvieses acceso al repositorio. Lo incluimos en la descripción del mismo:

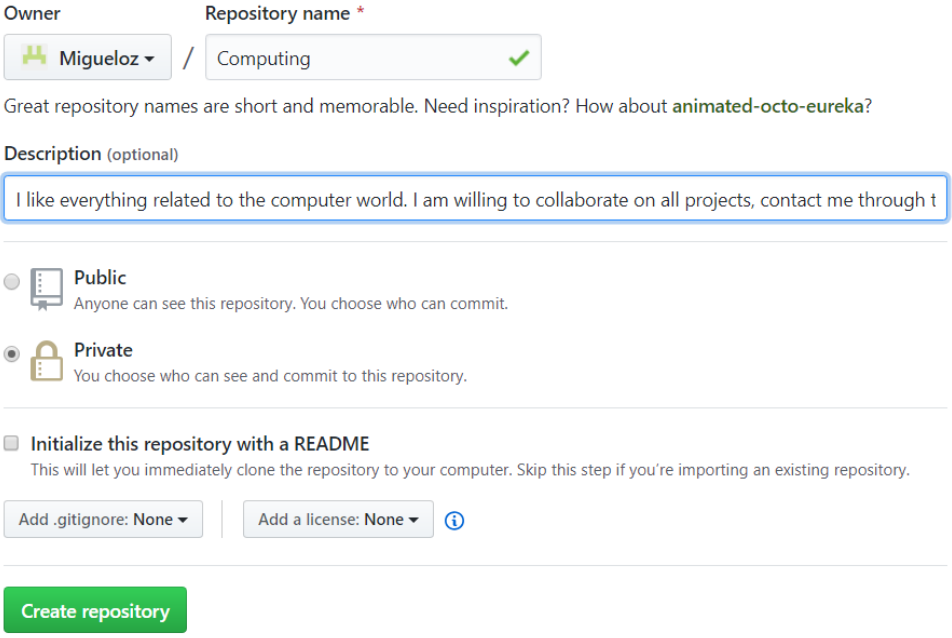


Figura 123. Creación de un repositorio privado en Github.

❖ Construcción de la quinta bandera:

La quinta bandera sería una dirección url. En la misma cuenta de github habría un documento Word, pero este en realidad sería una imagen a la que cambiaríamos su extensión. Ocultamos la url en su código hexadecimal (esto se hizo igual que en los retos anteriores).

Después cambiamos la extensión a la imagen como .docx. Subimos este documento a la cuenta de git, dentro del repositorio privado.

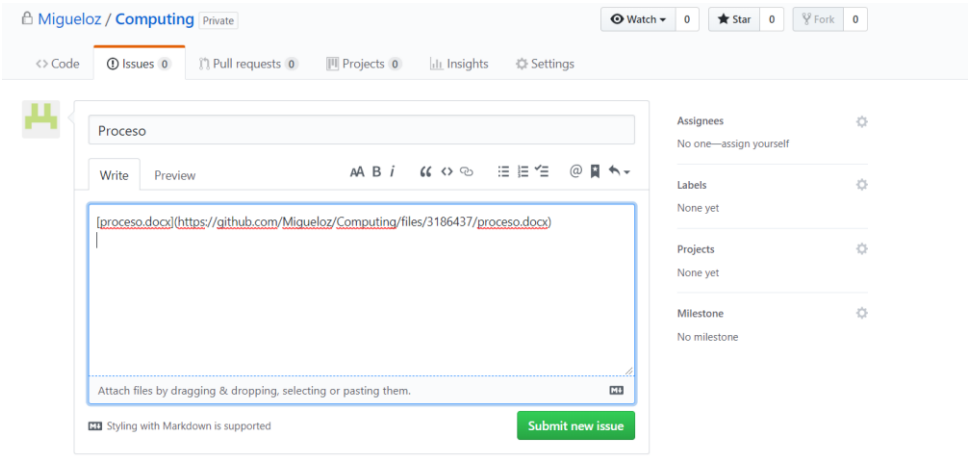


Figura 124. Subir un documento a un repositorio en Github.

❖ Construcción de la sexta bandera:

Para elaborar esta bandera, creamos un documento con información que pudiese pertenecer a una empresa, consultamos alguna página con ejemplos de proyectos empresariales. Y creamos uno con un plan de mercado.

Dicho documento, lo subimos a drive, e hicimos que se descargase de forma automática. Para hacer esto, seguimos los pasos utilizados en los retos anteriores. La url formada es la que hemos ocultado en la imagen anterior.

- Construcción de la séptima bandera:

Bueno como la intención era poder encontrar el nombre de la persona a la que se le había estado traspasando esta información, creamos un correo electrónico para él. La cuenta creada fue xxi.098712@gmail.com.

No le creamos ninguna cuenta en alguna red social. Solo creamos una en git, de esta manera podríamos recoger sus datos también. Los establecimos en su bibliografía.

5.4. PROGRAMACIÓN

No hay mucha información que aportar sobre estos tipos de retos, ya que son muy sencillos, por lo que los describiremos de una forma muy breve. Los retos de programación o PPC (Professional Programming Challenges), se tratan de desafíos en los que se nos presenta alguna tarea compleja que se debe realizar, y se busca hacerla en el menor tiempo posible con ayuda de un código o script que facilite este cometido.

En algunos casos únicamente requieren de un sencillo programa que resuelva un código enrevesado o complejo (donde se pueden tratar aspectos matemáticos como fórmulas o algoritmos), simplemente consistiría en optimizar dicho código, obteniendo una versión mejorada, más sencilla y que requiere un menor tiempo de ejecución.

En otras ocasiones, requieren hacer software más complejo. Son ejemplos de estos, la creación de bots para juegos, solucionadores de captcha (imagen, audio o texto), solucionadores de juegos lógicos (sudoku, monogramas, rompecabezas)…

Para el desarrollo de los mismos se puede hacer uso de innumerables lenguajes de programación como C, Python, Java, PHP… Cualquiera es bueno mientras consigamos alcanzar nuestro objetivo.

En el siguiente apartado conoceremos de qué forma hemos enfocado este tipo de retos en nuestro proyecto.

5.4.1. Una lección de matemáticas

Se ofrece un captcha al usuario que tiene que resolver mediante la programación de un código, es decir, un programa que lo resuelve.

Su ficha de descripción:

Nombre: Una lección de matemáticas.

Historia:

Todos sabemos multiplicar, incluso aunque estos sean números de 10 dígitos, podríamos hacerlo, pero ¿lo haríamos en menos de 10 segundos?

Categoría: Programación.

Nivel: Medio.

Descripción:

Se proporciona una dirección a una página con un captcha, tenemos que resolverlo en menos de 10 segundos, mejor empezar a programar. Esta es la dirección: https://challengeppc.blogspot.com/2019/06/ChallengePCC.html.

Bandera: Habilidad.

Pistas:

- No existen límites para tu creatividad.

Para evitar tener que crear un dominio, y colgar el captcha que habría que resolver, utilizamos Blogger, como hicimos en un reto anterior. Por lo que accedemos a la web y una vez dentro, elegimos la opción de crear blog.

Tuvimos que configurar el blog, para que no mostrase ninguna entrada. Estas solo se verían en el caso de que el usuario tuviese el enlace de la misma. En la pestaña de diseño, modificamos el sidebar (lo dejamos vacío):

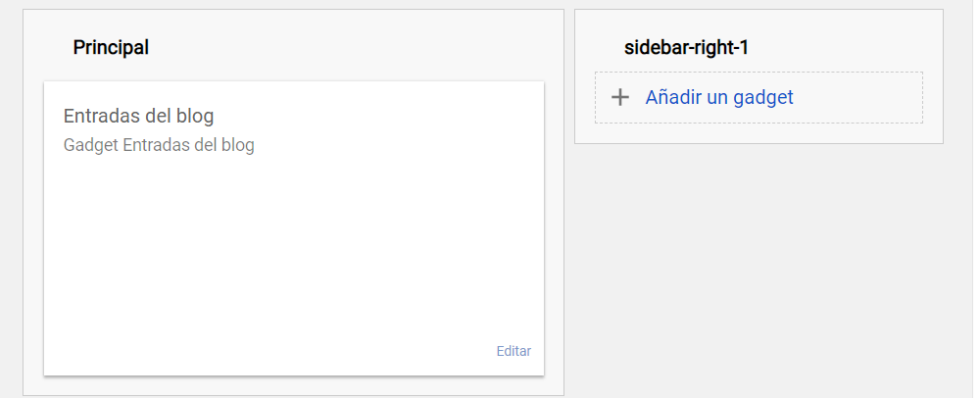


Figura 125. Configurar blog para que no muestre entradas con Blogger.

❖ Construcción del Captcha:

El siguiente paso, sería crear una entrada con el código correspondiente al captcha que habría que resolver. El código fue javascript integrado en html, desde la vista html de la entrada.

Antes de proceder a la explicación de código, algunos detalles correspondientes a la configuración. Modificamos el enlace, estableciendo uno personalizado para que no apareciese el título de la entrada, sino algo como ChallengePPC. Además evitamos que se mostrasen los comentarios.

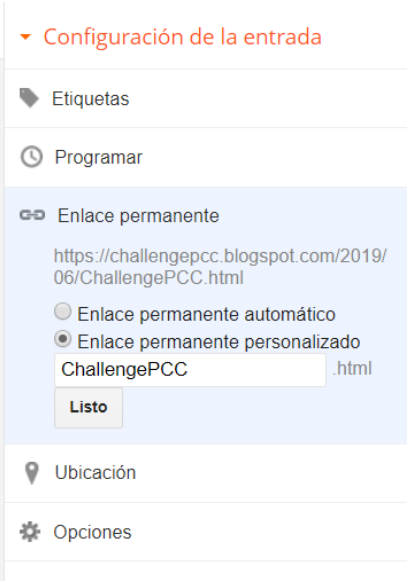


Figura 127. Creación de un enlace personalizada con Blogger.

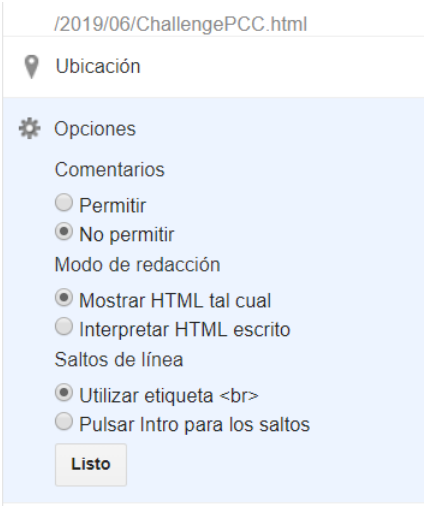


Figura 126. Evitar la publicación de comentarios en una entrada con Blogger.

Con respecto al código, comentamos algunas de las partes más características, el resto solo brevemente. En primer lugar, tenemos la parte correspondiente al css, el aspecto que presenta el captcha, el formulario donde se veía el captcha como tal y donde el usuario tenía que ingresar la respuesta.

```
<!-- START EXAMPLE CAPTCHA FORM -->
<form onsubmit="return checkform(this);" class="formmargin">

<div class="capbox">

<div id="CaptchaDiv"> </div>

<div class="capbox-inner">
Introduce el resultado:<br>

<input type="hidden" id="txtCaptcha">
<input type="text" name="CaptchaInput" id="CaptchaInput" size="15"><br>

</div>
</div>
<br>

<br>

<input type="submit" id="Test Captcha" value="Test Captcha"><br>

</form>
<!-- END EXAMPLE CAPTCHA FORM -->
```

Figura 128. Código del formulario utilizado en el Captcha.

Elegimos un captcha de operaciones matemáticas, donde había que resolver una multiplicación de dos números de gran longitud. Creamos los números de forma aleatoria utilizando la operación Math.random, y los incluimos en el apartado correspondiente al captcha, identificado con el id = txtCaptcha.

```
var a = Math.floor(1000000000 + Math.random() * 9000000000)+ '';
var b = "*"
var c = Math.floor(1000000000 + Math.random() * 9000000000);+ '';


var code = a + b + c;
document.getElementById("txtCaptcha").value = code;
document.getElementById("CaptchaDiv").innerHTML = code;
```

Figura 129. Código para generar números aleatorios de gran longitud.

Para validar el captcha, realizamos la multiplicación, tomamos el valor introducido en el input, y comparamos ambos:

```
// Validate input against the generated number
function ValidCaptcha(){

var str1 = a*c;
var str2 = removeSpaces(document.getElementById('CaptchaInput').value);
if (str1 == str2){
return true;
}else{
return false;
}
}
```

Figura 130. Código para validar el captcha.

Esos valores de true y false, según si el resultado introducido en el captcha era correcto o no, pasaban a ser gestionados aquí:

```
function checkform(theform){
var why = "";


if(theform.CaptchaInput.value == ""){
why += "Por favor, debe introducir el resultado de la operación.\n";
}
if(theform.CaptchaInput.value != ""){
if(ValidCaptcha(theform.CaptchaInput.value) == false){
why += "El resultado introducido no es correcto.\n";
}
if(ValidCaptcha(theform.CaptchaInput.value) == true){
if (tiempo == 1){
//setTimeout(result,00000);
why += "Demasiado lento.\n";
}else{
//setTimeout(result,00000);
why += "Eres un robot.\n";
}
}
}
if(why != ""){
alert(why);
return false;
}
}
```

Figura 131. Código para gestionar los resultados introducidos por el usuario en el Captcha.

Se mostraría una pestaña emergente indicando al usuario distintos mensajes. Si el resultado no era correcto significaría que el método anterior había devuelto false, por lo que mostraría “El resultado introducido no es correcto”.

Si era correcto, podían pasar dos cosas, que el usuario hubiese realizado la operación de forma manual o que hubiese utilizado un programa para hacerlo. Esto sería controlado con la variable tiempo, que actuaría como un booleano. Si su valor es 1, quería decir que lo había hecho el usuario de forma manual y mostraría “Demasiado lento”. En caso contrario, indicaría que es correcto y que somos un robot.

Esta variable tiempo la contralamos mediante la operación setTimeout. La variable se inicializaría a 0, y al pasar 10 segundos desde que se entró en la página, se llamaría a una función que modificaría su valor:

```
/*se la llama a los 10 segundos*/
setTimeout(pasotiempo,10000);

var tiempo = 0;

/*función que sólo cambia un valor*/
function pasotiempo(){
 tiempo = 1;
}
```

Figura 132. Código para controlar el tiempo.

Otro caso controlado, era cuando no se introducía nada, la ventana emergente le anunciaría "Por favor, debe introducir el valor de la operación".

Por último, se resetearían algunos valores, como el correspondiente al cálculo de la operación, mediante la operación removeSpaces:

```
str1 = removeSpaces();

// Remove the spaces from the entered and generated code
function removeSpaces(string){
return string.split(' ').join('');
}
```

Figura 133. Código para resetear todos los valores.

6. RESULTADOS Y DISCUSIÓN

En la siguiente sección se ofrece una solución para los retos planteados y elaborados en el apartado anterior. Además, también se lleva a cabo una pequeña discusión para cada uno, ofreciendo una especie de análisis de lo que puede proporcionarnos dicho reto.

6.1. Una imagen vale más que mil palabras

Procedemos a mostrar la forma en la que se le puede poner solución al reto presentado en el apartado 5.1.1. Una imagen vale más que mil palabras.

Se nos proporcionaba un documento con una imagen, pero esta al analizarla no nos proporcionaba absolutamente nada. Por lo que, consultamos la estructura interna del documento, porque tal y como se indicaba en la descripción del reto, todo lo necesario nos lo proporcionaría este.

Modificamos la extensión de nuestro archivo a .zip. Una vez hecho, ya pudimos extraer las carpetas correspondientes:

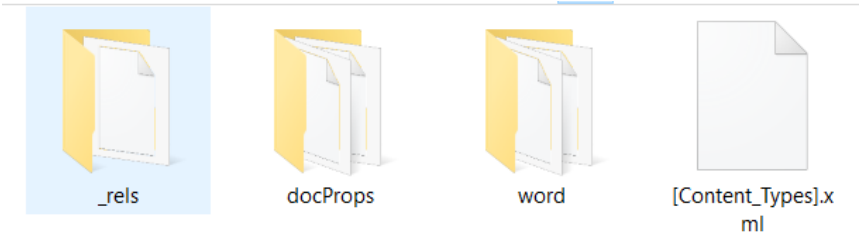


Figura 134. Reto 1: Carpetas contenidas dentro de un documento docx.

Dentro de la carpeta Word, buscamos la carpeta media y existían dos pistas que nos indicaban cual era el siguiente paso que teníamos que realizar. La primera, nos venía en la ficha de descripción "Sería bonito que existiese algo que nos hiciese a todos felices" y una imagen de herramientas de edición. Por lo tanto, necesitábamos utilizar algún programa de edición para modificar la imagen donde aparecían las caras tristes. La imagen smile contenía una url, que solo se podía visualizar modificando las curvas de color de la misma:

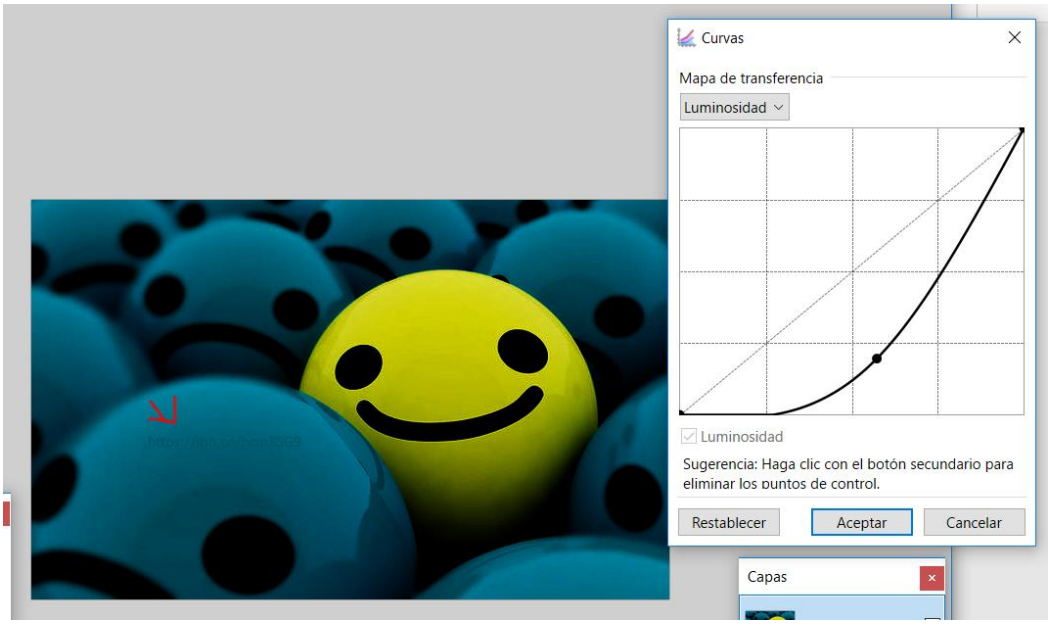


Figura 135. Reto 1: Modificación de las curvas para obtener la url oculta.

Esa dirección nos llevó a una imagen que se trataba de Madrid y el hecho de que se hubiese escogido ese lugar se debía a una razón. Como sabemos, algunos Smartphone al realizar fotos, almacenan la ubicación del lugar en el que se hicieron. Pudimos recuperar estas coordenadas, consultando los metadatos de la imagen:

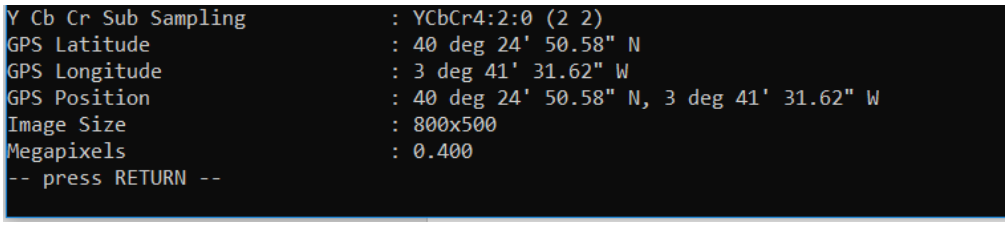


Figura 136. Reto 1: Coordenadas ocultas en una imagen.

Estas coordenadas, nos llevaron al Paseo del Prado.

El reto trataba sobre todo lo que podía ocultar una imagen, y un lugar que está repleto de imágenes, puede ser un museo. Más concretamente, el museo del Prado de Madrid. Esa es la bandera: **PRADO**.

Existen varias cosas que comentar sobre este reto, la primera, es lo realmente divertido que es tanto hacerlo como resolverlo. Es fascinante todo lo que se puede hacer con una imagen y las innumerables herramientas que existen para poder hacerlo. Se puede considerar un reto sencillo, con respecto al nivel de los demás que han sido desarrollados durante el proyecto, aunque no exista ninguna evidencia para poder afirmarlo, debido a que no ha sido presentado en ningún evento ni publicado para que otros participantes puedan resolverlo. Pero los pasos necesarios para hallar la bandera son mucho más breves, que otros de los que hemos descrito.

A través de este desafío, hemos podido trabajar con los metadatos tanto de un documento como de una imagen, conociendo todo lo que estos nos pueden ofrecer, así como la posibilidad de modificarlos, para eliminar un rastro en ellos.

6.2. El mapa de… ¿la bandera?

En el siguiente apartado, mostraremos la solución correspondiente al reto presentado en el apartado 5.1.2. El mapa de… ¿la bandera?

Se nos proporcionaba un archivo wireshark, que se trataba de una captura de una conversación por skype, y podíamos ver algunos de los mensajes que esta incluía. Estos eran pistas:

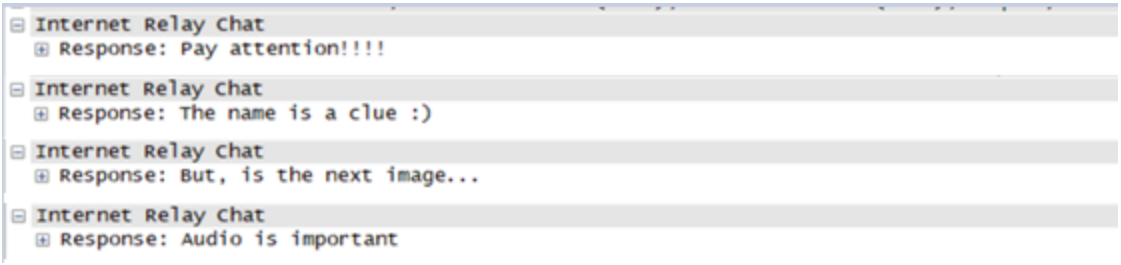


Figura 137. Reto 2: Mensajes localizados entre el tráfico del archivo pcap.

Según la descripción se habían intercambiado varios archivos, así que había que buscar posibles descargas, filtrando paquetes http. Encontramos una imagen y un audio, de los cuales se hablaba en los mensajes. Una vez localizados, los guardamos mediante la opción Export selected packet bytes:

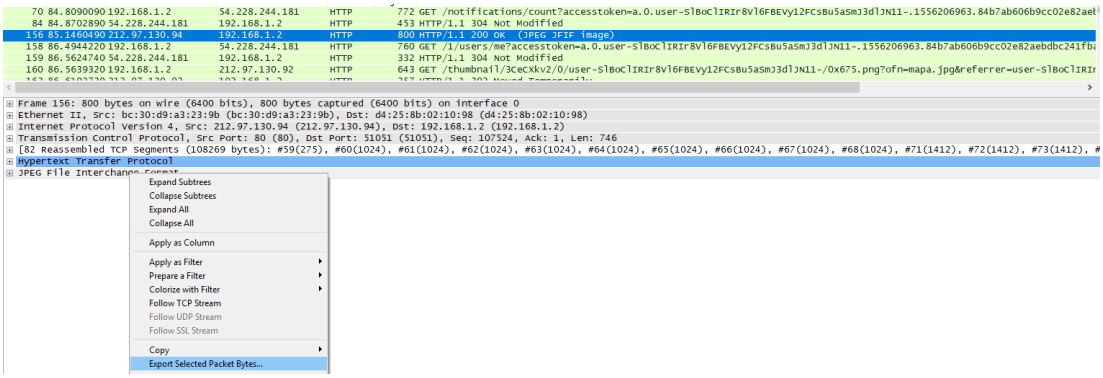


Figura 138. Reto 2: Guardar un archivo mediante la opción Export Select Packet Bytes.

La imagen que encontramos entre el tráfico del archivo pcap, incluía una url que nos conduciría a la siguiente pista. Al abrir dicha imagen con un lector hexadecimal pudimos ver al final de la misma la dirección url almacenada.

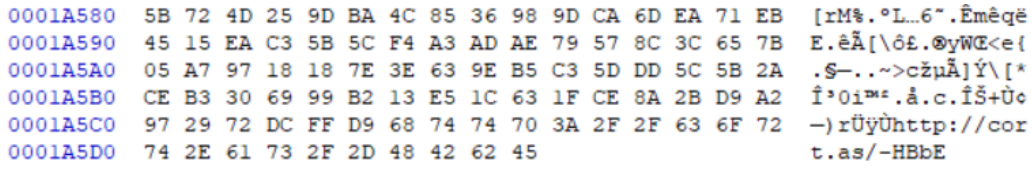


Figura 139. Reto 2: Url oculta en el código hexadecimal de una imagen.

Al introducir dicha dirección en el navegador, nos descargó un archivo zip con otras dos imágenes, una con el nombre cnffjbeq, y la otra solo era una referencia al audio.

Los mensajes anteriores nos habían indicado que el nombre era una pista y que el audio era importante, así como también nos habían hablado de otra imagen (debía ser esta).

Al tratar de abrir la imagen, nos mostraba un error, esto se debía a que le faltaba la cabecera PNG de la misma. Con un lector hexadecimal, se le pudo volver a colocar:

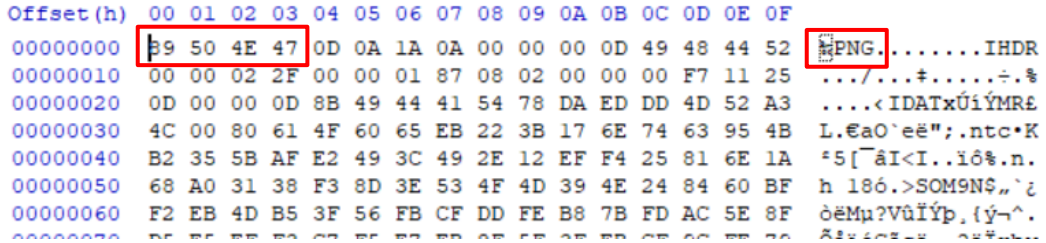


Figura 140. Reto 2: Cabecera PNG de una imagen.

Y ya se pudo abrir la imagen, la cual nos mostró el siguiente texto, que dado su aspecto parecía ser un texto cifrado:

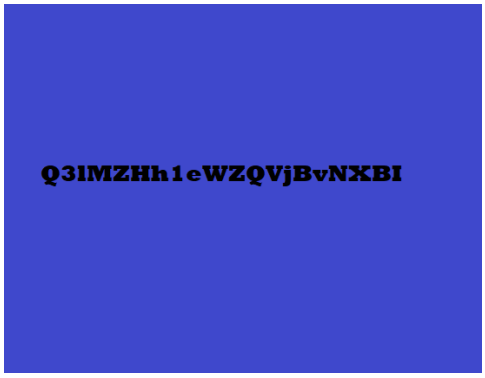


Figura 141. Reto 2: Imagen con texto cifrado en base64.

El tipo de cifrado elegido resultó ser una encriptación en base64, la cual se podía descifrar con ayuda de páginas como superpatanegra.

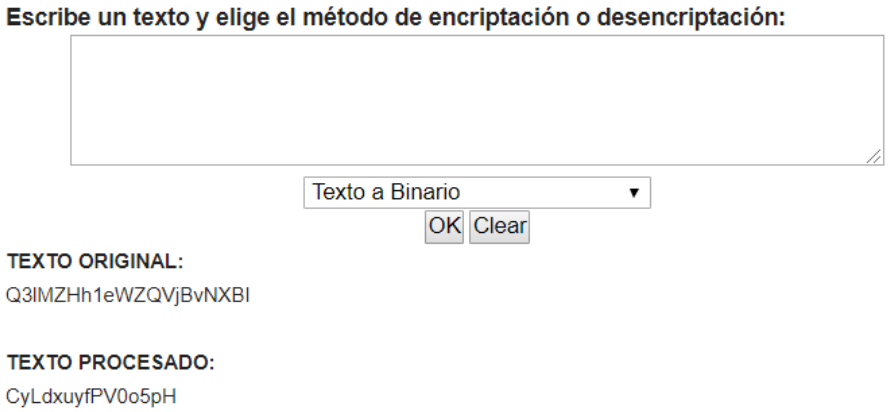


Figura 142. Reto 2: Descifrar texto encriptado en base64 con superpatanegra.

¿Y qué significaba este texto? Para descubrirlo volvimos sobre el audio anterior. Este audio era una canción en alemán que trataba sobre el abecedario.

Utilizamos el programa DeepSound, y pudimos extraer un fichero oculto en la misma.

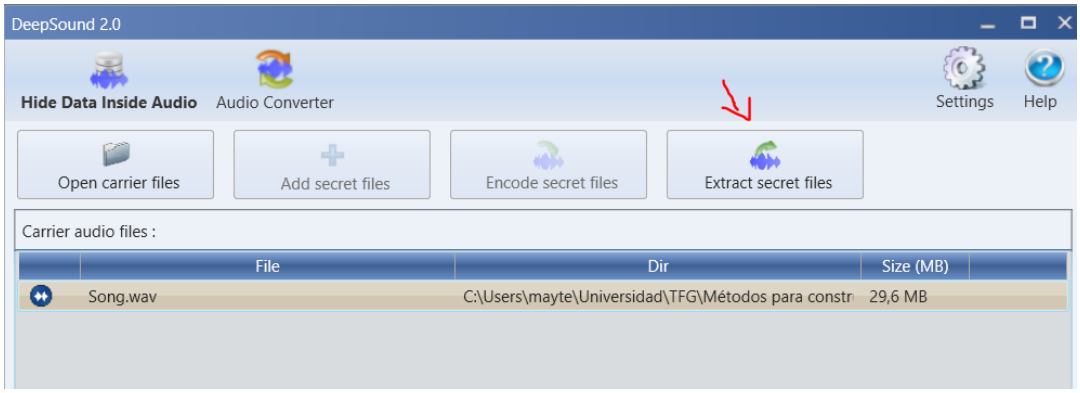


Figura 143. Reto 2: Extracción de archivo oculto en un audio con DeepSound.

El fichero contenido en él se trataba de un txt con el siguiente acertijo: “Por mi núcleo me has de doblar, para encontrar una nueva forma de encriptar”.

El propio nombre de la imagen se trataba de una pista, ya que nos indicaba de que se trataba la cadena que acabábamos de descifrar (CyLdxuyfPV0o5pH).

La canción anterior trataba del abecedario, si doblábamos el abecedario a la mitad de la siguiente forma:

a b c d e f g h i j k l m

n o p q r s t u v w x y z

E íbamos sustituyendo cada letra del nombre de la imagen, por su correspondiente. Si el nombre era cnffjbeq, la c se correspondía con la p, la n con la a, y así sucesivamente. Obtuvimos la palabra password. Por tanto, la cadena CyLdxuyfPV0o5pH, era una contraseña.

La imagen con texto cifrado ocultaba un fichero que pudimos extraer mediante la herramienta openstego. Indicando dicha contraseña.

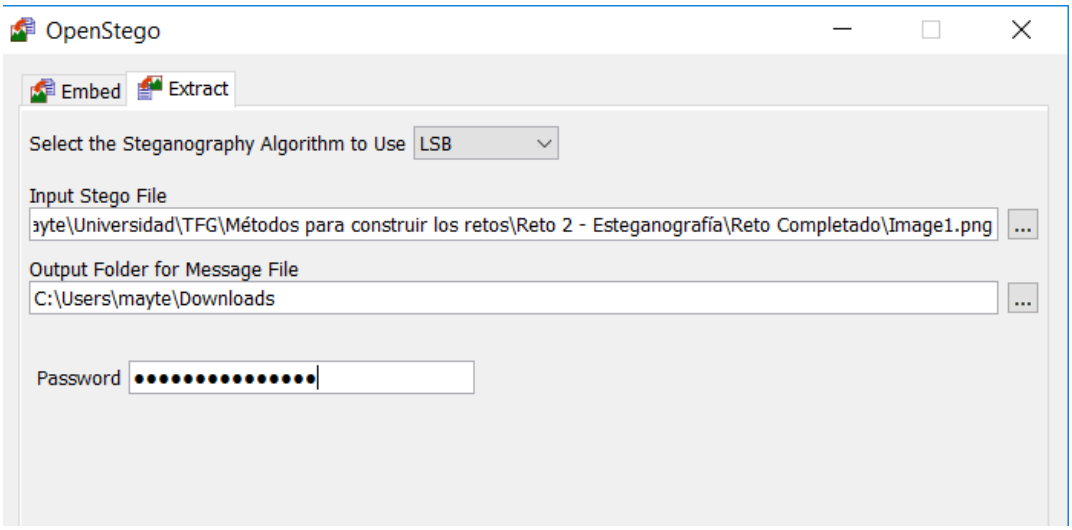


Figura 144. Reto 2: Extracción de archivo oculto en imagen con Openstego.

El fichero resultado nos mostró la bandera pasión.

Mediante este reto, pudimos conocer un poco más de cerca todo lo que un fichero pcap puede ofrecer, es decir, todo el potencial que tiene para albergar archivos, así como la facilidad para poder analizar el tráfico de red.

En el reto anterior solo habíamos conocido como poder ocultar información en imágenes, pero ahora podemos ver que esto también es posible en cualquier otro archivo multimedia, como un audio en este caso. Igualmente, también existen varias herramientas que lo hacen posible, cumpliendo el principal objetivo de la esteganografía, el elemento guardado pasa totalmente desapercibido.

Además, es importante comentar, que ha sido utilizado en las JNIC, y como su ficha de descripción marcaba se trató de un reto difícil. En función de las calificaciones, los resultados obtenidos nos informaron de que diecisiete personas trataron de resolverlo, y ninguna lo logró.

6.3. El mundo multimedia

Durante el desarrollo de este punto, haremos un recorrido por la solución correspondiente al reto presentado en el apartado 5.1.3. El mundo multimedia.

Se nos proporcionaba una imagen inicial. Si analizábamos esta imagen con una herramienta como imagehide. Al cargarla en el programa y darle a Read Data, podíamos ver este enlace:

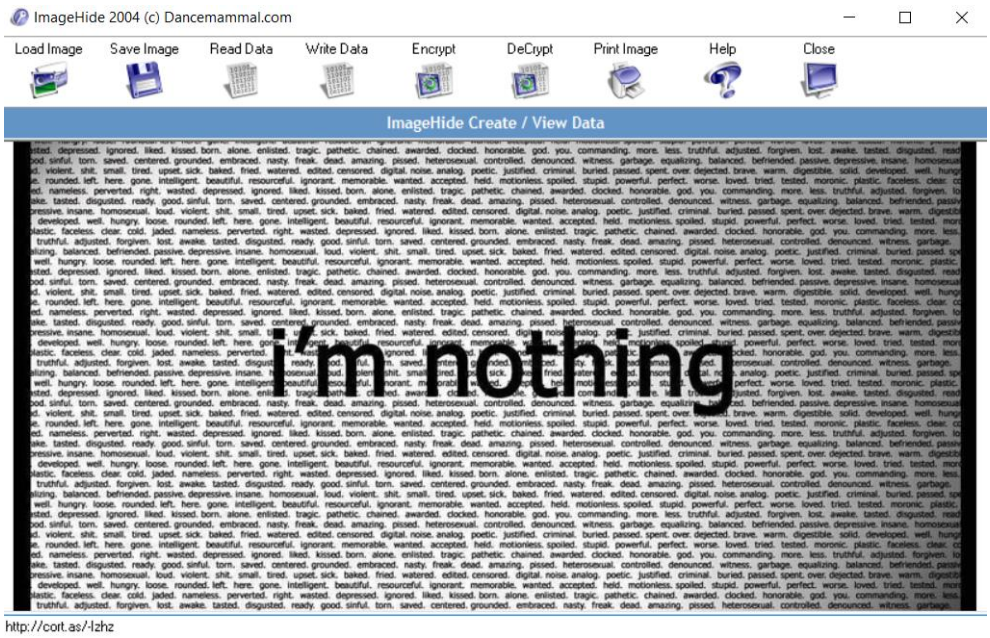


Figura 145. Reto 3: Extracción de mensaje oculto en una imagen con Imagehide.

Al colocar este enlace en el navegador, nos descargaba un archivo zip, que contenía una imagen y un video. Al ver el video podíamos escuchar que el sonido del mismo era un poco raro. Por lo que, al observar el espectrograma de dicho audio pudimos ver esto:

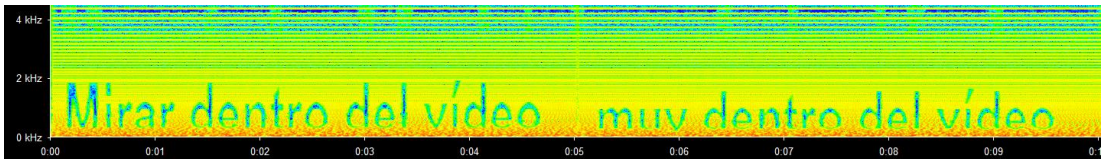


Figura 146. Reto 3: Espectrograma con mensaje secreto en un audio.

Nos indicaba que teníamos que mirar dentro del video, así como las pistas ofrecidas en la descripción, que nos indicaban que debíamos observar los huesos. Por tanto, vimos dicho video con un lector hexadecimal:

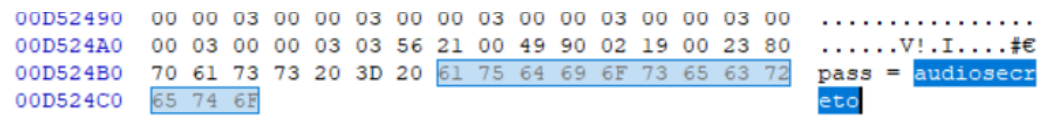


Figura 147. Reto 3: Contraseña oculta en el código hexadecimal de un vídeo.

Nos proporcionaba una contraseña. Había que saber a qué pertenecía dicha contraseña, empezamos a analizar los fotogramas del video, y en uno de ellos vimos un mensaje oculto:

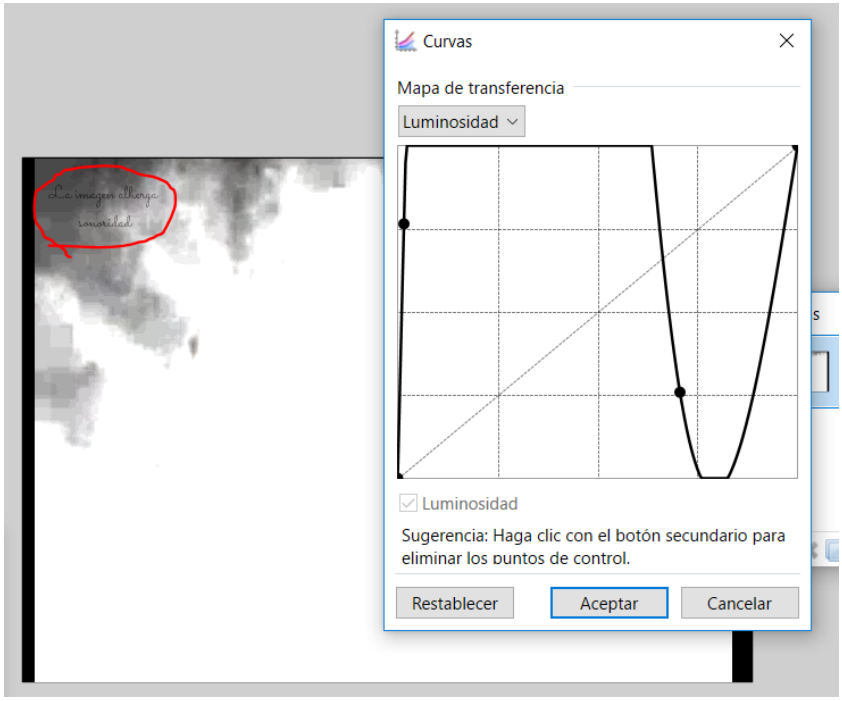


Figura 148. Reto 3: Mensaje oculto en uno de los fotogramas de un vídeo.

Que nos decía “la imagen alberga sonoridad”. Dentro del zip, también estaba otra imagen, esta debía tener oculto un audio en su interior. El cual pudimos extraer mediante la herramienta Openstego.

Al escuchar el audio, oímos varios golpecitos. Observamos el espectrograma de ese audio, y vimos que existía un código morse oculto en su interior:

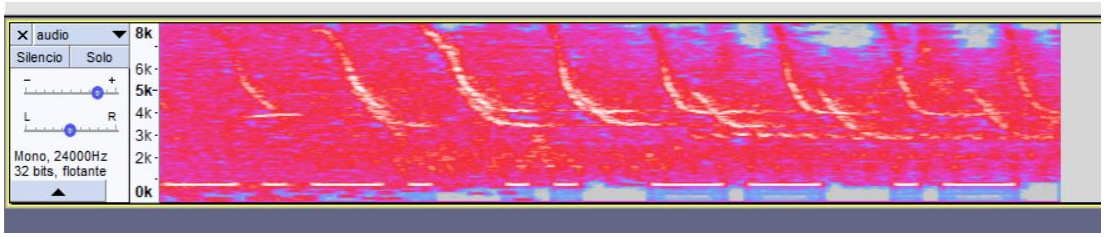


Figura 149. Reto 3: Espectrograma de Audacity donde se visualiza código morse oculto dentro de un audio.

Este código se trataba de la palabra cima, sería la clave para obtener una imagen oculta dentro del audio. La cual pudimos recuperar con el programa DeepSound y haciendo uso de dicha contraseña (cima).

Nos devolvió un codigo QR, con este aspecto:

Figura 150. Reto 3: QR con la bandera oculta en su interior.

Observamos que sus colores estaban invertidos, y con ayuda de Paint pudimos restaurarlo. Leímos dicho código y obtuvimos la bandera **ooSZbSvcjkFCiPZ.**

Con este reto, conseguimos construir una combinación de los dos anteriores, solo que en este caso, incluimos como sería la manipulación con un video.

A partir del mismo, hemos llegado a conocer, que si para los audios e imágenes existen muchas herramientas, para los videos son relativamente escasas. Solo existe una, y no funciona muy bien. De ahí que nos hayamos limitado a editarlo y a modificar su código hexadecimal.

6.4. Podría ser algo más…

En este apartado, pondremos solución al reto expuesto en el apartado 5.2.1. Podría ser algo más.

El siguiente reto se trataba de un caso propuesto a un perito con la intención de localizar a un usuario malicioso que había entrado en un sistema y eliminado algunos documentos. Además, se conocía que habían desaparecido un total de 200000 euros. Se nos proporcionaba una imagen del servidor de la empresa, esta la analizamos con la herramienta Autopsy. Nada más abrirla, pudimos ver:

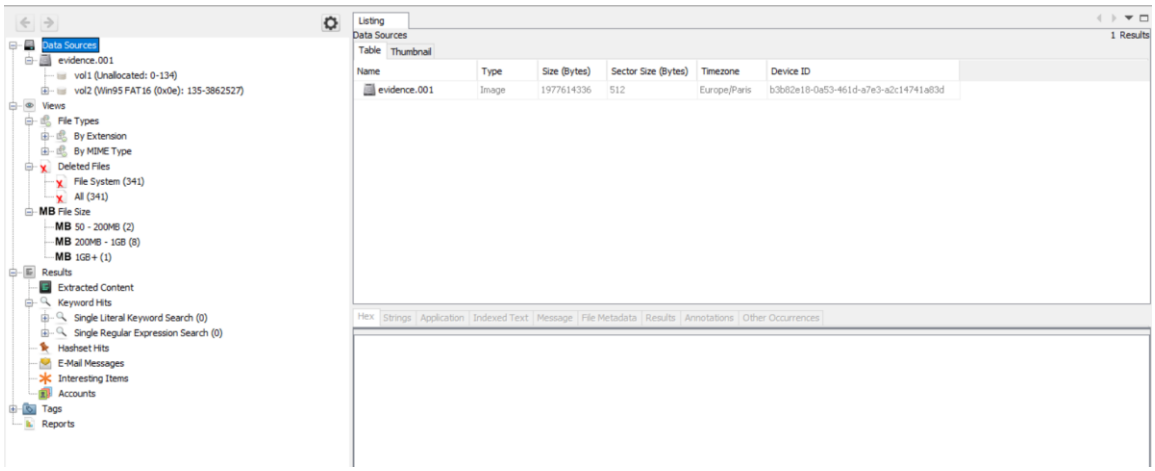


Figura 151. Reto 4: Imagen visualizada desde Audacity.

Al acceder a sus ficheros logs, consultamos el fichero en el que se registran todos los accesos al sistema:

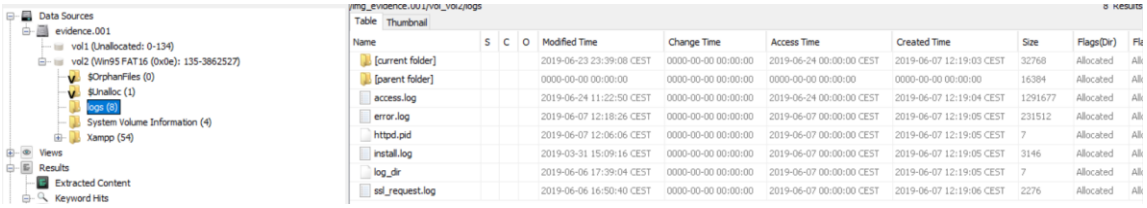


Figura 152. Reto 4: Ficheros logs de un servidor clonado.

Pudimos guardarlo, con clic derecho, export files. En él, observamos que efectivamente alguien había estado intentando acceder al sistema:

```
134.0.11.132 - admin [07/Jun/2019:11:45:31 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:31 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:31 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:33 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:33 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:33 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:34 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:35 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:36 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
134.0.11.132 - admin [07/Jun/2019:11:45:38 +0200] "HEAD / HTTP/1.1" 302 - "" "Mozilla/3.0 (Compatible);Brutus/AET"
```

Figura 153. Reto 4: Fichero log de accesos al sistema con indicios de un crackeo de contraseñas.

Entre los archivos borrados, encontramos una imagen y un documento docx.

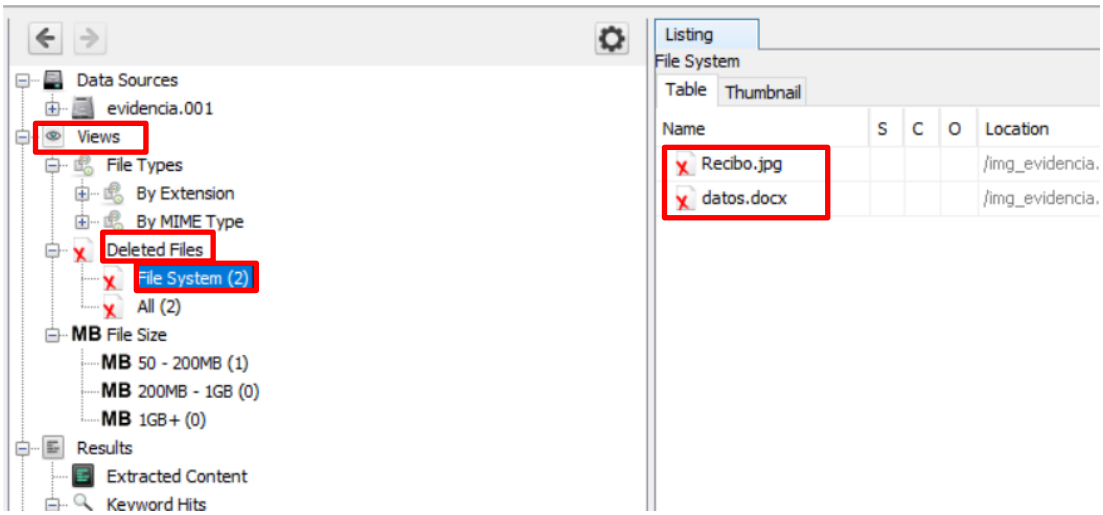


Figura 154. Reto 4: Rastro de archivos borrados en el servidor.

Pudimos guardarlos de la misma forma que el archivo log. Al abrir la imagen, vimos que se trataba de un recibo, y dentro de este, observamos una retirada de 18000 euros.

Hasta ese momento, ya sabíamos que unos 18000 euros de los 200000 euros habían ido a parar a un tal Bob. Ahora, tocaba ver que había en el documento Word, que al intentar abrirlo, no se podía. Descubrimos de qué tipo de archivo se trataba:

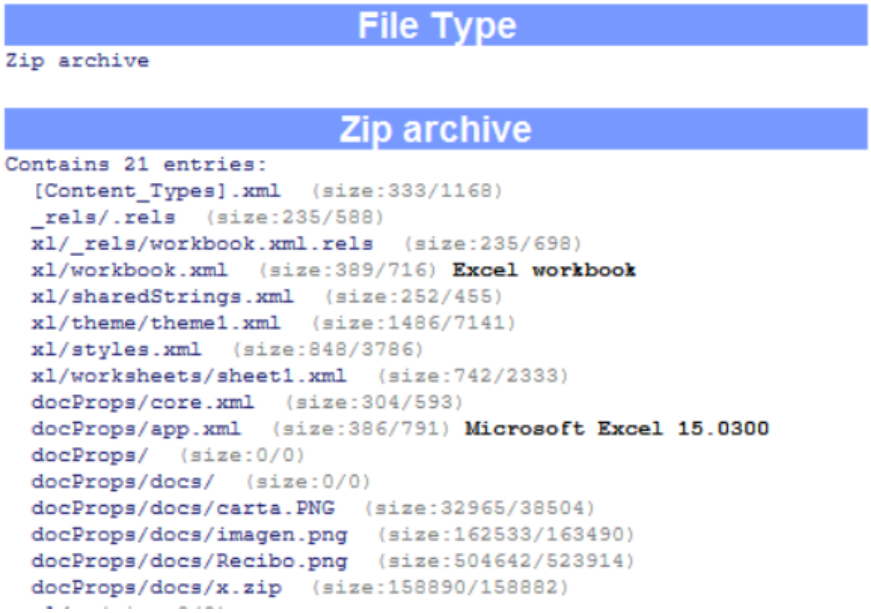


Figura 155. Reto 4: Descifrar el tipo de archivo con HexBrowser.

Se trataba de un archivo Excel y si le cambiábamos la extensión ya podríamos abrirlo.

En este fichero aparecían unos gastos de 750 euros. Además, como se veía en la captura anterior, dentro del fichero Excel, había otros archivos (varias imágenes y un zip). Al extraer la carpeta correcta, esta contenía un correo, dónde se indicaba el personaje al que fueron a parar los 18000 euros del primer recibo. No obstante, hasta el momento, habríamos averiguado dónde habían ido a parar 18750 euros de 200000, ahora había que saber dónde estaban los 181250 restantes.

Junto a esta carta, había una imagen borrosa que parecía ser un recibo. Pudimos conseguir recuperar dicha imagen, mediante el programa SmartDeblur. El cual nos permite enfocar imágenes, al menos para poder ver un poco mejor alguno de los valores. Esta se correspondía con 1250 euros invertidos en el alquiler de una limusina lujosa:

Figura 156. Reto 4: Imagen desenfocada.

Figura 157. Reto 4: Imagen enfocada con SmartDeblur.

El archivo zip que aparecía, necesitaba de una contraseña para poder abrirse. Y entonces entendimos una imagen de un candando, que se encontraba junto a las otras. La imagen era de un candado donde ponía rock.

Una forma de poder averiguar la contraseña del zip, era utilizar el diccionario rockyou. Utilizamos el comando fcrackzip, de esta forma:

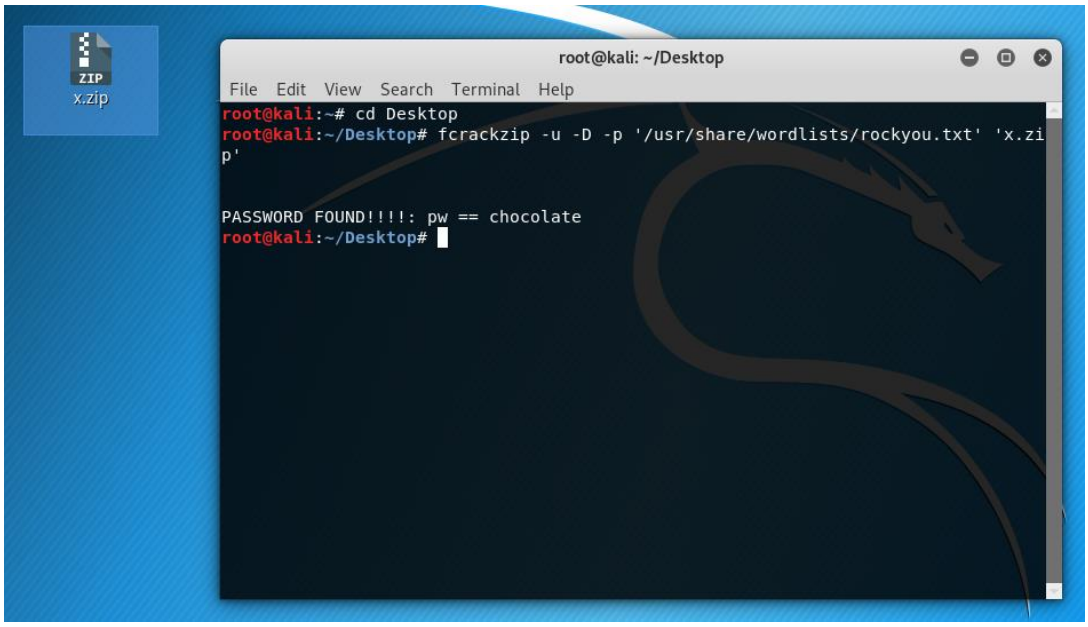


Figura 158. Reto 4: Descifrar contraseña de un archivo zip mediante el diccionario rockyou.

Ya pudimos abrir el zip, y dentro encontramos un documento y una imagen. Teníamos la imagen de una tableta de chocolate (como la contraseña).

El documento tenía contraseña. Si observábamos dentro de la imagen, con un lector hexadecimal podíamos ver “leche”.

Probamos si era la contraseña del Word, y efectivamente, ya pudimos abrirlo. El documento, se trataba de un contrato de compraventa de una vivienda, que era donde había ido a parar el resto del dinero (180000 €) y nuevamente pudimos ver quien había sido el que había gastado todo ese dinero. Si mirábamos todo el documento, al final del mismo, podíamos ver la bandera: **ética**.

Mediante este reto, abarcamos una parte de la informática forense, centrándonos en el uso de imágenes de sistemas, en este caso de un servidor. Profundizamos en las formas en las que se puede crear una imagen, y como se pueden analizar. Tanto para la creación como para el análisis existen múltiples herramientas, aunque algunas nos ofrecen más datos que otras, que nos pueden ayudar de guía en la investigación.

Por otra parte, también hemos entrado en aspectos más peliagudos como ataques de fuerza bruta, en este caso con la intención de acceder a un servidor, o crackear para descubrir la contraseña de un archivo. Sin duda, dos situaciones que pueden encontrarse perfectamente en el mundo real, bastante comunes. Además, sin olvidar que todos estos casos, se encuentran tocando aspectos éticos, que pueden

ayudar a pensar un poco más las cosas antes de intentar hacer algunas de estas, salvo que sea con carácter didáctico, como es el caso.

Finalmente, si tuviésemos que indicar la complejidad que puede suponer resolver este reto, deberíamos partir de algunos datos estadísticos. Estos los hemos podido reunir a partir del evento de las JNIC, donde este reto fue presentado. El número de personas que trataron de resolverlo fue entorno a unas dieciséis, de las cuales, solo alcanzaron la bandera, ocho. Esto quiere decir, que un 50% de participantes consiguieron solventarlo.

6.5. Tráfico de armas

Procedemos a mostrar la forma en la que se le puede poner solución al reto presentado en el apartado 5.2.2. Tráfico de armas.

Se nos proporcionó una imagen clonada de una memoria USB. Lo primero que hicimos, fue montar dicha memoria y, para montar la imagen proporcionada utilizamos el programa OSFMount. Una vez ejecutado, apareció la siguiente ventana, donde señalamos la opción “mount new”.

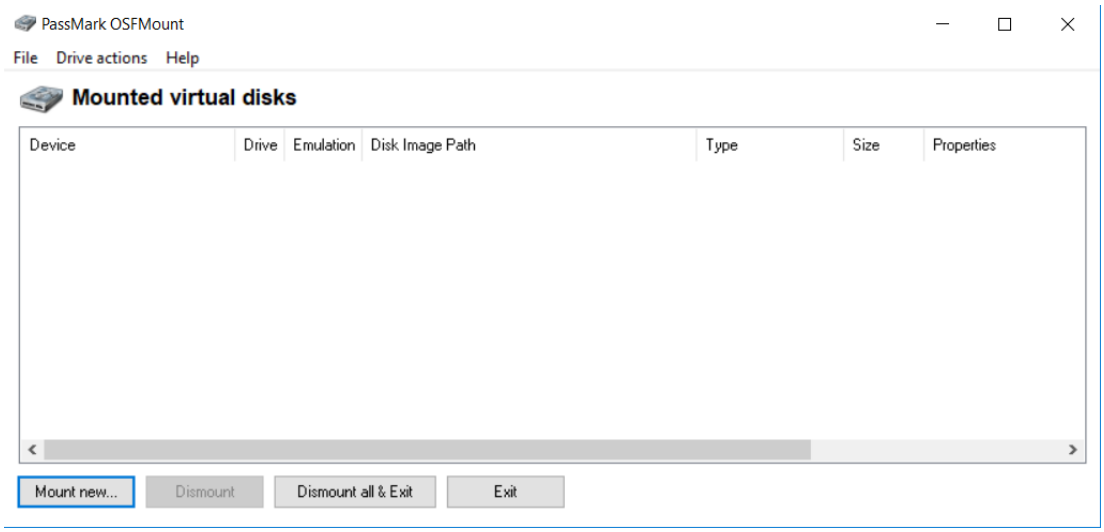


Figura 159. Reto 5: Montar una imagen con OSFMount 1.

En la ventana siguiente debíamos seleccionar la imagen que queríamos montar (en este caso, la imagen clonada) y las siguientes opciones:

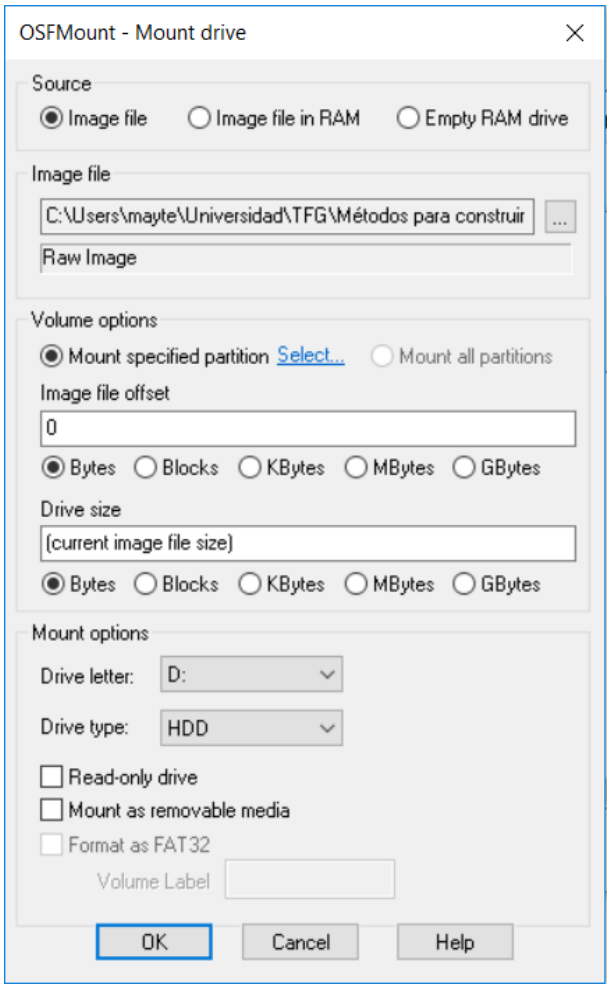


Figura 160. Reto 5: Montar una imagen con OSFMount 2.

Pulsamos ok, y nuestra imagen quedó montada. Al abrirla solo nos mostró varias imágenes. Como las preguntas nos estaban informando de que había ciertas cosas que teníamos que descubrir, pues sería porque estos elementos habían sido borrados, por tanto, los recuperamos. Para ello, utilizamos la herramienta Recuva. Al abrir esta herramienta, se nos presentó la siguiente ventana:

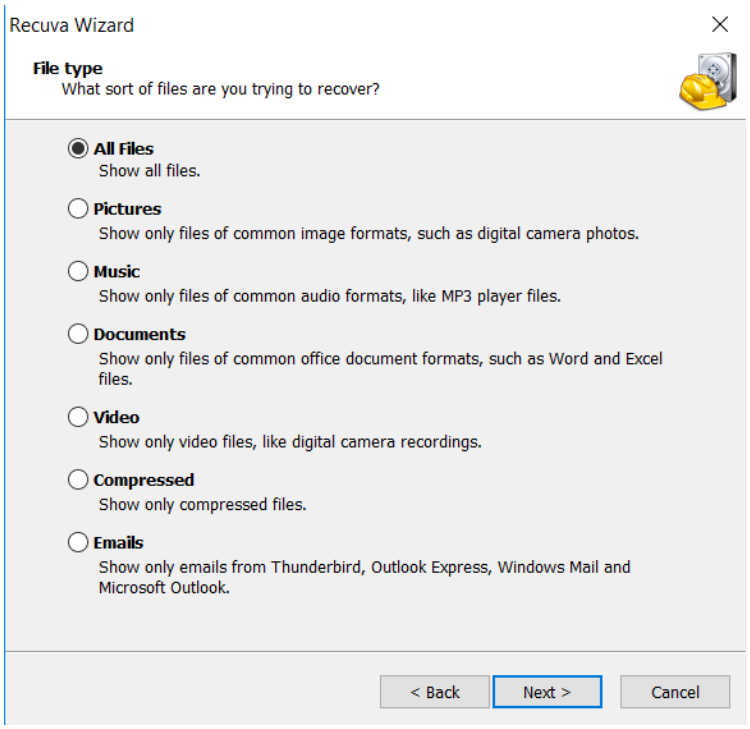


Figura 161. Reto 5: Recuperación de archivos borrados con Recuva 1.

Mantuvimos la opción all files, pues desconocíamos los archivos que habían podido ser borrados y pulsamos Next. En la siguiente ventana, marcamos la opción "in a specific location", y elegimos la localización de nuestra imagen montada.

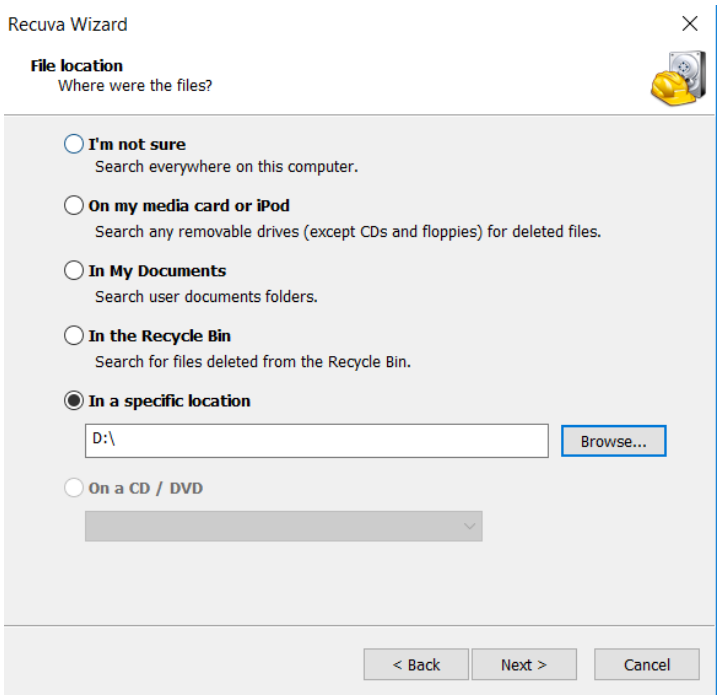


Figura 162. Reto 5: Recuperación de archivos borrados con Recuva 2.

Continuamos y para que se llevara a cabo la recuperación, debíamos pulsar "start" y marcar la opción "Enable Deep Scan", para que pudiesen ser recuperados todos los archivos.

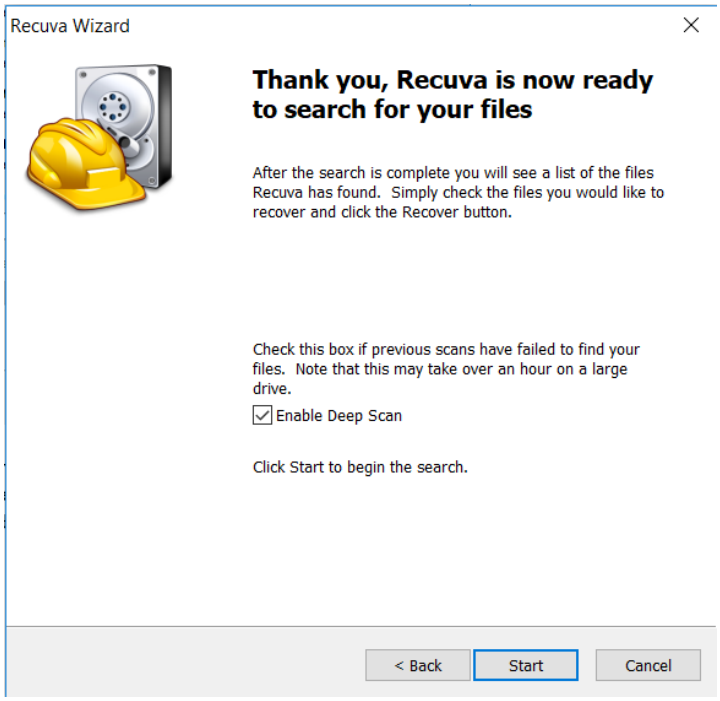


Figura 163. Reto 5: Recuperación de archivos borrados con Recuva 3.

Al terminar el proceso, obtuvimos esto:

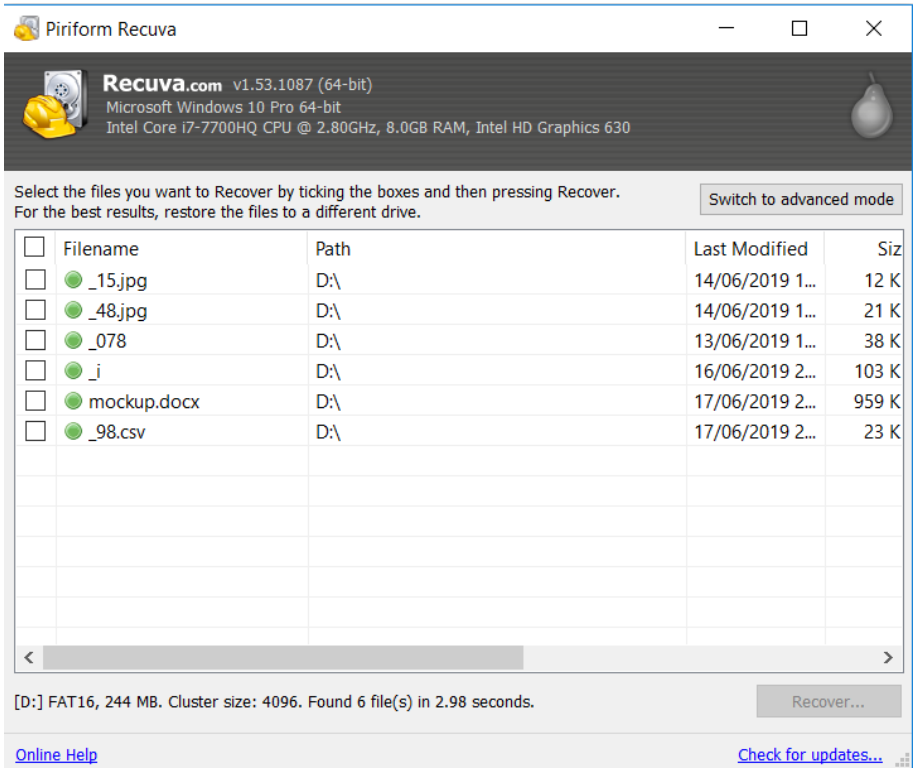
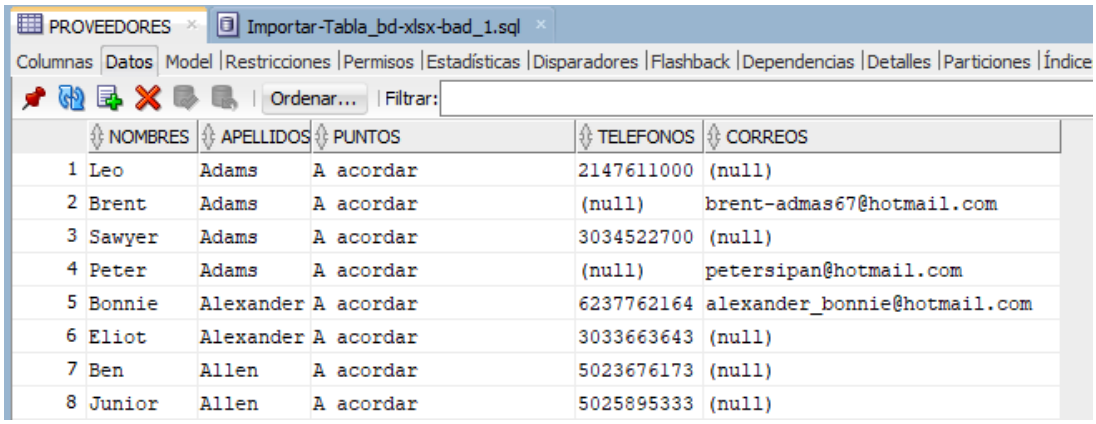


Figura 164. Reto 5: Recuperación de archivos borrados con Recuva 4.

Seleccionamos todos los archivos, pulsamos Recover y elegimos donde queríamos guardarlos.

Primero, averiguamos la extensión de los dos archivos que no la tenían haciendo uso de HexBrowser. El primero (_78) resultó ser un archivo de texto, más concretamente un archivo sql, y el segundo (_i), una imagen.

El siguiente paso, fue ir respondiendo las distintas preguntas. En cuanto a los proveedores, dado que el archivo sql, se trataba de la inserción en una base de datos de varios nombres, teléfonos y correos, intuimos que sería este.


PROVEEDORES | Importar-Tabla_bd-xlsx-bad_1.sql

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índice

Ordenar... | Filtrar:

	NOMBRES	APELLIDOS	PUNTOS	TELEFONOS	CORREOS
1	Leo	Adams	A acordar	2147611000	(null)
2	Brent	Adams	A acordar	(null)	brent-admas67@hotmail.com
3	Sawyer	Adams	A acordar	3034522700	(null)
4	Peter	Adams	A acordar	(null)	petersipan@hotmail.com
5	Bonnie	Alexander	A acordar	6237762164	alexander_bonnie@hotmail.com
6	Eliot	Alexander	A acordar	3033663643	(null)
7	Ben	Allen	A acordar	5023676173	(null)
8	Junior	Allen	A acordar	5025895333	(null)

Figura 165. Reto 5: Tabla generada por un fichero sql en SQL Developer.

De esta manera logramos responder a quienes eran los proveedores, ya que como podemos ver, aparecían con nombres y apellidos. Pero aún nos faltaba reunir los distintos estados involucrados. Para hacer esto, fuimos consultando los prefijos de los números de teléfono, sabiendo que el prefijo de EEUU es +1. Estos nos aportaban mucha información:

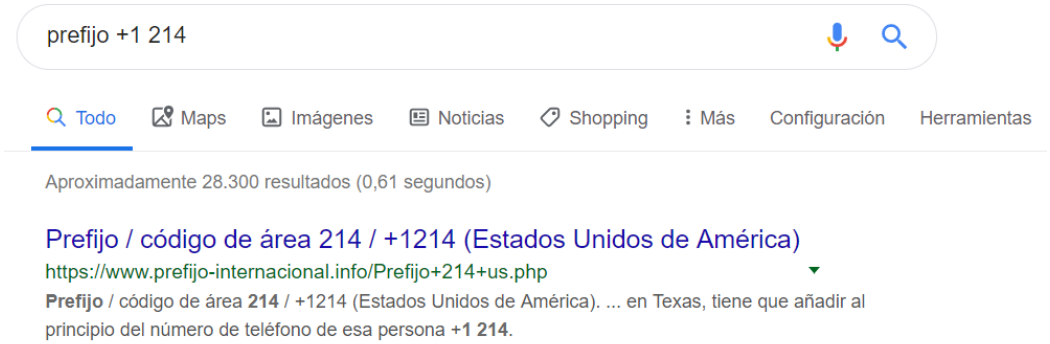


Figura 166. Reto 5: Consultar prefijo de teléfono en Google.

Este primer prefijo se correspondía con Texas.

Además, de los números de teléfono, en algunos se incluían puntos de encuentro (coordenadas), que al introducir en google maps, nos señalaba:

Figura 167. Reto 5: Consultar coordenadas en Google Maps.

Este punto nos situaba en Janesville, Wisconsin. Este proceso, se repitió con todos, finalmente reunimos el siguiente listado: Florida, Dakota del sur, Indiana, Missouri, Oklahoma, Utah, Tennessee, Virginia, Vermont, Texas, Wisconsin, Alabama, Arizona, Colorado, Carolina del Norte, Georgia, Kansas, Kentucky, Luisiana, Massachusetts, Nebraska, Minnesota, Nueva Jersey, Nueva York, Pensilvania, Nevada, Washington, Wyoming, Hawái. Con ciudades como Miami, Orlando, Kansas City, Phoenix, Atlanta, Tucson, Louisville, Boston, Minneapolis, Yuma, Lakewood, Omaha, Philadelphia, Carson City, Las Vegas,…

La siguiente pregunta a responder, era sobre los proveedores. Nos encontramos un fichero csv, con el siguiente contenido:

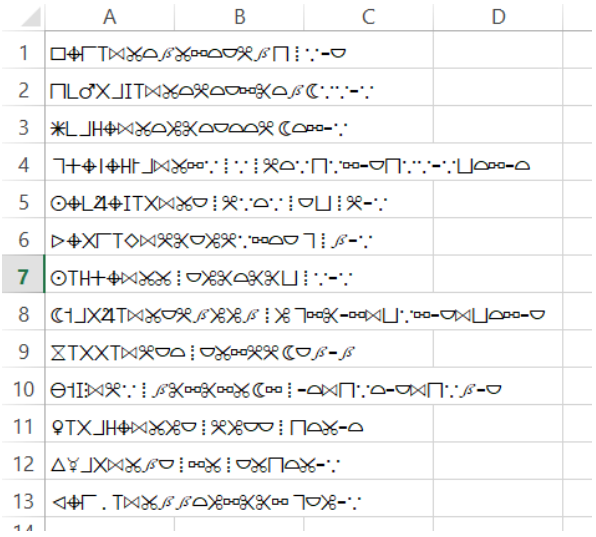


Figura 168. Reto 5: Texto cifrado con símbolos illuminatis.

Este parecía estar cifrado con símbolos illuminatis, así que procedimos a traspasar dichos símbolos a un lenguaje entendible, con ayuda de:

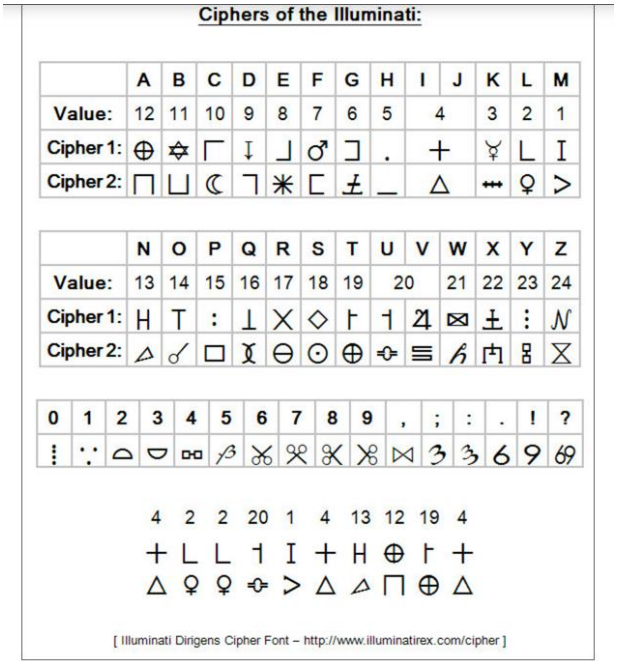


Figura 169. Reto 5: Cifrado de los Illuminatis.

El resultado final, fue este:

	A	B	C	D
1	Paco, 625642375 A01 -3			
2	Alfredo, 627234825 C11-1			
3	Elena, 629823227 C24-1			
4	Diamante, 641010721 A14-3 A11-1 B24-2			
5	Salvador, 630712103 B07-1			
6	Marcos, 783971423 D05-1			
7	Sonia, 660398288 B01-1			
8	Cuervo, 637599509 D48-4, B14-3, B24-3			
9	Zorro, 732036477 C35-5			
10	Rudy, 710584846 C40-2, A12-3, A15-3			
11	Lorena, 693079330 A26-2			
12	Iker, 653046036 A26-1			
13	Nacho, 655294884 D39-1			

Figura 170. Reto 5: Texto descifrado.

Teníamos nombres, número de teléfonos, y lo que parecían unos números de series, que podían estar haciendo referencia a las armas. De estas, disponíamos de algunas imágenes.

Al buscar por internet con ayuda de dichas imágenes, pudimos averiguar que se trataban de: una pistola m82, una carabina sig sauer 522, un rifle, mini Hécate (PGM

338), zastava master flg, revólver Beaumont-Adams, Walter p88,… Y así es como lo averiguábamos:

Figura 171. Reto 5: Búsqueda por imágenes en Google.

La otra imagen, resultó ser una tabla, que teniendo en cuento el nombre las fotos que aparecían en las imágenes, o en el archivo csv, parecía estar haciendo referencia a las armas. Se trataba de un registro, donde contabilizaban cantidades y posibles ventas. Por lo que sin tener en cuenta las compradas, el número total de armas parecía estar rondando un total de 176.

El último documento que nos quedaba por revisar, se trataba de un Word, pero este tenía contraseña, por lo que desciframos dicha contraseña con ayuda de John the ripper, con este comando:

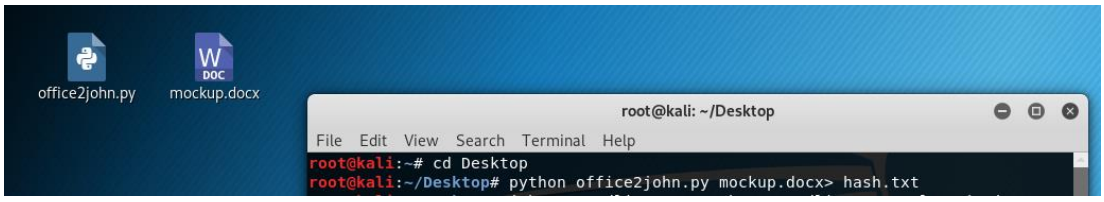


Figura 172. Reto 5: Generación de código hash para descifrar contraseña de un documento.

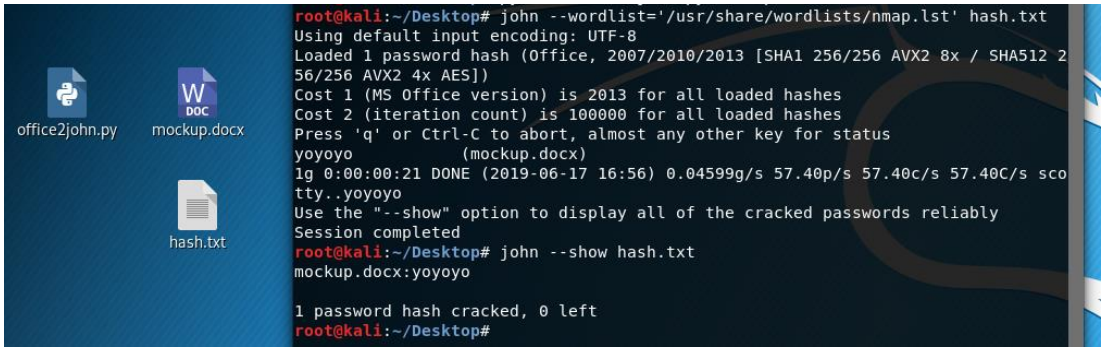


Figura 173. Reto 5: Descifrar contraseña de un documento hash con John the ripper.

La contraseña resultó ser yoyoyo. La ingresamos, y ya pudimos abrir el documento. Dentro de este, encontramos unos prototipos de lo que parecían ser unas licencias y guías de armas. Por lo que también estaban intentando falsificar este tipo de documentos.

A través del siguiente desafío hemos adquirido nuevos conocimientos relacionados con la preservación de evidencias. Ya que, con la intención de no perjudicar la memoria original, se lleva a cabo un clonado de la misma.

Hemos trabajado con otros métodos, como la recuperación de elementos (aunque estos se eliminen, no lo hacen por completo), descubrir extensiones de archivos, crackear contraseñas de documentos de texto (hasta el momento nos habíamos limitado a los archivos zip), nuevas formas de cifrado, relacionar elementos y conceptos. Sin duda, podemos afirmar que este reto permite trabajar y mejorar muchas de las cualidades necesarias en la informática forense, y que nos puede ayudar a conocer un poco más el tipo de circunstancias que nos podemos encontrar en una investigación de esta índole.

6.6. Un reto explosivo

En el desarrollo de esta sección, dibujaremos la solución correspondiente al reto presentado en el apartado 5.2.3. Un reto explosivo.

Se proporcionaba un archivo pcap con el tráfico del equipo de un químico que había estado desarrollando un explosivo. Era necesario localizar los componentes de dicho explosivo y el lugar de reunión con el comprador del mismo.

Dentro del archivo pcap, se registraba la subida de un documento. Pudimos extraer dicho documento mediante la opción de Export select packet bytes. Y obtuvimos un documento pdf. Al elemento guardado había que borrarle todo lo que se ponía delante en su código hexadecimal, para así poder abrirlo:

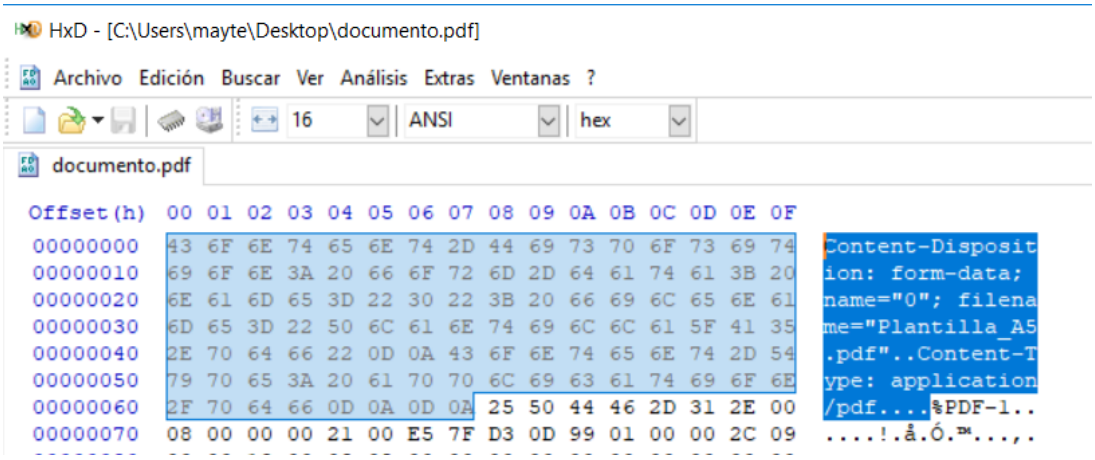


Figura 174. Reto 6: Información basura situada en la cabecera de un documento.

Sin embargo, seguíamos sin poder abrirlo. Por lo que seguimos observando dicho documento con el lector, y vimos que en realidad se trataba de un Word, pero que aunque cámbiesenos la extensión, seguíamos sin poder abrirlo. Esto se debía a que le faltaba la cabecera [content-types].xml y bastaba con volvérsela a añadir:

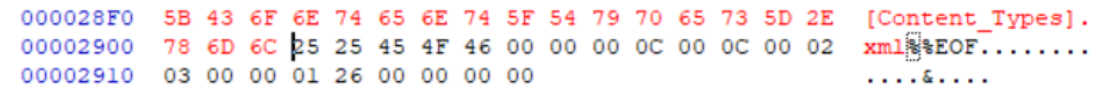


Figura 175. Reto 6: Cabecera Content-types necesaria en un documento docx.

Nuevamente, nos lanzó un aviso al intentar abrirlo, se le daba a reparar y ya se podía.

Estos eran los elementos que habían sido utilizados en el explosivo. Al fijarnos con atención en la información del documento vimos que nos indicaba más palabras de las que en realidad tenía, esto significaba que había texto oculto. Para recuperarlo, seleccionamos todo el texto e hicimos clic derecho, Fuentes:

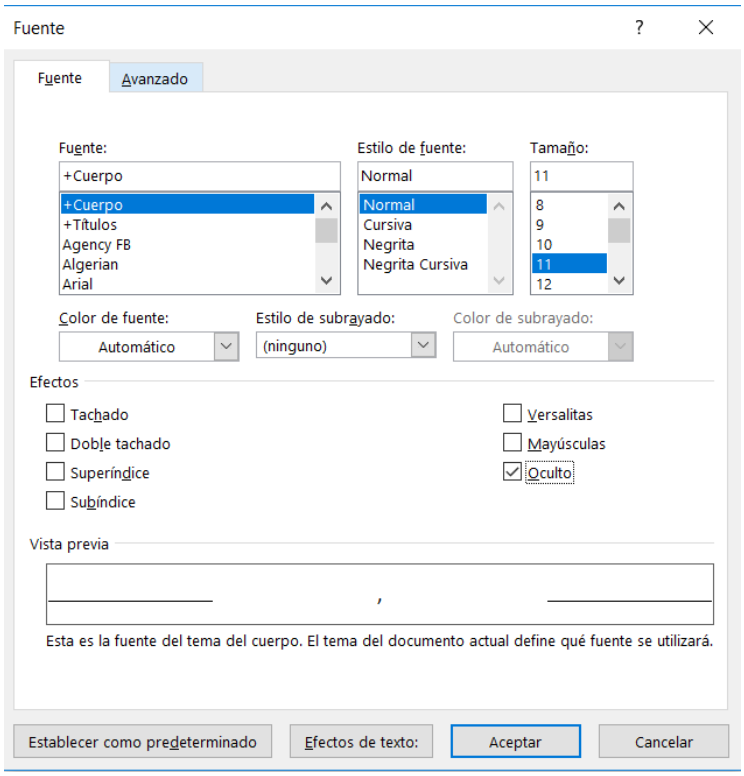


Figura 176. Reto 6: Mostrar información oculta en documento docx 1.

Desmarcamos la opción de Oculto:

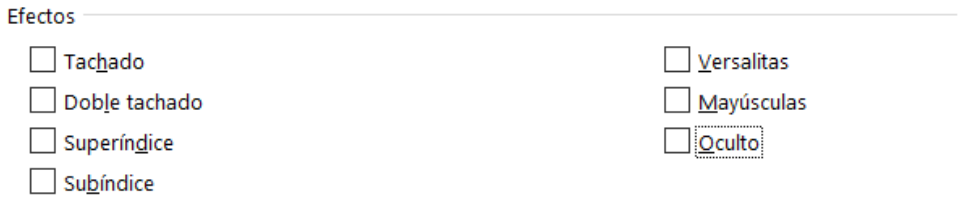


Figura 177. Reto 6: Mostrar información oculta en documento docx 2.

Y ya se podía ver todo el listado completo.

Con respecto, al punto de encuentro. Si analizábamos los correos del archivo pcap podíamos ver varias pistas ocultas en el como:

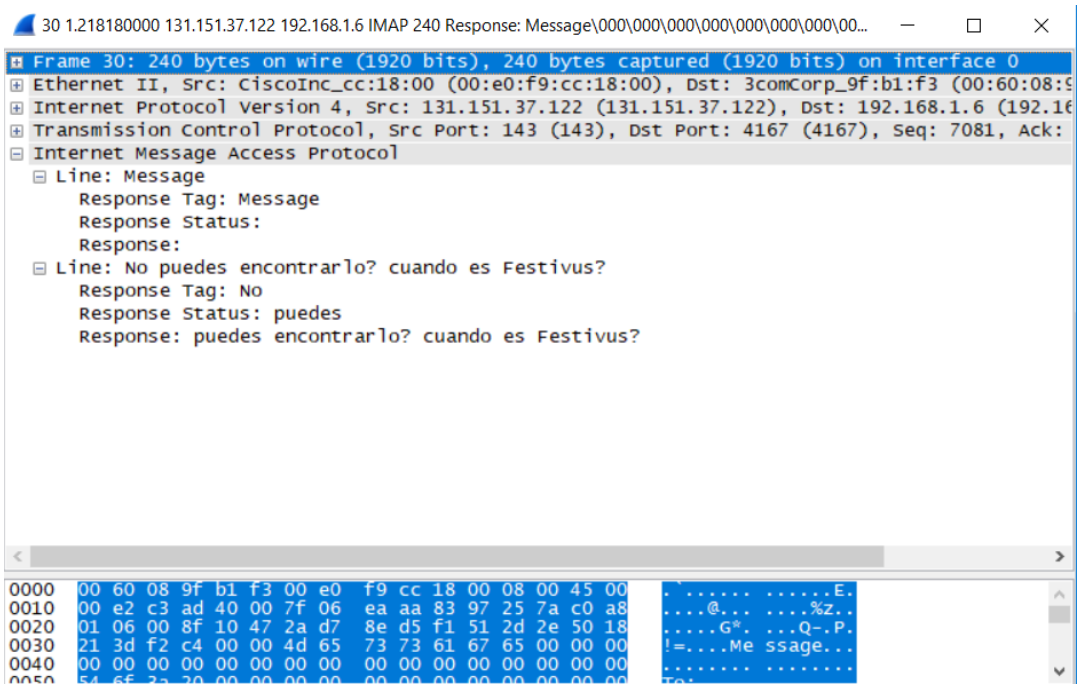


Figura 178. Reto 6: Mensaje de correo presente en un archivo pcap.

Así como otras pistas del tipo “blog”, “google”, “pass”, “donde?”.

Al buscar Festivus en google:

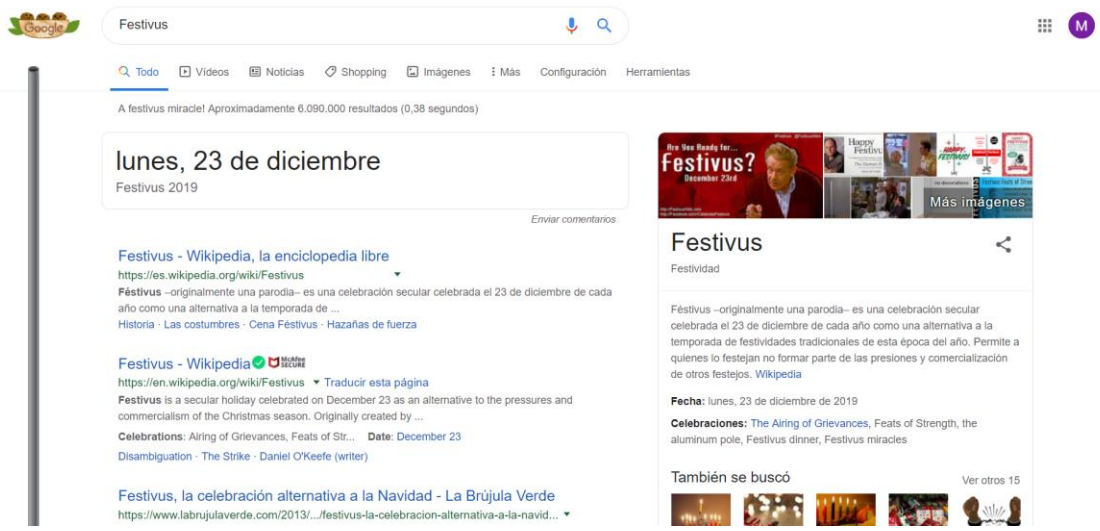


Figura 179. Reto 6: Resultado mostrado por Google al buscar Festivus.

Se trataba de una celebración que se hacía de broma.

En el fichero también nos hablaba de un blog de google, por lo que buscamos blogs que tuvieran que ver con Festivus, y no había nada. Si observábamos, el mensaje nos preguntaba ¿cuándo es Festivus? Por lo que buscamos un blog del 23 de diciembre, y encontramos una con una página llamada Festivus:

Figura 180. Reto 6: Página Festivus del blog 23 de Diciembre.

Si pulsábamos sobre dicho enlace nos llevaba a una entrada que nos pedía una contraseña, aquí utilizamos la contraseña indicada antes “pass” “donde?”.

En esta entrada obtuvimos un enlace, que nos descargó un zip. El cual, contenía un conjunto de archivos:

Encuentro.001	19/06/2019 13:32	Archivo WinRAR	1 KB
Encuentro.002	19/06/2019 13:32	Archivo 002	1 KB
Encuentro.003	19/06/2019 13:32	Archivo 003	1 KB
Encuentro.004	19/06/2019 13:32	Archivo 004	1 KB
Encuentro.005	19/06/2019 13:32	Archivo 005	1 KB
Encuentro.006	19/06/2019 13:32	Archivo 006	1 KB
Encuentro.007	19/06/2019 13:32	Archivo 007	1 KB
Encuentro.008	19/06/2019 13:32	Archivo 008	1 KB

Figura 181. Reto 6: Archivo separada en ocho partes.

Parecía que se trataba de un archivo dividido en varias partes. Con ayuda de la herramienta HJ-Split, volvimos a unir el archivo:

Figura 182. Reto 6: Unión de varias partes en un único archivo.

Averiguamos la extensión del archivo unido con HexBrowser. Vimos que era un archivo XML, KML. Estos archivos son generados con Google Earth. Por lo que cambiamos la extensión y abrimos dicho archivo con él:

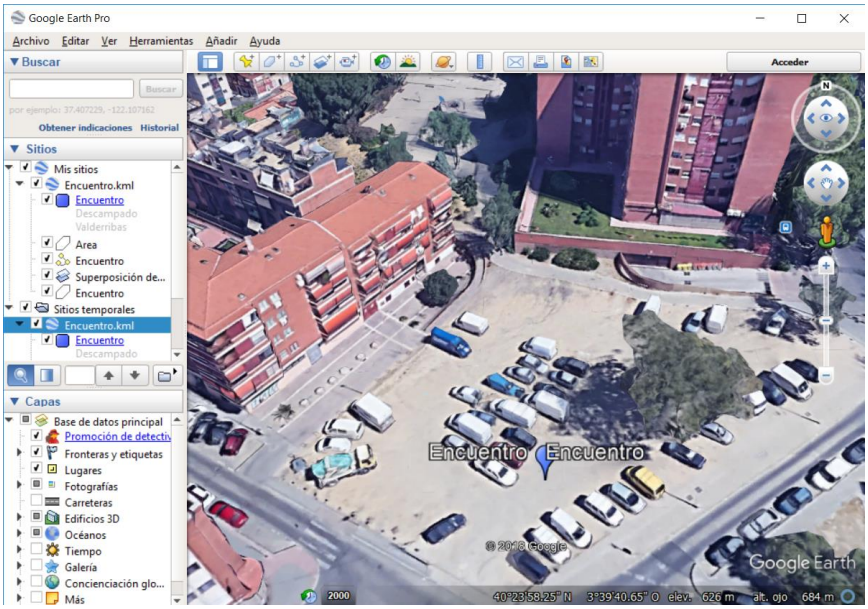


Figura 183. Reto 6: Archivo KML abierto con Google Earth.

Por lo tanto, el punto de encuentro era un Descampado de Madrid.

Con este reto, hemos salido del clásico reto forense donde nos limitamos a analizar una imagen, ya que en esta ocasión partimos de un fichero que registra un tráfico de red.

Nuevamente volvemos a trabajar la necesidad de relacionar varias pistas o en un caso real, información proporcionada por despiste. Es con esta forma, con la que podemos localizar el blog que nos permita encontrar el punto de encuentro. Y además, con este encontrarnos ante un nuevo desafío, un conjunto de partes que pertenecen a un archivo y se necesita volver a unirlas para formarlo nuevamente (con este también volvemos a trabajar las formas de descifrar una extensión).

Por lo que podemos afirmar, que realmente, lo más importante de este reto es el aprender a trabajar prestando especial atención en los detalles de una investigación. Pues cada dato, por pequeño que parezca, nos puede hacer avanzar a pasos agigantados.

6.7. Todo lo que ofreció

Durante el desarrollo de este punto, haremos un recorrido por la solución correspondiente al reto presentado en el apartado 5.3.1. Todo lo que ofreció.

Se proporcionaba un archivo wireshark con el intercambio de varios mensajes, y entre ellos, el de un documento. El correo que aparecía era falso, por ello había que obtener el correcto a partir del documento. Existían varias banderas que encontrar.

La primera era el nombre de usuario en una red social, más concretamente twitter. Una vez obtuvimos el documento del fichero mediante la opción export selected packet bytes como en retos anteriores.

Al consultar sus metadatos, pudimos ver este correo:

```
Creator                         : migueloz.1231@gmail.com
Keywords                        : ..
Description                     : ..
Last Modified By                : migueloz.1231@gmail.com
Revision Number                 : 3
Create Date                     : 2013:08:22 19:17:00Z
Modify Date                     : 2013:08:22 19:17:00Z
```

Figura 184. Reto 7: Metadatos de un documento docx.

Buscamos su cuenta de usuario en twitter a partir de dicho correo. Para ello, cargamos el correo en la lista de contactos de Gmail:

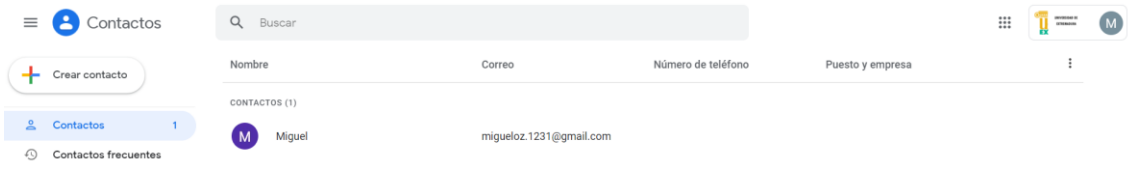


Figura 185. Reto 7: Encontrar usuario en Twitter a través de su correo 1.

Después desde una cuenta de twitter, y en la parte de encontrar amigos:

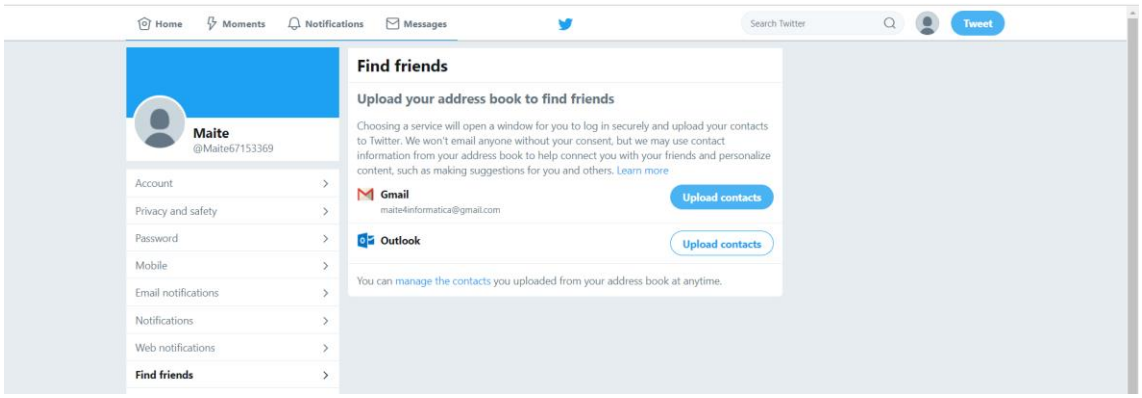


Figura 186. Reto 7: Encontrar usuario en Twitter a través de su correo 2.

Indicamos que cargábamos desde Gmail. Y al cargar los contactos, ya nos lo encontró:

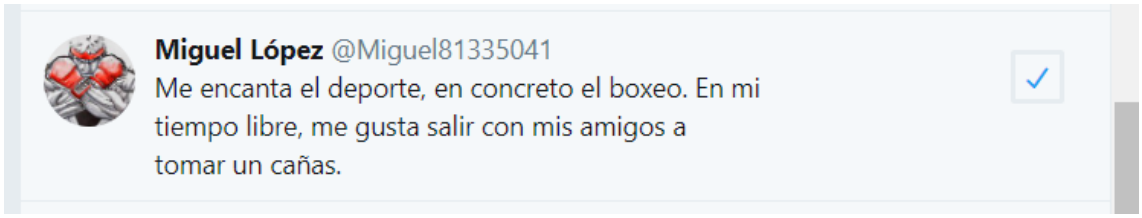


Figura 187. Reto 7: Encontrar usuario en Twitter a través de su correo 3.

Primera bandera: **@Miguel81335041.**

La segunda bandera consistía en la localización de donde se encontraba. Existían varias pistas, como imágenes subidas en la cuenta (unos chocolates, un hotel, un zoo). Por el chocolate, situamos que el país era Suiza.

Al consultar los metadatos de la imagen del hotel, encontramos la localización del mismo en Zúrich (Suiza), y más concretamente en **47.388307, 8.513936.**

La tercera bandera era una contraseña. Entre los tweet existía uno en el que se decía que le gustaba github, por lo que se entendió que tenía una cuenta en él. Seguidamente, de ese tweet existía una imagen con un codigo de barras. Algo que era bastante sospechoso, por lo que leímos dicho código de barras mediante una app de este tipo:

Figura 188. Reto 7: Información obtenida a través de un código de barras.

La tercera bandera era: **Xw8x0Xq2EL8gzpV.** Y esta sería la contraseña para acceder a git.

La cuarta bandera se trataba del número de teléfono. Al acceder a la cuenta de git con la contraseña obtenida anteriormente, encontramos un repositorio privado en la misma. Al ver su descripción, encontramos esto: "I like everything related to the Computer World. I am willing to collaborate on all Projects, contact me through the number 654734040".

Cuarta bandera: **654734040**.

La quinta bandera era la información proporcionada. En la misma cuenta de github, había un documento Word, si lo descargábamos e intentábamos abrirlo, comprobábamos que no se podía (porque realmente no era un docx). Pudimos descubrir de qué se trataba mediante la herramienta hexBrowser o por sus metadatos. Se trataba de una imagen. Al abrirla con un lector hexadecimal pudimos ver al final de la misma la dirección url almacenada. Por tanto la quinta bandera: http://cort.as/-IHEb.

Cuando introdujimos esa url en el navegador, nos descargó un documento. Esta información fue la sexta bandera.

Por último, había que encontrar el nombre de la persona a la que se le había estado traspasando esta información. En el archivo pcap, el correo destino era xxi.098712@gmail.com.

Pudimos comprobar que existía con un verificador de email. Que nos dijo que la cuenta era válida. Sin embargo, no encontramos ninguna red social asociada a ese correo, por lo que comprobamos si existía una cuenta en git, lo hicimos a través de la url:

Figura 189. Reto 7: Búsqueda de una cuenta en git a través de la url.

Y efectivamente, existía una cuenta en git. La sexta bandera era **Antonio Ruíz Castro**.

Con este reto, hemos podido realizar un recorrido, por distintas herramientas open source, pues todas las web que aparecen durante el desarrollo del mismo son públicas y además muy comunes, pues prácticamente todo el mundo tiene una cuenta en alguna red social.

De forma, un poco más desapercibida, tratamos el tema de la falta de seguridad en las mismas, ya que a partir de un correo hemos podido localizar a una persona. Existen redes en las que no ocurre esto, por ejemplo, Facebook, aunque antes también se podía hacer y además de una forma más directa (a través del propio buscador).

Este reto es bastante sencillo, si prestamos especial atención a todas las pistas que se nos van proporcionando en cada una las partes del mismo que tenemos que ir recorriendo.

6.8. Una lección de matemáticas

En este apartado, pondremos solución al reto expuesto en el apartado 5.4.1. Una lección de matemáticas.

Se nos proponía resolver un captcha. En primer lugar, si pulsábamos enviar sin haber escrito nada, nos mostraba este resultado:

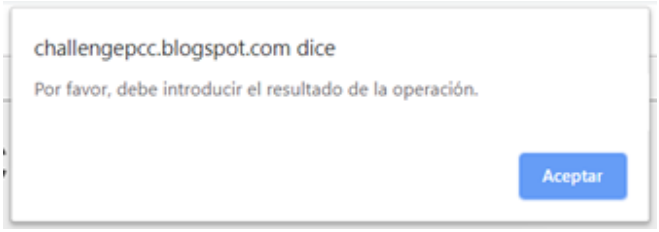


Figura 190. Reto 8: Resultado mostrado sino se resuelve el captcha.

Si el resultado introducido no era correcto, nos indicaba esto:

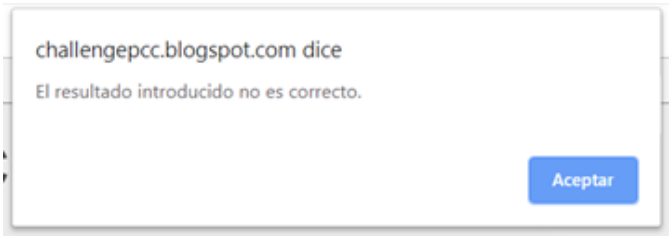


Figura 191. Reto 8: Resultado mostrado si se resuelve el captcha de forma errónea.

Al recargar la página (se cargaban nuevos números), para empezar nuevamente. Si calculamos el resultado de forma manual, nos mostraba una burla como la que podemos ver aquí:

Figura 192. Reto 8: Resultado mostrado si se resuelve el captcha de forma manual.

Por lo que resolvimos el reto, con la ayuda de un programa. El código del mismo lo escribimos en lenguaje java y utilizando Eclipse como IDE.

En primer lugar, fue necesario crear un proyecto maven, para añadir una serie de dependencias. Esta nos permitiría utilizar Selenium en java, para usar el webdriver y así conectarnos a la página que albergaba el código:

```
<artifactId>Solucion</artifactId>
<version>0.0.1-SNAPSHOT</version>
<dependencies>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-server</artifactId>
        <version>3.0.1</version>
    </dependency>
</dependencies>
```

Figura 193. Reto 8: Dependencias necesarias para utilizar Selenium en Java.

Además, también descargamos el ejecutable correspondiente al driver que nos facilitaba dicha conexión y lo guardamos en el disco local C. Este fue descargado para usar con el navegador Chrome (también lo hay disponible para Firefox y Opera). Es importante comentar, que fue necesario descargar aquel que coincidía con nuestra versión del navegador instalada.

Una vez comentado esto, procedemos con el código. Primero establecimos la conexión con la página de la siguiente forma:

```
System.setProperty("webdriver.chrome.driver","C:\\chromedriver.exe");

// Inicializamos el WebDriver
WebDriver driver = new ChromeDriver();
//Maximizamos la ventana del navegador
driver.manage().window().maximize();
//Accedemos a la web:
driver.get("https://challengeppc.blogspot.com/2019/06/ChallengePCC.html");
```

Figura 194. Reto 8: Código para establecer conexión con una página.

Mediante la opción SetProperty cargamos el driver. Estos métodos nos abrían una nueva pestaña en el navegador. Lo siguiente era recuperar la operación matemática, a través del id correspondiente al elemento input en el que esta se almacenaba:

```
// Obtenemos la operación matemática
String mathquestionvalue = driver.findElement(By.id("txtCaptcha")).getAttribute("value");
```

Figura 195. Reto 8: Código para obtener el texto del captcha.

Después obtuvimos ambos números, y para hacer esto, necesitábamos separarlos del símbolo de multiplicación (*), y recuperarlos cada de forma independiente:

```
// Obtenemos los dos números
String[] parts = mathquestionvalue.split("\\*");
String part1 = parts[0];
String part2 = parts[1];
```

Figura 196. Reto 8: Código para obtener las partes de un texto de forma independiente.

A continuación, procedimos a realizar la multiplicación entre ambos. Como eran números muy grandes utilizamos el tipo de datos BigInteger, tanto para los operandos como para el resultado:

```
// Multiplicamos ambos números
BigInteger uno = new BigInteger(part1);
BigInteger dos = new BigInteger(part2);
BigInteger multiplication = uno.multiply(dos);
```

Figura 197. Reto 8: Código para realizar la operación de multiplicación entre dos números de gran longitud.

Cuando ya teníamos el resultado debíamos insertarlo en el formulario:

```
// Insertamos el resultado de la operación en el formulario
WebElement capta = driver.findElement(By.id("CaptchaInput"));
capta.clear();
capta.sendKeys("" + multiplication);
```

Figura 198. Reto 8: Código para insertar un resultado en un formulario.

Y finalmente lo enviamos:

```
//Enviamos el resultado
driver.findElement(By.id("Test Captcha")).click();
```

Figura 199. Reto 8: Código para enviar un formulario.

Al lanzar el programa, en nuestra consola nos mostraba esto:

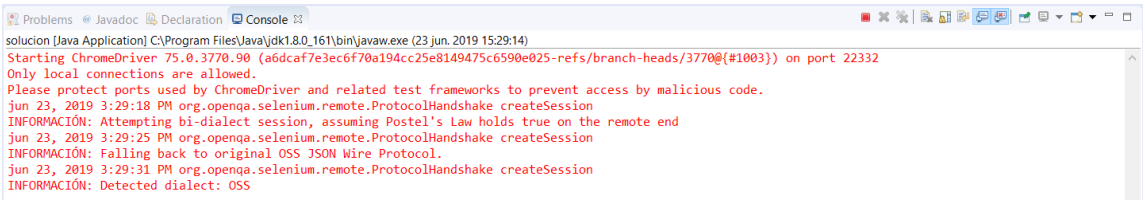


Figura 200. Reto 8: Resultado mostrado por la consola durante la ejecución del programa.

Y en la ventana del navegador que se nos había creado, podíamos ver esto:

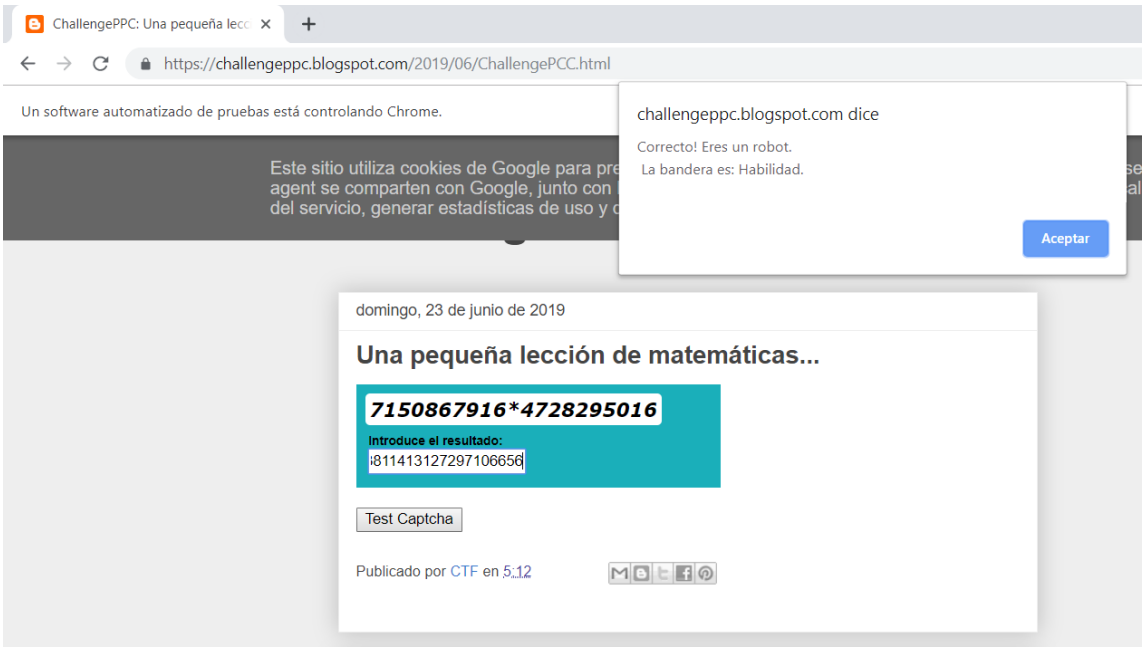


Figura 201. Reto 8: Resultado mostrado si se resuelve el captcha mediante un programa.

Mediante este reto, hemos podido poner en práctica el principal conocimiento adquirido en esta carrera. Teníamos un problema, lo hemos analizado y hemos encontrado la forma de solventarlo. Además, la forma ha sido mediante un código.

Sin duda, este tipo de retos nos permiten practicar en la programación, pero de una forma entretenida. Sin olvidar, lo mucho que aprendemos con ellos, porque podemos encontrar un montón de casos diferentes.

6.9. Reflexiones

La Informática es un fiel reflejo de la evolución tecnológica que se ha producido a nivel mundial y a lo largo de los años. Con la incorporación de Internet, el desarrollo de las comunicaciones ha seguido una evolución diferente trayendo consigo una nueva problemática hasta el momento impensable, como son los ciberdelitos.

Entendemos por delitos informáticos aquellos de acceso no autorizado, destrucción de datos, hacking, ciberterrorismo, infracción de los derechos de autor, infracción del copyright, interceptación de e-mail, estafas electrónicas, transferencia de fondos y phising. No obstante, podemos utilizar este tipo de delitos para aprender de ellos y desarrollar las competencias necesarias, evitando que estos puedan perjudicarnos de alguna forma.

Es por ello, que surge la idea del hacking ético, una forma de analizar los sistemas informáticos asumiendo el rol del ciberdelincuente con la intención de encontrar vulnerabilidades en los mismos, e informar sobre estas para que se puedan solventar. Ya que como decía Kevin Mitnick:

> "Los verdaderos hackers siguen un cierto conjunto de reglas éticas, que les impiden lucrarse o causar daño en sus actividades."

También es una forma de concienciar sobre la importancia de la seguridad informática, al igual, que el desarrollo de los retos de este tipo. Con estos, podemos impartir esta enseñanza desde un punto de vista más entretenido, ya que no partimos de una situación real sino una ficticia creada a partir de una historia presentada al participante.

Es indudable que solo con esta práctica vayamos a conseguir que las cosas cambien completamente, sino que también es necesario cambiar nuestra concepción

ante las distintas problemáticas que nos podemos encontrar, ya que esto es lo que nos permite evolucionar y definir nuestro propio método de acción. Por esto, es importante apoyarnos en otras investigaciones e ideas con la intención de mejorar nuestras técnicas. Siempre necesitamos más ayuda porque como dice Eric S. Raymond:

> "Ser capaz de superar la seguridad no te convierte en un hacker, de la misma forma que hacer un puente a un coche no te convierte en ingeniero mecánico."

Podemos utilizar los retos como un marco de conocimiento, convirtiéndolos en una herramienta útil que utilizar en la práctica. Sin embargo, es llamativo el desconocimiento acerca de los mismos, siendo la única familiarización con estos, el verlos como un tipo de competencia y no como un modelo de aprendizaje y práctica. Lo cual se puede afirmar, ya que los retos no están diseñados de forma regular, sino que las temáticas que abarcan son múltiples y variadas, y debemos transformarlos y desarrollarlos en función de las necesidades que se quieren cubrir o abarcar en ese momento, como hemos visto con anterioridad.

Esto pone de manifiesto la importancia de seguir realizando investigaciones que profundicen más en el tema y contribuyan a una mejora y utilización de los mismos de forma gradual. Siempre entendiendo que estos son un complemento y no la solución definitiva.

> "Si crees que la tecnología puede solventar tus problemas de seguridad, entonces no entiendes los problemas y no entiendes de tecnología." Bruce Schneier.

En esta época, donde la tecnología está presente en todo, resulta imprescindible la figura del informático, y este debe estar preparado para encontrarse cualquier situación ya que el abanico de posibilidades dentro de la misma es enorme. Por lo que, este concepto debería estar muy presente en la cabeza de todos y cada uno de nosotros.

Es un hecho que el hacking ético está preparado para el cambio, dejar de ser una herramienta de investigación, para convertirse en una profesión más. Pero para ello, es imprescindible contar con el entendimiento de lo que esto significa y de lo que podemos conseguir a través de él. En este proceso, puede hacerse vital llevar a

cabo modificaciones en el campo de la formación, incluyéndolo como una disciplina más, y haciendo responsables tanto a docentes como instituciones.

La evolución del mismo nos abrirá diferentes puertas a nuevos campos de actuación, suponiendo una oportunidad para reinventarnos en nuevas técnicas y conocimientos. Sin embargo, no será la solución definitiva, ya que parafraseando a Gene Spafford:

> "El único sistema completamente seguro es aquel que está apagado, encerrado en un bloque de cemento y sellado en una habitación rodeada de alambradas y guardias armados."

No obstante, no debemos olvidar el duro camino recorrido para llegar hasta cada solución, y luchar para mejorarla, de forma que en cada futuro que se nos presente, se convierta en una prácticamente concluyente.

7. CONCLUSIONES Y LÍNEAS FUTURAS

En este Trabajo Fin de Grado se ha conseguido diseñar un conjunto de ejercicios y retos relacionados con hacking ético e informática forense. Estos retos se han agrupado en tres categorías, de modo que se dispone de dos o tres ejercicios dentro de cada una de las categorías propuestas.

Como complemento de esta conclusión principal, puede añadirse que se han presentado varias herramientas que permiten enfrentarse a problemas de hacking ético e informática forense, estudiando las posibilidades que ofrecen para los retos propuestos y para otros similares. Muchas de estas herramientas, podían resultar más desconocidas porque pertenecían a sistemas Windows, un tema que en la informática forense, prácticamente se ha destinado a Linux. Por lo que ha sido una buena forma de practicar con ambos.

El planteamiento y resolución de los retos ha permitido aprender conceptos y metodologías relacionadas con las temáticas expuestas, lo cual ha resultado muy enriquecedor. Y al haberlo hecho, desde una visión meramente práctica, ha provocado que se convierta en un aprendizaje mucho más efectivo.

Finalmente, la realización del TFG ha resultado gratificante y entretenida, máxime cuando algunos de los ejercicios propuestos han sido presentados en el CFT de las JNIC 2019, esto es, en un CTF real, ante estudiantes universitarios de toda

España. Algo que sin duda, ha hecho que el esfuerzo haya sido aún mayor, para que los resultados obtenidos fueran de la altura de este certamen y nadie se sintiera decepcionada con los mismos.

Esto último, se convirtió en algo positivo, ya que logró sacar el máximo potencial y que este fuese plasmado en el trabajo, logrando como objetivo final el diseño y resolución de ocho desafíos, que resultasen únicos y de los más innovadores.

Como líneas futuras, podría indicarse la ampliación en el conjunto de ejercicios propuestos en las categorías en las que se centra este TFG, así como el diseño de nuevos retos para otras categorías diferentes. También la posibilidad de presentar ejercicios CTF que no pertenezcan a la modalidad jeopardy, sino que sean de defensa-ataque, por ejemplo.

REFERENCIAS BIBLIOGRÁFICAS

Equipo de InfSeg. (8 de Marzo de 2017). "Captura la bandera (CTF)" en InfSeg, [fecha de consulta: 7 febrero 2019]. Disponible en: https://infseg.com/seguridad/captura-la-bandera-ctf/.

Jornadas Nacionales de Investigación en Ciberseguridad [en línea] [fecha de consulta: 20 mayo 2019]. Disponible en: https://2019.jnic.es/.

LANZ L. (22 de Mayo de 2018). "¿Qué es la ciberseguridad?" en OpenWebinars, [fecha de consulta: 15 marzo 2019]. Disponible en: https://openwebinars.net/blog/que-es-la-ciberseguridad/.

GIMÉNEZ SOLANO, V. Hacking y Ciberdelito [en línea]. Trabajo Fin de Grado. Universidad Politécnica de Valencia, 2011. [Fecha de consulta: 20 mayo 2019]. Disponible en: https://riunet.upv.es/handle/10251/11856.

MOTOS, V. (18 de Octubre de 2013). "Crea tu propio CTF con Mellivora" en Hack Players, [fecha de consulta: 10 febrero 2019]. Disponible en: https://www.hackplayers.com/2013/10/crea-tu-propio-ctf-con-mellivora.html.

Equipo de ilpABOGADOS. (5 de Febrero de 2019). "¿Qué es el Hacking Ético?" en ilpABOGADOS, [fecha de consulta: 20 mayo 2019]. Disponible en: https://www.ilpabogados.com/que-es-el-hacking-etico/

FREEMAN, ROB. (12 Octubre 2016). "Ethical hacking: what is it, and why would I need it?" en IT Governance Blog, [fecha de consulta: 20 mayo 2019]. Disponible en: https://www.itgovernance.co.uk/blog/ethical-hacking-what-is-it-and-why-would-i-need-it.

SUTHERLAND, I. (2013). "Is Ethical Hacking Actually Ethical or even Legal?" en IAN SUTHERLAND, [fecha de consulta: 20 mayo 2019]. Disponible en: https://ianhsutherland.com/ethical-hacking/.

Instituto Nacional de Ciberseguridad de España. Qué es INCIBE [en línea] [fecha de consulta: 20 mayo 2019]. Disponible en: https://www.incibe.es/que-es-incibe.

PABLOS, R. (26 de Febrero de 2014). "CTF: Entrenamiento en seguridad informática" en Instituto Nacional de Ciberseguridad de España (blog), [fecha de consulta: 15 febrero 2019]. Disponible en: https://www.incibe-cert.es/blog/ctf-entrenamiento-seguridad-informatica.

IGLESIA, P. (28 de Abril de 2015). "#MundoHacker: Esteganografía, el arte de ocultar información sensible" en PabloYglesias, [fecha de consulta: 24 abril 2019]. Disponible en: https://www.pabloyglesias.com/mundohacker-esteganografia/.

PAUS, L. (16 de Marzo de 2015). "Esteganografía: ¿cómo ocultar archivos para proteger tu información?" en welivesecurity, [fecha de consulta: 24 abril 2019]. Disponible en: https://www.welivesecurity.com/la-es/2015/03/16/como-ocultar-archivos-para-proteger-informacion/.

SAN EMETERIO, P y ÁVILA, C. (9 de Noviembre de 2017). "La cara oculta de la esteganografía" en Telefónica Digital España (eleven path), [fecha de consulta: 24 abril 2019]. Disponible en: https://www.elevenpaths.com/es/noticias-y-eventos/elevenpaths-talks/la-cara-oculta-de-la-esteganografia/index.html.

Colaboradores de EcuRed. Esteganografía [en línea]. EcuRed: Enciclopedia cubana, 2011 [fecha de consulta: 24 abril 2019]. Disponible en: https://www.ecured.cu/Esteganograf%C3%ADa.

Internet Security Auditor S.L. Informática Forense y Peritajes [en línea] [fecha de consulta: 27 abril 2019]. Disponible en: https://www.isecauditors.com/informatica-forense-peritajes.

SÁNCHEZ, A. (29 de Agosto de 2018). "¿Qué es la informática forense?" en Proteger mi pc, [fecha de consulta: 27 abril 2019]. Disponible en: https://protegermipc.net/2018/08/29/que-es-la-informatica-forense/.

ELAINE. (16 de Mayo de 2018). "Cuáles son los objetivos de la informática forense" en OnRetrieval, [fecha de consulta: 27 abril 2019]. Disponible en: https://onretrieval.com/objetivos-de-la-informatica-forense/.

Consejo General de la Abogacía Española. Análisis forense de los dispositivos móviles [en línea] [fecha de consulta: 27 abril 2019]. Disponible en: https://www.abogacia.es/2017/07/28/analisis-forense-de-los-dispositivos-moviles/.

LogRythm. Análisis forense de la red [en línea] [fecha de consulta: 27 abril 2019]. Disponible en: https://es.logrhythm.com/solutions/security/network-forensics/.

SÁNCHEZ CORDERO, P. (4 de Octubre de 2012). "Wireshark: Editing A Packet" en Conexión inversa, [fecha de consulta: 19 febrero 2019]. Disponible en: http://conexioninversa.blogspot.com/2012/10/cloud-forensics-la-nube-amazon-y-el.html

NUGROHO, Y. (27 de marzo de 2019). "Recon" en Yohan.es, [fecha de consulta: 30 abril 2019]. Disponible en: https://yohan.es/ctf/recon/

FORTUNATO, T. (27 de Octubre de 2015). "CLOUD FORENSICS (La nube, Amazon y el análisis forense)" en Network Computing, [fecha de consulta: 27 abril 2019]. Disponible en: https://www.networkcomputing.com/networking/wireshark-editing-packet

Wireshark. Editcap [en línea] [fecha de consulta: 16 febrero 2019]. Disponible en: https://www.wireshark.org/docs/man-pages/editcap.html.

PRIETO FERNÁNDEZ, R. (3 de Febrero de 2015). "Cómo juntar y dividir ficheros pcap" en Raúl Prieto Fernández, [fecha de consulta: 16 abril 2019]. Disponible en: https://www.raulprietofernandez.net/blog/gnu-linux/como-juntar-y-dividir-ficheros-pcap.

Instituto Internacional de Seguridad Cibernética. ¿Cómo ocultar mensajes secretos en archivos de música? [en línea] [fecha de consulta: 28 febrero 2019]. Disponible en: http://www.iicybersecurity.com/audio-esteganografia.html.

Equipo de Feroa93. (11 de Junio de 2017). "Como crear enlace de descarga directa de Google drive." en Feroa93, [fecha de consulta: 11 marzo 2019]. Disponible en: https://www.feroa93.com/como-crear-enlace-para-descarga-directa-de-google-drive/.

Ocultando mensaje en archivo de audio y posterior descifrado con espectrograma [en línea]. En: Gera2Channel, presentado y dirigido por Gera2Channel, 27 Diciembre 2017 [fecha de consulta: 10 junio 2019]. Disponible en: https://www.youtube.com/watch?v=2vVOvcjRpNA.

KOECKLIN, B. (1 de Agosto de 2014). "Instalación local de Joomla (Con XAMPP)" en Educadictos, [fecha de consulta: 12 abril 2019]. Disponible en: https://www.educadictos.com/instalacion-local-de-joomla-con-xampp/.

Colaboradores de wikiHow. Cómo ponerle una contraseña a un documento Word [en línea]. WikiHow, 2008 [fecha de consulta: 7 febrero 2019]. Disponible en: https://es.wikihow.com/ponerle-una-contrase%C3%B1a-a-un-documento-Word.

MIRA, F. (26 de Octubre de 2014). "Llevar una tabla de Word a una base de datos de Access" en Computer Hoy, [fecha de consulta: 23 abril 2019]. Disponible en: https://computerhoy.com/paso-a-paso/software/llevar-tabla-Word-base-datos-access-18639.

MARTÍNEZ, I. (7 de Mayo de 2016). "Como Migrar tabla de Access a Oracle usando Excel" en Formación Web Online, [fecha de consulta: 23 abril 2019]. Disponible en: http://www.formacionwebonline.com/exportar-tabla-access-oracle-usando-Excel/.

Colaboradores de Wikipedia. Armas de fuego [en línea]. Wikipedia, La Enciclopedia Libre, 2018 [fecha de consulta: 25 abril 2019]. Disponible en: https://es.wikipedia.org/wiki/Categor%C3%ADa:Armas_de_fuego.

BRD_. (15 de marzo de 2019). "Crack Password-Protected Microsoft Office Files, Including Word Docs & Excel Spreadsheets" en Wonder how to, [fecha de consulta: 26 abril 2019]. Disponible en: https://null-byte.wonderhowto.com/how-to/crack-

password-protected-microsoft-office-files-including-Word-docs-Excel-spreadsheets-0193959/.

Como mostrar e ocultar texto no Word [en línea]. En: Conteudotope, presentado y dirigido por Conteudotope, 16 Diciembre 2015 [fecha de consulta: 14 junio 2019]. Disponible en: https://www.youtube.com/watch?v=RyH2KzYAu-o.

VEHON, M. "La mejor forma de recuperar un archivo dañado de Word" en Techlandia. Disponible en: https://techlandia.com/mejor-forma-recuperar-archivo-danado-Word-como_534954/.

TYAGI, T. Data Recovery with & without Programming. En: *Programming for "Raw File" Recovery*. *2004*: *Headers and footers of some important file types* [en línea]. pp. 1-6 [consultado 12 febrero 2019]. ISBN 81-7656-922-4. Disponible en: https://www.datadoctor.biz/author.htm.

Cómo poner contraseña a las entradas de tu blog de blogger 2014 [en línea]. En: teodorus maximus, presentado y dirigido por teodorus maximus, 27 Mayo 2014 [fecha de consulta: 17 junio 2019]. Disponible en: https://www.youtube.com/watch?v=a1MMSDIEtkU.

Como crear un archivo KML/KMZ en Google Earth [en línea]. En: Marcelo Paillali, presentado y dirigido por Marcelo Paillali, 17 Noviembre 2015 [fecha de consulta: 17 junio 2019]. Disponible en: https://www.youtube.com/watch?v=ep3B67K5LzA.

LÓPEZ ARREDONDO, J. (29 de Abril de 2015). "Cómo dividir un archivo en varias partes para compartirlo más fácilmente" en Smart Life, [fecha de consulta: 17 junio 2019]. Disponible en: https://cincodias.elpais.com/cincodias/2015/04/29/lifestyle/1430305496_275674.html.

ERICH, V. (2016). "Simple Captcha Script" en Allwebco, [fecha de consulta: 25 junio 2019]. Disponible en: http://allwebco-templates.com/support/script-simple-captcha.htm.

SÁNCHEZ, A. (4 de Abril de 2017). "Las mejores frase sobre seguridad informática" en Proteger mi PC, [fecha de consulta: 2 julio 2019]. Disponible en: https://protegermipc.net/2017/04/04/las-mejores-frases-sobre-seguridad-informatica/.

ANEXOS

Enlaces de descarga para las herramientas utilizadas en el TFG:

- Exiftool, herramienta utilizada para consultar los metadatos: https://sourceforge.net/projects/exiftool/.
- MetadataTouch, permite modificar los metadatos: http://www.digitalconfidence.com/metadatatouch.html.
- Wireshark, herramienta para el análisis del tráfico de red, todas las versiones: https://www.filehorse.com/es/.
- DeepSound, herramienta de esteganografía para ocultar información en audios: http://jpinsoft.net/deepsound/download.aspx.
- Openstego, herramienta de esteganografía para ocultar información en imágenes: https://openstego.softonic.com/?ex=DSK-1327.3.
- Imagehide, herramienta de esteganografía para ocultar mensajes en imágenes: https://softfamous.com/imagehide/.
- Audacity, programa utilizado para la edición de audio: https://sourceforge.net/projects/audacity/.
- Eraser, herramienta utilizada para el borrado de datos de forma segura, con completa destrucción: https://sourceforge.net/projects/eraser/.
-XAMPP, software para la gestión de bases de datos Mysql, servidor web e intérpretes de lenguajes: https://www.apachefriends.org/es/download.html.
- Joomla, sistema de gestión de contenidos para desarrollar sitios web dinámicos e interactivos: https://downloads.joomla.org/es/
- Gantry, plugin para proporcionar un marco temático a CMS's como Joomla, Wordpress,…: http://gantry.org/downloads.
- Phoca Download, software de gestión de descargas para Joomla: https://www.phoca.cz/download/category/4-phoca-download-component.
- Extplorer, software de gestión de archivos para Joomla: https://extplorer.net/projects/extplorer/files.
- Brutus, herramienta para el crackeo de contraseña, fuerza bruta: https://www.darknet.org.uk/2006/09/brutus-password-cracker-download-brutus-aet2zip-aet2/.

- FTK Imager, herramienta para la construcción y visualización de réplicas de datos: https://accessdata.com/product-download/ftk-imager-version-3.4.3.

- Illuminati-dirigens-cipher, fuente para cifrar textos con símbolos illuminatis: https://fontmeme.com/fonts/illuminati-dirigens-cipher-berlin-version-font/

- Spreadsheets, hoja de cálculo: http://www.byedesign.co.uk/.

- Photo-Stamp-Remover, editor de imágenes, para borrar partes de la misma: https://photo-stamp-remover.uptodown.com/windows).

- ImageUSB (para crear una imagen), OSFClone (para realizar el clonado de una memoria) y OSFMount (para montar una imagen), todas podemos obtenerlas desde: https://www.osforensics.com/tools/.

- Recuva, permite recuperar archivos eliminados: https://www.ccleaner.com/recuva.

- Autopsy, para analizar imágenes de memorias: https://www.autopsy.com/download/.

- HjSplit, para dividir y unir archivos: http://www.hjsplit.org/.

- Google Earth, versión para PC: https://www.google.es/earth/download/gep/agree.html.

- HexBrowser, herramienta para detectar extensiones de archivos: http://hexbrowser.com/.

- HxD, lector hexadecimal: https://hxd-hex-editor.soft32.es/.

- SmartDeblur, software para restaurar imágenes: https://smartdeblur.softonic.com/.

- Spek, para visualizar espectrogramas: http://spek.cc/.

- Coagula Light, herramienta utilizada para crear espectrogramas: http://www.hitsquad.com/smm/programs/Coagula_win32/.

- Chrome driver, driver para conectar con el navegador Chrome desde Eclipse: http://chromedriver.chromium.org/downloads.

Herramientas online:

- Befunky, editor online de imágenes: https://www.befunky.com/es/crear/difuminar-fotos/.

- Geolmgr, herramienta online para incluir coordenadas en una imagen: https://tool.geoimgr.com/

- Superpatanegra, web para codificar y decodificar: http://superpatanegra.com/texto/index.php.

- Cortas, página para acortar urls: http://cortas.elpais.com/.

- ea8brw, web para generar sonidos con código morse: http://www.ea8brw.es/index.php/recursos/cw-telegrafia/generador-de-codigo-morse.

- QR-Code-Generator, página para generar códigos QR: https://es.qr-code-generator.com/.

- Alazar, página para generar cadenas aleatorias: http://www.alazar.info/.

- Barcode Generator, generador de códigos de barras: https://barcode.tec-it.com/es

- Online Barcode reader, lector de códigos de barras: https://www.onlinebarcodereader.com/es.html

Plataformas para subir archivos:

- Imgbb, web donde subir imágenes de forma gratuita: https://es.imgbb.com/.

- Drive, servicio de alojamiento de archivos: https://www.google.es/drive/apps.html.

- Blogger, componente para crear blogs, estos sirven para alojar archivos: https://www.blogger.com.

Algunas de las páginas con información, con las que se completó el trabajo:

- Netresec, archivos pcap gratis: https://www.netresec.com/?page=pcapfiles. Otras como estas, son la página oficial de Wireshark (https://wiki.wireshark.org/) y asecuritysite (https://asecuritysite.com/forensics/pcap).

- elongsound, almacén de sonidos gratis: https://www.elongsound.com/sonidos.html.

- Video utilizado para el reto de Esteganografía: https://www.youtube.com/watch?v=oZ3UmEQHRjI.

- Páginas para consultadas para los nombres ficticios de los retos:

https://www.lifeder.com/apellidos-estadounidenses/, https://www.wattpad.com/96981695-nombres-y-apellidos-para-tus-personajes, https://www.wattpad.com/96973602-nombres-y-apellidos-para-tus-personajes y https://www.enterat.com/actualidad/nombres-apellidos-comunes.php.

- Página consultada para los prefijos de los teléfonos: http://www.abctelefonos.com/indice_usa.