UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Predicción de dominios maliciosos utilizando técnicas de Machine Learning

Trabajo Fin de Máster

Máster Universitario en Ciberseguridad y Ciberinteligencia

AUTOR/A: Palop Alcaide, Fernando

Tutor/a: Monserrat Aranda, Carlos

CURSO ACADÉMICO: 2021/2022

Resumen

La investigación de los dominios que se conectan a las distintas organizaciones y su clasificación como benignos o maliciosos es una tarea que consume una enorme cantidad de recursos en un SOC. Este trabajo trata de facilitar esta tarea mediante su clasificación automática, basándose en técnicas de Machine Learning con aprendizaje supervisado. El objetivo es construir una herramienta que, dada una serie de dominios a investigar, obtenga información de fuentes abiertas y la compare con un set de entrenamiento para ofrecer la probabilidad de que estos dominios sean maliciosos o no. Los resultados de precisión obtenidos por el algoritmo desarrollado permiten concluir que esta técnica no solo es adecuada sino también muy efectiva, por lo que podría ser muy útil para ayudar a priorizar las investigaciones y optimizando así los recursos del SOC.

Palabras clave: aprendizaje automático, aprendizaje supervisado, ciberseguridad, clasificación, dominio, internet, regresión logística.

Abstract

The investigation of domains to establish whether they are dangerous or not for any organization network is a task that consumes a lot of resources within a SOC. This work tries to facilitate the task by making an automatic classification based on Machine Learning techniques with Supervised Learning. The objective is to build a tool that, given a domain list to investigate, obtains information from open sources and compares it with a training set to offer the probability that these domains are malicious or not. The precision results obtained by the developed algorithm allow us to conclude that this technique is not only adequate but also very effective, so it could be very useful to help prioritize investigations and thus optimize SOC resources.

Keywords: machine learning, supervised learning, cybersecurity, classification, domain, internet, logistic regression.

Tabla de contenidos

Índice de figuras

1. Introducción

1.1. Motivación.

A la hora de defender una red, los Centros de Operaciones de Seguridad (SOC por sus siglas en ingles “Security Operations Centers”) se encuentran con multitud de conexiones entrantes y salientes hacia o desde diferentes dominios. Lo habitual es que estos dominios estén realizando tareas legítimas. Sin embargo, puede que detrás de alguna de esas conexiones haya intenciones dañinas que pongan en peligro el sistema. Clasificar rápidamente los dominios como legítimos o maliciosos en función de su intencionalidad permitiría bloquear conexiones potencialmente peligrosas.

Cuando un analista de ciberseguridad se topa con un dominio que no ha visto previamente, necesitará recopilar una gran cantidad de información desde numerosas fuentes hasta ser capaz de emitir un juicio sobre su intencionalidad. El problema es que la tipología de este tipo de dominios es muy variada y no hay una regla fija para clasificarlos. Los mismos parámetros pueden indicar cosas distintas en función de cómo se combinen entre ellos. Al final, será el “olfato” y la experiencia del analista experto el que, a la vista de los datos recopilados, le permita tomar una decisión.

El autor dirige el SOC para redes sin clasificar del ESP-DEF-CERT, uno de los 3 CERTs gubernamentales de referencia en España[1]. La red a la que da servicio este SOC es enorme y está muy expuesta a Internet por lo que la cantidad de conexiones a la que se enfrenta un analista supera con creces la capacidad de su personal para gestionarlas adecuadamente, por lo que la priorización es fundamental. Para el analista sería de enorme ayuda poder automatizar la recopilación de información, pero sería aún más ventajoso si pudiera tener una predicción que le indique la probabilidad de que cada uno de esos dominios sean maliciosos. Esto le permitiría dirigir sus primeros esfuerzos a los más sospechosos.

El bloque de Machine Learning, dentro de la asignatura de Generación de Ciberinteligencia del Máster Universitario en Ciberseguridad y Ciberinteligencia (MUCC), despertó en el autor la duda de si sería posible utilizar algoritmos de machine learning para identificar esos dominios peligrosos, aprovechando la gran cantidad de muestras ya recolectadas por sus sistemas de defensa.

1.2. Objetivos.

El objetivo de este trabajo es desarrollar una prueba de concepto que valide la posibilidad de predecir dominios maliciosos con técnicas de machine learning.

[1] Junto al CCN CERT y al INCIBE CERT, responsables respectivamente de la Administración Pública y del Sector Privado y la ciudadanía, el ESP DEF CERT tiene asignada, principalmente, la protección de las redes del Ministerio de Defensa.

Para que un algoritmo emita un veredicto de clasificación, malicioso/benigno, es necesario recopilar una enorme cantidad de datos previamente. Por ello, en este proyecto se pretende automatizar ese proceso de forma que el analista pueda pasar a la herramienta un listado de dominios a investigar y, de forma totalmente automática, ésta le devuelva los dominios ya clasificados, con la probabilidad resultante y, además, ponga a su disposición toda la información recogida para facilitar la toma de decisiones.

1.3. Limitaciones.

El proyecto pretende ser de utilidad en el SOC del ESP-DEF-CERT, por lo que está orientado a las posibles amenazas para las redes de Defensa y tratará de usar las mismas fuentes de datos que actualmente se usan en él.

Es necesario considerar que este SOC, por la naturaleza de las redes a las que presta servicio, debe centrar sus esfuerzos en la detección de Amenazas Persistentes Avanzadas (APT). Por este motivo, se considera una línea roja la interacción con el dominio investigado, al menos en las fases iniciales. De esta forma evitaremos que el posible atacante se sienta descubierto y abandone el dominio. Por lo tanto, solo usaremos fuentes de datos indirectas que recopilan información de forma automática y no levantan sospechas en el agresor.

Alguna de las fuentes de información, especificadas en el Capítulo 5, no son de uso público y tan solo están disponibles para Organismos del Sector Publico. Las API-keys usadas serán anonimizadas para evitar su uso no autorizado.

1.4. Impacto Esperado.

En el caso de que se obtenga algún algoritmo que ofrezca unos resultados aceptables, por encima del 80%, la herramienta serviría para orientar el trabajo de los analistas hacia los dominios con más probabilidad de ser malignos, minimizando el riesgo de exposición a éstos. Además, la automatización de la recolección de información ahorraría mucho tiempo y permitiría afrontar mayor número de investigaciones. En este caso, el impacto final sería un incremento notable de la capacidad del SOC para identificar y bloquear conexiones peligrosas.

En el supuesto de que no se logre un algoritmo de clasificación fiable, la herramienta que recopila los datos aún seguiría siendo válida y útil para los analistas.

1.5. Metodología.

Se pretende implementar un algoritmo basado en aprendizaje supervisado, es decir, que utilice muestras ya clasificadas. Para la obtención de dominios maliciosos se recurrirá a listas negras mientras que los benignos se obtendrán de los diversos listados disponibles en internet.

Una vez identificados los dominios que constituirán el set de entrenamiento, es el momento de identificar qué información necesita un analista durante su investigación. Además, al tener que descargar información de varios miles de dominios, lo más adecuado es recurrir a fuentes

que dispongan de su propia API. Se desarrollará una herramienta que haga peticiones sucesivas de cada uno de los dominios y guarde sus respuestas en archivos de texto para su parseo posterior.

De forma paralela, se estudiará cada uno de los tipos de archivos de texto que proporciona cada API para identificar, dentro de ellos, la información valiosa para el analista y para el algoritmo de machine learning. Se creará una segunda herramienta que, una vez finalizada la recolección de archivos, pasará por cada uno de ellos extrayendo la información, normalizándola y grabándola en una hoja de cálculo.

La tabla final contendrá mucha más información de la que sería razonable introducir en el algoritmo de aprendizaje automático. Habrá que decidir qué campos de la hoja de calculo van a formar el dataset. En otras palabras, qué datos consideramos no sólo relevantes sino también aptos para un algoritmo de aprendizaje. Para ello, ser realizará un estudio de las diferentes propiedades de un dominio y la justificación de su inclusión o eliminación del dataset.

Se elegirá un algoritmo de clasificación y se evaluará su precisión. En función de los resultados obtenidos, se modificarán los campos y/o su peso específico hasta conseguir un resultado satisfactorio. En caso de no conseguirlo, se cambiará el algoritmo y se volverá a comenzar el proceso de ajuste.

Una vez obtenido un algoritmo satisfactorio, se desarrollará una nueva herramienta para el usuario final que deberá tomar el listado de dominios a investigar, recopilar la misma información obtenida para el entrenamiento, parsearla y presentarla al algoritmo para que este haga sus predicciones y etiquete las nuevas muestras. Esta última aplicación también deberá ordenar los resultados de mayor a menor probabilidad de ser maliciosos y mostrar al analista toda la información recopilada para un análisis más profundo.

1.6. Estructura.

En el Capítulo siguiente “2. Estado del Arte” se analizará cómo se aborda hasta ahora el problema de la identificación de dominios maliciosos.

En el Capítulo “3. Análisis del problema” se analizarán las dos principales problemáticas que se encuentran. Por un lado, la dificultad para identificar y clasificar dominios y, por otro, la complejidad para obtener información normalizada de forma automática.

El Capítulo “4. Selección de muestras” explica el proceso y los criterios utilizados para elegir las muestras etiquetadas que compondrán el set de entrenamiento

El Capítulo “5. Descarga de información sin procesar” se centra en la selección de las distintas fuentes y el uso de sus APIs para automatizar la recogida de información.

El análisis de la ingente cantidad de información recogida para detectar qué elementos serán relevantes tanto para el investigador como para el algoritmo se trata en el Capítulo “6. Extracción de información relevante”.

Seguidamente, en el Capítulo “7. Algoritmo de aprendizaje” se hace una breve descripción del algoritmo de Regresión Logística.

En el Capítulo “8. Construcción del dataset y validación del algoritmo” se estudian los distintos tipos de datos recogidos y se hace la selección de los que conformarán el dataset final. Se dividirá el dataset en “entrenamiento” y “test” y se evaluará su funcionamiento.

Ya con el algoritmo testado y validado, en el Capítulo “9. Predicciones sobre muestras no etiquetadas” se aborda el proceso que permitirá al analista introducir una lista de dominios a investigar y obtener como resultado un listado ordenado con la probabilidad de que cada uno de ellos sea malicioso, junto con toda la información recopilada automáticamente.

El Capítulo 10 aborda el problema del reentrenamiento, o como añadir muestras al set de entrenamiento para evolucionar su capacidad de clasificación.

Por último, en los Capítulos “11. Conclusiones” y “12. Trabajos futuros” se analizan los resultados del trabajo y se proponen posibles áreas de mejora de la herramienta.

2. Estado del arte.

Actualmente, los administradores de red suelen configurar sus dispositivos para que bloqueen conexiones a dominios que previamente han sido identificados como maliciosos y aparecen en listas negras. En este caso, el firewall de salida de la organización leería las listas negras a las que estuviera subscrito, como por ejemplo las del Centro Criptológico Nacional (CCN), y bloquearía todas las conexiones entrantes o salientes relacionadas con esos dominios. Como se puede ver, es necesario que otra organización haya investigado previamente un dominio y lo haya categorizado como maligno. Es cierto que la comunidad de ciberseguridad es muy ágil en la compartición de inteligencia y rápidamente se difunden estos hallazgos. Este sistema es útil porque nos protege de amenazas conocidas, al igual que un antivirus clásico[2] nos defiende del malware conocido, pero es ineficaz contra el nuevo. El problema es cuando el ataque es dirigido y las armas han sido diseñadas específicamente para atacar nuestra organización. En este caso hablamos de dominios creados o adaptados para interaccionar solo con nuestros sistemas y, por lo tanto, no han sido detectados, no están incluidos en listas negras y nuestra defensa perimetral no los detendrá.

También es frecuente bloquear conexiones basándose en la reputación y/o categorización de un dominio. Por ejemplo, no se permitirán conexiones a dominios categorizados como pornografía, malware, compras, etc. o a todos aquellos que tengan una reputación menor que "x", siendo "x" un valor numérico que dependerá de la fuente consultada. La fiabilidad de ambos parámetros depende del número de consultas al dominio y de las "votaciones" que realicen los usuarios. En otras palabras, dominios populares tendrán una categorización bien definida y suficientes reseñas para establecer un nivel de reputación, mientras que los dominios de nueva creación es muy probable que no tengan reputación y/o categorización o sea muy difusa y poco fiable. La ausencia de categorización podría ser un indicador sospechoso en si mismo, pero sería insuficiente para decidir un bloqueo automático. Una vez más, este sistema no es adecuado para combatir ataques dirigidos ya que el atacante puede camuflar su dominio bajo la apariencia de una web de categoría inocua. Incluso podría comprar un dominio ya abandonado que en el pasado tuviera buena reputación[3].

En cuanto a la recopilación de información sí existen soluciones. En el ámbito del sector público, el referente es REYES, desarrollada por el CCN. REYES permite hacer búsquedas por diferentes Indicadores de Compromiso (IoC), recopilando información de numerosas fuentes. Su entorno web es muy útil para investigaciones secuenciales, mostrando distintos tipos de datos en varias pestañas. REYES tiene una API creada para automatizar las búsquedas que, actualmente, está en proceso de actualización.

[2] Los antivirus modernos no se limitan solo a recopilar firmas de malware, sino que analizan el sistema para identificar comportamientos sospechosos.
[3] Ref. 12.

Las técnicas de Machine Learning son cada vez más utilizadas en ciberseguridad[4], tanto en clasificación, por ejemplo, ayudando a determinar si un archivo es malware o no, como en regresión, prediciendo el número de ataques que sufrirá una red, entre otras cosas. También son frecuentes las aplicaciones que, mediante algoritmos de clustering y aprendizaje no supervisado, tratan de detectar anomalías es los sistemas o identificar familias de malware utilizando análisis de comportamiento basado en técnicas heurísticas. Se podría decir que el objetivo inicial del Machine Learning en ciberseguridad sería predecir los ataques antes de que se produzcan para poder implementar medidas preventivas. Sin embargo, la complejidad de los ataques hace que, cada vez más, se centren en su detección en las fases iniciales de forma que se pueda limitar su impacto al mínimo posible. Además de las ya citadas, los principales campos en los que se está aplicando Machine Learning en ciberseguridad son la detección de intrusiones de red, la detección de phishing[5] y spam[6], la evaluación de riesgo, la detección de deepfakes[7] o la detección de ataques contra las Terminales de Punto de Venta (TPV).

Lo que el autor no ha encontrado es una herramienta como la que plantea este trabajo, que une la recopilación automática de información sobre dominios con su posterior tratamiento con técnicas de machine learning para hacer predicciones sobre su intencionalidad.

[4] Refs. 13 y 14.
[5] Estafa que tiene como objetivo obtener datos privados de los usuarios, especialmente sus cuentas o datos bancarios, a través de correos electrónicos en los que se suplanta a un remitente legítimo.
[6] Correo electrónico no solicitado que se envía a un gran número de destinatarios con fines publicitarios o comerciales.
[7] Contenido falso, frecuentemente imágenes o videos, producidos utilizando técnicas de inteligencia artificial.

3. Análisis del problema

3.1. Identificación de dominios maliciosos

El tiempo de reacción y los recursos humanos y materiales en un SOC son siempre escasos. Sus actividades deben perseguir la máxima eficacia. Para lograrlo es conveniente seguir el modelo conocido como "pirámide del dolor"[8]. Este concepto muestra la relación entre los tipos de indicadores que se pueden utilizar para detectar las actividades de un adversario y la cantidad de "dolor" que le infringimos cuando logramos bloquear esos indicadores. Según esto, los SOC deberían centrarse en la detección y el descubrimiento de las técnicas, tácticas y procedimientos (TTP) del atacante.

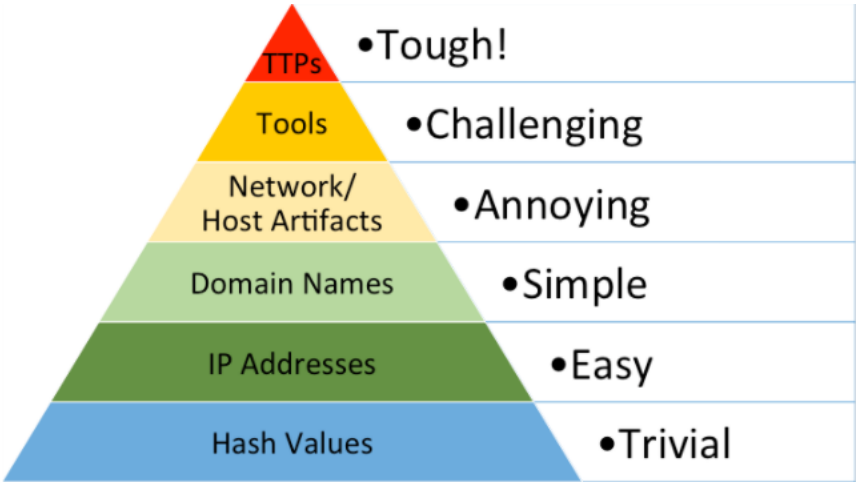

Figura 1. Pirámide del dolor

Aunque los dominios están situados varios niveles por debajo en la cúspide de la pirámide, si lográsemos analizarlos masivamente y obtener una predicción de su peligrosidad, se podrían descubrir ataques complejos. En un SOC que protege las redes de Ministerio de Defensa, el objetivo último de la detección de dominios maliciosos no es tanto su bloqueo inmediato, sino la posibilidad de descubrir la infraestructura de mando y control de grupos APT. Al ser capaz de establecer relaciones y patrones entre algunos de los dominios detectados, se puede tratar de identificar las TTPs del atacante para causarle el mayor "dolor" posible.

El tiempo de detección de una amenaza persistente avanzada (APT) está estimado en dos años, periodo más que suficiente para causar un tremendo daño, tanto económico como reputacional. Automatizar el análisis y la predicción para descubrir los dominios que el grupo APT está usando podría reducir drásticamente el impacto del ataque o incluso evitar que supere las fases iniciales.

[8] La "*pirámide del dolor*" es un concepto creado por David Bianco en 2013. Ver Ref. 2.

El mayor problema al que se enfrenta un analista es la variedad de tipologías que existen entre los dominios maliciosos, que hace imposible establecer una regla fija que permita clasificarlos atendiendo a la presencia o ausencia de determinadas características. Será su experiencia y su instinto lo que le haga inclinarse por una opción u otra. Nos moveremos siempre en el campo de las sospechas, los pequeños indicios que hacen desconfiar al analista. Ahora bien, el proceso es diferente en cada caso.

Para ejemplificar esta problemática, se describen a continuación algunas tipologías de dominios maliciosos con características muy distintas entre si:

- Dominios de reciente creación: Técnica usada en ataques para esquivar las listas negras de las organizaciones o proveedores de inteligencia de los diferentes dispositivos desplegados en la defensa perimetral.
- Dominios sin categorización y/o reputación: Técnica similar a la anterior con la diferencia de que no siempre son dominios de nueva creación, también pueden ser dominios creados hace tiempo y usados solo por atacantes con capacidades avanzadas en determinados momentos de la explotación. Con esta técnica es fácil evadir la defensa perimetral de las organizaciones.
- Dominios registrados en determinados registradores de nombres, en algunos TLD (Top Level Domain[9]) concretos o en algunos países que no ejerzan mucho control sobre ellos. En muchos casos, éste será un indicador de la capacidad del adversario tanto económica como técnica.
- Dominios antiguos con buena reputación y categorización pero que recientemente han cambiado de dueño al no renovar su contrato el anterior propietario: Técnica muy avanzada usada por atacantes con grandes capacidades ofensivas. Necesita de un estudio previo de los dominios que se dan de baja, análisis de la información que contenía, de la reputación e incluso de la información que ofrecía para poder hacer un clonado del contenido tratando de confundir al analista en caso de investigación. Son muy peligrosos ya que pasarían inadvertidos tanto en los dispositivos de la defensa perimetral como a los ojos de los analistas.
- Dominios sin copias en Archive.org[10] o que, de repente, dejan de permitirlas: Esto puede ser un indicador de reciente creación, baja reputación y/o cambio de propietario.
- Dominios que aparentemente ofrecen un servicio comercial pero que no responden a conexiones web (http o https): Técnica usada para ocultar dominios de mando y control que solo permiten las conexiones bajo un patrón determinado por el atacante, como, por ejemplo, un "*User-Agent*"[11] especifico.
- Dominios alojados en un servidor cloud con determinados puertos abiertos: La existencia de algunos puertos abiertos como el 22/shs o 389/rdp son indicadores de

[9] Para más información sobre el TLD y su importancia consultar Ref. 4.
[10] Archive.org es una enorme biblioteca multimedia que almacena contenidos tales como música, libros, películas, fotografías, etc. Uno de sus servicios es la Wayback Machine, que realiza copias periódicas de la mayoría de las páginas web de Internet.
[11] El User-Agent es la firma que deja el navegador o la aplicación con el que se accede a un recurso web.

que puede tratarse un servidor virtual privado (VPS). Estos puertos normalmente son usados para administrar el servidor. Este indicador es muy importante para ayudar al analista a determinar si puede tratarse de un servidor de la infraestructura del atacante.

- Un dominio comercial que permite conexiones https, pero cuyo certificado está autofirmado o ha expirado: Este indicador muestra al analista que el atacante no ha sido lo suficiente cuidadoso a la hora de configurar el dominio, bien sea por la prisa en desplegar una campaña, por falta de conocimientos e incluso de recursos económicos, subestimando las capacidades defensivas y técnicas de los analistas y en mayor medida la de los objetivos.
- Dominios con nombres aparentemente aleatorios: Técnica usada por algunos tipos de malware.
- Cantidad de dominios alojados en una misma IP: En muchos casos, los atacantes se ocultan en proveedores de servicios para pasar inadvertidos, pero en otras ocasiones el hecho de que sólo un dominio responda en la IP puede ayudar al analista a determinar que es malicioso.

Cada uno de estos indicios por sí solo podría ser un indicador de la peligrosidad del dominio, pero al considerarlos todos en conjunto, cada uno con su peso específico, se puede llegar a establecer con un mayor grado de certeza si estamos ante un dominio malicioso o no.

La tipología es tan amplia como la imaginación de los atacantes para burlar la detección. Hay que tener en cuenta que cada vez que un dominio malicioso es identificado se convierte en inservible y el atacante debe generar otro para remplazarlo.

Como hemos visto, no existe una metodología claramente definida. Serán las características del dominio las que irán determinando la dirección de la investigación[12]. De este modo, un analista suele pasar por las siguientes fases durante su análisis (se añade entre paréntesis el servicio que suele usarse en el SOC del ESP-DEF-CERT):

a) Buscar apariciones del dominio en incidentes anteriores (LUCIA).
b) Confirmar que el dominio no está incluido en listas negras, ni implicado en incidentes de ciberseguridad compartidos por otras entidades (REYES: CIF, MISP).
c) Comprobar su categorización y reputación (Cisco TALOS).
d) Comprobar la clasificación que le otorgan diversos antivirus (VirusTotal).
e) Consultar la información de Whois, creación, caducidad, actualización, país de registro, organismo registrador, datos del registrante, servidores DNS que resuelven el dominio, etc. (REYES: Whois + DomainTools).
f) Consultar las IPs que históricamente han alojado el dominio y buscar posibles escaneos a la IP. Analizar los puertos abiertos, su geolocalización, el número y tipo de dominios que aloja. (REYES: Shodan + Censys)

[12] Un ejemplo interesante de investigación sobre dominios maliciosos puede encontrarse en la Ref. 3.

g) Consultar los certificados SSL de sus diferentes servicios en busca de elementos sin firmar o auto-firmados, pre-certificados, o firmados por CAs poco fiables (Reyes: Shodan+Censys, Crt.sh[13])
h) Consultar copias de la página web si las hubiera (Archive.org).
i) Consultar las respuestas a peticiones web en servicios externos que navegan hasta el dominio solicitado y muestran una captura de lo que devuelven. Este tipo de interacción podría alertar al atacante por lo que es mejor evitarla (UrlScan[14]).

Por otra parte, no todos los dominios maliciosos persiguen el mismo objetivo. Baste como ejemplo el repositorio de listas negras del CCN. En él encontramos listados de dominios de descarga de malware, de distribución de phishing, de mando y control para ransomware o botnets, balizas de Cobal Strike[15], etc. Aunque este trabajo parte de la premisa de que todos ellos tienen ciertas características comunes, lo ideal sería desarrollar un algoritmo que fuera capaz de distinguir entre diferentes tipos de intencionalidad.

Como se ve, la clasificación de dominios es una tarea compleja que consume tiempo y recursos en un SOC. El tiempo es el factor más crítico ya que la rápida identificación y bloqueo del ataque es vital para minimizar su impacto. Teniendo en cuenta que el número diario de dominios a investigar suele ser elevado, disponer de una herramienta que oriente al analista hacia los que de forma automática se identifican como sospechosos supondría una ayuda muy importante.

3.2. Automatización de la recogida de información.

El primer escollo que se ha encontrado al desarrollar este trabajo ha sido la dificultad para obtener información de forma normalizada. Unos de los principales requisitos de los dataset en machine learning es que los datos de las diferentes muestras estén siempre en el mismo formato y que se minimicen los campos sin valor. Conseguir esto ha sido un proceso muy costoso.

Al realizar acciones tan sencillas como hacer un *whois* sobre un dominio, se obtenía información en formatos completamente diferentes en función de que servidor devolviera el resultado. Esto, que no es un problema cuando es un humano el que lee los datos, se convierte en un serio inconveniente cuando es un software el que debe interpretar los resultados.

Para automatizar la búsqueda de información es necesario recurrir a fuentes que dispusieran de una API, lo cual, salvo alguna excepción, no es un problema. Lo normal es que las APIs devuelvan la información solicitada en formato *json*, fáciles de trabajar excepto cuando tratan las ausencias de determinados campos de forma distinta, lo que provocaba errores en el programa que parseaba los datos (por ejemplo, la ausencia de información sobre la fecha de

[13] https://crt.sh
[14] https://urlscan.io
[15] Cobal Strike es una herramienta de seguridad legitima diseñada para simular el comportamiento de atacantes durante pruebas de penetración. En los últimos años se está haciendo muy popular entre los ciberdelincuentes reales que la utilizan en sus ataques reales.

actualización del dominio puede aparecer como “updatedDate: nil” o simplemente no aparecer la etiqueta “updatedDate”). El nombre de las etiquetas variaba de unos servidores a otros, los formatos de las fechas eran muy diversos, la estructura de los *json* cambia continuamente, la forma de reportar errores no siempre coincide, etc. Una gran proporción el desarrollo se ha invertido en parsear la información recibida hasta obtener un set bastante completo y con formatos normalizados.

Otro inconveniente en la recopilación de información es que las principales fuentes funcionan por suscripción (Shodan, VirusTotal, Censys, RiskIq, etc.). Aunque generalmente disponen de cuentas gratuitas, éstas están bastante limitadas en el número de peticiones que se puede realizar al día. Cuando hablamos de un dataset de varios miles de dominios, este hecho obliga a fragmentarlo e irlo descargando progresivamente. Inicialmente REYES pareció la solución a este problema ya que el autor consiguió del CCN una cuenta privilegiada con suficientes peticiones para reducir la fragmentación a pocos días. Sin embargo, la API de REYES resulto tener serios problemas que han reducido su uso a unas pocas fuentes, obligando a recurrir a suscripciones gratuitas para las demás.

4. Selección de muestras

La construcción de un buen set de adiestramiento resulta esencial para conseguir que el algoritmo de machine learning aprenda correctamente. Como ya se ha mencionado, se va a realizar un aprendizaje supervisado por lo que deberemos etiquetar las muestras como maliciosas o benignas. Para ello, hay algunos factores que es necesario considerar:

- El número de muestras será determinante. Cuanto mayor sea el set, mayores posibilidades habrá de que el algoritmo aprenda a categorizar los dominios. Sin embargo, este aspecto va a ser muy limitativo ya que la disponibilidad de dominios maliciosos de los que extraer información es limitada.
- Como se vio en el Capítulo anterior, la información sobre los dominios no esta estandarizada, ni en contenido ni en formato. El mayor reto para este trabajo consiste en automatizar la descarga de información de forma que se pueda construir una tabla uniforme.
- La cantidad de dominios maliciosos que circulan por internet es enorme y está en continuo crecimiento. Sin embargo, su proporción es ridícula si la comparamos con el número de dominios benignos. Tratar de reflejar esta proporción en el set de adiestramiento haría que las muestras estuvieran muy desbalanceadas, con lo que el algoritmo no aprendería adecuadamente. La solución de compromiso ha sido construir un set con un 55% de muestras benignas y un 45% de muestras maliciosas. Este factor será también determinante en las sucesivas fases de reentrenamiento pues si solo añadimos muestras ya identificadas como peligrosas, el algoritmo acabará pensando que todo Internet se ha vuelto maligno.

4.1. Selección de dominios maliciosos.

Para la obtención de los dominios maliciosos se ha recurrido a las listas negras disponibles en REYES. En estas listas se van incluyendo regularmente los dominios que el CCN ha clasificado como maliciosos a través de sus diversos canales de inteligencia. Estas listas, agrupadas en diferentes categorías, son habitualmente usadas por los administradores de seguridad para bloquear las navegaciones desde sus redes hasta estos dominios o viceversa.

Los atacantes son muy ágiles a la hora de crear nuevos dominios con intenciones ilícitas. También son eficaces a la hora de detectar si sus dominios han sido identificados y rápidamente los abandonan o los reconvierten en legítimos o inocuos. Por este motivo, es necesario actualizar las listas de bloqueo de dominios. De lo contrario, éstas crecerán hasta el punto de sobrepasar la cantidad gestionable por los dispositivos de red encargados de implementarlas (rúters, firewalls, etc.). Un dominio malicioso encontrado hace un año probablemente haya dejado de ser utilizado por los atacantes hace muchos meses. Incluso es muy probable que, si buscamos su información meses después de su detección, la que encontremos ya no se parezca a la que tenía cuando estaba activo. Por este motivo es necesario establecer una fecha de descarga de estas listas y proceder inmediatamente con la obtención automática de sus propiedades.

Para este trabajo se eligió la fecha del 06/02/2022. En ese momento, el CCN ponía a disposición de sus usuarios registrados diferentes listados de dominios, de entre los que se seleccionaron los siguientes:

- Dominios asociados a ataques de ransomware
- Dominios de distribución de malware
- Dominios relacionados con ataques mediante balizas de Cobalt Strike
- Dominios relacionados con ataques de malware Emotet[16]
- Dominios asociados a las vulnerabilidades de Log4j de Apache[17]

De entre estos listados se ha hecho una selección de dominios, que incluían la práctica totalidad de los de ransomware, Cobalt, Emotec, Log4j y una buena parte de los de distribución de malware. Se han eliminado duplicidades y aquellos que dieron problemas al obtener información sobre ellos. Finalmente, quedó un listado de 1562 dominios que fueron etiquetados genéricamente como "maliciosos" para el adiestramiento del sistema de aprendizaje.

Es interesante estudiar la distribución de TLDs de los dominios descargados. Como se observa en la Figura 2, entre los 1562 dominios obtenidos encontramos 96 TLDs distintos, siendo el principal representante "com" con 846 dominios, lo que equivale al 54,16% del total. Les sigue a mucha distancia "xyz" con 85 muestras (5,44%), "top" con 66 muestras (4,22%) y "net" con 59 (3,78%).

Esta distribución es muy significativa pues nos obliga a seleccionar dominios legítimos con representación de, al menos, todos estos TLDs. De lo contrario, si en todo el set de adiestramiento solo hubiera un dominio con un TLD determinado, por ejemplo ".kr", y éste estuviera etiquetado como malicioso, habría muchas posibilidades de que el algoritmo de aprendizaje dedujera que todos los dominios ".kr" también lo son.

[16] Emotet fue originalmente un troyano que robaba información bancaria pero que ha ido evolucionando y añadiendo servicios a su catalogo como el envío de spam o la distribución de otros malwares.
[17] Dominios que han sido relacionados con la explotación de las vulnerabilidades CVE-2021-44228, CVE-2021-45046, CVE-2021-4104 y CVE-2021-45105

#	TLD	nº muestras	#	TLD	nº muestras	#	TLD	nº muestras
1	com	846	33	quest	5	65	tw	1
2	xyz	85	34	club	5	66	re	1
3	top	66	35	co	5	67	fun	1
4	net	59	36	us	4	68	investments	1
5	org	46	37	uk	4	69	fr	1
6	de	44	38	bar	4	70	be	1
7	tk	34	39	eu	4	71	su	1
8	website	32	40	za	3	72	at	1
9	site	31	41	ca	3	73	wtf	1
10	ml	18	42	ovh	3	74	hk	1
11	ru	15	43	digital	3	75	ec	1
12	online	13	44	ga	3	76	it	1
13	cf	11	45	cyou	3	77	my	1
14	info	11	46	click	3	78	casa	1
15	live	10	47	mx	2	79	studio	1
16	space	9	48	nl	2	80	io	1
17	shop	9	49	art	2	81	rest	1
18	cloud	9	50	agency	2	82	blog	1
19	tech	9	51	gq	2	83	red	1
20	me	8	52	world	2	84	best	1
21	pw	8	53	au	2	85	mg	1
22	cab	8	54	zone	2	86	work	1
23	in	8	55	pro	2	87	trade	1
24	fi	8	56	ir	2	88	systems	1
25	link	8	57	fyi	2	89	tf	1
26	cn	7	58	az	1	90	asia	1
27	nu	7	59	se	1	91	vip	1
28	cfd	7	60	gr	1	92	app	1
29	one	7	61	nz	1	93	tr	1
30	today	6	62	biz	1	94	lu	1
31	cc	5	63	ro	1	95	cl	1
32	br	5	64	id	1	96	kr	1
							Total general	**1562**

Figura 2. TLDs con dominios maliciosos en el set de entrenamiento

4.2. Selección de dominios benignos

El número de dominios no maliciosos viene definido por la necesidad de balancear las muestras. Se ha intentado que la diferencia entre ambas categorías sea alrededor del 10%, lo que nos da una cantidad de 1938 dominios benignos. Con estas cifras, el dataset estaría formado por 3500 dominios.

En la selección de los dominios no maliciosos se ha atendido a los siguientes factores:

- Para evitar interpretaciones erróneas del algoritmo, es necesario que haya representación de, al menos, todos los TLDs que tienen representantes maliciosos entre la muestra (todos aquellos que aparecen en la Figura 2). Para esto se ha recurrido a la información que proporciona la página web de W3Technologies[18]. En esta web encontramos un análisis estadístico del uso de los TLDs, como muestra la Figura 2.

[18] https://w3techs.com/technologies/overview/top_level_domain

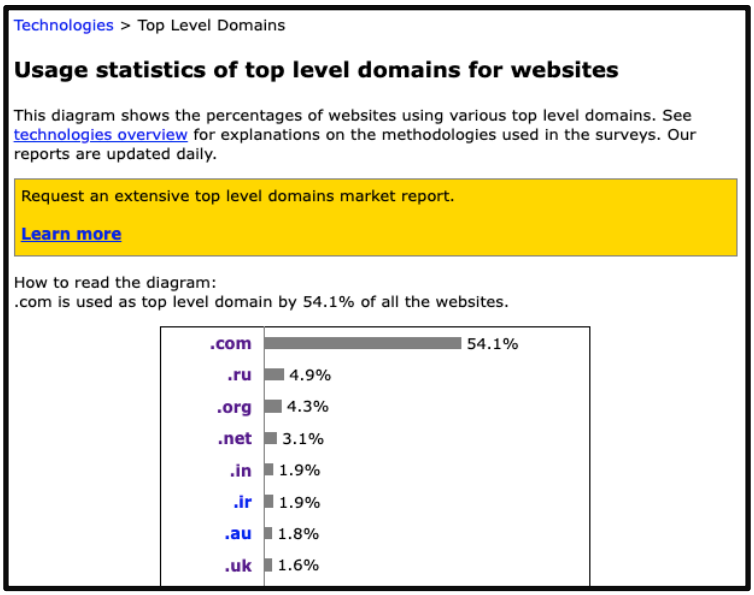

Figura 3. Distribución de TLDs en internet

Esta página también presenta enlaces a todos y cada uno de los posibles TLDs en los que se muestran sus estadísticas de uso, un listado de entre 10 a 15 ejemplos de dominios populares y otros 5 a 10 dominios aleatorios que usan este TLD. Esto es importante porque cuando se recurre a rankings de dominios populares, normalmente no aparecen representantes de los menos representativos pero que son importantes para el algoritmo de aprendizaje.

- Se ha utilizado otra fuente popular en el análisis de dominios, "*The top 500 sites on the web*"[19] de Alexa (Amazon). En esta web encontramos un listado de popularidad global y otro por países. Se han incorporado al set los 50 dominios listados en el ranking global más los que aparecen en una selección de países próximos o significativos por su relevancia. En concreto se recopilaron listados de España, Francia, Estados Unidos, Rusia, Marruecos, China, Gran Bretaña e Irán. Estos listados presentaban múltiples duplicidades que fueron eliminadas. Hasta este punto se obtuvieron aproximadamente 1000 dominios.

- El set se completó hasta las 1938 muestras con otra fuente muy popular: el listado de "Majestic Million"[20]. Este recurso permite descargar un fichero *csv* con el ranking del millón de sitios web más visitados del mundo, actualizado diariamente. Para huir de los dominios extremadamente populares, ya recogidos en los listados anteriores, se seleccionó una muestra aleatoria de 1000 dominios entre los 500.000 primeros resultados. Las duplicidades existentes con los listados anteriores también fueron eliminadas.

[19] https://www.alexa.com/topsites
[20] https://majestic.com/reports/majestic-million

Es importante señalar que, al recurrir a dominios populares, prácticamente se asegura que no se introducen muestras maliciosas y se etiqueten como benignas. Por otro lado, debemos incluir dominios que no estén entre los primeros puestos de popularidad, ya que presumiblemente tendrán otro tipo de características que el algoritmo también necesita aprender.

Analizando el set de muestras benignas final, según muestra la Figura 4, encontramos que los dominios "com" copan "*tan solo*" el 29,2% de los registros, frente al 54% aproximado que existe en internet. Si bien las muestras recogidas con los factores expuestos anteriormente sí reflejaban este porcentaje, se ha decido reducir su número hasta casi un 30%, para así dar cabida a una mayor representación de TLDs.

#	TLD	nº muestras	#	TLD	nº muestras	#	TLD	nº muestras	#	TLD	nº muestras
1	com	566	35	az	15	69	lu	8	103	tr	1
2	org	96	36	tech	15	70	it	8	104	by	1
3	ru	50	37	co	15	71	click	7	105	tw	1
4	edu	42	38	top	15	72	ml	7	106	gd	1
5	net	40	39	live	15	73	gq	7	107	tools	1
6	uk	36	40	online	15	74	ovh	6	108	ms	1
7	gov	34	41	world	15	75	cyou	6	109	travel	1
8	fr	27	42	fi	14	76	cab	5	110	sk	1
9	cn	25	43	be	14	77	tf	5	111	bnpparibas	1
10	br	24	44	shop	14	78	systems	5	112	ee	1
11	de	23	45	hk	14	79	bar	5	113	gal	1
12	ca	22	46	gr	14	80	tk	5	114	no	1
13	nl	21	47	one	14	81	cfd	5	115	my	1
14	eu	19	48	nz	14	82	tv	5	116	su	1
15	in	19	49	re	13	83	ga	4	117	md	1
16	info	19	50	mx	13	84	au	4	118	is	1
17	us	18	51	jp	13	85	investments	3	119	pe	1
18	me	18	52	blog	13	86	hr	3	120	lv	1
19	io	17	53	space	13	87	za	3	121	ph	1
20	es	16	54	asia	13	88	cz	3	122	earth	1
21	vip	16	55	digital	11	89	trade	3	123	dev	1
22	club	16	56	ir	11	90	agency	3	124	wiki	1
23	link	16	57	ma	11	91	ua	3	125	vn	1
24	biz	16	58	fun	11	92	pro	3	126	ec	1
25	id	16	59	nu	10	93	rest	3	127	mil	1
26	cc	15	60	cf	10	94	ro	3	128	ar	1
27	site	15	61	zone	10	95	fyi	3	129	fm	1
28	pw	15	62	best	9	96	red	3	130	wtf	1
29	cl	15	63	studio	9	97	quest	3	131	to	1
30	today	15	64	website	9	98	ch	2	132	lb	1
31	xyz	15	65	mg	9	99	int	2	133	at	1
32	app	15	66	art	9	100	pl	2	134	kpmg	1
33	cloud	15	67	work	9	101	news	2	135	kr	1
34	se	15	68	casa	8	102	ly	2		**Total general**	**1938**

Figura 4. Distribución de TLDs en las muestras no maliciosas

Finalmente, en la Figura 5 se puede observar el set completo y como no existe ningún TLD con solo representantes maliciosos.

#	TLD	Benigno	Malicioso	Total	#	TLD	Benigno	Malicioso	Total	#	TLD	Benigno	Malicioso	Total
1	com	566	846	1412	47	es	16		16	93	trade	3	1	4
2	org	96	46	142	48	az	15	1	16	94	red	3	1	4
3	xyz	15	85	100	49	app	15	1	16	95	ua	3		3
4	net	40	60	100	50	cl	15	1	16	96	cz	3		3
5	top	15	66	81	51	gr	14	1	15	97	hr	3		3
6	de	23	44	67	52	hk	14	1	15	98	ec	1	1	2
7	ru	50	15	65	53	mx	13	2	15	99	pl	2		2
8	site	15	31	46	54	be	14	1	15	100	wtf	1	1	2
9	edu	42		42	55	nz	14	1	15	101	tw	1	1	2
10	website	9	32	41	56	digital	11	3	14	102	news	2		2
11	uk	36	4	40	57	re	13	1	14	103	tr	1	1	2
12	tk	5	34	39	58	asia	13	1	14	104	su	1	1	2
13	gov	34		34	59	blog	13	0	13	105	my	1	1	2
14	cn	25	7	32	60	cab	5	8	13	106	int	2		2
15	info	19	11	30	61	ir	11	2	13	107	ly	2		2
16	br	24	5	29	62	jp	13		13	108	ch	2		2
17	fr	27	1	28	63	zone	10	2	12	109	at	1	1	2
18	online	15	13	28	64	fun	11	1	12	110	kr	1	1	2
19	in	19	8	27	65	cfd	5	7	12	111	ee	1		1
20	me	18	8	26	66	ma	11		11	112	lv	1		1
21	live	15	10	25	67	art	9	2	11	113	lb	1		1
22	ca	22	3	25	68	studio	9	1	10	114	pe	1		1
23	ml	7	18	25	69	best	9	1	10	115	ar	1		1
24	tech	15	9	24	70	work	9	1	10	116	ph	1		1
25	cloud	15	9	24	71	mg	9	1	10	117	no	1		1
26	link	16	8	24	72	click	7	3	10	118	to	1		1
27	eu	19	4	23	73	lu	8	1	9	119	dev	1		1
28	shop	14	9	23	74	gq	7	2	9	120	by	1		1
29	pw	15	8	23	75	cyou	6	3	9	121	md	1		1
30	nl	21	2	23	76	bar	5	4	9	122	tools	1		1
31	us	18	4	22	77	ovh	6	3	9	123	kpmg	1		1
32	fi	14	8	22	78	casa	8	1	9	124	earth	1		1
33	space	13	9	22	79	it	8	1	9	125	vn	1		1
34	club	16	5	21	80	quest	3	5	8	126	mil	1		1
35	cf	10	11	21	81	ga	4	3	7	127	wiki	1		1
36	today	15	6	21	82	za	3	3	6	128	fm	1		1
37	one	14	7	21	83	au	4	2	6	129	sk	1		1
38	co	15	5	20	84	tf	5	1	6	130	travel	1		1
39	cc	15	5	20	85	systems	5	1	6	131	gal	1		1
40	io	17	1	18	86	pro	3	2	5	132	ms	1		1
41	world	15	2	17	87	tv	5		5	133	gd	1		1
42	vip	16	1	17	88	agency	3	2	5	134	is	1		1
43	biz	16	1	17	89	fyi	3	2	5	135	bnpparibas	1		1
44	nu	10	7	17	90	investments	3	1	4		**Total general**	**1938**	**1562**	**3500**
45	id	16	1	17	91	ro	3	1	4					
46	se	15	1	16	92	rest	3	1	4					

Figura 5. Distribución global de TLDs en el dataset

5. Descarga de información sin procesar

Tras definir los dominios que conformarán el dataset, llega el momento de buscar fuentes que suministren la información requerida. Partiendo de los pasos descritos en el Capítulo 3.1, se automatizó el proceso usando las mismas aplicaciones.

Se desarrolló un script de Python llamado "1_obtencionArchivos_DataSet.py" que lee secuencialmente los nombres de dominios y comienza a hacer peticiones a diferentes APIs. En este punto surgían dos opciones de trabajo: parsear los datos directamente y guardar solo la información relevante en una tabla Excel o, como finalmente se decidió, guardar todos los ficheros de texto en una carpeta y posteriormente analizarlos y parsearlos. Esta opción permitía iniciar el proceso de descarga incluso antes de tener claro qué información se iba a extraer de los archivos.

De cada dominio se generaron 4 archivos con información, lo que para un dataset de 3500 dominios suponen 14000 archivos. Por cada dominio se tardaba entre 15 y 30 segundos en recopilar sus 4 archivos correspondientes, después se hace una pausa de 10 segundos para no saturar a las APIs. Como las suscripciones a las distintas APIs estaban limitadas, el script corría hasta que alguna de ellas daba error por superar el número de peticiones autorizado. En ese momento se detenía y había que esperar al día siguiente para volver a lanzar el proceso desde donde se había parado.

Todos los archivos que se crean se almacenan en la carpeta "infoDataset". Además de la creación de los archivos de texto, el script muestra una salida por pantalla que informa del progreso de la recogida de datos y permite saber al recolector del dataset donde se ha interrumpido el proceso para continuar una vez levantadas las restricciones.

A continuación, se muestra la salida por pantalla para los dos primeros dominios descargados.

Las APIs utilizadas fueron Reyes, VirusTotal, Shodan y Archive. Seguidamente se verán cada una de ellas en detalle.

```
==========================================================
numero de filas: 3500
DOMINIO: cdn-jquery.nl ( 1 / 3500 )
No hay errores, continua la extración
numero de resoluciones historicas 1
Ultima IP asociada: 34.244.201.1
Obteniendo información en VIRUSTOTAL sobre  cdn-jquery.nl
Obteniendo información en SHODAN sobre IP: 34.244.201.1
No hay información en Shodan, buscando en historico
Informacion histórica encontrada en Shodan
Obteniendo información en ARCHIVE sobre  cdn-jquery.nl
Fecha primera copia: 9999-12-31
Total desde 2021: 0
9999-12-31,9999-12-31,0
==========================================================
esperando 10 segundos...
==========================================================
DOMINIO: www.agoegations.com ( 2 / 3500 )
No hay errores, continua la extración
numero de resoluciones historicas 8
Ultima IP asociada: 104.21.30.53
Obteniendo información en VIRUSTOTAL sobre  www.agoegations.com
Obteniendo información en SHODAN sobre IP: 104.21.30.53
Obteniendo información en ARCHIVE sobre  www.agoegations.com
Fecha primera copia: 9999-12-31
Total desde 2021: 0
9999-12-31,9999-12-31,0
==========================================================
esperando 10 segundos...
==========================================================
DOMINIO: tubaho.com ( 3 / 3500 )
```

Figura 6. Salida por pantalla del scritp "1_obtencionArchivos_Dataset.py"

5.1. Reyes

Los analistas del SOC del ESP-DEF-CERT basan sus investigaciones en la herramienta Reyes[21]. Reyes facilita la investigación de ciberincidentes, recopilando información de diversas fuentes y cotejando en plataformas de cooperación como MISP[22]. Además, está conectada a la instancia central de la herramienta de Ticketing[23] del CCN (LUCIA). Al introducir un Indicador de Compromiso (IoC), la herramienta muestra gran cantidad de información relacionada y, también, proporciona información sobre si está ligado con algún incidente de los compartidos en MISP o de los ciberincidentes registrados en LUCIA en el ámbito de la Administración. Esta herramienta recopila información de servicios como VirusTotal, Whois, Censys, RiskIq, Shodan, DomainTools, etc., con los que tiene suscripciones contratadas. Su API permite hacer búsquedas por dominio y especificarle que servicios se van a utilizar[24]. Para acceder a Reyes es necesario recibir del CCN un certificado digital de cliente, además de credenciales para autenticarse en la aplicación. Una vez dentro, es posible obtener una API Key. El Certificado de cliente resulta necesario para trabajar tanto en web como con la API.

---

[21] https://www.ccn-cert.cni.es/soluciones-seguridad/reyes.html

[22] MISP (Malware Information Sharing Platform) es la principal plataforma de inteligencia contra ciberamenazas. Esta orientada a la compartición, almacenamiento y correlación de IoCs. https://www.misp-project.org

[23] Una herramienta de "Ticketing" sirve para tener in registro y control de los incidentes que se gestionan en un SOC.

[24] Ref. 1.

Reyes iba a ser inicialmente la fuente principal de obtención de información para el dataset. Teóricamente, con un par de peticiones se obtendrían datos provenientes de casi todos los servicios necesarios, evitando tener que suscribirse individualmente a cada uno de ellos. Sin embargo, diversos problemas en su API hicieron que fuera perdiendo protagonismo: la información que devolvía de Shodan era muy incompleta, Censys daba errores debido a una actualización reciente de su API y VirusTotal consumía varias peticiones por cada llamada, con lo que se agotaba muy rápidamente el cupo.

Finalmente, la petición a Reyes solo requería información de Whois y RiskIq. Con la primera se obtiene información básica del registro[25], mientras que la segunda ofrece un listado de las IPs históricas que han resuelto el dominio. La única información que extrae este script es, precisamente, la última IP asociada al dominio, ya que es necesaria para la consulta a la API de Shodan.

Reyes devuelve un archivo por cada dominio analizado en formato json.

5.2. VirusTotal

VirusTotal[26] es una herramienta online que, tomando como entrada un dominio, IP, hash o url, devuelve el resultado del análisis que hacen alrededor de 70 antivirus y servicios de bloqueo de url/dominios. Además, aporta información relevante como la categorización, la reputación, información de certificados, comentarios que aporta la comunidad de ciberseguridad, etc. El servicio es gratuito para uso no comercial, permitiendo hasta 500 consultas diarias a un ritmo máximo de 4 por minuto. Para necesidades mayores ofrece suscripciones Premium.

VirusTotal dispone de una API que devuelve la información solicitada en formato json. El script "1_obtencionArchivos_DataSet.py", mediante la librería Requests[27], hace una llamada a la API y guarda la información en un archivo de texto siguiendo el mismo patrón anterior.

Al observar los archivos descargados para la creación del dataset se aprecia que faltan aproximadamente 600 archivos procedentes de VirusTotal. El motivo es que las primeras versiones del script no llamaban directamente a la API de VirusTotal, sino que lo hacían a través de REYES.

5.3. Shodan

Shodan[28] es conocido principalmente como un motor de búsqueda de equipos conectados a internet. Mientras que buscadores como Google rastrean la World Wide Web (WWW),

[25] Ref. 11.
[26] https://www.virustotal.com
[27] https://docs.python-requests.org/en/latest/
[28] https://www.shodan.io

Shodan busca todo tipo de dispositivos conectados para obtener una imagen completa y actualizada de Internet ya que rastrea todo Internet al menos una vez a la semana.

Para obtener información, el buscador pregunta directamente a los servicios que está ejecutando un equipo. Estos responden información publica conocida como "*banners*", que son metadatos sobre el software que está ejecutando el dispositivo. La mayor parte de la información que proporciona Shodan proviene de los banners.

Existen varios tipos de suscripción al servicio, entre ellos uno gratuito que limita considerablemente los servicios disponibles. También hay una modalidad llamada "Academic Membership" que eleva considerablemente las prestaciones y sigue siendo gratuita. Para obtenerla hay que registrarse con un correo electrónico con extensión ".edu" o similar. Se utiliza un sistema basado en "*credits*", según el tipo de membresía se dispone de más créditos y según el tipo de búsqueda se consumirán más o menos créditos. Básicamente se pueden realizar 2 tipos de peticiones: *querys* y *scans*. Con la primera se accede a la base de datos de escaneos ya realizados, con la segunda se ordena un escaneo sobre el recurso consultado y se obtienen los resultados. Siguiendo las limitaciones expuestas en el Capítulo 1.3., para la recolección de información solo se usarán *querys*, nunca scans, para evitar alertar al atacante de que su dominio puede haber sido detectado. Las búsquedas con *querys* devolverán los datos recogidos en los últimos 30 días. En caso de que en los últimos 30 días no haya información, la respuesta será nula. En ese caso, el script repetirá la petición solicitando datos históricos.

Como se mencionó en el punto 5.3, el único parseo que hace el script "1_obtencionArchivos_DataSet.py" es la búsqueda de la última IP asociada al dominio. Esta IP será el parámetro sobre el que se solicite información a la API de Shodan. El objetivo es obtener información del servidor que alberga al dominio, donde está localizado, qué puertos tiene abiertos, qué servicios corren en él, qué servicios hay tras esos puertos, qué otros dominios aloja, vulnerabilidades, certificados y un largo etc.

La respuesta de Shodan es un archivo de texto en formato json. Siguiendo la misma técnica que en las APIs anteriores, se generará un archivo por cada dominio analizado.

5.4. Archive

Internet Archive (Archive[29]) es una biblioteca sin ánimo de lucro que almacena millones de libros, películas, música, software, contenido multimedia, etc. Una de sus funcionalidades más conocidas es la "Wayback Machine", un servicio que almacena más de 668.000 millones de copias de páginas web (668 billones en la escala numérica anglosajona). Gracias a esto, es posible consultar la evolución de las páginas web en el tiempo. Al introducir en su buscador una url, nos devuelve un calendario donde se pueden ver todas las copias que Archive tiene almacenadas de dicha url. Pulsando sobre un día en concreto podremos ver las copias realizadas ese día. La Wayback Machine hace más copias cuanto más popular es la página.

[29] https://archive.org

Sin embargo, no se puede decir que este servicio proporcione una copia exacta de la evolución de Internet ya que los sitios web pueden evitar ser copiados por Archive mediante determinada configuración del archivo robots.txt[30].

La Wayback Machine dispone de varias APIs para automatizar consultas a sus bases de datos. El script hace uso de dos de ellas: la "*Wayback Availability JSON API*" permite saber si una url ha sido copiada en algún momento y está disponible; por su parte la "*Wayback CDX Server API*" proporciona un método para hacer consultas complejas a la base de datos y poder analizar los resultados.

Utilizando ambas APIs, se pregunta a la Wayback Machine por la fecha en que se realizó la primera copia, la fecha de la última y el número de copias realizadas en el último año. En el Capítulo 6 se explica el motivo de estas 3 preguntas. El almacenamiento de la información sigue la estructura ya descrita en puntos anteriores.

[30] Introduciendo en el archivo robots.txt lo siguiente: "*User-agent: ia_archiver Disallow: /*"

6. Extracción de información relevante

En paralelo a la recolección de información se hizo un análisis de los ficheros *json* que devolvían las diferentes APIs y se seleccionó la información que se quería extraer. En los siguientes puntos se detalla este proceso con una pequeña explicación del motivo por el que se considera relevante cada dato seleccionado y el nombre de la columna correspondiente en el dataset. Para facilitar la comprensión se ha dividido en 4 secciones correspondientes a los 4 archivos obtenidos para cada dominio.

Una vez que se tuvo claro que información se va a extraer de los 4 archivos de texto, se construyó una tabla de Excel, de nombre "Dataset.xlsx", en la que las filas corresponden a los dominios y las columnas a las características obtenidas. En total hay 47 columnas para 47 clases de información. A estas hay que añadir 3 columnas, la primera es simplemente un índice numérico que resultó útil durante el proceso de extracción de los datos; la segunda columna es el nombre del dominio y la tercera es la etiqueta (malicioso=1, benigno=0) para el entrenamiento supervisado. El Anexo 1 muestra una pequeña porción de la tabla del archivo "Dataset.xlsx".

Cuando se ejecuta el script de parseo, llamado "2_parseo_DataSet.py", se importa la tabla Excel como un *dataframe* de *Pandas*[31]. El script va leyendo cada fila, obteniendo el nombre de dominio correspondiente y comienza a buscar los archivos de texto correspondientes. Al encontrarlos, los carga y extrae la información, le da un formato único para cada campo y la guarda en el dataframe. Al finalizar con la última fila, exporta el dataframe a un Excel con el mismo nombre "Dataset.xlsx", con lo que obtenemos la misma tabla de Excel inicial, pero con todos los campos ya rellenos.

En un Macbook Pro de 2020 con procesador 2 GHz Intel Core i5 de 4 núcleos y 16Gb de RAM, este proceso de leer y parsear unos 13500 archivos tarda 1 minuto y 8 segundos en ejecutarse al completo.

```
ferpalop@MacBook-Pro-de-Fernando-2 TFM Software % /usr/local/bin/python3 "/Users/ferpalop/Documents/MASTER UPV/doc TFM/TFM Software/2_parseo_DataSet.py"

================================================================================
Importando Dataset.xlsx...
Número de filas: 3500
Parseando datos...
Parseo completado en  1 minutos y  8 segundos
Guardando tabla...
Proceso finalizado
```

Figura 7. Output del script "2_parseo_DataSet.py"

6.1. Reyes

A Reyes se le marcan como aplicaciones a consultar tanto Whois como RiskIq:

[31] El DataFrame es la estructura de datos fundamental de Pandas, representa una tabla de datos panel con indexación integrada. Cada columna contiene los valores de una variable y cada fila un conjunto de valores de cada columna

- Fechas de creación, actualización y caducidad del dominio (*RD_createdDate, RD_updatedDate, RD_expiresDate, Antigüedad, Actualizado*): Todas estas fechas parecen muy relevantes pues permiten conocer si un dominio es antiguo o de nueva creación, si ha sido actualizado recientemente o está próximo a caducar o incluso ya ha caducado. Como ya se explicó, los atacantes usan varias técnicas para confundir a los analistas y jugar con la antigüedad de los dominios es una de las principales. Con la finalidad de facilitar el trabajo del algoritmo, el script crea dos campos calculados: Antigüedad y Actualizado. Estos contienen el número de días desde la creación y actualización del dominio respectivamente. Se descartó un campo análogo para la caducidad porque es un dato ausente en un número significativo de registros y no se pensaba incluir en el dataset final.
- Servidores DNS que resuelven el dominio (*n_DNS_Servs, DNS_servers*): Es un dato que proporciona Whois y que inicialmente podría ser interesante para el analista.
- Registrador del dominio (*registrarName*): Un dominio puede ser registrado a través de varias empresas[32] que suelen cobrar por el servicio y compiten entre si. Cada registrador solicitará una serie de datos técnicos y de contacto al registrante, los almacena y presenta al registro central aquellos necesarios para que éste responda a las consultas de Whois. La cantidad de datos que se harán públicos, el precio del registro, el control sobre la veracidad de la información aportada, etc. serán claves para que un atacante elija a un registrador u otro.
- País de registro (*TC_country*): A priori, el país donde el atacante registra su dominio parece significativo, por la idea subyacente de que hay países más activos que otros en este tipo de actividades maliciosas. Hay unos 700 dominios en el dataset que no incluyen esta información, lo que se considera significativo en si mismo.
- Email del registrante y nº de caracteres del mismo (*TC_email, long_email*): Con este dato se pretende buscar emails generados aleatoriamente y con un grado de entropía muy elevado. Sin embargo, la realidad es que la mayoría de los servidores Whois ocultan este dato por protección de datos y privacidad, con lo que es poco útil para el algoritmo de aprendizaje. Aun así se mantuvo ya que si está presente puede dar información complementaria al analista.
- Dominio de primer nivel (Top Level Domain) (*TLD*): El TLD es el último segmento del nombre de un dominio. Es un campo calculado automáticamente por el script, obteniendo la cadena posterior al último punto que aparece en el nombre del dominio. La IANA (Internet Assigned Numbers Authority) reconoce 3 tipos de TLDs:
    - gTLD: Generic Top-Level Domains. Relacionados con el contenido de forma más o menos vaga, por ejemplo “.com”, “.org” o “.net”

[32] Un listado de los registradores acreditados se puede encontrar en https://www.icann.org/en/accredited-registrars?filter-letter=a&sort-direction=asc&sort-param=name&page=1

  - sTLD: Sponsored Top-Level Domains. Patrocinados por una entidad u organización especifica. Algunos ejemplos son ".edu", ".gov" o ".mil".
  - cTLD: Country Code Top-Level Domains. Representan a países concretos como ".es", ".us" o ".fr"

  Cada TLD tiene un precio asociado. La combinación entre la necesidad de crear muchos nuevos dominios y la de hacerlos pasar desapercibidos serán indicativos de la capacidad y la voluntad del atacante.

- Número de IPs que históricamente resolvieron el dominio (*num_IPs*): Puede dar una idea de la movilidad del dominio entre diferentes servidores. Este dato se obtiene del servicio RiskIq.
- Última IP asociada al dominio (*Ult_IP*): Tras analizar todas las IPs históricas, el script se queda con la que tiene la fecha de detección más reciente. El dato en si mismo es irrelevante para el algoritmo de machine learning, pero permite hacer una investigación sobre la IP que aloja al dominio.
- Primera y última vez que se vio el dominio (*HI_firstSeen, HI_lastSeen*): Son las fechas en que RiskIq detectó por primera y última vez el dominio. Comparando estas fechas con las de registro y actualización se puede obtener una idea de su actividad.
- Número de subdominios (*num_Subdomains*): El número de subdominios que alberga el dominio parece un dato relevante, ya que puede indicar si el dominio es plenamente funcional o solo se ha creado con una finalidad maliciosa muy concreta. El nombre de los subdominios no se ha considerado importante.

6.2. VirusTotal

A VirusTotal se pregunta por el dominio y se recoge la siguiente información:

- Categorización del dominio (*category, cats*): El hecho de que un dominio tenga categorización ya es significativo pues demuestra que tiene cierta antigüedad y que alguno de los servicios que establecen la categorización han podido determinar que tipo de servicio ofrece. VirusTotal recoge la categorización de numerosos servicios, como Komodo, Sophos, BitDefender y un largo etcétera[33]. Además, se recopila en un único campo los valores de todos los servicios que dan algún resultado, separados por comas.
- Reputación del dominio (*reputation*): Es un caso análogo al anterior. La reputación se basa en la media de las votaciones que recibe el dominio por parte de los usuarios de VirusTotal. Un valor alto indica que los usuarios lo consideran confiable, un valor negativo significaría lo contrario y la ausencia de valor (0) puede indicar que el dominio apenas ha tenido interacción con usuarios y no ha recibido votaciones.

[33] Un listado de los servicios que colaboran con VirusTotal se puede encontrar en https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors

- Certificados ssl (*Cert_https*): Cuando el dominio responde a peticiones https por el puerto 443 está obligado a poseer un certificado que vincule digitalmente una clave criptográfica con los datos de una organización[34]. La ausencia total de información sobre el certificado podría indicar que el propietario del dominio no tiene auténtico interés en que se acceda al dominio por el protocolo https. De la misma forma, certificados temporales, auto-firmados, caducados, etc., son indicios que el analista debe considerar. Muchos de ellos son recopilados en los campos siguientes:
    - Fecha del certificado (*LastCertDate*): Fecha en la que se expidió el certificado.
    - Algoritmo de cifrado de la clave pública (PKalgorithm): Algoritmo criptográfico usado por la clave pública del certificado.
    - Algoritmo de firma (CertSignAlg): Indica el algoritmo usado para la clave privada y la función hash usada para la prueba de integridad. Por ejemplo, para un valor "sha256RSA", el algoritmo será RSA y la función hash SHA 256.
    - Validez del certificado (*ValNoAntes, ValNoDespues*): Expresa el momento a partir de cuando el certificado es válido y cuando finaliza la validez.
    - Autoridad de certificación (CA) (*CA_Issuers*): Empresa u organización que valida la identidad de la entidad para la que se emite el certificado. No todas las CAs son igual de estrictas a la hora de certificar ni sus honorarios son los mismos, por lo que la elección de una u otra puede tener algún significado.
    - Emisor del certificado (*Issuer_CN*): Este campo contiene información respecto a la Autoridad de Certificación subordinada que se ha empleado para generar el certificado final.
    - Titular del certificado (*subject_CN*): Organización para la que se expide el certificado.

6.3. Shodan

A Shodan se le pregunta por la última IP asociada al dominio y de entre la enorme cantidad de información que proporciona, en principio se recopilan los siguientes datos:

- Puertos abiertos (*num_Ports, ports*): el número total de puertos y un campo donde se relacionan todos separados por comas. El tipo de servicios que usa un dominio para sus comunicaciones dice mucho sobre su actividad. Lo normal es que tengan abiertos el 80 y el 443 si disponen de un servidor web; si es necesario controlar el servicio en remoto suelen tener el 22, si disponen de un servicio ftp será 21 y si tiene servidor de correo usará el 25. Algunos puertos son más sospechosos que otros, por lo que un usuario avanzado tratará de utilizar puertos poco sospechosos[35].

[34] Ref. 15.
[35] Ref. 10.

- Puertos significativos (*port21ftp, port23telnet, port53dns, port22ssh, port80, port443*): Campos binarios que reflejan si estos puertos concretos están en uso en la IP investigada.
- Respuesta a peticiones http y https (*80_http_status, 443_http_status*): Respuesta que devuelve el dominio ante peticiones http por el puerto 80 o https por el 443. Han sido redondeadas a su centena para evitar múltiples resultados que dificulten el algoritmo de aprendizaje. Se pueden encontrar los siguientes valores:
    - 100 - Respuestas informativas
    - 200 - Respuestas satisfactorias
    - 300 – Redirecciones
    - 400 - Errores de los clientes
    - 500 - errores de los servidores
- Propietario de la IP (*IP_Organization*): Organización dueña de la IP.
- Proveedor de Servicios de Internet (ISP) (*IP_isp*): Es la empresa que brinda conexión a Internet al que pertenece esta dirección IP y que conecta el dispositivo a la Internet pública.

6.4. Archive

Haciendo uso de dos de las APIs disponibles (Wayback Availability JSON API y Wayback CDX Server API) se pregunta a Archive por la siguiente información:

- Primera copia en Archive y Antigüedad (*Archive_first, AntigArch*): Fecha en la que Archive (Wayback Machine) hizo la primera copia de la página web que responde al dominio. Para facilitar el trabajo del algoritmo, se introduce un campo calculado que representa la antigüedad de la primera copia en número de días transcurridos hasta la fecha de creación del dataset.
- Última copia en Archive y días transcurridos desde la misma (*Archive_first, AntigLast*): Fecha en la que Archive hizo la última copia de la página web y la diferencia con respecto a la fecha del parseo de los datos.
- Copias en Archive durante el último año (*Archive_year*): Número de copias realizadas de la página web desde el 1-1-2021. Lo que se trata de descubrir son aquellos dominios con años de antigüedad y un comportamiento normal, que han dejado de autorizar las copias en los últimos meses.

7. Selección del algoritmo

La clasificación basada en aprendizaje supervisado es una de las aplicaciones más frecuentes que realizan los llamados *Sistemas Inteligentes*. Existen diversos algoritmos especializados en esta tarea. Podemos clasificarlos entre aquellos desarrollados por la Estadística, como la Regresión Logística o el Análisis Discriminante, o los que se originan en el campo de la Inteligencia Artificial, como las Redes Neuronales, los Árboles de Decisión o las Redes Bayesianas. De entre todos ellos se ha optado por empezar el estudio por el de *Regresión Logística*. En los Capítulos siguientes se describe brevemente este algoritmo, sus características y limitaciones principales, sin llegar a entrar en sus fundamentos matemáticos por quedar fuera del alcance de este trabajo.

En función de los resultados que se pudieran obtener se contemplaba la posibilidad de probar otros algoritmos hasta dar con uno con suficiente precisión.

7.1. Introducción a la Regresión Logística

La regresión logística es un método estadístico que trata de modelar la probabilidad de una variable cualitativa binaria en función de una o más variables independientes. Su principal aplicación es la creación de modelos de clasificación binaria[36].

Llamaremos *regresión logística simple* cuando solo hay una variable independiente y *regresión logística múltiple* cuando hay más de una. Dependiendo del contexto, a la variable modelada se le conoce como variable dependiente o variable respuesta, y a las variables independientes como *regresores, predictores* o "*features*".

7.2. Condiciones para el uso del modelo

El desarrollo matemático del modelo de regresión logística se basa en una serie de asunciones que deben verificarse para que sus resultados y conclusiones sean validas. Aunque en la práctica es difícil que estas asunciones se cumplan completamente, esto no quiere decir que debamos ignorarlas o que el modelo vaya a dejar de ser útil si estas no se verifican. En cualquier caso, debemos estudiarlas y ser conscientes del impacto que esto podría tener en las conclusiones del modelo.

- No colinealidad o multicolinealidad

En los modelos de regresión logística múltiple, los predictores deben ser independientes, no debe de haber colinealidad entre ellos. Esto significa que un predictor no puede estar linealmente relacionado con ningún otro del modelo. Cuando hay colinealidad, no es posible identificar claramente el efecto que tiene cada predictor sobre la variable respuesta y, además, pequeños cambios en los datos provocarían grandes cambios en las estimaciones de los

[36] Ref. 6.

coeficientes. Tan difícil será encontrar una no-colinealidad perfecta como una colinealidad perfecta. Lo más frecuente es encontrar la llamada casi-colinealidad o multicolinealidad no perfecta. La colinealidad puede provocar que cuando se intenta establecer relaciones causa-efecto, se obtengan conclusiones erróneas, como que una variable es la responsable de determinado resultado cuando en realidad es otra que es colineal con ella.

Si se encuentra colinealidad entre predictores, podemos excluir uno de los predictores problemáticos o bien tratar de combinar las variables colineales en un único predictor, aunque con el riesgo de perder su interpretación.

- Relación entre los predictores numéricos y el *logaritmo de las predicciones (log of odds)*

Cada predictor numérico tiene que estar linealmente relacionado con el *logaritmo de odds* de la variable respuesta "*y*", mientras los demás predictores se mantienen constantes, de lo contrario no se deben introducir en el modelo.

- No autocorrelación (Independencia)

Los valores de cada observación son independientes de los otros. Esto es especialmente importante de comprobar cuando se trabaja con mediciones temporales.

- Valores atípicos

Es importante identificar observaciones que sean atípicas o que puedan estar influenciando al modelo.

- Tamaño de la muestra

En el caso de que no se disponga de suficientes observaciones, predictores que no son realmente influyentes pueden llegar a parecerlo (overfitting). Una recomendación frecuente es que el número de observaciones sea como mínimo entre 10 y 20 veces el número de predictores del modelo.

- Parsimonia

La Parsimonia hace referencia a que, el mejor modelo, es aquel capaz de explicar con mayor precisión la variabilidad observada en la variable respuesta empleando el menor número de predictores, por lo tanto, con menos asunciones.

7.3. Probabilidad vs Clasificación

La salida de un modelo logístico es una probabilidad. Dado que una de las principales aplicaciones de este tipo de modelos es la clasificación, deberemos establecer un límite a partir del cual se considera que la variable pertenece a uno de los niveles. Por ejemplo, se puede asignar una observación al grupo 1 si la probabilidad estimada es mayor de 0.5 y al grupo 0 de lo contrario.

En el modelo que se está desarrollando, se clasificará inicialmente el dominio cuando la probabilidad de ser malicioso sea mayor que la de ser benigno. En cualquier caso, se

presentará el valor de la probabilidad obtenida para que el analista comience su investigación por los valores más altos.

7.4. Tipos de datos

La mayoría de los datos usados en aprendizaje automático se pueden agrupar en 4 categorías:

- Datos numéricos: también conocidos como datos cuantitativos, son cualquier dato donde los puntos de datos son números exactos. Se pueden caracterizar en datos continuos o discretos. Los datos continuos pueden asumir infinitos valores dentro de un rango (o intervalo cerrado) mientras dicho número de valores es finito en el caso de datos discretos.
- Datos Categóricos: representan características, para lo cual pueden tomar valores numéricos, aunque estos números no tendrían un significado matemático. En el contexto de la súper-clasificación, los datos categóricos serían la etiqueta de la clase. Dentro de los datos categóricos se encontrarían los datos ordinales, que son una mezcla de los numéricos y los categóricos ya que, aun estando clasificados en categorías, estas se encuentran ordenadas de alguna manera en particular.
- Datos de series temporales: son una secuencia de números recopilados a intervalos regulares durante un período de tiempo, por lo que este tipo de datos tienen un valor temporal adjunto que permite ordenarlos.
- Datos de texto: son básicamente palabras. Para que tengan utilidad en aprendizaje automático es necesario tratarlos con funciones como las bolsas de palabras.

Desarrollar una comprensión profunda de los diferentes tipos de datos es un requisito previo crucial en aprendizaje automático.

En ocasiones se necesitará convertir tipos de datos de algunas variables para facilitar el trabajo de los algoritmos y adecuarlos al modelo que se está desarrollando[37].

7.5. Variables Dummy

Cuando se trabaja con predictores en forma de variables categóricas con más de dos niveles o categorías, se deberán crear lo que se conoce como variables "*dummy"* o "*one-hot-encodding"*, que son variables creadas para cada uno de los niveles del predictor categórico y que pueden tomar el valor de 0 o 1. De esta forma, cada vez que se emplee el modelo, solo una variable dummy por predictor tomará el valor 1, tomando el resto el valor 0.

La creación de variables dummy resulta sencilla cuando se conoce el número de valores que puede tomar el predictor y este número no es muy elevado. Ahora bien, éste se puede tornar complejo cuando no se está seguro de esa cifra o los valores posibles son muy numerosos.

[37] Más información en Ref. 8.

Este problema se pondrá de manifiesto en el siguiente Capítulo con varios campos de datos incorporados al dataset.

8. Construcción del dataset y validación del algoritmo

Al estudiar los datos recopilados en el dataset, resultó obvio que no sería posible incluir toda esa información en un algoritmo de machine learning, principalmente porque una buena parte de ellos eran datos de texto que requerirían un tratamiento especial e individualizado.

En primer lugar, se hizo un análisis de los tipos de datos recogidos de acuerdo con la clasificación expuesta en el Capítulo 7.4:

Numéricos	Categóricos	Series temporales	Texto
• HI_lastSeen • HI_firstSeen • num_IPs • RD_createdDate • Antigüedad • RD_expiresDate • RD_updatedDate • RD_updatedDate • long_email • num_Ports • num_Subdomains • LastCertDate • ValNoDespues • ValNoAntes • Archive_first • AntigArch • Archive_last • Archive_year	• MALICIOSO • TC_country • TLD • port21ftp • port23telnet • port53dns • port22ssh • port80 • port443 • 80_http_status • 443_http_status • category • reputation • Cert_https • PKalgorithm • CertSignAlg		• Ult_IP • DNS_servers • registrarName • TC_email • ports • IP_Organization • IP_isp • cats • CA_Issuers • Issuer_CN • subject_CN

Figura 8. Clasificación de los datos recopilados

La clasificación de algunos de los campos no es tan intuitiva como podría parecer, por ejemplo, el país donde se registró el dominio: el número de países en el mundo es relativamente estable y conocido pero la forma de referirse a ellos, según el idioma del registrador, es muy variada y no hay un formato estandarizado para referirse a ellos en la información que proporciona Whois. Como este parámetro se cree que puede resultar importante para la clasificación del dominio se ha optado por hacerlo categórico y crear variables dummy para todos y cada uno de los países que aparecen en el dataset (105). Exactamente lo mismo ocurre con el TLD y el número de sus variantes también es muy elevado (135).

Otros posibles predictores que también podrían ser tratados como categóricos, como el Registrador del dominio (417 distintos), la organización propietaria de la IP (919) el proveedor de servicios de internet (681) o la Autoridad de Certificación, pero el número de variables dummy se dispararía. En estos casos se mantiene la información para que la evalué el analista, pero no se incorporan como predictores al dataset.

Por otro lado, las fechas han sido consideradas como datos numéricos y no como series temporales, como intuitivamente podría parecer. Para facilitar su integración en el algoritmo, algunas de ellas han sido convertidas a un número entero que representa la diferencia en días entre la fecha y el momento en el que se produce el parseo de los datos.

Después de varias pruebas y teniendo en cuenta los condicionantes anteriores, se seleccionaron los siguientes campos de la tabla Excel "Dataset.xlsx" para ser incorporados al dataset para el aprendizaje automático:

nº	Campo	nº	Campo	nº	Campo
1	MALICIOSO	9	port21ftp	17	num_Subdomains
2	num_IPs	10	port23telnet	18	category
3	Antiguedad	11	port53dns	19	reputation
4	Actualizado	12	port22ssh	20	Cert_https
5	n_DNS_Servs	13	port80	21	AntigArch
6	TC_country	14	port443	22	AntigLast
7	TLD	15	80_http_status	23	Archive_year
8	num_Ports	16	443_http_status		

Figura 9. Campos incluidos en el dataset para el aprendizaje automático

En este punto, solo quedaba construir el dataset y probarlo. Para esta tarea se creó el script "3_evalua_RegresionLogistica.py", que utiliza la librería "Scikit-Learn[38]" de python[39].

En primer lugar, importa del archivo dataset.xlsx solo las 23 columnas especificadas y lo guarda como un dataframe de pandas al que llama "*tabla*":

```
tabla = pd.read_excel('dataset3500.xlsx', usecols= "C,G,I,L,M,P,S,U,W,X,Y,Z,AA,AB,AC,AD,AG,AH,AJ,AK,AU,AW,AX")
```

Figura 10. Salida por pantalla 2 del Script "3_evalua_RegresionLogistica.py"

En segundo lugar, se crean las variables dummy para los campos *TC_country* y *TLD*. El script devuelve las dimensiones del dataset inicial y después de la creación de las variables dummy:

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Importando el Dataset
Dimension del Dataset original: (3500, 23)
Dimensión del Dataset con dummies: (3500, 261)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Figura 11. Salida por pantalla 1 del Script "3_evalua_RegresionLogistica.py"

[38] Scikit-Learn, o simplemente "sklearn", es la librería más popular sobre machine learning en python. Contiene numerosas funciones y herramientas tanto para machine learning como para análisis estadístico, incluyendo algoritmos de clasificación, regresión, clustering o reducción dimensional.
[39] Ejemplos que se tomaron como base para la implementación del algoritmo de Regresión Logística en python se pueden encontrar en las Refs. 5 y 7.

Como se aprecia en la Figura, se han creado 240 variables dummy para cubrir los 105 países y los 135 TLDs. Como estas dos variables (*TC_country* y *TLD)* desaparecen en favor de sus 240 dummies, hay que restar 2 a los 23 originales, con lo que obtenemos 261=240+21.

El siguiente paso es dividir aleatoriamente el dataset en 2 grupos, el primero estará formado por un 80% de las muestras y se utilizará para entrenar al algoritmo. El segundo grupo lo constituirá el 20% restante y será el que pruebe la eficacia del algoritmo. Además, se especifica cual es la etiqueta (MALICIOSO) para el entrenamiento supervisado y se establece una semilla aleatoria para asegurar que podemos repetir el proceso con los mismos datos (Figura 12).

```
X_train, X_test, y_train, y_test = train_test_split(
    dataset.drop('MALICIOSO', axis=1), dataset['MALICIOSO'],
    test_size=0.20, random_state=5)
```

Figura 12. Código para dividir el dataset en Script "3_evalua_RegresionLogistica.py"

Para un "random_state=5" se ha generado un set de entrenamiento (training set) con 2800 muestras (de las que un 45,214% del total son maliciosas) y un set de prueba (test set) con 700 muestras (de las que un 42.286% son maliciosas).

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Dividiendo el data set 80% training / 20% test --- Random State_5
Dimension de training set (X_train)= (2800, 260) --- 80.000 %
maliciosos: 1266 ---- benignos: 1534
45.214 % de maliciosos en el set de entrenamiento
Dimension de test set (X_test)= (700, 260) --- 20.000 %
maliciosos: 296 ---- benignos: 404
42.286 % de maliciosos en el set de test
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Figura 13. Salida por pantalla 2 del Script "3_evalua_RegresionLogistica.py"

Llega el momento de seleccionar el algoritmo y entrenarlo (Figura 14).

```
clf = LogisticRegression(max_iter=100000).fit(X_train, y_train)
```

Figura 14. Código para entrenar el algoritmo en Script "3_evalua_RegresionLogistica.py"

Este proceso se completa en 5 segundos, con el mismo equipo especificado en el Capítulo 6.

Por último, se procede a validar el modelo sobre las 700 muestras del set de prueba obteniendo una precisión (*accuracy score*) conforme se muestra en la Figura 15

```
Haciendo predicciones sobre test set...
Prediciones realizadas: 700
Acuracy score: 0.9514285714285714
Matriz de Confusion:
[[382  22]
 [ 12 284]]
```

Figura 15. Salida por pantalla 4 del Script "3_evalua_RegresionLogistica.py"

Como se observa en la Figura, el algoritmo ha funcionado con una precisión muy elevada del 95,14%. Atendiendo a la matriz de confusión y teniendo en cuenta que Scikit-Learn no presenta los datos como de la forma habitual sino siguiendo el esquema que se muestra en la Figura 16.

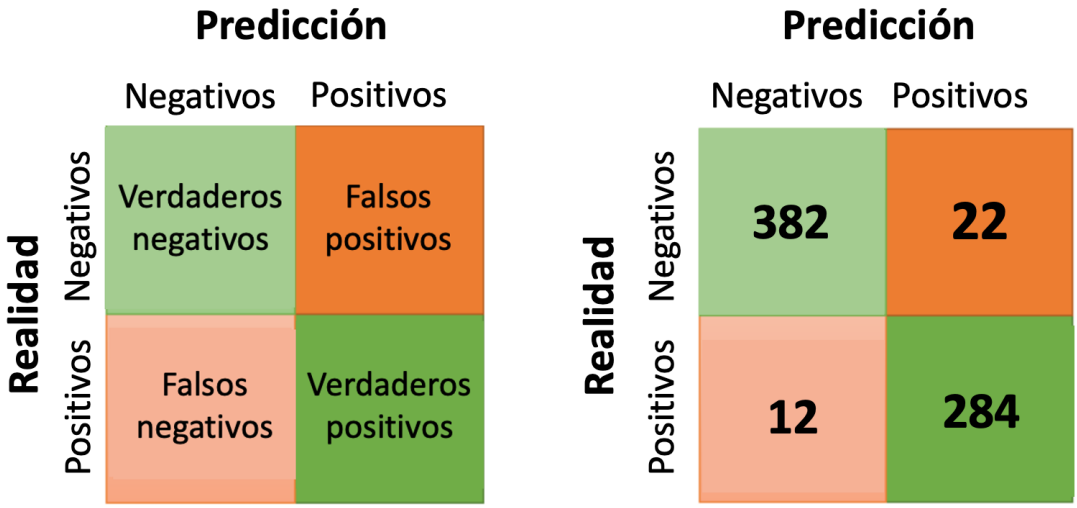

Figura 16. Matriz de Confusión en Scikit-Learn y resultado para el training set

Donde la diagonal verde muestra los aciertos y la naranja los errores. Es decir, de las 404 muestras benignas en el training set, el algoritmo ha clasificado correctamente 382 y se ha equivocado en 22. En cuanto a las 296 muestras maliciosas, han sido clasificadas como tal 284 mientras que ha errado en 12.

Se repite el proceso 6 veces más con diferentes semillas aleatorias obteniéndose los resultados de la Figura 17, donde se aprecia que la precisión del algoritmo está entre el 93,57% y el 97,57%, con una media de 94,98%.

A continuación, el script hace un análisis de las 10 variables más significativas tanto para el resultado positivo (malicioso) como negativo (benigno) y hace una representación gráfica de la importancia de las variables (Figuras 18 y 19)[40].

[40] Ref. 9.

Random_State	Accuracy Score	Confusion Matrix
5	0.9514	[[382 22] [12 284]]
10	0.9471	[[351 17] [20 312]]
15	0.9357	[[347 23] [22 308]]
20	0.9757	[[363 7] [10 320]]
25	0.94	[[369 19] [23 289]]
30	0.9443	[[389 28] [11 272]]
35	0.9543	[[363 18] [14 305]]
Media	**0.9498**	

Figura 17. Precisión y Matriz de Confusión para distintas semillas

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx ANALISIS DE LA IMPORTANCIA DE LAS VARIABLES xxxxx

-------Variables mas significativas para MALICIOSO-------------
(0, ('TLD_com', 2.846581183123438))
(1, ('TLD_net', 2.7746192109879235))
(2, ('TLD_tk', 1.5946756056136242))
(3, ('category', 1.549082925091602))
(4, ('TLD_xyz', 1.20306794175769))
(5, ('TLD_info', 1.0206656623778103))
(6, ('TC_country_south africa', 1.013403727759039))
(7, ('TLD_za', 0.9961611209465034))
(8, ('TLD_mx', 0.8470584480352079))
(9, ('TC_country_pakistan', 0.8324616957076096))
-------Variables mas significativas para BENIGNO-------------
(0, ('TLD_fun', -1.2560268355848982))
(1, ('port21ftp', -1.2401735478070681))
(2, ('TLD_vip', -1.2027966595007262))
(3, ('TLD_biz', -1.0562272326554838))
(4, ('TLD_gq', -1.0277779473849933))
(5, ('TLD_world', -0.960366274456094))
(6, ('TC_country_viet nam', -0.9443852753329471))
(7, ('TLD_work', -0.8957116649378652))
(8, ('TLD_systems', -0.8394407494863744))
(9, ('TC_country_turkey', -0.793181158927161))
```

Figura 18. Análisis de la importancia de los predictores

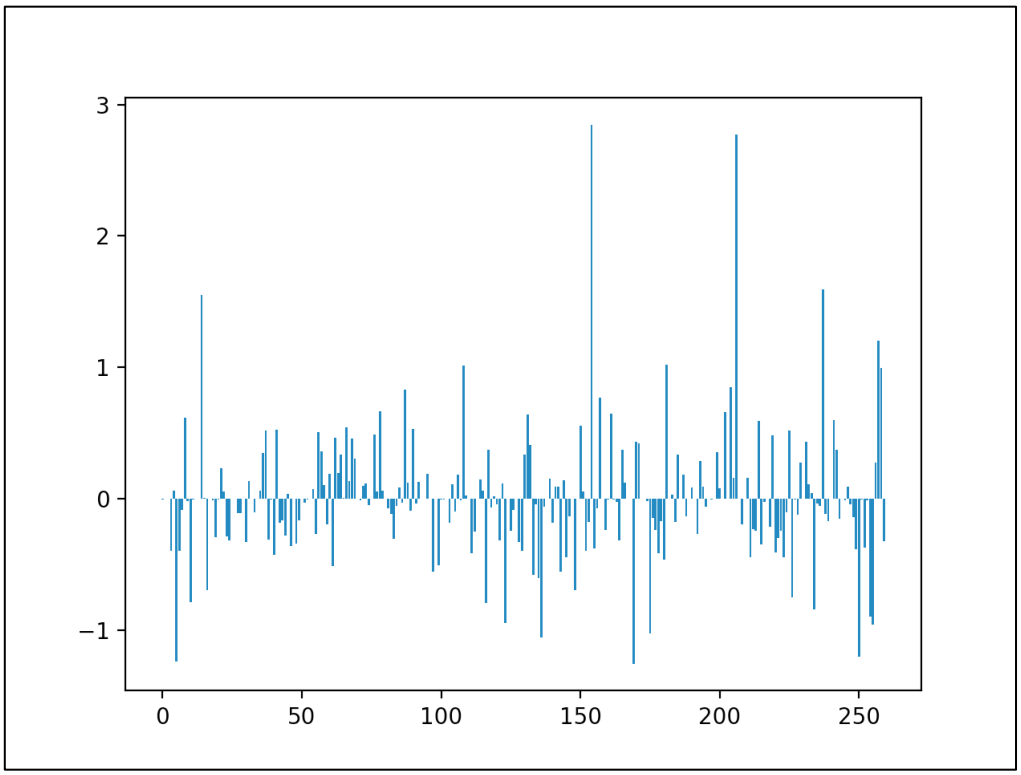

Figura 19. Representación gráfica de la importancia de los predictores

Se puede interpretar que el peso de los predictores TLDs ".com" y ".net" es bastante significativo, lo que podría suponer que cuando el algoritmo ve un TLD ".com" asume que se trata de un dominio malicioso. Para descartarlo se ha forzado a que el propio script haga un análisis de los dominios con TLD ".com" y ".net" en el set de test, para ver si los etiqueta correctamente.

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Analisis de posibles proxies - .com
TLD .com en set test= 303
maliciosas= 170  - benignas= 133
predicciones correctas sobre .com= 290
predicciones correctas sobre .com benignas= 125 sobre  133  = 93.98496240601504 %
```

Figura 20. Análisis de muestras ".com" en el set de prueba

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Analisis de posibles proxies - .net
TLD .net en set test= 23
maliciosas= 15  - benignas= 8
predicciones correctas sobre .net= 23
predicciones correctas sobre .net benignas= 8 sobre  8  = 100.0 %
```

Figura 21. Análisis de muestras ".net" en el set de prueba

Como se aprecia en los resultados, el algoritmo mantiene una precisión de aproximadamente el 94% a la hora de predecir dominios ".com" y la aumenta hasta el 100% en el caso de los ".net", que además son mucho menos numerosos. Esto permite descartar que el algoritmo esté condicionado por estas dos variables, pese a su importancia en el proceso de clasificación.

9. Predicciones sobre muestras no etiquetadas.

Ya se ha comprobado que el algoritmo es suficientemente eficaz para el cometido que ha sido desarrollado. Es el momento de ponerlo en funcionamiento con muestras no etiquetadas.

No se debe olvidar que el objetivo de la aplicación final no es tanto clasificar sino ordenar los dominios bajo investigación por orden de peligrosidad, de forma que sea el analista el que posteriormente valide o corrija la clasificación automática. Esto es debido a las implicaciones que pueden tener tanto los falsos positivos como los falsos negativos en el sistema.

En esta fase se ha pensado principalmente en la comodidad del analista. Se requiere que este introduzca los dominios que quiere investigar en una tabla de Excel muy simple, llamada "investigar.xlsx", con una única columna llamada "DOMINIO", como se ve en la Figura 22.

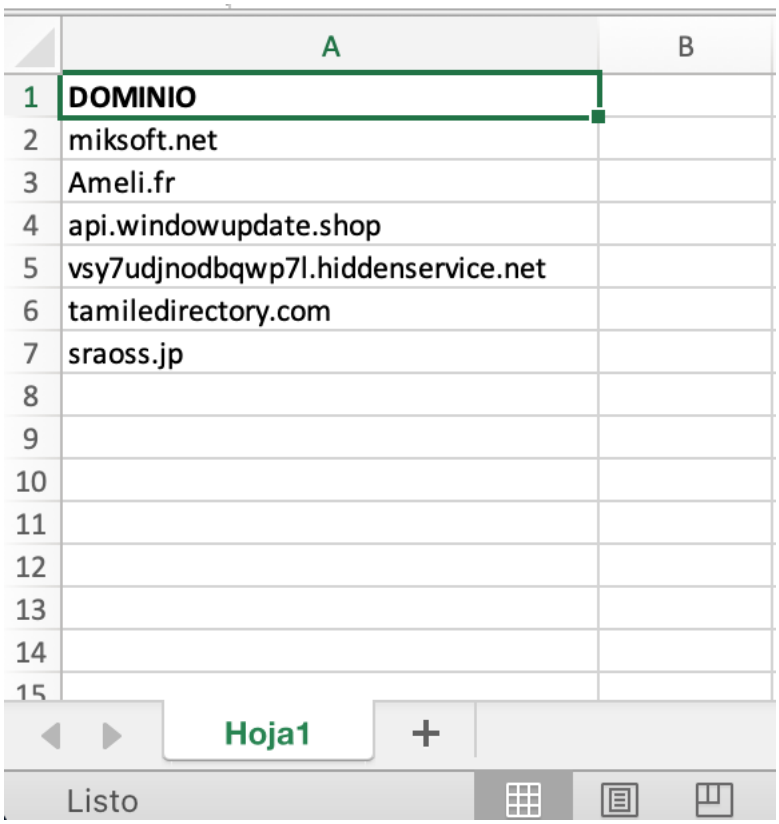

Figura 22. Archivo investigar.xlsx con los dominios sin etiquetar

El script "4_prediccion.py" es un compendio de todos los anteriores. En primer lugar, comprueba que ninguno de los dominios del archivo "investigar.xlsx" está incluido entre los 3500 del dataset. En caso de que alguno lo estuviera, lo notifica por pantalla y lo bloquea, introduciendo el resto en una tabla con la misma fila de encabezados que "dataset.xlsx" pero sin datos, que recibe el nombre de "datasetreal.xlsx". En el ejemplo, el dominio "*Ameli.fr*" forma parte del dataset y está clasificado como "Benigno", por lo que se excluye en la tabla de dominios a investigar, como muestra la Figura 23.

```
------------------------------------------------------
Proceso finalizado
------------------------------------------------------
ferpalop@MacBook-Pro-de-Fernando-2 TFM Software % /usr/local/bin/python3 "/Users/ferpalop/Documents/MA

Total de dominios a investigar: 6
================================================================================
Comprobando que los dominios a investigar no estan ya en el dataset...
Ameli.fr -- Dominio ya en Dataset. Está clasificado como BENIGNO
================================================================================
Nuevo total de dominios a investigar: 5
```

Figura 23. Eliminación de dominios ya clasificados por estar incluidos en el Dataset

A continuación, se itera sobre las filas chequeando que el dominio no ha sido investigado previamente y, tal y como hacia el script "1_obtencionArchivos_DataSet.py", recopila información de las distintas APIs, generando los mismos 4 archivos que se creaban para cada dominio cuando se recopiló la información para el entrenamiento del modelo de aprendizaje automático. Todos los archivos obtenidos se guardan en la carpeta "infoDominios". El script tiene la precaución de comprobar que el dominio no ha sido previamente investigado, verificando que no existen los archivos correspondientes en la carpeta "infoDominios". En el ejemplo se ha incluido el dominio "*miksoft.net*" que se había descargado en una investigación anterior, para verificar esta protección. Se muestra la salida por pantalla de esta primera fase del script en la Figura 24.

Cuando ha terminado la recopilación de información comienza el parseo de forma automática. Éste consiste en un proceso igual al descrito en el Capítulo 6 para el script "2_parseo_DataSet.py". En el caso de que para alguno de los dominios no se haya podido recopilar información esencial para hacer la predicción, por ejemplo la IP asociada al dominio o información esencial de Shodan, este será eliminado para no interferir en el proceso, lo cual será notificado por pantalla. En el ejemplo, para el dominio "*api.windowupdate.shop*" el script no es capaz de encontrar información en Shodan y por tanto no cumple con los requisitos mínimos para hacer una predicción. La salida por pantalla se muestra en la Figura 25

En ese momento el script abre el fichero "datasetreal.xlsx" en Excel ya con los datos rellenos y permite al analista verificar que la información obtenida no contiene errores significativos, como por ejemplo un error ortográfico en el país de registro o la falta de datos por fallo en las APIs. En caso de que los tuviera, el analista debe corregirlos y guardar el archivo. En este Excel no aparecen los dominios sobre los que no se va a realizar predicción, ya sea por estar previamente clasificado o por falta de información esencial. En la Figura 26 se muestra una porción del archivo "dasetreal.xlsx" en el que se aprecian las columnas de datos rellenas a excepción del campo "MALICIOSO" que todavía está vacío.

Por último, una vez guardados los posibles cambios en el Excel y cerrada la aplicación, el analista deberá volver sobre la consola de python y pulsar ENTER para que el script comience a hacer las predicciones.

```
Total de dominios a investigar: 6
================================================================================
Comprobando que los dominios a investigar no estan ya en el dataset...
Ameli.fr -- Dominio ya en Dataset. Está clasificado como BENIGNO
================================================================================
Nuevo total de dominios a investigar: 5

DOMINIO: sraoss.jp ( 1 / 5 ) - NO ESTA DESCARGADO. DESCARGANDO...
No hay errores, continua la extración...
Número de resoluciones históricas 5
Última IP asociada: 99.84.74.70
Obteniendo información en VIRUSTOTAL sobre  sraoss.jp
Obteniendo información en SHODAN sobre IP: 99.84.74.70
Obteniendo información en ARCHIVE sobre  sraoss.jp
Fecha primera copia: 2006-05-03
Fecha última copia: 2022-04-30
Total desde 2021: 4
----------
esperando 6 segundos
----------
DOMINIO: miksoft.net ( 2 / 5 ) - YA DESCARGADO
----------
DOMINIO: vsy7udjnodbqwp7l.hiddenservice.net ( 3 / 5 ) - NO ESTA DESCARGADO. DESCARGANDO...
No hay errores, continua la extración...
Número de resoluciones históricas 11
Última IP asociada: 82.192.82.228
Obteniendo información en VIRUSTOTAL sobre  vsy7udjnodbqwp7l.hiddenservice.net
Obteniendo información en SHODAN sobre IP: 82.192.82.228
Obteniendo información en ARCHIVE sobre  vsy7udjnodbqwp7l.hiddenservice.net
Fecha primera copia: 9999-12-31
Total desde 2021: 0
----------
esperando 6 segundos
----------
DOMINIO: api.windowupdate.shop ( 4 / 5 ) - NO ESTA DESCARGADO. DESCARGANDO...
No hay errores, continua la extración...
Número de resoluciones históricas 1
Última IP asociada: 127.0.0.1
Obteniendo información en VIRUSTOTAL sobre  api.windowupdate.shop
Obteniendo información en SHODAN sobre IP: 127.0.0.1
No hay informacion en Shodan sobre esta IP  Invalid IP
Obteniendo información en ARCHIVE sobre  api.windowupdate.shop
Fecha primera copia: 9999-12-31
Total desde 2021: 0
----------
esperando 6 segundos
----------
DOMINIO: tamiledirectory.com ( 5 / 5 ) - NO ESTA DESCARGADO. DESCARGANDO...
No hay errores, continua la extración...
Número de resoluciones históricas 3
Última IP asociada: 50.87.151.119
Obteniendo información en VIRUSTOTAL sobre  tamiledirectory.com
Obteniendo información en SHODAN sobre IP: 50.87.151.119
Obteniendo información en ARCHIVE sobre  tamiledirectory.com
Fecha primera copia: 2015-05-31
Fecha última copia: 2022-03-14
Total desde 2021: 14
----------
```

Figura 24. Salida por pantalla del proceso de recolección de información

```
================================================================================================================
Parseando datos...
Eliminando dominios sin información suficiente...
Parseo completado en  0 minutos y  0 segundos
Guardando tabla...
========================================================

- Verifique datos en archivo "datasetreal.xlsx"
- Se han eliminado directamente dominios para los que no hay suficientes datos para hacer la predicción
- Pulse ENTER cuando esten grabadas las modificaciones y cerrado el Excel
```

Figura 25. Salida por pantalla del script "4_prediccion.py"

	A	B	C	D	E	F	G	H	I	J
1	ID	DOMINIO	MALICIOSO	Ult_IP	HI_lastSeen	HI_firstSeen	num_IPs	RD_createdDate	Antiguedad	RD_expiresDate
2	0	sraoss.jp		99.84.74.70	2021-08-29	2021-08-29	5	2022-06-16	4	2022-08-31
3	1	miksoft.net		49.12.204.104	2022-06-15	2022-04-23	52	2004-11-30	6411	2022-11-30
4	2	vsy7udjnodbqwp7l.hiddenservice.net		82.192.82.228	2022-06-17	2022-02-11	11	2019-09-08	1016	2022-09-08
5	5	tamiledirectory.com		50.87.151.119	2022-06-16	2020-05-11	3	2014-08-27	2854	2022-08-27
6										
7										
8										
9										

Figura 26. Archivo datasetreal.xlsx tras el parseo de los datos

Con la información de los dominios a investigar ya completada y verificada, el script sigue el mismo proceso descripto en el Capítulo 8, con la salvedad de que ahora no tenemos un único dataset que hay que dividir, sino que tenemos un dataset completo de training con 3500 muestras etiquetadas y un dataset para clasificar con los dominios que estamos investigando (4 dominios en el ejemplo que se está mostrando). En este punto surgió un problema: para hacer las predicciones, los 2 dataset deben tener la misma estructura. El dataset de training tiene una dimensión de 3500x261 debido a las variables dummy. Si creamos las variables dummy directamente para el segundo dataset, nos daría una dimensión de 4x26, ya que en los 4 dominios a investigar solo hay 2 países (United States y Japan) y 3 dominios (com, net y jp). La solución fue concatenar ambas tablas, anexando las 4 filas del segundo dataset al final del primero, crear las variables dummy y, finalmente, separarlas de nuevo, con lo que obteníamos 2 dataset con el mismo número de columnas (3500x262 y 4x262).

A continuación, se separan los sets suprimiendo y/o añadiendo las etiquetas donde corresponda:

```
X_train = tabla.drop('MALICIOSO', axis=1)
y_train = tabla['MALICIOSO']
X_test = prediccion.drop('MALICIOSO', axis=1)
```

Figura 27. Código para separar los sets en Script "4_prediccion.py"

Se inicializa y entrena el modelo (Figura 28).

```
# Inicializamos y entrenamos el modelo
clf = LogisticRegression(max_iter=50000).fit(X_train, y_train)

# Hacemos las predicciones y sacamos las probabilidades
y_predp = clf.predict_proba(X_test)
y_pred = clf.predict(X_test)
```

Figura 28. Código para separar entrenar y hacer las predicciones en Script "4_prediccion.py"

Seguidamente se muestran por pantalla todos los dominios con su probabilidad y la etiqueta asignada, ordenandos para mostrar en primer lugar los que con mayor probabilidad son maliciosos.(Figura 29)

```
============================================================
Haciendo predicciones...

Prediciones ordenadas por peligrosidad:
96.576 % -- MALICIOSO -- vsy7udjnodbqwp7l.hiddenservice.net
32.466 % -- benigno -- tamiledirectory.com
5.310 % -- benigno -- sraoss.jp
0.013 % -- benigno -- miksoft.net
============================================================
```

Figura 29. Predicciones efectuadas por el Script "4_prediccion.py"

El algoritmo clasifica directamente si la probabilidad del valor 1 (“malicioso”) es mayor que la de 0 (“benigno”). Esto podría ajustarse, pero no es muy relevante ya que lo realmente importante es la probabilidad de la predicción.

Además, se listan los dominios incluidos en el archivo “investigar.xlsx” para los que no se ha realizado predicción y el motivo correspondiente, bien por ya estar en el dataset, y por tanto ya clasificados, o por no disponerse de información esencial para el proceso de predicción (Figura 30).

```
============================================================
No se ha realizado predicciones sobre:

Ameli.fr --  Ya clasificado: BENIGNO
api.windowupdate.shop --  No hay información en Shodan
------------------------------------------------------------
Proceso finalizado
------------------------------------------------------------
```

Figura 30. Dominios para los que no se ha realizado predicción y motivo

Por último, el script escribe el valor de la etiqueta en el dataset y se exporta de nuevo al Excel “datasetreal.xlsx”. El proceso finaliza abriendo nuevamente el Excel ya con el campo “MALICIOSO” relleno con la predicción realizada y el resto de los dominios para los que no se realizó predicción, ya que se considera que la información recopilada, aunque insuficiente para hacer predicciones, puede ser interesante para la investigación del analista. Los dominios que sí se han eliminado definitivamente son aquellos que ya estaban clasificados en el Dataset (en el ejemplo Ameli.fr).

	A	B	C	D	E	F	G	H
1	ID	DOMINIO	MALICIOSO	Ult_IP	HI_lastSeen	HI_firstSeen	num_IPs	RD_createdDate
2	0	sraoss.jp	0	99.84.74.70	2021-08-29	2021-08-29	5	2022-06-16
3	1	miksoft.net	0	49.12.204.104	2022-06-15	2022-04-23	52	2004-11-30
4	2	vsy7udjnodbqwp7l.hiddenservice.net	1	82.192.82.228	2022-06-17	2022-02-11	11	2019-09-08
5	5	tamiledirectory.com	0	50.87.151.119	2022-06-16	2020-05-11	3	2014-08-27
6	3	api.windowupdate.shop		127.0.0.1	2021-11-10	2021-11-10	1	2022-06-16
7								
8								

Figura 31. Archivo datasetreal.xlsx tras la clasificación de los dominios

10. Reentrenamiento del modelo

El proceso de añadir nuevas muestras al dataset de entrenamiento resulta muy sencillo a partir del "datasetreal.xlsx", ya que solo tendríamos que añadir las líneas ya etiquetadas al final de la tabla "dataset.xlsx", siempre y cuando se cumplan dos condiciones importantes:

- 1ª.- Que el analista haya verificado que las etiquetas asignadas automáticamente por el algoritmo son correctas. En caso contrario deberá modificar el valor de la etiqueta. Hay que recordar una vez más que la principal utilidad de esta herramienta no es clasificar directamente sino orientar al analista hacia los dominios más susceptibles de ser maliciosos.
- 2ª.- Que se añadan dominios maliciosos y benignos en una proporción no muy diferente a la inicial del dataset (55% benignos – 45% maliciosos).

Por otra parte, es difícil que la disponibilidad de recursos del SOC permita llegar a investigar hasta el último dominio para verificar su etiquetado. Lo lógico sería empezar por los más peligrosos e ignorar los catalogados como benignos con probabilidad media-alta de que lo sean. En este caso, para mantener la proporción sería más conveniente elegir los dominios con muy alta probabilidad de ser benignos y completarlos con dominios sacados de entre los 5000 o 10000 dominios más populares de, por ejemplo, *Majestic Million*[41]. Estos nuevos dominios deberán ser sometidos al proceso de predicción no para obtener su etiqueta, sino para recopilar la información necesaria.

Al ir incorporando nuevas muestras al set de entrenamiento, nos aseguramos de que las nuevas TTPs de los atacantes se van reflejando en el set y, por tanto, aumenta la capacidad del algoritmo de detectarlas.

[41] Cuanto más popular sea el dominio menos probabilidades habrá de que sea malicioso y por tanto menor probabilidad de que introduzcamos etiquetas erróneas en el set de entrenamiento.

11. Conclusiones

Este trabajo se ideó con la finalidad de intentar demostrar una intuición basada en la experiencia de trabajo en un SOC: que los atacantes siguen ciertas reglas o patrones a la hora de levantar dominios con fines maliciosos y que utilizando técnicas de machine learning, se podría llegar a hacer una predicción en función de determinadas características del dominio sin llegar a interaccionar con él de forma reconocible.

Los resultados obtenidos no solo permiten verificar esta hipótesis, sino que lo hacen con una precisión muy por encima de lo esperado por el propio autor.

El hecho de que el dataset para el entrenamiento supervisado esté confeccionado con datos reales y actuales, que incluyen la práctica totalidad de los dominios maliciosos disponibles en listas negras del CCN y con una amplia representación de dominios benignos para equilibrar las muestras, hace que la herramienta resultante pueda ser utilizada inmediatamente en un SOC.

Pese a la fiabilidad que demuestra el algoritmo en las pruebas de validación (94,98% de clasificaciones correctas de media), no resulta aconsejable confiarle, por ejemplo, el bloqueo automático de dominios en función de la predicción, principalmente porque las TTPs del atacante evolucionan muy rápido y podrían aprender la forma de confundir al algoritmo. Esto se solucionaría alimentando el dataset de entrenamiento con nuevas muestras con las últimas TTPs y, sobre todo, con la supervisión del analista que, ahora sí, tendrá una idea muy precisa de qué dominios deben ser investigados primero y cuales pueden esperar.

En el proceso de recogida de información para el entrenamiento y la predicción, se decidió incluir mucha más información de la que necesita el algoritmo de machine learning, pero que resulta muy útil al analista para completar la investigación. Esta área presenta aún mucho espació de mejora, tanto en la forma de presentar la información, como en la cantidad de información recogida.

Nuevamente se ha confirmado que la parte más compleja cuando se trabaja con machine learning es la construcción del set de entrenamiento. Si bien la disponibilidad de los datos no es compleja, la diversidad de las fuentes y los distintos criterios a la hora de presentarlos convierten la automatización del proceso en una tarea mucho más ardua de lo esperado.

En caso de decidir usar la aplicación en producción, es recomendable disponer de suscripciones no gratuitas, ya que éstas limitan drásticamente el número de consultas diarias (o por hora) que pueden servir. Incluso el nivel de servicio que ofrece el CCN a sus usuarios resulta insuficiente si se pretende analizar aquellos dominios desconocidos a los que se enfrenta diariamente una gran organización.

12. Trabajos futuros

Hay dos líneas de trabajo que el autor está decidido a acometer con la intención de que este trabajo tenga utilidad práctica:

I. La construcción de una interfaz gráfica que permita al analista:
   - Introducir los dominios a investigar de forma cómoda e intuitiva.
   - Presentar adecuadamente las predicciones calculadas.
   - Presentar la información recogida de cada uno de los dominios, incluyendo enlaces a páginas en las que seguir profundizando.

II. La integración de la herramienta con el SIEM de la organización, de forma que éste identifique los dominios no conocidos, haga un primer filtrado para no colapsar al sistema y los pase a la herramienta para su análisis y clasificación. La mayor dificultad de esta tarea no es la comunicación entre ambas herramientas, sino decidir en base a qué variables se hace ese primer filtrado que debe realizar automáticamente el SIEM.

Además, a corto plazo, se podrían incorporar al algoritmo otros predictores que podrían tener cierta importancia como el Registrador del dominio, la Autoridad de Certificación (CA) de sus certificados SSL, etc. Estas variables generarían un número excesivo de dummies si no se hace un tratamiento previo que ha quedado fuera del alcance inicial de este trabajo.

También sería interesante explorar la capacidad de clasificación de los dominios maliciosos en función de su tipología u objetivos, es decir, separarlos como dominios de mando y control, de distribución de malware, de distribución de phishing, etc. Esto requeriría aumentar considerablemente el número de muestras maliciosas de todas las categorías, así como las benignas.

Por último, teniendo un usuario con suficientes privilegios, sería posible consultar a Reyes sobre la existencia de tickets de incidentes en la herramienta LUCIA o eventos de MISP en los que aparezca el dominio investigado, lo que daría al analista un elemento clave para la decisión. No se ha incorporado en este trabajo inicialmente porque se daba por hecho que los dominios investigados no habían sido reportados previamente como relacionados con ataques o incidentes.

Agradecimientos:

A Javier Reina, analista de ciberseguridad, compañero de trabajo y amigo, quien inspiró este trabajo al sugerir la necesidad de una herramienta como la presentada y que, durante todo su desarrollo, puso a mi disposición sus conocimientos y amplia experiencia en la búsqueda de ciberamenazas y protección de redes.

Al Centro Criptológico Nacional (CCN), por su colaboración a la hora de facilitar el acceso a REYES, eliminando limitaciones y/o restricciones de cuota cuando fue necesario y solucionando todos aquellos problemas con la API que fueron surgiendo.

Referencias:

1. CENTRO CRIPTOLÓGICO NACIONAL. 2021. Reyes 3.4.13 Manual de API.

2. DAVID J. BIANCO. 2014. The Pyramid of Pain. [Última consulta 21-04-22] https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

3. MICHAEL KOCZWARA. 2021. Cobalt Strike Hunting, Red Teams/Threat Actors TTP's. [Última consulta 20-04-22]. https://michaelkoczwara.medium.com/cobalt-strike-hunting-aefe1c5d1ec5

4. WIKIPEDIA. 2021. Dominio de nivel superior. [Última consulta 21-04-22] https://es.wikipedia.org/wiki/Dominio_de_nivel_superior

5. VÍCTOR ROMÁN. 2019. Machine Learning: Cómo desarrollar un modelo desde cero. [Última consulta 20-04-22]. https://medium.com/datos-y-ciencia/machine-learning-cómo-desarrollar-un-modelo-desde-cero-cc17654f0d48

6. JOAQUÍN AMAT RODRIGO. 2020. Regresión logística con Python. [Última consulta 21-04-22]. https://www.cienciadedatos.net/documentos/py17-regresion-logistica-python.html

7. JUAN IGNACIO BAGNATO. 2017. Regresión Logística con Python paso a paso. [Última consulta 20-04-22]. https://www.aprendemachinelearning.com/regresion-logistica-con-python-paso-a-paso/

8. SITIOBIGDATA. 2019. Tipos de datos de aprendizaje automático con ejemplos. [Última consulta 20-04-22]. https://sitiobigdata.com/2019/12/24/tipos-de-datos-de-aprendizaje-automatico-con-ejemplos/#

9. JASON BROWNLEE. 2020. How to Calculate Feature Importance with Python. [Última consulta 20-04-22]. https://machinelearningmastery.com/calculate-feature-importance-with-python/

10. ÁNGEL EULISES ORTIZ. 2019. Asegurando puertos de red riesgosos, seguridad web, informática. [Última consulta 20-04-22]. https://www.hostdime.com.ar/blog/asegurando-puertos-de-red-riesgosos-seguridad-web-informatica/

11. JULIÁN GUTIÉRREZ. 2018. Whois: cómo interpretarlo para OSINT. [Última consulta 21-04-22]. https://ciberpatrulla.com/whois-osint/

12. BLOG DE LINUBE. Fecha desconocida. Comprobar la reputación del dominio y evitar que acabe en spam. [Última consulta 21-04-22] https://linube.com/blog/reputacion-dominio/

13. SOL GONZÁLEZ. 2021. ¿Por qué el Machine Learning es un gran aliado para la ciberseguridad? [Última consulta 21-04-22] https://www.welivesecurity.com/la-es/2021/12/10/por-que-machine-learning-aliado-para-ciberseguridad/

14. ESED BLOG. Fecha desconocida. Machine Learning aplicado en ciberseguridad. Todo lo que necesitas saber. [Última consulta 21-04-22] https://www.esedsl.com/blog/machine-learning-aplicado-en-ciberseguridad

15. DAVID CUTANDA. 2014. Fundamentos sobre Certificados Digitales – El estándar X.509 y estructura de certificados I y II. [Última consulta 21-04-22] https://www.securityartwork.es/2014/04/09/fundamentos-sobre-certificados-digitales-el-estandar-x-509-y-estructura-de-certificados/

Anexo I - Representación de la tabla “dataset.xlsx”

ID	DOMINIO	MALICIOSO	Ult_IP	HI_lastSeen	HI_firstSeen	num_IPs	RD_createdDate	Antiguedad	RD_expiresDate	RD_updatedDate	Actualizado	n_DNS_Servs	DNS_servers	registrarName	TC_country	TC_email	long_email	TLD	Shodan	num_Ports	ports	port21	ftport23	telnport53	dnport22	ssh
0	cdn-jquery.nl	1	34.244.201.1	2022-02-09	2021-09-16	1	2022-02-09	49	-----	2022-02-09	49	3	nsn1.mijndomein.nl, nsn	metaregistrar b.v.	-----	-----	0	nl	YES	6	22, 80, 111, 443, 193	0	0	0	1	
1	www.agoegations.com	1	104.21.30.53	2022-02-09	2021-12-31	7	2021-05-03	331	2022-05-03	2021-12-30	90	2	HARMONY.NS.CLOUDFLA	godaddy.com, llc	china	-----	0	com	YES	10	80, 443, 2053, 2082,	0	0	0	0	
2	tubaho.com	1	23.82.128.160	2021-09-12	2021-08-27	7	2021-08-27	215	2022-08-27	2021-09-12	199	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	30eff7d1f82e404fb7	63	com	YES	7	21, 22, 80, 443, 3306	1	0	0	1	
3	asa.bace.mg	1	170.10.163.4	2022-02-09	2020-04-09	1	2022-02-08	50	2022-09-17	2022-02-08	50	2	dns1.supremepanel.com	stileex	madagascar	bace.mef@gmail.cor	18	mg	YES	19	21, 26, 53, 80, 110, 1	1	0	1	0	
4	itts.apname.org	1	10.10.215.2	2022-02-09	2021-10-18	2	2022-02-09	49	2022-08-30	2022-02-09	49	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	-----	0	org	YES	19	21, 26, 53, 80, 110, 1	1	0	1	0	
5	ferewo.com	1	173.254.235.15	2022-02-05	2022-02-05	1	2021-09-11	200	2022-09-11	2021-09-11	200	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	1e5c8f7dc9584fb09	63	com	YES	20	22, 25, 53, 80, 110, 1	0	0	1	1	
6	fransisgu.com	1	41.77.113.173	2022-01-18	2022-01-04	2	2021-11-09	141	2022-11-09	2021-12-03	117	3	1-YOU.NJALLA.NO, 2-CAN	tucows domains inc.	saint kitts and n	-----	0	com	YES	4	22, 53, 80, 111	0	0	1	1	
7	www.adwlabs.top	1	104.21.18.179	2022-02-09	2021-12-24	4	2022-02-09	49	2022-12-22	2022-02-09	49	2	sullivan.ns.cloudflare.co	dnspod, inc.	china	-----	0	top	YES	10	80, 443, 2082, 2083,	0	0	0	0	
8	vigave.com	1	172.241.29.47	2021-08-05	2021-08-05	1	2021-08-05	237	2022-08-05	2021-09-12	199	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	76e1c285e2ae4aaea	63	com	YES	3	80, 443, 3389	0	0	0	0	
9	citrixworcspace.com	1	46.166.161.148	2021-12-10	2021-12-09	2	2021-12-09	111	2022-12-09	2021-12-11	109	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	45b96dca95814083	63	com	YES	1	22	0	0	0	1	
10	auth.limanowa.top	1	104.21.84.249	2022-02-09	2021-12-01	5	2022-02-09	49	2022-12-01	2022-02-09	49	2	simone.ns.cloudflare.cor	namesilo,llc	united states	-----	0	top	YES	10	80, 443, 2053, 2082,	0	0	0	0	
11	shellcodes.systems	1	83.97.20.246	2022-02-09	2021-09-23	1	2022-02-09	49	2022-09-23	2022-02-09	49	3	1-you.njalla.no, 2-can.nj	tucows domains inc.	canada	-----	0	systems	YES	1	22	0	0	0	1	
12	ssl.chromeupdates.space	1	104.21.71.192	2022-02-09	2021-11-18	4	2022-02-07	51	2022-11-19	2022-02-07	51	2	BRIAN.NS.CLOUDFLARE.C	go daddy, llc	china	-----	0	space	YES	8	80, 443, 2082, 2083,	0	0	0	0	
13	samsunng.tk	1	104.21.69.88	2022-02-09	2021-04-26	9	2022-02-09	49	-----	2022-02-09	49	2	NED.NS.CLOUDFLARE.CO	-----	netherlands	copyright@freenom.	21	tk	YES	11	80, 443, 2082, 2083,	0	0	0	0	
14	wget-upd.com	1	170.130.28.39	2021-10-20	2021-08-26	1	2021-08-26	216	2022-08-26	2021-10-20	161	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	67acfe853cec420cb3	63	com	YES	6	22, 80, 443, 3389, 55	0	0	0	1	
15	cobalt.crimsoncore.be	1	46.101.238.148	2022-02-09	2021-10-18	2	2022-02-09	49	-----	2022-02-09	49	4	ns3.eurodns.com, ns2.eu	eurodns s.a.	-----	-----	0	be	YES	4	22, 80, 443, 50050	0	0	0	1	
16	ksdb.ru	1	91.236.120.238	2022-02-09	2021-09-13	4	2022-02-09	49	2022-09-08	2022-02-09	49	2	ns1.bitwebdns.net, ns2.b	regru-ru	-----	-----	0	ru	YES	4	22, 80, 443, 1200	0	0	0	1	
17	sx.bypass.net.cn	1	104.21.31.66	2022-02-09	2021-01-25	2	2022-02-09	49	2023-09-04	2022-02-09	49	2	edward.ns.cloudflare.cor	帝思普网络科技有限	-----	hackwenyu77@gmai	21	cn	YES	10	80, 443, 2053, 2082,	0	0	0	0	
18	rurofo.com	1	45.147.230.236	2021-08-22	2021-08-22	2	2021-08-22	220	2022-08-22	2021-09-12	199	2	DNS1.REGISTRAR-SERVER	namecheap, inc.	iceland	c5b231fcf8c740a39	63	com	YES	1	3389	0	0	0	0	
19	www.edge-chrome.com	1	47.243.22.29	2022-02-09	2021-06-07	2	2021-04-25	339	2022-04-25	2021-06-23	280	2	NS29.DOMAINCONTROL.	godaddy.com, llc	china	-----	0	com	YES	1	443	0	0	0	0	
20	migrdeb.com	1	139.60.161.215	2022-02-10	2022-02-10	3	2022-02-01	57	2023-02-01	2022-02-01	57	2	NOW1.DNS.COM, NOW2	net international limit	united states	-----	0	com	YES	3	22, 80, 443	0	0	0	1	

port80	port443	80_http_status	443_http_status	IP_Organization	IP_isp	num_Subdomains	category	cats	reputation	Cert_https	LastCertDate	PKalgorithm	CertSignAlg	ValNoDespues	ValNoAntes	CA_Issuers	Issuer_CN	subject_CN	Archive_first	AntigArch	Archive_last	AntigLast	Archive_year
1	1	0	0	Amazon Data Services	Amazon.com, Inc.	0	0	-----	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	400	400	Cloudflare, Inc.	Cloudflare, Inc.	1	1	newly registered website	0	1	2022-01-21	EC	sha256ECDSA	2022-12-29	2021-12-30	http://cacerts.digicert.com/(	Cloudflare Inc	sni.cloudflaressl.(	9999-12-31	0	9999-12-31	0	0
1	1	0	0	Leaseweb USA, Inc.	Leaseweb USA, Inc.	5	1	Malware Sites, media sha	0	0	-----	-----	-----	-----	-----	-----	-----	-----	2013-08-09	3155	9999-12-31	0	0
1	1	200	200	LiquidNet US LLC	Steadfast	29	1	government, media shari	0	1	2022-01-30	RSA	sha256RSA	2022-03-09	2021-12-09	http://crt.comodoca.com/cP	cPanel, Inc. Ce	asa.bace.mg	2021-04-14	350	2022-01-23	66	2
1	1	200	200	LiquidNet US LLC	Steadfast	3	1	newly registered website	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	0	0	QuadraNet Enterprise	QuadraNet Enterpri	3	1	malicious web sites, com	0	1	2022-02-06	EC	sha256RSA	2022-05-07	2022-02-06	http://r3.i.lencr.org/	R3	ferewo.com	9999-12-31	0	9999-12-31	0	0
1	0	200	0	Genious Cloud Platfo	GloboTech Commu	1	1	malware, Malware Sites,	0	1	2021-11-22	EC	sha256RSA	2022-02-07	2021-11-09	http://r3.i.lencr.org/	R3	fransisgu.com	9999-12-31	0	9999-12-31	0	0
1	1	400	400	Cloudflare, Inc.	Cloudflare, Inc.	1	1	newly registered website	0	1	2022-01-06	RSA	sha256RSA	2022-03-22	2021-12-22	http://r3.i.lencr.org/	R3	*.adwlabs.top	9999-12-31	0	9999-12-31	0	0
1	1	0	0	Leaseweb USA, Inc.	Leaseweb USA, Inc.	58	1	Malware Sites, spyware a	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
0	0	0	0	UAB Cherry Servers	UAB Cherry Servers	0	1	newly registered website	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	400	400	Cloudflare, Inc.	Cloudflare, Inc.	1	1	spyware and malware, m	0	1	2022-01-29	EC	1.2.840.1004	2022-04-29	2022-01-29	http://e1.i.lencr.org/	E1	*.limanowa.top	9999-12-31	0	9999-12-31	0	0
0	0	0	0	OVO Systems Ltd.	M247 Ltd	0	1	malware, malicious web	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	400	300	Cloudflare, Inc.	Cloudflare, Inc.	1	0	-----	0	1	2021-11-19	RSA	sha256RSA	2022-02-17	2021-11-19	http://r3.i.lencr.org/	R3	*.chromeupdates	9999-12-31	0	9999-12-31	0	0
1	1	400	300	Cloudflare, Inc.	Cloudflare, Inc.	0	0	-----	0	1	2021-12-09	EC	sha256ECDSA	2022-04-25	2021-04-26	http://cacerts.digicert.com/(	Cloudflare Inc	sni.cloudflaressl.(	9999-12-31	0	2021-12-23	97	1
1	1	0	0	SSDBlaze, LLC	Eonix Corporation	2	1	bot networks, spyware ar	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	400	400	DigitalOcean, LLC	DigitalOcean, LLC	11	1	Phishing and Other Fraud	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	0	0	BitWeb LLC	BitWeb LLC	1000	1	malware, media sharing,	0	1	2021-11-13	RSA	sha256RSA	2022-02-11	2021-11-13	http://r3.i.lencr.org/	R3	ksdb.ru	9999-12-31	0	9999-12-31	0	0
1	1	400	200	Cloudflare, Inc.	Cloudflare, Inc.	1	0	-----	0	1	2021-12-30	EC	sha256ECDSA	2022-12-03	2021-12-04	http://cacerts.digicert.com/(	Cloudflare Inc	sni.cloudflaressl.(	9999-12-31	0	9999-12-31	0	0
0	0	0	0	combahton GmbH	combahton GmbH	300	1	Malware Sites	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
0	1	0	400	Alibaba.com LLC	Alibaba (US) Techno	1	0	-----	0	0	-----	-----	-----	-----	-----	-----	-----	-----	9999-12-31	0	9999-12-31	0	0
1	1	400	400	HOSTKEY	HOSTKEY	2	1	malware, newly registere	0	1	2022-03-02	RSA	sha256RSA	2022-05-11	2022-02-10	http://r3.i.lencr.org/	R3	migrdeb.com	9999-12-31	0	9999-12-31	0	0

Anexo II – Software empleado en el desarrollo.

- VISUAL STUDIO CODE. Versión 1.66.2 (para macOS)
- PYTHON. Versión 3.8.5 64-bit.

  Se han instalado las siguientes librerías de python:

  - Pandas. Versión 1.2.0
  - Pycurl. Versión 7.44.1
  - Requests. Versión 2.26.0
  - Shodan. Versión 1.25.0
  - Waybackpy. Versión 3.0.2
  - Matplotlib. Versión 3.4.2
  - Sklearn (Scikit-learn). Versión 0.24.1.
    Se importaron los módulos:
    - linear_model
    - metrics

  Además, es necesario importar los siguientes módulos de la Librería Standard:

  - Operator.
  - Os.
  - Time.
  - Datetime.
  - Io.
  - Json.

Anexo III - Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**				X
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**		X		
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades**.			X	X
ODS 11. **Ciudades y comunidades sostenibles**.				X
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**	X			
ODS 17. **Alianzas para lograr objetivos.**				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Este TFM tiene un grado de relación alto con el ODS N.º 9 “Industria, innovación e infraestructuras”. En primer lugar, por proponer una aplicación diferente del Machine Learning, una rama de la inteligencia artificial que no solo es innovadora si no incluso disruptiva. Si bien el aprendizaje automático se está consolidando en el área de la ciberseguridad, su uso para identificar dominios maliciosos en base a sus características constituye un uso innovador que permitirá utilizar los recursos con mayor eficacia (Meta 9.4). En segundo lugar, porque su finalidad última es conseguir un internet más seguro, lo cual ayudaría a aumentar significativamente el acceso a la tecnología de la información y las comunicaciones, facilitando un acceso universal, asequible y seguro a Internet (Meta 9.c).

También se considera que está muy relacionado con el ODS N.º 16 “Paz, justicia e instituciones sólidas”. Los conflictos, la inseguridad, las instituciones débiles y el acceso limitado a la justicia continúan suponiendo una grave amenaza para el desarrollo sostenible. No se debe olvidar que la ciberdelincuencia es ya el negocio ilícito más lucrativo. En este sentido, la identificación precoz de las herramientas que usan los ciberdelincuentes para cometer sus delitos debería ayudar a complicarles sus objetivos. La alta especialización que requiere esta actividad hace que los delincuentes suelan constituirse en grupos organizados, siendo la lucha contra todas las formas de delincuencia organizada una de las metas de este ODS (Meta 16.4). La reducción del soborno, que sin duda es una de las manifestaciones más comunes de ciberdelincuencia, es otra de las metas que persigue el ODS 16 (Meta16.5). En última instancia, la motivación del TFM era optimizar el trabajo en el SOC que da soporte a las redes del Ministerio de Defensa, colaborando así a unas instituciones públicas más sólidas y seguras.

Por otra parte, es posible encontrar una relación media con el ODS Nº8 “Trabajo decente y crecimiento económico”. Uno de los objetivos principales del TFM es lograr optimizar el trabajo en los SOCs, que de forma generalizada se enfrentan a recursos de personal muy escasos en comparación con el número de ciberamenazas a las que se enfrentan. Esto que inicialmente podría implicar una menor necesidad de personal es, en realidad, una forma de hacer que el trabajo de los técnicos sea más eficaz y de mayor calidad, eliminando tareas de triaje simples pero que consumen mucho tiempo y energía. Además, si este trabajo colabora a evitar o reducir ciberataques a empresas, se estaría también fomentando el crecimiento de las microempresas y las pequeñas y medianas empresas, que son generalmente las que tienen mayor dificultad para superar el impacto de un ciberataque. Todo esto está recogido en las Metas 8.2 y 8.3 de este ODS.

Por último, este TFM podría relacionarse de forma colateral con el ODS Nº 10 “Reducción de las desigualdades”. Los países más ricos y desarrollados tecnológicamente son, no solo los que mejor pueden soportar las consecuencias de las actividades delictivas de los ciberdelincuentes, también suelen ser el lugar de origen de estos grupos, que no dudan en buscar las victimas más vulnerables en entornos menos favorecidos. Esto a la postre acaba fomentando las desigualdades contra las que lucha este ODS.

En resumen, el TFM puede contribuir a mejorar la detección de actividades maliciosas en internet, minimizando el impacto de los ciberataques en las empresas e instituciones, fomentando un acceso a la información más seguro y luchando contra la delincuencia organizada.