ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN INGENIERÍA INDUSTRIAL

ANÁLISIS FORENSE DE UN DATASET INDUSTRIAL Y PROPUESTA DE UN ESTÁNDAR GRÁFICO PARA LOS REGISTROS EN LA CIBERSEGURIDAD

Autor: Jorge Buil García
Director: Álvaro López López

Madrid

Junio 2018

Análisis forense de un dataset industrial y propuesta de un estándar gráfico para los registros en la ciberseguridad

Autor: Jorge Buil García
Director: Álvaro López López

***Abstract*—La Industria 4.0 incluye muchas nuevas características que han llevado a la aparición de nuevos paradigmas y caminos para vislumbrar la realidad que nos rodea. Actualmente, en cada instalación industrial se pueden encontrar sistemas de monitorización y supervisión que están al tanto de todo tipo de actividad. Todos estos procesos dejan un rastro en forma de registros llamados *logs*, los cuales se obtienen, codificados en hexadecimal, con programas que se encargan de monitorear el tráfico de la red. Después, se pueden también interpretar y de esa manera que puedan ser analizados por expertos. El principal inconveniente es que la información recogida está presentada en un gran número de campos, así como la diversidad de los caracteres que los forman. Esto conlleva a que este entorno presenta una gran dificultad para cualquier persona que se encargue de revisarlos y de analizarlos. Pero ya no sólo para usuarios, sino también para cualquier tipo de técnica de machine learning, si se quiere trabajar con los logs ya interpretados. Por ello, con este Trabajo fin de Máster se quiere construir un puente que una el mundo del análisis de datos (tanto forense o normal, dentro del campo de *computer science*) con el de las técnicas de machine learning.Por ello se realizará un análisis forense de los registros de un conjunto de datos de ciberseguridad industrial no tratados y se propondrá, finalmente, un estándar en formato imagen que sirva como puente entre estos dos mundos.**

***Index Terms*—Cybersecurity, Industry 4.0, SIEM, logs, SOC, Machine Learning, Deep Learning, ICS, IoT, Forensic Analysis, Graphic Standard, Network Forensics**

I. Introduction

EL fenómeno de la Industria 4.0 fue inicialmente mencionado en Alemania, 2011[1], como una propuesta de desarrollo para el nuevo concepto de política económica alemana basado en estrategias de gran tecnología. Este incluye los llamados sistemas ciber-físicos, el *Internet of Things* (IoT)[2] y el *Internet of Services* (IoS)[3] entre otros, así como un cambio de mentalidad e infraestructuras de gran importancia en todos los sentidos.

Actualmente, en cada instalación industrial se encuentran sistemas de monitorización y supervisión que están al tanto de todo tipo de actividad sobre lo que está ocurriendo dentro de la planta y de llevar un control adecuado de ésta. Tanto SCADAs[4], DCSs y PLCs tienen que estar conectados a través de una red de trabajo para así poder recibir información de todos y cada uno de los dispositivos y sensores distribuidos por ésta, así como generar un control adecuado sobre cada uno de ellos[5]. Si no se disponen de las medidas de ciberseguridad necesarias, estos elementos pueden ser una puerta de acceso para cualquier agente de amenaza. Los más reprsentativos se describen en el informe del CCN-CERT[6]. El daño que son capaces de causar, si no se está bien protegido puede ser fatal, no sólo por la pérdida de servicios o disponibilidad, sino también por la filtración de datos de usuarios, datos privados que, debido a la gran difusión que existe a través de internet, equivaldría a una pérdida en la confianza y en la imagen que sería desastrosa para la compañía y el sector en sí mismo.

La ciberseguridad puede definirse como el cuerpo de las tecnologías, procesos y prácticas diseñadas para proteger las redes, ordenadores y los datos de ataques, daños y accesos no autorizados[7]. Los recursos más importantes a analizar son los *logs*, estos son un registro de todos los eventos que tienen lugar en la red de trabajo y en su análisis está la clave para poder diagnosticar y conocer el estado de ésta. Debido a esto, actualmente se está buscando cómo poder analizar estos *logs* mediante diferentes técnicas de *machine learning*[8], [9], [10], [11]. El objetivo final es que estos dispositivos puedan diagnosticar, alertar, prevenir y remediar cualquier tipo de amenaza o ataque que se produzca.

Pero existe un problema, y es que los *logs*, al ser grabaciones de huellas de diferentes eventos, no se encuentran estandarizados y esto es un proceso que cada organización ha de resolver por su cuenta. Además, los programas que permiten el análisis de estos *logs* los tratan en un formato apenas legible para ningún módulo de *machine learning*. Y es aquí donde éste paper se centra, con el análisis forense de un dataset industrial y con la propuesta de un estándar gráfico para el análisis de los registros de ciberseguridad, se pretende construir un puente entre estos dos mundos.

Este paper se estructura de la siguiente manera: la Sección II muestra el análisis forense realizado del conjunto de datos industriales, en el cual se comentará cómo se ha llevado a cabo y los principales resultados de éste. Seguido, en la Sección III se hablará de la propuesta del estándar en formato imagen que servirá de unión entre los dos mundos ya comentados. La Sección IV se centrará en los resultados de este estándar y en el análisis campo por campo de las imágenes y, por último, las Secciones V y VI presentarán las conclusiones del análisis forense y de las imágenes así como el futuro trabajo a desarrollar.

II. Análisis forense de un conjunto de datos industriales

Con el objeto de poder construir un puente entre el mundo del análisis de registros (orientados a la ciberseguridad) y el mundo del *machine learning*, en primer lugar se comenzará realizando un análisis forense de un conjunto de datos industriales.

El conjunto de datos del que se dispone pertenece al utilizado en la conferencia sueca sobre ciberseguridad industrial (4SICS GeekLounge). La dimensión de este dataset es de 350MB de tráfico capturado en la red. Entre los dispositivos industriales se encuentran SCADAs y otros ICS (*industrial control systems*).

A. Agentes de la amenaza

A lo largo de este apartado, con el objetivo de contextualizar la situación actual, se van a nombrar los principales agentes de la amenaza recogidos en el informe CCN-CERT IA-09/18 "Ciberamenazas y Tendencias"[6].

- Los estados como agentes de las amenazas
- Ciberdelincuentes
- Terrorismo y ciberyihadismo
- Los Hacktivistas
- Cibervándalos y Script kiddies
- Actores internos
- Organizaciones privadas

B. 4SICS GeekLounge Dataset

El dataset utilizado corresponde al de la conferencia sueca de 4SICS sobre ciberseguridad industrial. Consta de 3 archivos .pcap (formato en el que se exportan los registros del tráfico monitoreado de una red). Estos son:

- 4SICS-GeekLounge-151020.pcap
  - 25MB
  - 246.137 paquetes
- 4SICS-GeekLounge-151021.pcap
  - 136MB
  - 1.253.100 paquetes
- 4SICS-GeekLounge-151022.pcap
  - 204MB
  - 2.274.747 paquetes

C. Analizadores de red

Los analizadores de redes también son conocidos como analizadores de paquetes para los logs¸ los registros. Estos pueden ser definidos como aquellos programas que se encargan de "escuchar" todo el tráfico que se transmite dentro de la red de trabajo. Se pueden determinar qué tipos de amenazas se han sufrido en la red y cómo detectarlas.

El proceso de *network sniffing* (que es como se le llama a tener uno de estos programas monitoreando a tiempo real una red), se muestra a continuación:

- Recolección
- Conversión
- Análisis

Una vez conocidas sus tres principales funciones, es importante saber que actualmente existe una gran variedad de programas analizadores de redes. A continuación, se nombrarán tres analizadores open source (código abierto) de gran repercusión en este mundo:

- Tcpdump
- Windump
- Wireshark

Estas dos peculiaridades: la posibilidad de seguir una transmisión, así como la interfaz gráfica de usuario, han resultado fundamentales para la elección del programa Wireshark de entre los otros. Como base para el estudio y valoración de la propuesta que se plantea en este proyecto se utilizará éste como analizador de redes.

D. Wireshark

Wireshark, como se ha comentado en la Sección 2.4, se trata de un analizador de paquetes. Es una herramienta básica para observar los mensajes intercambiados entre aplicaciones.

1) Principales funciones: Consta principalmente de dos elementos,, Los cuales, se mencionan a continuación:

- Librería de captura de paquetes
- Analizador de paquetes

2) Características distintivas:

- Interfaz gráfica
- Capacidad de filtrado

E. Análisis forense con Wireshark

A lo largo de esta sección se presentará el análisis forense realizado en el dataset. Para realizar el análisis forense, el punto de inicio fundamental es fijarse en el protocolo ya que según cuál se haya utilizado habrá que tener en cuenta unas cosas u otras.

1) Protocolos del dataset: Aquí se van a enumerar los protocolos encontrados dentro del dataset para que, en la siguiente sección, se pueda hablar de los más representativos y cómo descubrir las posibles amenazas.

TCP	UDP	ICMP	ARP	SIP	OpenVPN
HTTP	DNS	FTP	SSH	DCE/RPC	COTP
TELNET	SSL	TLS	RSH	S7COMM	ESP
NTP	NBNS	NBSS	Portmap	LOOP	DNP3
Modbus/TCP					

TABLE I: Protocolos encontrados en el dataset

De dónde se obtiene que los protocolos más representativos son: TCP, ICMP, ARP, UPD, DNP3 y FTP.

2) Amenazas: A continuación se enumerarán los protocolos más representativos junto con los principales ataques y amenazas que se han ido identificando durante el análisis forense:

a) TCP:

- Ack scanning: se atacan puertos y su objetivo es determinar si el objetivo está protegido por algún tipo de filtrado.
- Xmas scan: se utiliza para descubrir si un determinado puerto está cerrado.
- Fin scan: se utiliza para descubrir si un determinado puerto está cerrado.

- Fin-Ack scanning: se utiliza con la intención de saturar la red o cerrar determinados puertos TCP (denegación de servicio).
- Syn scan / Stealth scan / Half open attack: se utiliza para saber si un puerto está abierto o cerrado.
- Rst attack: es un tipo de ataque de denegación de servicio.
- Syn flood attack: se utiliza tanto para saber si un puerto está abierto como para realizar una denegación de servicio.
- Tcp ping sweep: se utiliza para descubrir que IPs están disponibles en la red.
- Null scan: se utiliza para averiguar si un determinado puerto está cerrado.
- Psh-Ack flood: se utiliza para realizar una denegación de servicio.
- Tcp-Dup-Ack attack / Duplicate ack spoofing: se utiliza para realizar un ataque de denegación de servicio.

b) ICMP:

- Icmp scan / ping sweep: se utiliza para averiguar que IPs están disponibles.
- Icmp route direction / redirect: se utiliza para realizar ataques tipo *man-in-the-middle*.
- IP protocol scan: se utiliza para descubrir protocolos qué protocolos se pueden ejecutar sobre la IP objetivo.
- Smurf attack: es un tipo de ataque de denegación de servicio basado en *pings*.
- Ping of death: se utiliza para realizar ataques de denegación de servicio.
- Icmp tunneling: se utiliza para llevar a cabo comunicaciones encubiertas.
- Icmp get timestamp: se utiliza para conocer la latencia a lo largo de la red.

c) ARP:

- Arp poisoning / spoofing: se utiliza para realizar ataques de suplantación de identidad, y luego ya perpetrar el verdadero ataque.
- Arp sweep / scan: se utiliza para obtener las IPs de los *host* de una red.

d) UDP:

- Udp scan: se utiliza para descubrir si un determinado puerto está disponible.
- Udp flood attack: se utiliza para realizar un ataque de denegación de servicio.

e) DNP3:

- Dnp3 warm reset: se utiliza para realizar un reinicio parcial del sistema.
- Dnp3 cold start: se utiliza para realizar un reinicio total del sistema.
- Dnp3 time change: se utiliza para realizar un ataque de denegación de servicio.

f) FTP:

- Brute force attack: se utiliza para entrar y conseguir acceso al servidor.
- Packet Capture / Sniffing: se utiliza para conseguir usuarios y contraseñas de usuarios de la red.

III. Propuestaa del estándar en formato imagen

Tras haber presentado las posibles amenazas detectadas en el dataset mediante el programa capturador de paquetes Wireshark, lo siguiente es exponer cómo se ha ido conformando la propuesta en formato imagen. Para ello a lo largo de este capítulo se repasarán las bases sobre las que se empezó el proceso iterativo para la conformación del estándar. Tras éstas se presentará la solución final propuesta junto con los campos que se pueden visualizar en ella.

A. Bases del formato imagen

En esta sección se pretenden establecer las bases que conforman el estándar en formato imagen

- Matriz de 128x128, tamaño pequeño que permite análisis visual.
- Escala de grises UINT-8, de 0-255 matrices de 2 dimensiones.
- Diccionario de conversión, para pasar cada elemento a la escala de 0-255.

B. Solución final del estándar propuesto

A continuación se mostrará la solución final del estándar propuesto para visualizar las imágenes. En primer lugar se observan los campos que se han incluido, los originales de Wireshark y los extra añadidos tras un análisis de información intrínseca de esos dos protocolos. Tras estas dos se puede ver la conformación final de la imagen.

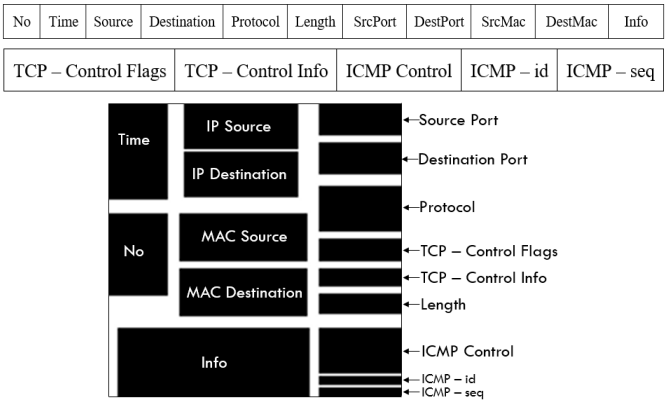


Fig. 1: Imagen Final

IV. Resultados y análisis de las imágenes

A lo largo de esta sección se mostrarán los resultados obtenidos del análisis forense representados en las imágenes. Para ello se mostrará un análisis, campo por campo, del estándar de las imágenes.

A. Análisis de las imágenes

Para este propósito se cogerán las imágenes correspondientes a una transmisión TCP entre dos dipositivos, en este caso corresponde a un ataque de escaneo de puertos Syn scan.

Fig. 2: Transmisión TCP convertida a imágenes

1) Dirección IP: Al estar evaluando una transmisión, se tiene una petición y una respuesta, respectivamente.

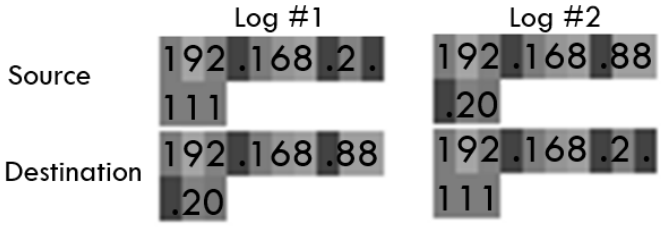


Fig. 3: Campos de dirección IP

2) Puertos & Protocolos: De nuevo, la petición y la respuesta, a nivel puertos y protocolos, se muestra a continuación:

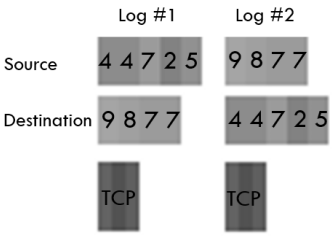


Fig. 4: Campos de puertos y protocolo

3) Dirección MAC: La dirección MAC actúa como idenficador físico del dispositivo. A efectos prácticos es como la dirección IP pero única.

Fig. 5: Campos de la dirección MAC

4) Longitud de la trama & TCP - Control: El campo Length ayuda a saber la extensión de la trama mientras que el campo de TCP Control fue añadido tras analizar información implícita.

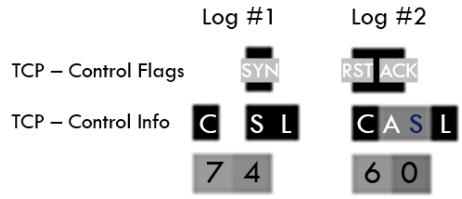


Fig. 6: Longitud de la trama y campos de TCP - Control

A continuación se explica el campo TCP - Control:

- **TCP - Control Flags:** RST, ACK, SYN, FIN, PSH, URG.
- **TCP - Control Info:** este campo se encarga de evaluar información contenida en la trama, permitirán con mayor facilidad la identificación de un posible ataque. Los parámetros que se evalúan son:
  - **C** → Control: En negro si algunos de los 3 siguientes está presente.
  - **A** → ack: si ack = 0 adoptará un color negro y si ack = 1 será gris.
  - **S** → seq: si seq = 0 adoptará un color negro y si seq = 1 será gris.
  - **L** → len: si len = 0 adoptará un color negro y si len = 1 será gris.
  - **T** → TCP: si aparece ”TCP” en la trama será negro, si no blanco.
  - **D** → Dup: si aparece ”Dup” en la trama será negro, si no blanco.

5) Campos de control protocolo ICMP: Para poder evaluar este campo es necesario estar frente a un registro cuya comunicación fuera mediante el protocolo ICMP. Por ello se mostrarán los pertenecientes a otra imagen:

Fig. 7: Log perteneciente al protocolo ICMP

Este campo contienen información de gran utilidad para detectar y analizar sus registros. La información que queda representada se muestra a continuación:

- ICMP – Control: campo que corresponde a determinados códigos de error y tipos del protocolo ICMP.
  - Echo (ping) request → ICMP Type 8

  - Echo (ping) reply → ICMP Type 0
  - Destination unreachable → ICMP Type 3
  - No response found → dentro del ICMP Type 8 cuando no se registra ninguna respuesta
- ICMP – id: se encarga de representar el valor del "id" correspondiente a este protocolo, LE/BE.
- ICMP – seq: se encarga de representar el valor de "seq", LE/BE.

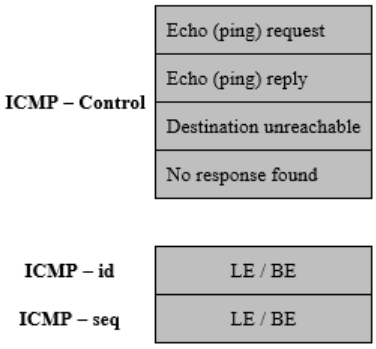


Fig. 8: Distribución de los campos de análisis ICMP

V. Conclusiones

Las conclusiones se pueden dividir principalmente en dos grandes grupos, debido a la dualidad de este proyecto:

A. Conclusiones sobre el análisis forense

- El conjunto de datos provienen de una convención que consistía en perpetuar ataques a un conjunto de dispositivos industriales. Por esto no se está ante una situación "normal" y los datos, así como sus resultados, podrían estar sesgados a determinados protocolos.
- El 93% del volumen de los datos pertenecen a los protocolos TCP e ICMP, de ahí que sean los que mayor ataques se han detectado, si se hubieran tenido más ataques de los 23 protocolos restantes se habrían producido más de estos últimos.
- Se han estudiado los protocolos de manera independiente para facilitar su análisis, como muchos están presenten en varias capas del modelo OSI igual se ha perdido parte de la visión global.

B. Conclusiones generales sobre las imágenes

- El estándar en formato imagen permite de forma rápida la distinción entre protocolos. Así como una inspección visual acertada de distintos problemas como escaneos de puertos, direcciones IP y MAC.
- Al haberse incluido campos extras para el análisis de los protocolos ICMP y TCP se han obtenido mejoras sustanciales a la hora de identificar amenazas dentro de estos. Se añadieron debido a su elevado volumen pero si se hubieran tenido otros datos, los campos especializados habrían cambiado de protocolo.
- El formato UINT-8 puede presentar dificultades visuales a la hora de distinguir entre tonalidades de grises muy cercanas entre ellas, lo cual puede suponer un problema a nivel visual y quizás también para el módulo de *machine learning* puesto que no se ha podido testear.
- Los campos, al estar estructurados desde su concepción de forma modular, permiten de forma clara la unión entre los dos mundos ya planteados y además una fácil implementación a la hora de redistribuir y modificar cualquiera de ellos.
- Los *logs* siguen transmisiones de comunicación, esto hace que un conjunto de 2-3-5 registros formen parte de la misma transmisión. Para poder analizar las amenazas y los posibles ataques es necesario estudiar la transmisión completa.

VI. Futuros desarrollos

Este proyecto deja la puerta abierta a futuros desarrollos. Los principales se presentan a continuación:

- **Realizar las imágenes a color(RGB)**: pasando de las dos dimensiones de escala de grises a las correspondientes a una imagen a color.Habría que estudiar si las ventajas obtenidas en el análisis visual al tener ahora una gama de colores compensarían las posibles complicaciones para poder analizar las imágenes como matrices de píxeles RGB.
- **Representar la información desde la *raw binary data* (en hexadecimal)**: utilizar la información obtenida de manera bruta capturada por los *packet sniffers* para conformar las imágenes de forma que sea fácilmente interpretada por un análisis visual sería un gran avance. Actualmente es necesario pasar por el analizador de protocolos para interpretar la información y poder exportarla para así conformar las imágenes.
- **Construir un módulo de *machine learning* para analizar y clasificar las imágenes**:ahora que se tienen ya las imágenes estructuradas y preparadas para el análisis visual y numérico (por los píxeles), uno de los siguientes puntos debería ser construir un módulo de *machine learning* que utilice técnicas de análisis de imágenes o *deep learning* para poder aprender de las transmisiones y poder clasificar y detectar posibles amenazas y ataques. Es importante construir el módulo de tal forma que tenga "memoria" para poder ser capaz de analizar las transmisiones de 2-3-5 imágenes.
- **Ampliar el número de campos de información ya analizada de más protocolos**: al igual que se ha hecho con los protocolos TCP e ICMP se pueden desarrollar más campos con información implícita
- **Estudiar la comunicación de un módulo de *machine learning* con un SIEM**: lo cual sería el objetivo último a nivel logístico y operativo. Ser capaz de poder cotejar a tiempo real las distintas amenazas con el módulo de *machine learning*.

References

[1] V. Roblek, M. Meško, and A. Krapež, "A complex view of industry 4.0," *SAGE Open*, vol. 6, no. 2, pp. 1–11, 2016.

[2] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0007681315000373

[3] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Key Challenges in Cloud Computing: Enabling the Future Internet of Services," *IEEE Internet Computing*, vol. 17, no. 4, pp. 18–25, Jul. 2013.

[4] Y. Zhang, L. Wang, Y. Xiang, and C.-W. Ten, "Power System Reliability Evaluation With SCADA Cybersecurity Considerations," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1707–1721, Jul. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7042739/

[5] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri, "The Cybersecurity Landscape in Industrial Control Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, May 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7434576/

[6] CCN-CERT, "CCN-CERT_ia-16-17 Ciberamenzas and Tendencias," Resumen ejecutivo Edición 2017, 2017.

[7] N. Diakun-Thibault, "Defining Cybersecurity," *Technology Innovation Management Review*, vol. 2014, pp. 13–22, Oct. 2014.

[8] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey," *arXiv:1701.02145 [cs]*, Jan. 2017, arXiv: 1701.02145. [Online]. Available: http://arxiv.org/abs/1701.02145

[9] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7307098/

[10] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, Jan. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1084804512001178

[11] Y. Chang, W. Li, and Z. Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine." IEEE, Jul. 2017, pp. 635–638. [Online]. Available: http://ieeexplore.ieee.org/document/8005870/

Forensic Analysis of an industrial dataset and standard graphic proposal for cybersecurity logs

Author: Jorge Buil García
Director: Álvaro López López

***Abstract*—Industry 4.0 includes many new features that have led to the emergence of new paradigms and paths to glimpse the reality that surrounds us. Currently, in each industrial facility it can be found monitoring and supervision systems that are aware of all types of activity. All these processes leave a trace in the form of records called logs, which are obtained, encoded in hexadecimal, with programs that are responsible for monitoring network traffic. Then, they can also be interpreted and in that way they can be analyzed by experts. The main drawback is that the information collected is presented in a large number of fields, as well as the diversity of the characters that form them. This leads to an environment that presents a great difficulty for anyone who is responsible for reviewing and analyzing them. But not only for users, but also for any type of machine learning technique, in order to work with logs, they must be already interpreted. For this reason, with this Master's Thesis a bridge is going to be built. A bridge that unites the world of data analysis (both forensic or normal, within the field of computer science) with machine learning techniques. A forensic analysis of the records of an untreated industrial cybersecurity data set will be carried out and finally, a image standard format will be proposed that will serve as a bridge between these two worlds.**

***Index Terms*—Cybersecurity, Industry 4.0, SIEM, logs, SOC, Machine Learning, Deep Learning, ICS, IoT, Forensic Analysis, Graphic Standard, Network Forensics**

I. Introduction

THE phenomenon of Industry 4.0 was initially mentioned in Germany, 2011[1], as a development proposal for the new concept of German economic policy based on high-tech strategies. This includes the so-called cyber-physical systems, the Internet of Things (IoT)[2] and the Internet of Services (IoS)[3] among others, as well as a change of mentality and infrastructures of great importance in all senses.

Currently, in each industrial installation there are monitoring and supervision systems that are aware of all types of activity about what is happening inside the plant and to keep an adequate control of it. Both SCADAs[4], DCSs and PLCs have to be connected through a network of work in order to receive information on each and every one of the devices and sensors distributed by it, as well as to generate an adequate control over each one of them[5]. If the necessary cybersecurity measures are not available, these elements can be a gateway for any threat agent. The most representative are described in the report of the CCN-CERT[6]. The damage they are capable of causing, if they are not well protected, can be fatal, not only due to the loss of services or availability, but also due to the leakage of user data, private data that, due to the wide dissemination that exists to through the Internet, it would amount to a loss of confidence and image that would be disastrous for the company and the sector itself.

Cybersecurity can be defined as the body of technologies, processes and practices designed to protect networks, computers and data from attacks, damages and unauthorized access[7]. The most important resources to analyze are the logs, these are a record of all the events that take place in the network of work and in its analysis is the key to be able to diagnose and know the state of it. Due to this, they are currently looking to analyze these logs using different Machine Learning techniques[8], [9], [10], [11]. The ultimate goal is that these devices can diagnose, alert, prevent and remedy any type of threat or attack that may occur.

But there is a problem, and that is that the logs, being track recordings of different events, are not standardized and this is a process that each organization has to solve on its own. In addition, the programs that allow the analysis of these logs treat them in a barely legible format for any module of machine learning. And this is where this paper focuses, with the forensic analysis of an industrial dataset and with the proposal of a graphic standard for the analysis of cybersecurity records, it is intended to build a bridge between these two worlds.

This paper is structured as follows: Section II shows the forensic analysis made of the industrial dataset, which will comment on how it has been carried out and the main results of it. Next, Section III will discuss the proposal of the standard in image format that will serve as a link between the two worlds already commented. Section IV will focus on the results of this standard and the field-by-field analysis of the images and, finally, Sections V and VI will present the conclusions of the forensic analysis and the images as well as the future work that might be developed.

II. Forensic analysis of an industrial data set

In order to build a bridge between the world of the analysis of records (oriented to cybersecurity) and the world of machine learning, a forensic analysis of a set of industrial data will be performing at the beginning.

The dataset available belongs to that used in the Swedish conference on industrial cybersecurity (4SICS GeekLounge). The dimension of this dataset is 350MB of traffic captured in the network. Among the industrial devices are SCADAs and other ICS(industrial control systems) devices.

A. Threat Agents

Throughout this section, in order to contextualize the current situation, the main threat agents included in the report CCN-CERT IA-09/18 "Cyber Threats and Trends" will be presented.

- The states as agents of threats
- Cyber criminals
- Terrorism and cyberjayism
- The Hacktivists
- Cybervandals and Script kiddies
- Internal players
- Private organizations

B. 4SICS GeekLounge Dataset

The dataset used corresponds to that of the Swedish 4SICS conference on industrial cybersecurity. It consists of 3 .pcap files (format in which records of monitored traffic of a network are exported). These are:

- 4SICS-GeekLounge-151020.pcap
  - 25MB
  - 246.137 packets
- 4SICS-GeekLounge-151021.pcap
  - 136MB
  - 1.253.100 packets
- 4SICS-GeekLounge-151022.pcap
  - 204MB
  - 2.274.747 packets

C. Network analyzers

Network analyzers are also known as packet analyzers for logs¸ registers. These can be defined as those programs that are in charge of "listening" to all the traffic that is transmitted within the work network. With them it can be determined what types of threats have been suffered in the network and how to detect them.

The network sniffing process (which is how it is called when one of these programs is monitoring a network in real time), is shown below

- Collection
- Conversion
- Analysis

Once these three main functions are known, it is important to know that there is currently a wide variety of network analyzer programs. Next, emphasis will be placed on three open source analyzers (open source) of great impact in this world:

- Tcpdump
- Windump
- Wireshark

Two peculiarities: the possibility of following a transmission, as well as the graphic user interface, have been fundamental for the choice of Wireshark program from among the others. As a basis for the study and evaluation of the proposal proposed in this project, Wireshark will be used as a network analyzer.

D. Wireshark

Wireshark, as discussed in previous section, is a packet analyzer. It is a basic tool to observe the messages exchanged between applications.

1) Main functions: It mainly consists of two elements, a packet capture library and an analyzer of these. Which, are explained below:

- Package capture library
- Packet analyzer

2) Distinctive features:

- Graphic Interface (GUI)
- Filtering capacity

E. Forensic analysis with Wireshark

Throughout this section the forensic analysis carried out in the dataset will be presented. To perform the forensic analysis, the fundamental starting point is to look at the protocol because depending on which one has been used, some things or others must be taken into account.

1) Protocols of the dataset: Here a list the protocols found within the dataset is presented, in the next section, a brief talk about the most representative ones and how to discover the possible threats will be discussed.

TCP	UDP	ICMP	ARP	SIP	OpenVPN
HTTP	DNS	FTP	SSH	DCE/RPC	COTP
TELNET	SSL	TLS	RSH	S7COMM	ESP
NTP	NBNS	NBSS	Portmap	LOOP	DNP3
Modbus/TCP					

The most representative protocols are: TCP, ICMP, ARP, UDP, DNP3 and FTP.

2) Threats & Attacks: Below are the most representative protocols as well as the attacks and threats that have been characteristic during the forensic analysis:

a) TCP:

- Ack scanning:it ports are attacked and their objective is to determine if the objective is protected by some type of filtering.
- Xmas scan: it is used to discover if a certain port is closed.
- Fin scan: it is used to discover if a certain port is closed.
- Fin-Ack scanning: it is used with the intention of saturating the network or closing certain TCP ports (denial of service).
- Syn scan / Stealth scan / Half open attack: it is used to know if a port is open or closed.
- Rst attack: it's a type of denial of service attack.
- Syn flood attack: it is used both to know if a port is open and to perform a denial of service.
- Tcp ping sweep:it is used to discover which IPs are available on the network.
- Null scan: it is used to find out if a certain port is closed.
- Psh-Ack flood: it is used to perform a denial of service.
- Tcp-Dup-Ack attack / Duplicate ack spoofing: it is used to perform a denial of service attack.

b) ICMP:

- Icmp scan / ping sweep: it is used to find out which IPs are available.
- Icmp route direction / redirect: it is used to perform man-in-the-middle attacks.
- IP protocol scan: it is used to discover protocols which protocols can be executed on the target IP.
- Smurf attack: it is a type of denial-of-service attack based on pings.
- Ping of death: it is used to perform denial of service attacks.
- Icmp tunneling: it is used to carry out covert communications.
- Icmp get timestamp: it it is used to know the latency throughout the network.

c) ARP:

- Arp poisoning / spoofing: it is used to perform phishing attacks, and then perpetrate the real attack.
- Arp sweep / scan: it is used to obtain the IPs of the hosts of a network.

d) UDP:

- Udp scan: it is used to discover if a certain port is available.
- Udp flood attack: it is used to perform a denial of service attack.

e) DNP3:

- Dnp3 warm reset: it is used to perform a partial system restart.
- Dnp3 cold start: it is used to perform a total system reset.
- Dnp3 time change: it is used to perform a denial of service attack.

f) FTP:

- Brute force attack: it is used to enter and get access to the server.
- Packet Capture / Sniffing: it is used to get users and passwords of network users.

III. Proposal

After presenting the possible threats detected in the dataset through the Wireshark packet capture program, the following is to show how the proposal has been shaped in image format. For this, throughout this chapter, the bases on which the iterative process for the conformation of the standard was begun, will be reviewed. After these, the proposed final solution will be presented along with the fields that can be displayed in it.

A. Background of the image format

In this section the background of the image format will be presented.

- Matrix of 128x128, small size that allows visual analysis.
- Grayscale UINT-8, from 0-255 2-dimensional matrices.
- Conversion dictionary, to convert each element to the 0-255 scale.

B. Final proposal

Next, the final solution of the proposed standard to display the images will be shown. First, the fields that have been included, the Wireshark originals and the extra ones added after an analysis of intrinsic information of these two protocols are observed. After these two, the final conformation of the image is presented.

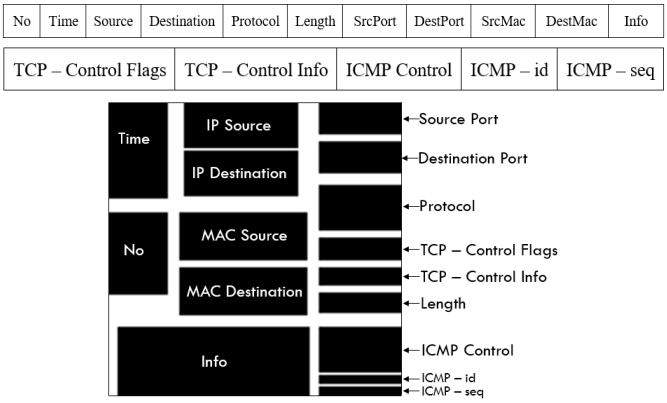


Fig. 1: Final Proposal

IV. Results and image analysis

Throughout this section the results obtained from the forensic analysis represented in the images will be shown. To do this, an analysis will be shown, field by field, of the standard of the images.

A. Image Analysis

For this purpose the images corresponding to a TCP transmission will be taken between two devices, in this case it corresponds to a scan attack of Syn scan ports.

Fig. 2: TCP Stream transmission

1) IP address: When evaluating a transmission, a request and a response is obtained, respectively.

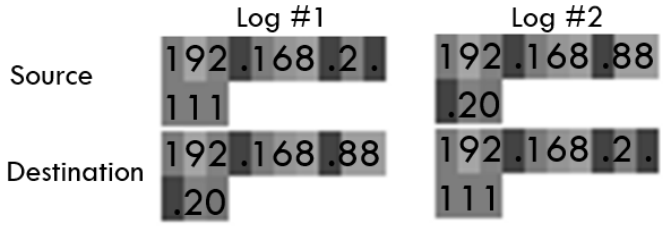


Fig. 3: IP address fields

2) Ports & Protocols: Again, the request and the response, at the ports and protocols level, is shown below:

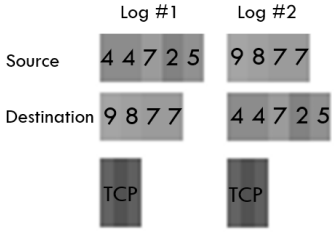


Fig. 4: Ports and protocol fields

3) MAC address: The MAC address acts as the physical identifier of the device. For practical purposes it is like the IP address but unique for each device.

Fig. 5: MAC address fields

4) Length & TCP - Control: The Length field helps to know the extension of the frame while the TCP Control field was added after analyzing implicit information. The Length field helps us to know the extension of the frame while the TCP Control field was added after analyzing implicit information.

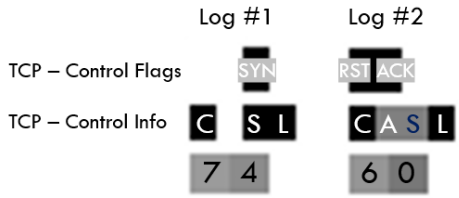


Fig. 6: Length and TCP - Control fields

The TCP - Control field is explained below:

- **TCP - Control Flags:** RST, ACK, SYN, FIN, PSH, URG.
- **TCP - Control Info:** This field is responsible for evaluating information contained in the plot, will allow more easily the identification of a possible attack. The parameters that are evaluated are:
  - **C** → Control: Printed in black if any of the other three is present.
  - **A** → ack: if ack = 0 then this will be black and if ack = 1 then it will be gray.
  - **S** → seq: if seq = 0 then this will be black and if seq = 1 then it will be gray.
  - **L** → len: if len = 0 then this will be black and if len = 1 then it will be gray.
  - **T** → TCP: if appears "TCP" through the frame this will be black, if not, white.
  - **D** → Dup: if appears "Dup" through the frame this will be black, if not, white.

5) ICMP - Control fields: In order to evaluate this field it is necessary to be in front of a registry whose communication was through the ICMP protocol. Therefore, those belonging to another image will be shown:

Fig. 7: ICMP protocol log

This field contains very useful information to detect and analyze the logs. The information that is represented is shown below:

- ICMP – Control: field that corresponds to certain ICMP error codes and types.
  - Echo (ping) request → ICMP Type 8
  - Echo (ping) reply → ICMP Type 0
  - Destination unreachable → ICMP Type 3
  - No response found → dentro del ICMP Type 8 when no answer is got.
- ICMP – id: is responsible for representing the value of the "id" corresponding to this protocol, LE/BE.
- ICMP – seq: is responsible for representing the value of "seq", LE/BE.

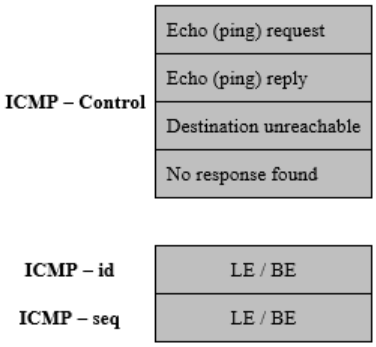


Fig. 8: ICMP fields distribution

V. Cconclusions

The conclusions can be divided mainly into two large groups, due to the duality of this project:

A. Forensic Analysis conclusions

- The data set comes from a convention that consisted in perpetuating attacks on a set of industrial devices. Therefore, this is not a "normal" situation and the data, as well as its results, could be biased to certain protocols.
- 93 % of the volume of the data belong to the TCP and ICMP protocols, hence they are the ones that have detected the most attacks, if there had been more attacks of the 23 remaining protocols, more of the latter would have occurred..
- The protocols have been studied independently to facilitate their analysis, as many are presented in several layers of the OSI model, and part of the global vision has been lost.

B. General conclusions about the image format

- The standard in image format allows quickly the distinction between protocols. As well as a successful visual inspection of different problems such as port scans, IP and MAC addresses.
- As extra fields have been included for the analysis of the ICMP and TCP protocols, substantial improvements have been obtained when identifying threats within them. They were added due to their high volume but if they had had other data, the specialized fields would have changed protocol.
- The UINT-8 format can present visual difficulties when distinguishing between shades of gray very close to each other, which can be a problem visually and perhaps also for the machine learning module since it has not been possible to test.
- The fields, being structured from their conception in a modular way, clearly allow the union between the two worlds already raised and also an easy implementation when redistributing and modifying any of them.
- The logs follow communication transmissions, this makes that a set of 2-3-5 registers are part of the same transmission. In order to analyze the threats and possible attacks it is necessary to study the complete transmission.

VI. Future work

This project leaves the door open to future developments. The main ones are presented below:

- **Make the color images (RGB)**: moving from the two gray scale dimensions to those corresponding to a color image. It should be studied if the advantages obtained in the visual analysis by having a range of colors now compensate for the possible complications to be able to analyze the images as RGB pixel arrays.
- **Represent the information from the raw binary data (in hexadecimal)**: use the information obtained in a raw way captured by the packet sniffers to conform the images so that it is easily interpreted by a visual analysis would be a great advance. Currently it is necessary to go through the protocol analyzer to interpret the information and be able to export it in order to conform the images.
- **Build a module of machine learning to analyze and classify the images**: now that the images are already structured and prepared for the visual and numerical analysis (by the pixels), one of the following points should be to build a module of machine learning that use techniques of image analysis or deep learning to be able to learn from transmissions and to classify and detect possible threats and attacks. It is important to build the module in such a way that it has "memory" in order to be able to analyze the transmissions of 2-3-5 images.
- **Expand the number of information fields already analyzed for more protocols**: as has been done with the TCP and ICMP protocols, more fields can be developed with implicit information.
- **Study the communication of a module of machine learning with a SIEM**: which would be the ultimate goal at logistic and operational level. Being able to check in real time the different threats with the machine learning module.

References

[1] V. Roblek, M. Meško, and A. Krapež, "A complex view of industry 4.0," *SAGE Open*, vol. 6, no. 2, pp. 1–11, 2016.

[2] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0007681315000373

[3] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Key Challenges in Cloud Computing: Enabling the Future Internet of Services," *IEEE Internet Computing*, vol. 17, no. 4, pp. 18–25, Jul. 2013.

[4] Y. Zhang, L. Wang, Y. Xiang, and C.-W. Ten, "Power System Reliability Evaluation With SCADA Cybersecurity Considerations," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1707–1721, Jul. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7042739/

[5] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri, "The Cybersecurity Landscape in Industrial Control Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, May 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7434576/

[6] CCN-CERT, "CCN-CERT_ia-16-17 Ciberamenzas and Tendencias," Resumen ejecutivo Edición 2017, 2017.

[7] N. Diakun-Thibault, "Defining Cybersecurity," *Technology Innovation Management Review*, vol. 2014, pp. 13–22, Oct. 2014.

[8] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey," *arXiv:1701.02145 [cs]*, Jan. 2017, arXiv: 1701.02145. [Online]. Available: http://arxiv.org/abs/1701.02145

[9] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7307098/

[10] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, Jan. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1084804512001178

[11] Y. Chang, W. Li, and Z. Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine." IEEE, Jul. 2017, pp. 635–638. [Online]. Available: http://ieeexplore.ieee.org/document/8005870/

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER INGENIERÍA INDUSTRIAL

ANÁLISIS FORENSE DE UN DATASET INDUSTRIAL Y PROPUESTA DE UN ESTÁNDAR GRÁFICO PARA LOS REGISTROS EN LA CIBERSEGURIDAD

MEMORIA

Autor: Jorge Buil García
Director: Álvaro López López

Madrid
Junio 2018

Agradecimientos

A la gente de Ricci, gracias por los ánimos y la ayuda

A mis amigos, por tragarse "mis imágenes" tanto tiempo

A las personas de mi día a día, que me han aguantado, soportado y aun así, han seguido a mi lado

A mi familia, padres, abuelos, tíos y primos. GRACIAS.

Índice de la memoria

Índice de figuras

Índice de tablas

Parte I

MEMORIA

Capítulo 1 INTRODUCCIÓN

En los últimos años se ha ido observando una nueva realidad en el mundo contemporáneo. La aparición de la llamada Industria 4.0 ha propiciado un cambio significativo en la concepción de la mayoría de los sistemas que se han venido utilizando en las pasadas décadas. Con la cantidad de dispositivos emergentes conocidos como *Internet of Things*, la expansión y estandarización del internet como medio básico de comunicación y servicios, así como el aumento en la capacidad de almacenamiento de gran cantidad de datos, con su consecuente análisis, ha guiado a este mundo a una serie de cambios significativos en la forma operativa de la industria y el resto de organismos.

Estas importantes implementaciones, que corren en favor de una mayor eficiencia en la producción de un mercado más global, no han querido profundizar de manera fehaciente en una parte fundamental, la ciberseguridad. En esta parte del trabajo se pretende realizar una evaluación de los llamados *Security Operation Center*, indagando en los tradicionales y en los cognitivos, hacia dónde se dirigen estos. Como elemento fundamental de estos se encuentran los *Security Information and Event Management*, que son los responsables de realizar las correlaciones de los *logs* para obtener niveles de alerta. Aquí se estudiará qué implican cada uno de estos elementos, así como qué retos pueden ser encontrados que permitan la conversión y adaptación de estos *logs* para su posterior implementación en un módulo de aprendizaje autónomo.

Para la realización de este proyecto se necesitará profundizar en la rama de *Computer Science*, especialmente en el sector de Análisis Forense de Datos, así como en técnicas de *machine learning* puesto que el objetivo último de éste será proponer la definición de un estándar gráfico para el análisis de registros de ciberseguridad.

Con este estándar se persigue construir un puente entre los dos mundos expuestos en el párrafo anterior. Que sirva para unir el sector de Análisis de los *logs* (tanto forense como normal) con el mundo del *machine learning* y *deep learning*.

1.1 ESTADO DEL ARTE

El fenómeno de la Industria 4.0 fue inicialmente mencionado en Alemania, 2011 [1], como una propuesta de desarrollo para el nuevo concepto de política económica alemana basado en estrategias de gran tecnología. Este incluye los llamados sistemas ciber-físicos, el *Internet of Things* (IoT) [2] y el *Internet of Services* (IoS) [3] entre otros, así como un cambio de mentalidad e infraestructuras de gran importancia en todos los sentidos.

Esto ha guiado hacia una revolución que afecta en todos los sentidos de la realidad contemporánea, el mundo, la sociedad y la industria. La Industria 4.0 incluye muchas nuevas características que han llevado a la aparición de nuevos paradigmas y caminos para vislumbrar la realidad. Significativamente, es importante contemplar qué cambios se han producido en el sector industrial, que es el campo que será estudiado en mayor detalle.

La Industria. Es importante hacer énfasis en que, en la mayoría de los casos, el intercambio de información se realiza entre distintas máquinas de forma autónoma. Están constantemente enviando datos de forma inalámbrica, utilizando sensores y a través de *Smart centers*, donde una gran cantidad de información está siendo analizada. Uno de los principales objetivos de la automatización hoy en día es la personalización y la adaptación al cliente, tanto para los productos como para los servicios. Si a todo esto se le añade la digitalización de los procesos, el uso de la electrónica y tecnologías de la información[4] se permite vislumbrar la nueva era que se ha instaurado en la sociedad de hoy en día.

A continuación, se realizará una descripción de la situación de la industria. En ella se hará inciso en qué consiste, por qué está formada y los principales cambios que se han sucedido en los últimos años, así como que implicaciones tienen y hacia dónde va. Éste es el por qué, a través de este trabajo, el llamado *Security Operation Center* (SOC), sección 1.1.2, se presentará en contraste con la estructura nueva emergente, *Cognitive SOC*. En la sección 1.1.3 se detallará de forma breve lo relacionado con el módulo de *machine learning* perteneciente a ésta nueva estructura y cómo ésta afectaría e interactuaría con el SOC. Para poder enmarcar todos estos elementos, es necesario un componente fundamental, el *Security information and event management (SIEM)*, el cual será explicado en la sección 1.1.4 y dónde se incidirá en el punto donde se centrará este proyecto: La Estandarización de los Logs, aquí se verán qué técnicas se utilizan y cómo se hace para gestionarlos. Para poder llevar a cabo esto, será necesario adentrarse en el campo de *Computer Science*, especialmente en el Análisis forense de datos, que se expondrá en la sección 1.1.5. Finalmente, la sección 1.1.6 se revisará, en términos generales, la estandarización de los *logs*, cómo gestionarlos y organizarlos.

1.1.1 SITUACIÓN DE LA INDUSTRIA

Los investigadores de la ciberseguridad, así como los que se encargan de diseñar las distintas infraestructuras para ésta, tienen como objetivo conservar la confidencialidad, la integridad y la disponibilidad de la información, así como la de aquellos dispositivos utilizados para gestionarla a través de distintos sistemas de ciber defensa. También buscan poder proteger aquellos ordenadores y redes de trabajo de *hackers* informáticos que quieran inmiscuirse en el sistema o robar cualquier tipo de datos[5].

Actualmente, en cada instalación industrial se encuentran sistemas de monitorización y supervisión que están al tanto de todo tipo de actividad sobre lo que está ocurriendo dentro de la planta y de llevar un control adecuado de ésta. Tanto SCADAs[6], DCSs y PLCs tienen que estar conectados a través de una red de trabajo para así poder recibir información de todos y cada uno de los dispositivos y sensores distribuidos por ésta, así como generar un control adecuado sobre cada uno de ellos[7]. Si no se disponen de las medidas de ciberseguridad necesarias, estos elementos pueden ser una puerta de acceso para cualquier agente de amenaza. De esta manera, información de gran importancia sobre todas las cosas que se están llevando a cabo podría acabar en manos de personas que no deberían tener acceso a ella. La realidad es que el número de dispositivos interconectados se está incrementando a un ritmo exponencial debido a los dispositivos IoT, almacenamiento y análisis de datos, y a las comunicaciones, los tres pilares de esta nueva era.

Con la ininterrumpida evolución de la tecnología, rápidas oportunidades están siendo creadas para establecer nuevas experiencias y prácticas en la vida cotidiana[8]. Por ello, la realidad en la que este proyecto se centra, el modo en el que todas estas innovaciones y estos cambios afectan al sector de la industria desde el punto de vista de la ciberseguridad. Es importante tener en cuenta que el excesivo aumento del número de sensores que están recogiendo información constantemente, y de los dispositivos intercomunicados a través de la red, traen consigo una serie de beneficios muy grandes; ahora es posible tener una gran relación con los clientes, un nuevo tipo de eficiencia corporativa. Ahora las cosas se hacen de formas que hace unos años hubieran sido impensables, lo cual, visto desde el punto de vista industrial es algo fantástico, pero lleva consigo una serie de contras de gran importancia, nuevos riesgos. Los agentes de amenaza se describen en el informe del CCN-CERT[9]. El daño que pueden causar si no se está bien protegido puede ser muy grave, no sólo por la pérdida de servicios o disponibilidad, sino también la filtración de datos de usuarios, datos privados que, debido la gran difusión que existe a través de internet, equivaldría una pérdida en la confianza y en la imagen que sería desastrosa para la compañía y el sector en sí mismo.

1.1.2 SOC

La ciberseguridad puede definirse como el cuerpo de las tecnologías, procesos y prácticas diseñadas para proteger las redes, ordenadores y los datos de ataques, daños y accesos no autorizados[10]. A pesar de la creciente importancia de la ciberseguridad, muchas organizaciones continúan enfocando el problema desde un punto de vista tecnológico. Pero la ciberseguridad es una cuestión mucho más amplia, pues debe estar integrada en diversas áreas de la organización para tener una protección integral de ésta contra potenciales amenazas y criminales que puedan atentar contra ella convirtiéndola en su objetivo. Todas las organizaciones comparten una visión, hay que saber tratar con las ciberamenazas, y para ello la gestión de riesgos o situaciones de emergencias es una práctica crítica y necesaria que ha de usarse para cumplir este objetivo[11].

Los conocidos como *Computer security incident response teams* (CSIRTs) son los responsables de recibir, revisar y responder a cualquier reporte y actividad relacionada con el monitoreo de la seguridad. La primera tarea de un CSIRT consiste en monitorear la seguridad de los eventos relacionados con las tareas de *Information Technology* (IT) de su organización. Realizando esta tarea se encuentra el SOC, que generalmente es una unidad centralizada en una organización de IT empresarial[12]. Los SIEM, que previamente se han introducido, son una herramienta de vital importancia dentro de los SOCs cuyo objetivo es recolectar, analizar y correlar, a tiempo real, información relevante para la seguridad de la información[13], (principalmente registros, logs). Tal y como se muestra en la Figura 1, los SOCs empresariales modernos están estructurados jerárquicamente alrededor de un sistema SIEM[12]:

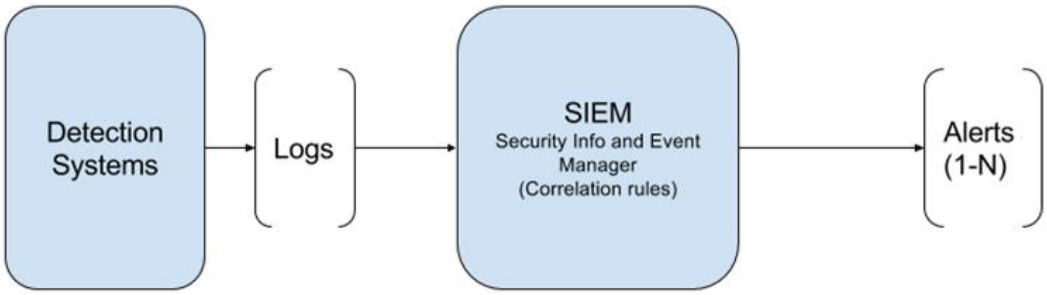


Figura 1 Esquema básico SOC Tradicional

Entre los principales objetivos de un SIEM, es importante mencionar los siguientes: encontrar y corregir errores en la configuración de la red de trabajo, revelar posibles actos de amenazas a la seguridad, determinar recursos críticos de la red y elegir una política segura y efectiva para manejar éstas[14].

En la actualidad es posible complementar el esquema que se muestra en la Figura 1 con un módulo de técnicas de inteligencia artificial (un módulo de *machine*

learning) que puede equipar al SOC con la capacidad de aprender. De esta forma, el sistema de seguridad podría llegar a predecir (o detectar) ataques desconocidos actualmente. Perfeccionar esta funcionalidad cognitiva se empieza a percibir como la tendencia principal en la investigación tanto a nivel corporativo como académico. La Figura 2 muestra el esquema funcional hacia el cual empiezan a tender los SOC:

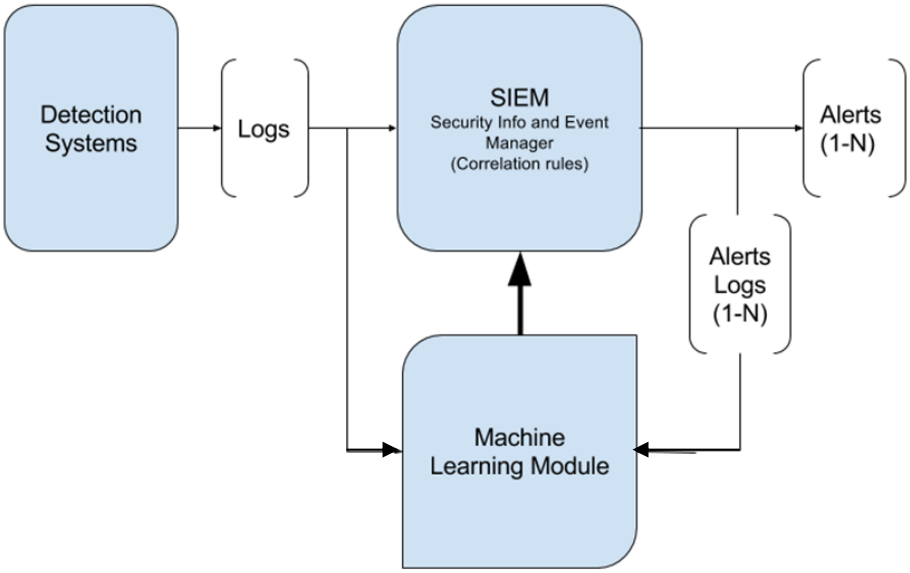


Figura 2 Esquema básico del SOC Cognitivo

Los diferentes sistemas de detección (como antivirus, *firewalls*, etc) generan *logs*[15] que van al SIEM y éste, de acuerdo con las reglas que le han definido, genera niveles de alerta (de 1 a N) en el caso de ser una posible amenaza. A diferencia del esquema tradicional, aparece un Módulo de Aprendizaje que tendría como entradas tanto los registros entrantes del SIEM como aquellos que han resultado en amenazas. Este módulo sería capaz de analizar los eventos y aprender de ellos, lo que abre un abanico de nuevas posibilidades porque, entre otras, ahora sería posible modificar las reglas del SIEM basándose en la experiencia de las operaciones del sistema.

Es importante tener en cuenta que, en la actualidad, las tecnologías y técnicas de seguridad que se han desarrollado para los sistemas IT comienzan también a aplicarse a los sistemas de *Operation Technology* (OT) de forma generalizada. Este proceso debe ser llevado a cabo de forma ordenada, tratando de evitar, en la medida de lo posible, volver a caer en los errores históricos cometidos a la hora de proteger sistemas de IT.

1.1.3 SIEM

Los SIEMs constituyen la plataforma central de los sistemas de seguridad modernos en los centros de operaciones. Se encargan de recopilar eventos de múltiples sensores (*Intrusion Detection Systems*, IDS, antivirus, *firewalls*, etc), relacionan dichos eventos y entregan nuevas vistas configuradas que permiten percibir niveles de alertas para proporcionar informes de seguridad y tratar con las amenazas[16].

Las siglas de SIEM surgen de la unión de los sistemas de *Security Information Manager* (SIM) y *Security Event Manager* (SEM)[17].

$$SIEM = SIM + SEM$$

Con esta unión lo que se pretende es cubrir los siguientes campos:

- Anomalías en la red de trabajo y detección de amenazas.
- Análisis antes, durante y después del ataque.
- Captura total de los paquetes de la red.
- Comportamiento de los usuarios, así como su contextualización.
- Cumplimiento de nuevas leyes y regulaciones.

Además, los servicios que se buscan son los que se presentan a continuación:

- Administración de registros.
- Cumplimientos de las regulaciones sobre los IT.
- Correlación de eventos
- Respuestas activas ante cualquier situación.
- Seguridad en los puntos críticos de la red y usuarios.

El sistema SIEM permite la identificación y clasificación de cada dispositivo de acuerdo con su tipo y rol dentro de la infraestructura de la red de trabajo. Una vez que estos se han categorizado dentro del sistema comienza la siguiente fase. Dependiendo de cada tipo, diferentes reglas se pueden configurar para realizar distintas acciones según de cual se trate, evaluando los factores de riesgo así como su criticidad. El valor de la información y sucesos acontecidos en el pasado del sistema pueden ayudar a determinar estos puntos fundamentales. A través de la correlación, que es un proceso que indaga en las relaciones entre los distintos eventos, e infiere en la significación de éstas, prioriza y proporciona un marco de referencia sobre el cual tomar acciones[18].

Los niveles en los que se estructura, a modo resumen, un SIEM pueden observarse en la Figura 3:

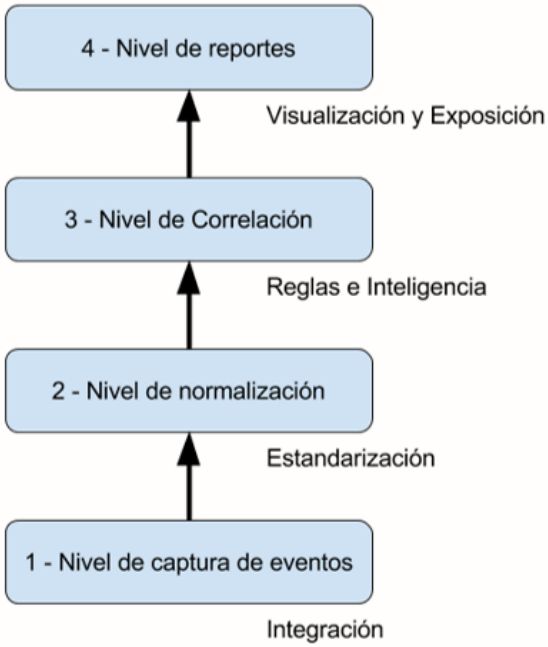


Figura 3 Estructura del SIEM

1.1.4 MÓDULO DE MACHINE LEARNING

Como se mostró en la Figura 2, la tendencia actual del *SOC* es hacia un sistema que, trabajando el paralelo, se encargue de analizar los *logs* que hayan entrado al *SIEM* frente a aquellos que hayan generado un nivel de alerta (un módulo de aprendizaje). El objetivo último, como meta final, sería conseguir que este módulo automáticamente se encargara de modificar las reglas del *SIEM*, mejorando así su rendimiento. Con respecto a la Figura 3, el módulo de aprendizaje actuaría entre los niveles 2 y 3.

A continuación, una serie de técnicas usadas en los llamados *intrusion detection systems (IDS)* serán mencionadas. Debido a la relevancia que tienen respecto a este problema, el foco de atención se centrará en aquellas que puedan usar técnicas de *machine learning*[19]–[22]:

- *Support Vector Machine (SVM)* [23].
- *Artificial Neural Network (ANN)* [24].
- *Association Rules and Fuzzy Logic (AR & FL)* [25].
- *Bayesian Network (BN)* [6].
- *Clustering (CL)* [26].
- *Decision Trees (DT)* [27].
- *Ensemble Learning (EL)* [28].
- *Meta-Heuristic (MH) / Evolutionary Computation* [29].
- *Inductive Learning (IL)* [30], [31].
- *Naïve Bayes (NB)* [32].

1.1.5 COMPUTER SCIENCE

Computer Science consiste en el estudio de la teoría, la experimentación y la ingeniería que forma la base para el diseño y uso de los ordenadores. Es el enfoque práctico y científico de estos, así como el estudio sistemático de su estructura y algoritmos[33].

Es importante indicar que hoy en día es vital el uso de las técnicas de *machine learning* en el campo de la informática[21], [22]. En el tema tratado a lo largo de este documento, la ciberseguridad (especialmente enfocado a los *logs* de las redes de trabajo), para ser capaz de utilizar estas técnicas es importante y necesario el tratamiento adecuado de los datos. A lo largo de los años se han ido evaluando diferentes enfoques, estos permitirían alumbrar la elección adecuada de la tecnología a utilizar. En el caso que concierne a este documento, los registros de ciberseguridad, *Computer Science* responde a tres preocupaciones principales: análisis de datos, análisis forense digital y visualización de la ciberseguridad. La primera abarcaría el marco de las técnicas a usar, la segunda cómo usarlas retrospectivamente para descubrir qué ha sucedido y, la última, cómo estar al tanto de aquellos ataques que se han sufrido y/o están ocurriendo.

1.1.5.1 Análisis de datos

Para hablar de análisis de datos en el campo de la ciberseguridad cuando se refiere a los *logs*, es importante explicar, por adelantado, un concepto: *network flow*. Este concepto se define como la secuencia de paquetes unidireccional o bidireccional entre dos puntos terminales[34]. Así es como el *flow monitoring* ha llegado a convertirse en el método predominante para el monitoreo de tráfico en redes de alta velocidad[35] que, hoy en día, son prácticamente todas las redes.

Por ello el análisis de datos es el punto final dentro de la configuración del f*low monitoring*, porque es donde los resultados de todos los pasos anteriores se reúnen. El análisis de paquetes con el objetivo de reforzar la seguridad en la red puede ayudar a los usuarios a estar prevenidos y a ser capaces de responder ante incidentes con mayor celeridad[36].

Es importante distinguir entre las tres áreas de aplicación en las que actúa el análisis de datos, las cuales se utilizan también para clasificar los softwares de análisis[35]:

- *Flow Analysis & Reporting:* los dispositivos exportadores de flujo suelen estar implementados en localizaciones estratégicas, aquellas donde un mayor flujo o tráfico puede ser observado entre dos *hosts*. Los datos resultantes proporcionan un informe completo de las interconexiones.
- *Threat Detection:* aquí pueden diferenciarse dos casos cuando los datos del

flujo son utilizados para la detección de amenazas. En primer lugar, pueden ser utilizados para analizar con qué servidor se ha comunicado cada usuario, incluyendo potentes informes sobre el número de paquetes y de bytes transmitidos. El segundo caso sería el uso de la corriente de flujo en sí para analizar ciertos tipos de amenazas. Éste serviría para modelar comportamientos de usuarios en la red.

- *Performance Monitoring:* esta área de aplicación tiene como objetivo observar el estado de los servicios que se llevan a cabo en la red. Las aplicaciones de análisis de datos, por lo general, reportan métricas. Además de esto, también llevan a cabo un post-procesado de los datos del flujo para un mejor rendimiento en el monitoreo. Como último, aparte, también pueden revelar distintos eventos en la red, así como evaluar el impacto final en la experiencia del usuario.

1.1.5.2 Análisis forense digital

El propósito del análisis forense digital es el de proporcionar las respuestas a las siguientes preguntas dentro de una investigación: quién, cómo, qué, por qué, cuándo y dónde[37]. Se tiene un principal objetivo: reconstruir eventos. Por otro lado, se encarga de utilizar métodos científicos para analizar e interpretar electrónicamente información almacenada. El software para el análisis forense digital permite analizar prácticamente cualquier tipo de información guardada en un ordenador o en una red, precisamente el campo en el que este documento se enfoca. En términos generales, se refiere a un conjunto de técnicas y métodos para recoger, analizar y preservar los datos digitales envueltos en un incidente y así poder extraer las evidencias que sean de utilidad[38]. La mayor fuente de evidencias son los *logs.*

El análisis forense de las redes (*network forensics*) permite explorar las evidencias digitales recogidas del tráfico de la red después de haber sucedido un evento sospechoso[39]. Se encarga del área del análisis forense digital que es responsable del tratamiento de las redes de trabajo, analizando dispositivos como *firewalls, switches, routers* y sistemas de gestión de la red. Los cibercriminales a menudo utilizan ordenadores conectados a internet para realizar ataques a las redes. Los dispositivos conectados a éstas, estando bien configurados, permitirán que la acción del ataque deje una serie de huellas digitales de todas las actividades que estén teniendo lugar en la red de la organización. Los paquetes capturados pueden revelar las firmas de estos ataques[36]. Después de rastrear las máquinas usadas en el ataque, los investigadores deben correlar los registros encontrados en los ordenadores utilizados para el ataque con aquellos que aún permanecen en la red comprometida[40].

1.1.5.3 Visualización de la ciberseguridad

La visualización de la información es una forma de mejorar el procesamiento cognitivo humano respecto a datos abstractos con la ayuda de representaciones

generadas por ordenadores[41]. Éste es un punto importante ya que está conectado con el propósito de este trabajo. Por lo tanto, es esencial el estudio del trasfondo que sigue al precepto de que los seres humanos tienen una capacidad especial para la percepción. Este hecho es único y se extiende más allá de un mero procesamiento algorítmico en el campo del descubrimiento y comprensión de patrones[42]. Por otro lado, la tecnología ha avanzado a un increíble ritmo en lo que a velocidad de computación se refiere. Cuando se combinan estas dos cualidades, se desarrolla un ciclo de resolución de problemas donde la visualización de los elementos en la ciberseguridad actúa como la interfaz entre esos dos componentes. El cerebro humano dedica más recursos al procesamiento visual que a cualquier otro sentido, lo que hace que esta habilidad sea una herramienta de gran poder. Esto permite, al ser humano, convertir datos en bruto en información y conocimiento[42]. El sistema informático puede generar enormes cantidades de datos de alerta y *logs* sin procesar que proporcionan detalles sobre el tráfico de la red y actividades maliciosas. Con esto, lo que se quiere transmitir es que una imagen que pueda comunicar los contenidos de un *log* podrá, por lo tanto, ser procesada por los usuarios de forma más rápida que lo que llevaría manualmente analizar esos registros de *logs.* Los datos se vuelven notablemente más fáciles de comprender cuando se presentan en un formato visual. Por esto se podría decir que la visualización de la ciberseguridad es efectiva apoyando al analista humano[43].

Diferentes tecnologías para procesar los *logs,* con el objetivo de la visualización de la ciberseguridad, serán presentadas a continuación:

- *Security Visualization Tools.*
- *Network Visualization Platform.*
- *Wireshark* [44].
- *Etherape.*
- *NetGrok.*
- *Time-based Network Visualizer*.

1.1.6 ESTANDARIZACIÓN DE LOS *LOGS*

Como se ha podido apreciar en la Figura 1 y en la Figura 2, hay un elemento que está siempre presente en los sistemas de seguridad, los *logs.* Cuando se habla de *logs* se hace referencia al registro de los eventos que suceden dentro de los sistemas de una organización a través de la red. Cada entrada de estos contiene información relacionada con un suceso ocurrido dentro del sistema o de la infraestructura de red[45].

El procesamiento de estos registros es de gran importancia ya que son los *inputs* principales del *SIEM,* tanto en el *SOC* tradicional como en el cognitivo. De acuerdo con la información recogida y detectada en estos *logs,* un conjunto de reglas de correlación tienen lugar, las cuales proporcionarán unos niveles de alerta relevantes. Con el caso del *SOC* cognitivo esto variaría un poco, ya que sería el *machine*

learning module el encargado de detectar las posibles amenazas y comunicárselas al SIEM para adoptar niveles de alerta y gestionar la seguridad.

Es importante tener en cuenta que todos estos registros, que son las principales entradas para el *SIEM*, llegan como información en bruto. Cada uno proviene de distintos dispositivos que están en la red o en supervisión, como servicios web, *firewalls*, antivirus, etc. Esto implica que cada uno genera y envía estos *logs* en base a sus propios estándares. Por desgracia, los *logs* para la ciberseguridad son conocidos por su elevado número a la hora de monitorear cualquier red, lo que provoca que su análisis sea una tarea complicada[46]. El *SIEM* es el encargado de elegir qué campos leer dependiendo de cómo su estructura esté programada para leer y correlar la información extraída de cara a la detección de amenazas. Esto lleva a una pérdida de información que en determinados casos puede ser significativa[47].

Las diferencias de formato entre los ficheros de *logs* para la variedad de servicios y aplicaciones en las que se usan suponen un problema para el análisis de la seguridad y de los desarrolladores de *IDSs*. Hoy en día cada aplicación tiene su propio formato de registros[47]. Por otro lado las fuentes de datos relevantes para la ciberseguridad constan de gran variedad de campos que requieren ser normalizados debido a las múltiples formas de expresar y definir estos atributos de interés[48].

1.1.6.1 Gestión de los registros

La principal fuente de datos del *SIEM* son los registros. A lo largo de este apartado se detallará el proceso utilizado para gestionarlos: organización, adquisición y aumento.

- Organización.

Los ficheros de los registros son generados automáticamente por los sistemas de operación o aplicaciones que se encargan de guardar todo lo que ocurre. Cuando sucede un evento de interés, un nuevo registro es añadido en la línea final del fichero del *log*. En términos generales se considera que cada registro consiste en una línea en el archivo del *log*[49].

- Adquisición.

Extraer los atributos o campos relevantes de cada uno de estos formatos de *logs* es un paso muy importante de cara al análisis de datos posterior. Una vez que se han interpretado aquellos de mayor relevancia desde los datos en bruto, es necesario un proceso de normalización[48]. Tras haber interpretado el archivo completo, etiquetado y tipificado, la fase de aumento tiene lugar de ser necesaria.

- Aumento

Existen dos tipos de aumento, el horizontal y el vertical[49]:

- **Aumento Horizontal,** que consiste en añadir campos extras a cada registro (nuevas columnas en el conjunto de datos) basados en la extensión de estos.
- **Aumento Vertical,** el cual se encarga de realizar cálculos estadísticos para los valores de cada campo, incluyendo aquellos añadidos en el aumento horizontal.

1.2 MOTIVACIÓN DEL PROYECTO

La Industria 4.0 incluye muchas nuevas características que han llevado a la aparición de nuevos paradigmas y caminos para vislumbrar la realidad que nos rodea. Actualmente, en cada instalación industrial se pueden encontrar sistemas de monitorización y supervisión que están al tanto de todo tipo de actividad. Tanto SCADAs, DCSs, PLCs y RTUs tienen que estar conectados a través de la red para poder recibir información de todos y cada uno de los dispositivos distribuidos. Esto es fundamental ya que, cada uno de estos, si no se han tomado las medidas de ciberseguridad apropiadas, pueden ser una puerta de acceso para cualquier tipo de ataque por parte de los *hackers*.

Para poder estar protegidos, actualmente se dispone de los los *SOCs* junto con los *SIEMs* (explicados en la Sección 1.1.2). Estos últimos se encargan de procesar los llamados *logs* (registros de los eventos de cualquier suceso que ocurra en la red) para correlarlos con un conjunto de reglas prestablecidas y así sacar niveles de alerta. Estos niveles, junto con la información de estos ya procesada, pasarían a un conjunto experto de analistas que se encargarían de verificar la información y, en función de las conclusiones, establecer medidas de seguridad.

Estos *logs* se obtienen, codificados en hexadecimal, con programas que se encargan de monitorear el tráfico de la red. Después, se pueden también interpretar y de esa manera que puedan ser analizados por expertos. El principal inconveniente es que la información recogida está presentada en un gran número de campos, así como la diversidad de los caracteres que los forman. Esto conlleva a que este entorno presenta una gran dificultad para cualquier persona que se encargue de revisarlos y de analizarlos. Pero ya no sólo para usuarios, sino también para cualquier tipo de técnica de m*achine learning*, si se quiere trabajar con los *logs* ya interpretados.

Por ello, partiendo de lo descrito en la literatura, con este Trabajo Fin de Máster se quiere construir un puente que una el mundo del análisis de datos (tanto forense o normal, dentro del campo de *computer science*) con el de las técnicas de *machine learning*. Para que en un futuro se pueda alcanzar la tendencia hacia la que se va, el *SOC* cognitivo. Por ello se realizará un análisis forense de los registros de un conjunto de datos de ciberseguridad industrial no tratados y se propondrá, finalmente, un estándar en formato imagen como puente entre estos dos mundos.

1.3 OBJETIVOS

Para poder obtener el objetivo principal de este Trabajo Fin de Máster, el enunciado en el último párrafo de la Sección 1.2 de la motivación, se han de cumplir los siguientes objetivos:

- **Objetivo 1:** Identificación de los posibles ataques ocurridos en el dataset analizado así como la forma en la que fueron perpetrados. De manera que se hayan obtenido los principales modos de ataque y los protocolos más vulnerables.
- **Objetivo 2:** Extracción de la información representativa de los registros industriales de ciberseguridad. Punto de control necesario, una vez que se tengan los conocimientos apropiados, para poder comenzar con el proceso de propuesta del estándar gráfico.
- **Objetivo 3:** Establecimiento de las bases sobre las que componer el estándar gráfico. Que vendrá como resultado de una profundización en los campos del análisis forense y en diferentes técnicas de *machine learning* y *deep learning*, para que estén acordes con la motivación del proyecto.
- **Objetivo 4:** Proposición del estándar gráfico que sirva como unión entre el análisis de registros (normal y forense) y el mundo del *machine learning* y *deep learning.* El cual vendrá dado como resultado final de todos los objetivos previamente citados.
- **Objetivo 5:** Análisis del estándar gráfico propuesto, y muestra de resultados, a nivel visual y sus posibilidades futuras.

1.4 METODOLOGÍA

Para facilitar la consecución de los objetivos enunciados, el trabajo se desglosará en las siguientes tareas:

- **Tarea 1:** Profundización de los protocolos de Internet, así como sus características dentro de la información grabada en los registros y sus particularidades que permitan la identificación y realización de ataques.

- **Tarea 2:** Estudio de técnicas de análisis de datos, así como análisis forense de redes y monitoreo del tráfico para la identificación de ciberataques o situaciones anómalas.
- **Tarea 3:** Documentación y aprendizaje sobre los analizadores de protocolos, de los analizadores de red y *packet sniffers*, para la captura, análisis y gestión de los eventos y *logs* a utilizar.
- **Tarea 4:** Familiarización y uso con el programa Wireshark para la lectura y procesamiento de registros. Aprendizaje del lenguaje utilizado para filtrar y representar información selectiva, que será necesaria para el análisis forense de datos.
- **Tarea 5:** Profundización en la investigación de técnicas de *machine learning* aplicadas al formato imagen, así como técnicas de d*eep learning* que permitan la identificación y clasificación de éstas.
- **Tarea 6:** Importación de los registros exportados desde Wireshark a Matlab, así como la elaboración del código en el entorno de desarrollo de este último para su respectivo tratamiento del dataset.
- **Tarea 7:** Estudio de diferentes técnicas de codificación de caracteres para la obtención de píxeles y elaboración del diccionario de conversión a escala UINT-8.
- **Tarea 8:** Elaboración del código de Matlab que permita la automatización del proceso de separación de los elementos del *log*, su respectiva codificación a escala de grises, la disposición en matrices de pixeles y su estructuración dentro de las imágenes.
- **Tarea 9:** Realización de un proceso iterativo para ir mejorando de forma progresiva la propuesta de estándar gráfico a medida que se vaya profundizando en el resto de campos hasta presentar la final.
- **Tarea 10:** Redacción de la memoria de este Trabajo Fin de Máster

Las tareas se llevarán a cabo siguiendo la planificación que aparece en la Tabla 1.

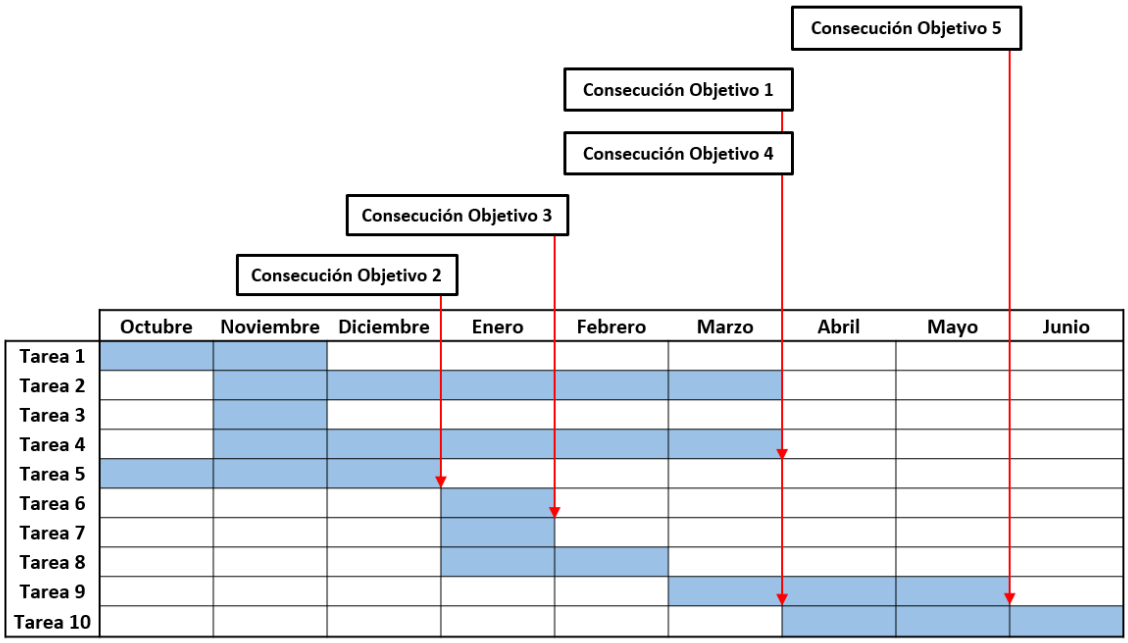


Tabla 1: Cronograma del proyecto

1.5 RECURSOS Y HERRAMIENTAS EMPLEADAS

Los materiales que se van a utilizar durante la realización del proyecto serán principalmente:

- El software de MATLAB-SIMULINK® para realizar el código necesario para la importación de los *logs*, la codificación a escala de grises a través del diccionario de conversión y la formación de las imágenes, así como su estructuración.
- El software de Wireshark como analizador de paquetes, para abrir los archivos .pcap y realizar el análisis forense de estos.
- El repositorio de la IEEE, Elservier y ResearchGate para conseguir información referente al análisis forense orientado a la ciberseguridad, a la utilización y gestión de los registros (*logs*).
- El dataset perteneciente a 4SICS-GeekLounge con unos cinco millones de registros de ciberseguridad sin analizar.
- Ordenador personal.

Capítulo 2 ANÁLISIS FORENSE DE UN CONJUNTO DE REGISTROS INDUSTRIALES

2.1 INTRODUCCIÓN

Con el objeto de poder construir un puente entre el mundo del análisis de registros (orientados a la ciberseguridad) y el mundo del *machine learning*, en primer lugar se comenzará realizando un análisis forense de un conjunto de datos industriales.

El conjunto de datos del que se dispone pertenece al utilizado en la conferencia sueca sobre ciberseguridad industrial (4SICS GeekLounge)[50]. La dimensión de este dataset es de 350MB de tráfico capturado en la red. Entre los dispositivos industriales se encuentran SCADAs y otros dispositivos ICS (*industrial control systems*).

Este capítulo es de vital importancia ya que corresponde a un trabajo que asentará las bases para la propuesta final en formato imagen. Un análisis de los *logs* disponibles (un análisis forense), permitirá discernir cómo reorganizar la información de la que se dispone para que, de forma efectiva y visual, pueda ser estructurada en la propuesta.

Para ello se comenzará profundizando en quiénes son los agentes causantes de la amenaza que actualmente realizan ataques. Se continuará explicando de forma breve el dataset utilizado, para contextualizar el tipo de datos del que se dispone, así como el número de registros. Siguiendo a esto, se hablará de los analizadores de red y sus funciones, ya que son herramientas fundamentales para llevar a cabo la monitorización de la red y su pertinente estudio y análisis (tanto normal como forense). Aquí también se comentarán algunos de los softwares disponibles y cuál se ha elegido para llevar a cabo el análisis de éste. Una vez que estos puntos queden claros se procederá al análisis forense de estos registros, explicando los protocolos encontrados y dando a conocer cuáles son las posibles amenazas (por el contexto en el que se sitúa). También cómo han sido detectadas o cómo podrían haberlo sido de haberse dado ese tipo de ataque.

2.2 AGENTES DE LA AMENAZA

A lo largo de este apartado se van a comentar los principales agentes de la amenaza recogidos en el informe CCN-CERT IA-09/18 "Ciberamenazas y Tendencias"[51]. El objetivo es contextualizar la situación actual, es importante conocer de dónde vienen los ataques, quién está detrás. Es importante tener en cuenta que determinar con precisión la persona o entidad que ha llevado a cabo el ataque es una tarea muy compleja, ya que muchas veces se dejan pistas falsas u ocultan las huellas dejadas. A continuación, se muestran los principales actores de la amenaza[9], [51]:

2.2.1 LOS ESTADOS COMO AGENTES DE LAS AMENAZAS

2.2.1.1 Crecimiento y especialización del ciberespionaje

El número de países que ha adquirido la capacidad de recopilar inteligencia a través de internet ha crecido enormemente. El ciberespionaje es un método que conlleva menores riesgos que el espionaje tradicional dada la dificultad de atribución de la autoría., además de ser relativamente más económico y rápido. Esta amenaza, utilizada principalmente por Servicios de Inteligencia, está dirigida tanto al sector público como al privado y suele provenir de países que desean posicionarse de manera más favorable, desde puntos de vista estratégicos, económicos o políticos.

2.2.1.2 La inversión creciente de los estados en capacidades ofensivas

Los Servicios de Inteligencia occidentales han identificado que muchos países están invirtiendo en la creación de capacidades digitales ofensivas. El objetivo parece ser influir en las operaciones de información. Se recopilan información confidencial que luego se publica por un tercero (aparentemente independiente), con el objetivo de sembrar confusión y división. Además, también se ha profesionalizado la ocultación por parte de los atacantes.

2.2.1.3 A la búsqueda de nuevos métodos

Los actores estatales se encuentran constantemente en búsqueda de nuevos métodos que les permitan infiltrarse en las redes sin ser detectados (muchas veces en combinación con los métodos tradicionales), al igual que ocurre con las organizaciones independientes. Cualquier infraestructura TI desarrollada sigue siendo un elemento atractivo como puerto de tránsito para los ciberataques. Se han detectados vectores de ataques que explotan infraestructuras tecnológicas

extranjeras para atacar a terceros países, convirtiéndoles en cómplices involuntarios.

2.2.2 CIBERDELINCUENTES

Las amenazas que las organizaciones profesionalizadas en este campo representan para la seguridad continúa creciendo a un ritmo acelerado. Los ciberdelincuentes siguen explorando incesantemente "modelos de negocio" más lucrativos.

2.2.2.1 La diversificación delincuencial al socaire del ransomware

El desarrollo de nuevos métodos de ataque se está evidenciando, entre otros, por el uso e impacto de *ransomwares*. Además de los ataques no dirigidos, sin ninguna víctima en concreto, los ciberdelincuentes están empleando el *ransomware* para atacar a organizaciones concretas en las que el impacto de este tipo de ataque sería mayor. De manera que estarán más inclinadas a satisfacer un rescate de mayor cuantía.

Diferentes equipos de investigación han demostrado que el *ransomware* también están empezando a usarse contra los ICS y todos aquellos elementos relacionados con el IoT. Por ello, es de esperar que los ciberdelincuentes apunten también a estas áreas en el futuro próximo.

2.2.2.2 Mayor frecuencia y sofisticación en los ataques a las entidades financieras

Ha aumentado el número de ataques dirigido contra sistemas de empresas, bancos y otras instituciones financieras, en lugar de dirigirse sólo a los consumidores. Analizando cómo explotar al máximo el acceso a la red y convertir tal acceso en dinero. El incremento de la sofisticación de tales ataques también ha sido más que evidente. La conclusión de todo esto parece apuntar a que los ataques dirigidos consumen más tiempo y recurso a los delincuentes, pero a su vez la ganancia económica es mucho mayor que los simples e innominados ataques a los consumidores.

2.2.3 TERRORISMO Y CIBERYIHADISMO

Los grupos yihadistas, junto con el uso de internet por parte de grupos terroristas, constituyen en la actualidad la principal amenaza de este tipo.

2.2.3.1 Intención de montar ataques cibernéticos

Aunque los yihadistas todavía no parecen ser capaces de desarrollar ataques sofisticados, parecen estar decididos a desarrollar esta vía de agresión. Aunque, hasta la fecha, los resultados más evidentes han sido las desfiguraciones y los ataques DDoS, todos ellos de naturaleza propagandística. Estas características también se encuentran en las listas que se han publicado a través de internet.

Además de lo anterior, los grupos yihadistas también pretenden desarrollar ciberataques con el fin del desenvolvimiento diario de los ciudadanos mediante tácticas violentas o de perturbación social. Por otro lado, la presencia de productos y servicios para desarrollar este tipo de ataques, fácilmente accesibles en foros concretos, incrementa la preocupación.

Finalmente, añadir que los ataques llevados a cabo hasta ahora por estos grupos mostraban una importante aleatoriedad.

2.2.4 LOS HACKTIVISTAS

Los grupos hacktivistas llevan a cabo sus ciberataques por razones ideológicas. Esto pude observar, entre otros, en marzo de 2017 con la creciente tensión diplomática en Turquía, realizaron ataques a pequeña escala aduciendo activismo político o motivaciones nacionalistas. Otro ejemplo podría ser julio de 2016, en Vietnam, las pantallas de información en varios aeropuertos mostraron lemas anti-vietnamitas y anti-Filipinas en relación con el conflicto en el Mar del Sur de China.

Aun cuando el acometimiento de ciertos ciberataques exige un importante conocimiento, la amenaza que supone el hacktivismo podría crecer a la vista de la mayor disponibilidad de productos, cada vez más creciente. Servicios y herramientas para desarrollar ataques con significativo impacto social cada vez son más accesibles.

2.2.5 CIBERVÁNDALOS Y SCRIPT KIDDIES

El desafío para demostrar sus propias capacidades, así como la broma, suelen ser los principales motivos que anima a estos individuos o grupos para llevar a cabo sus acciones.

Actualmente existe una amplia disponibilidad de herramientas, de forma pública, para la comisión de ataques. Un ejemplo de esta disponibilidad, de gran impacto, fue el ataque DDoS a gran escala que utilizaba la *botnet Mirai*, públicamente disponible.

2.2.6 ACTORES INTERNOS

Algo de elevada frecuencia es que, cuando se trata de acciones intencionadas, la motivación suele ser de naturaleza personal: razones económicas, políticas o estrictamente personales. También es cierto, sin lugar a dudas, que existen amenazas provenientes de actores internos que provienen de comportamientos inconscientes, descuidos o negligencias.

2.2.7 ORGANIZACIONES PRIVADAS

Los motivos y características de los ataques desarrollados por organizaciones privadas son diferentes, se encuentran los que se dirigen a la disponibilidad o a la confidencialidad de los sistemas de sus competidores, los que se realizan para obtener beneficios económicos, los que dirigen explotar comercialmente información obtenida de forma ilícita, etc.

Aunque los usuarios deberían conceder los permisos pertinentes para permitir el uso de su información privada, han abundado las noticias de prensa en relación con empresas que han comercializado con estos. Sin que quedara claro si el cliente había dado su consentimiento o no.

2.3 4SICS GEEKLOUNGE DATASET

El dataset utilizado corresponde al de la conferencia sueca de 4SICS[50] sobre ciberseguridad industrial. Consta de 3 archivos .pcap (formato en el que se exportan los registros del tráfico monitoreado de una red). Estos son:

- 4SICS-GeekLounge-151020.pcap
  - 25MB
  - 246.137 paquetes
- 4SICS-GeekLounge-151021.pcap
  - 136MB
  - 1.253.100 paquetes
- 4SICS-GeekLounge-151022.pcap
  - 204MB
  - 2.274.747 paquetes

Cada uno de estos archivos corresponde a uno de los tres días que duró la conferencia. A lo largo de todos los paquetes disponibles, se pueden encontrar los siguientes protocolos utilizados:

- ARP
- ICMP
- HTTP
- LOOP
- OpenVPN
- Modbus/TCP
- NBNS
- RSH
- SIP
- TELNET
- DCERPC
- COTP
- UDP
- DNS
- TCP
- FTP
- NTP
- S7COMM
- Portmap
- NBSS
- SSH
- SSL
- TLS
- DNP3
- ESP

Por último destacar que el "*Geek Lounge"* de 4SICS contiene un laboratorio *ICS* con PLCs, RTUs (unidades terminales remotas), servidores y equipos de redes industriales como *switches, firewalls, etc*. Los datos capturados están disponibles gracias a Netresec, que trabajaron con los organizadores para capturar el tráfico de la red durante el *ICS lab.*

2.4 ANALIZADORES DE RED

Tal y como se ha presentado en la introducción, los analizadores de red[52] son un componente clave para poder llevar a cabo el análisis de los *logs* y, con ellos, poder presentar la propuesta final del estándar gráfico. Estos proporcionarán las herramientas necesarias para establecer qué es lo verdaderamente relevante de cada registro, compondrán las bases de este trabajo. Por otro lado, será gracias al campo de *network forensics* [35], [37], [38] (análisis forense de redes) que se puedan determinar qué tipos de amenazas se han sufrido en la red y cómo detectarlas.

Los analizadores de redes también son conocidos como analizadores de paquetes para los *logs*¸ los registros. Estos pueden ser definidos como aquellos programas que se encargan de "escuchar" todo el tráfico que se transmite dentro de la red de trabajo. A través de ésta la información viaja de forma bruta en binario, y aquí es donde reside la verdadera importancia de estas herramientas: pueden convertir esa información binaria en bruto a un formato legible para las personas. Esto facilita, en gran medida, el análisis de las redes de trabajo. Por ello es importante tener en cuenta que el uso de estas analizadores es administrar, resolver problemas y mantener la seguridad por parte de los administradores. El proceso de *network sniffing* (que es como se le llama a tener uno de estos programas monitoreando a tiempo real una red), se muestra a continuación[52]:

- **Recolección:** ésta es la primera tarea de los analizadores de paquetes. Son los responsables de escuchar todo el tráfico que confluye en la red y capturar así los datos binarios en bruto que se transmiten. Esto es de vital importancia ya que pueden ser exportados para salvar dicha información, de manera que más tarde pueda ser analizada y/o procesada mediante otros programas. La extensión más común para estos archivos exportados es .pcap.
- **Conversión:** este proceso lo lleva a cabo el componente decodificador de los rastreadores de paquetes. En esta segunda tarea los datos binarios brutos que han sido capturados se decodifican, convirtiéndolos en formas legibles para el ser humano (son interpretados).
- **Análisis:** éste es el último paso del proceso. En esta tarea se realiza el análisis de los protocolos. Una vez que se tiene la información de los *logs* ya interpretada, realizada en el segundo paso, los protocolos utilizados durante el tráfico de la red se pueden visualizar. Con los campos ya interpretados, conociendo los protocolos utilizados, los paquetes ya pueden ser analizados y explicados.

Una vez conocidas sus tres principales funciones, es importante saber que actualmente existe una gran variedad de programas analizadores de redes. A continuación, se hará hincapié en tres analizadores *open source* (código abierto) de gran repercusión en este mundo:

- **Tcpdump:** reconocido como la herramienta fundamental de entre toda la colección de analizadores de paquetes. Se trata de una aplicación de código abierto que vienen instalada por defecto en casi todos los sistemas operativos de tipo Unix. Tcpdump es un programa excelente de recopilación y tiene un completo y complejo lenguaje de filtrado. Es importante saber cómo filtrar los datos a la hora de recopilarlos con el fin de poder manejar adecuadamente los datos a analizar.

- **Windump:** la mayoría de las herramientas de código abierto se acaban clonando eventualmente a otros sistemas operativos. Cuando se da este caso, se dice que se ha "portado" a otro sistema. Windump es un port de Tcpdump y, por tanto, se comporta de una forma muy similar. La principal diferencia entre los dos es que éste último necesita tener instalada la librería Winpcap para poder ser ejecutado.
- **Wireshark:** éste es un famoso analizador de paquetes de la red. Se trata de un programa de código abierto que se utiliza para el análisis de protocolos, la supervisión y la resolución de problemas en la red. Aparte de todo esto, Wireshark se encuentra presenta en prácticamente cualquier sistema operativo existente. Se puede ejecutar en cualquier lado, incluido de forma portable. Una de las herramientas más útiles que proporciona este programa es la posibilidad de seguir una transmisión. También tiene una interfaz de usuario (*GUI*) que es fácil de usar y facilita en gran medida el análisis de los *logs.*

Estas dos peculiaridades: la posibilidad de seguir una transmisión, así como la interfaz gráfica de usuario, han resultado fundamentales para la elección de este programa de entre los otros. Como base para el estudio y valoración de la propuesta que se plantea en este proyecto se utilizará Wireshark como analizador de redes.

2.5 WIRESHARK

En esta sección del capítulo se va a presentar de forma breve y concisa la información más relevante respecto al programa Wireshark. De manera que se tenga una pequeña base sobre el tipo de programa que es, cuáles son sus principales funciones y qué características son las lo han convertido en el programa utilizado en la realización de este proyecto.

2.5.1 ¿QUÉ ES?

Wireshark[35], [52]–[54], como se ha comentado en la Sección 2.4, se trata de un analizador de paquetes. Es una herramienta básica para observar los mensajes intercambiados entre aplicaciones. A este tipo de programas se les puede categorizar como elementos pasivos, únicamente observan mensajes que circulan por una red determinada pero nunca envían, ellos mismos, ningún mensaje. Es más, un analizador de protocolos (como también se les llama a estos programas), recibe una copia de los mensajes que están siendo recibidos o enviados en la terminal donde esté ejecutándose.

2.5.2 PRINCIPALES FUNCIONES

Consta principalmente de dos elementos, una librería de captura de paquetes y un analizador de estos mismos. Los cuales, se explican a continuación:

- **Librería de captura de paquetes:** es una librería que permite al programa capturar diversas tramas de la red para recolectarlas, y después ya analizarlas o exportarlas. Según la librería que se utilice se permiten capturar e interpretar diferentes extensiones de este tipo de paquetes capturados. La más conocida para Wireshark es *winpcap* que se encarga de capturar los paquetes en formato .pcap.
- **Analizador de paquetes:** esta funcionalidad permite que el programa pueda reconocer que la trama capturada pertenezca a un protocolo concreto (TCP, ICMP…) y mostrar al usuario la información decodificada, ya interpretada. Al mostrar los campos correspondientes a cada uno de los paquetes analizados se consigue, de esta forma, que el usuario sea capaz de ver todo aquello que en un momento concreto está circulando por la red. Esto es de gran importancia porque permite comprobar lo que realmente hacen los programas.

2.5.3 CARACTERÍSTICAS DISTINTIVAS

Aunque esto es algo que ya se ha comentado en la sección anterior, es importante recalcarlas ya que han sido dos funcionalidades principales a la hora de llevar a cabo el análisis forense del dataset así como la conformación del estándar en formato visual. Estas dos principales características, que distinguen al programa Wireshark del resto de analizadores de protocolos, son: Su interfaz gráfica y su potente capacidad de filtrado que presenta.

2.5.3.1 Interfaz Gráfica

De sus características fundamentales, gracias a tener una *GUI* es posible visualizar los registros de los *logs* de forma que sean fácilmente legibles por un usuario especializado. De esta manera se evita, si se quiere, tener que usar la consola de comandos para poder realizar las acciones propias de un analizador de protocolos. A continuación, se muestra una imagen de la interfaz gráfica mostrando un conjunto de *logs* del dataset en la Figura 4:

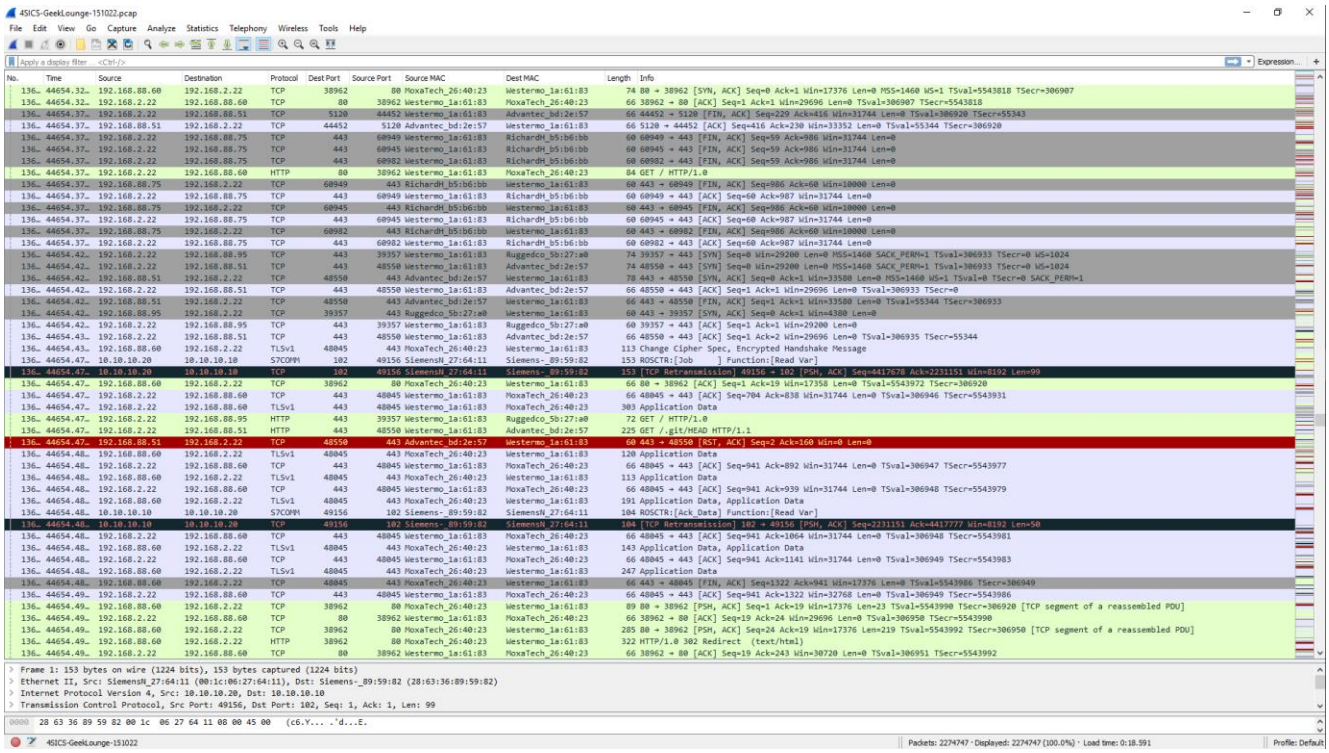


Figura 4 Interfaz gráfica de Wireshark

Cada una de las líneas que aparecen ahí representadas, corresponden realmente a un registro del *log*. Los que han sido capturados por la librería de captura de paquetes (por ejemplo en formato .pcap). Tal y como se ha comentado, al ser un analizador de paquetes, permite al usuario visualizar cada uno de los campos ya sea en la pantalla principal (en las cabeceras) o abajo dónde se le permite diseccionar cada parte correspondiente a la trama de estos. Esto provoca que sea un programa muy atractivo para el usuario a la hora de trabajar con los paquetes de *logs* capturados.

2.5.3.2 Capacidad de filtrado

La otra característica que le diferencia del resto de sus programas similares. Principalmente se puede observar en un doble entorno, por un lado las reglas de coloración (para representar los registros, si se cumplen determinadas reglas, de distintos colores como se puede apreciar en la Figura 4) y barra para aplicar filtros de búsqueda.

- **Reglas de coloración:** te permiten definir en qué color y cómo Wireshark va a representar los protocolos y los eventos de los datos capturados. Trabajar con estas reglas de coloración permite facilitar en gran medida la labor de análisis del dataset. No sólo se pueden visualizar los protocolos

en diferentes colores, sino que también se pueden configurar distintos colores para posibles eventos.

- **Filtros de búsqueda o de visualización:** consisten en establecer un criterio de filtro sobre los paquetes capturados y que se están visualizando en la pantalla principal de Wireshark. Constituyen un medio muy potente por sus posibilidades ya que con estos se podrá analizar exclusivamente el contenido que se desea. Los parámetros son propios del programa, pero la guía de usuario facilita la sintaxis en gran medida, también se pueden combinar mediante operaciones lógicos.

2.6 ANÁLISIS FORENSE CON WIRESHARK

A lo largo de esta sección se presentará el análisis forense realizado en el dataset con el programa Wireshark, ya justificado, en la Sección 2.4, el motivo del uso de éste para el análisis. Para el análisis forense, el punto fundamental de inicio es fijarse en el protocolo. Según el protocolo utilizado para el *log* habrá que tener en cuenta una serie de cosas u otras. Como Wireshark ya ha interpretado la información binaria en bruto contenida dentro de los ficheros .pcap esto es posible de forma rápida. Por ello, a continuación se explicarán de forma breve las bases de cada uno de los protocolos encontrados en el conjunto de datos de los que se dispone, los nombrados en la Sección 2.3.

2.6.1 PROTOCOLOS DEL DATASET

En primer lugar, antes de entrar a describir en brevedad cada uno de los protocolos encontrados, es importante dejar claro en qué consiste un protocolo de comunicaciones. En informática y telecomunicaciones, se trata de un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitirse información por medio de cualquier tipo de variación de una magnitud física. Los más representativos, principalmente a nivel industrial, son los siguientes: TCP, ICMP, ARP, UDP, DNP3, FTP.

2.6.1.1 TCP

Transmission Control Protocol[55], [56], es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. Se utiliza cuando se busca que la comunicación a través de la red sea segura. Este protocolo se asegura de que los datos que emite el cliente sean recibidos por el servidor sin errores y en el mismo

orden que fueron emitidos. Es un protocolo orientado a la conexión, puesto que el cliente y el servidor deben anunciarse y aceptar la conexión antes de comenzar a transmitir los datos a ese usuario que debe recibirlos. TCP da soporte a muchas aplicaciones de internet como FTP y de aplicación como HTTP, SSH y FTP.

2.6.1.2 UDP

User Datagram Protocol es el protocolo de datagramas de usuario es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de estos a través de la red sin que se haya establecido previamente una conexión ya que éste incorpora la información suficiente de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo por lo que los paquetes pueden adelantarse unos a otros. Tampoco se tiene confirmación de llegada correcta. Su principal uso es para protocolos como DHCP, DNS y demás protocolos en los que el intercambio de paquetes de la conexión y desconexión son mayores. También se utiliza para la transmisión de audio y vídeo real.

2.6.1.3 ICMP

Internet control message protocol[57] es el sub-protocolo de control y notificación de errores del protocolo de internet (IP). ICMP difiere del propósito de los otros dos ya que no se suele utilizar para las aplicaciones de usuario. La excepción reside en la herramienta de *ping ping* y *traceroute*, que envían mensajes de petición *Echo* Ping y recibe mensajes de respueta *Echo,* para determinar si un host está disponible entre otras cosas. Los mensajes de este protocolo se utilizan con fines de diagnóstico o control y se generan en respuesta a los errores en operaciones IP. Estos errores se dirigen a la dirección IP de origen del paquete originario.

2.6.1.4 ARP

Address Resolution Protocol[58] es un protocolo de comunicaciones de la capa de enlace. Es responsable de encontrar la dirección de hardware (dirección MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete a la dirección de difusión de la red que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina responda (ARP *reply*) con la dirección *Ethernet* que le corresponde. En *Ethernet*, la capa de enlace trabaja con direcciones físicas. El protocolo ARP se encarga de traducir las direcciones IP a direcciones MAC. Para realizar esa conversión, el nivel de enlace utiliza las tablas ARP, cada interfaz tiene tanto una dirección IP como una dirección física MAC.

2.6.1.5 HTTP

Hypertext Transfer Protocol es el protocolo de comunicación que permite las transferencias de información en la *World Wide Web* (www). Se trata de un protocolo sin estado, no guarda ninguna información de conexiones anteriores, no tiene memoria. Para eso se utilizan las llamadas *cookies*, la información que un servidor puede almacenar en el sistema cliente. HTTP conforma un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente realiza una petición enviando un mensaje con cierto formato al servidor, y éste último se encarga de enviar un mensaje de respuesta.

2.6.1.6 DNS

Domain Name System[59] es un sistema de bases de datos distribuidas que se utiliza para gestionar los nombres de los sistemas principales y sus direcciones IP asociadas. Es un protocolo dentro de la capa de aplicación que se relaciona directamente con la capa de transporte (TCP, UDP) y la de red. El uso de DNS implica que los usuarios pueden utilizar nombres sencillos para localizar un sistema principal, en lugar de emplear la dirección IP. Los servidores DNS pueden colaborar entre sí para correlacionar todos los nombres de dominio con sus direcciones IP, estos que colaboran entre sí son los que permiten que los sistemas se comuniquen a través de Internet.

2.6.1.7 FTP

File Transfer Protocol[58] es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, que está basado en la arquitectura cliente-servidor. Un cliente se puede conectar a un servidor para descargar archivos desde él o para enviárselos, independientemente del sistema operativo en cada equipo. El protocolo FTP es ofrecido por la capa de aplicación. El principal problema de éste es que no está pensado para la máxima seguridad ya que todo el intercambio de información se realiza en texto plano sin ningún tipo de cifrado. Para solucionar este problema son de gran utilidad protocolos como SSH que permiten transferir archivos per cifrando todo el tráfico.

2.6.1.8 SSH

Secure SHell es el nombre de un protocolo y del programa que lo interpreta. Sirve para acceder a servidores privados a través de una puerta de trasera. Permite manejar por completo el servidor mediante un intérprete de comandos. Se le asignó el puerto TCP 22 y se encuentra en la capa de aplicación. Además de la conexión a otros dispositivos, SSH permite copiar datos de forma segura (tanto archivos sueltos como simular sesiones FTP cifradas). SSH trabaja de forma similar a como se hace con TELNET, la diferencia es que éste utiliza técnicas de cifrado que hacen que la

información que viaja por el medio de comunicación vaya de manera no legible, evitando que terceras personas puedan estar observando.

2.6.1.9 TELNET

Telecommunation Network es el nombre de un protocolo de red que permite acceder a otra máquina para manejarla remotamente como si se estuviera delante de ella. Se sitúa en la capa de aplicación y para que la conexión funcione la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. Suele utilizarse el puerto TCP 23. TELNET sólo sirve para acceder en moro terminal, pero es una herramienta muy útil para arreglar fallos a distancia, también se usa para consultar datos e información accesibles por la red. Es un protocolo ya en desuso debido a su problema con la seguridad, ya que todos los nombres de usuarios y contraseñas necesarias para entrar en las máquinas viajan por la red como texto plano.

2.6.1.10 SSL

Secure Sockets Layer es un protocolo criptográfico que actúa en el nivel de sesión y que proporciona comunicaciones seguras en una red. Se usan certificados y criptografía asimétrica para autentificar a la contraparte con quien se está comunicando y para intercambiar una llave simétrica. Esta sesión es luego usada para cifrar el flujo de datos entre las partes, lo que permite la confidencialidad del dato/mensaje, códigos de autenticación de mensajes para integridad y, como producto lateral, autenticación del mensaje. El protocolo SSL proporciona autenticación y privacidad de la información entre extremos sobre internet mediante el uso de criptografía. Habitualmente sólo el servidor es el que se autentifica, el cliente se mantiene sin autentificar.

2.6.1.11 TLS

Transport Layer Security es una evolución del protocolo SSL, es un protocolo mediante el cual se establece una conexión segura por medio de un canal cifrado entre el cliente y servidor. El intercambio de información se realiza en un entorno seguro y libre de ataques. Al igual que ocurre con el SSL, normalmente es el servidor el único que es autenticado, garantizando así su identidad, ya que para la autentificación mutua se necesita una infraestructura de claves públicas para los clientes. Este protocolo permite prevenir escuchas, evitar la falsificación de la identidad del remitente y mantener la integridad del mensaje en una aplicación cliente-servidor.

2.6.1.12 RSH

Es un protocolo de *Shell* remoto, permite ejecutar comandos en ordenadores remotos, es capaz de lanzar una *Shell* del equipo remoto. Este programa tiene la principal dificultad de que toda la información que transmite lo realiza en texto claro, es decir sin cifrar, motivo por el cual su uso ha caído en desuso con la aparición de alternativas como el protocolo SSH. RSH ejecuta comandos de forma remota Enel host especificado, para ello debe conectarse a un servicio llamado RSHD en la máquina remota.

2.6.1.13 NTP

Network Time Protocol es un protocolo para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable. Éste utiliza UDP como su capa de transporte, usando el puerto 123. El funcionamiento de éste difiere de la mayoría del resto de protocolos. NTP no sincroniza todos los relojes conectados, sino que establece una jerarquía de servidores de tiempo y clientes. Cada nivel de esta jerarquía se denomina estrato, y el Estrato 1 constituye el nivel más alto. Es posible una sincronización de alta precisión gracias a las distintas referencias de tiempo.

2.6.1.14 NBNS

NetBios Name Service es un protocolo que se encarga de registrar los nombres de los miembros de una red NetBios sobre TCP. NBNS mantiene un registro centrar de todos los participantes en una red. Cuando un nuevo nodo necesita un nombre, el registro responde a una consulta rápida que establece la singularidad de la solicitud de nombre. Este protocolo define un formato de mensaje para la traducción de las solicitudes NBNS a direcciones IP, el servicio de nombres que se accede a través de un conjunto de mensajes de solicitud.

2.6.1.15 NBSS

NetBios Session Service es un protocolo que sirve para conectar dos ordenadores para la transmisión de mensajes largos o tráfico de peso de gran peso. Esto es así porque este protocolo está relacionado con la generación de tráfico a través del puerto TCP 139. Suele utilizarse mayoritariamente para impresión y servicios de ficheros en una red. Este protocolo está basado en el nivel de capa de transporte y de sesión.

2.6.1.16 Portmap

Port Mapper es un protocolo que se encarga de mapear el número o versión de un programa *Open Network Computing Remote Procedure Call (ONC RPC)* a un

puerto usado en la red. Tras el inicio, el servidor ONC RPC solicita el mapeador de puertos que asigne un número de puerto para cada programa utilizando el protocolo de transporte para la transferencia de archivos o fines de comunicación. Es importante tener en cuenta que los programas, antes de usar el puerto en cuestión, utilizan el mapeador de puertos para determinar que puerto se les está asignando.

2.6.1.17 SIP

Session Initiation Protocol es un protocolo que tiene la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el vídeo, voz, mensajería, realidad virtual. La sintaxis de sus operaciones se asemeja a las de HTTP y SMTP. Este protocolo se concentra en el establecimiento, modificación y terminación de las sesiones, y se complemente con otros como el SDP, que describe el contenido multimedia de la sesión, por ejemplo qué direcciones IP, puertos y códecs se usarán durante la comunicación. Es un protocolo de señalización.

2.6.1.18 DCE/RPC

Distributed Computing Environment / Remote Procedure Call es un protocolo que se encarga de utilizar un ordenador para ejecutar código en otra máquina remota sin tener que preocuparse por la comunicación entre ambas. Esta especialmente desarrollado para el entorno de la informática distribuida, permite a los programadores escribir software distribuido como si fueran todos el mismo ordenador, sin tener que preocuparse por el código de la red que subyace. Funciona de manera que el cliente envía una petición y se queda a la espera, si no llega la respuesta dentro del tiempo de espera, continúa si ejecución, el cliente no tiene realimentación en caso de fallo.

2.6.1.19 S7COMM

S7 Communication[60] es un protocolo propiedad de Siemens que sirve de comunicación entre los PLCs de la familia Siemens S7-300/400. Es utilizado para programación PLC, intercambiando datos entre ellos, permite también el acceso a sus datos desde un sistema SCADA, también con propósito de diagnóstico. Para conectarse al PLC se utiliza el protocolo TCP 102 y está situado en la capa de transporte. La comunicación sigue el modelo maestro-esclavo o cliente-servidor, dónde el PC (maestro/cliente) envía una petición al dispositivo (esclavo/servidor).

2.6.1.20 LOOP

Configuration Testing Protocol (Loop) es un protocolo de diagnóstico incluido en la especificación del Xerox Ethernet II. La funcionalidad que ofrece este protocolo

es similar a la que ofrece el sistema *ping*, pero éste opera en la capa de enlace de datos en vez de en la capa de red. El protocolo de pruebas de configuración se implementó en clientes DEC y *routers* Cisco.

2.6.1.21 OpenVPN

Es un protocolo de conectividad basado en software libre. Ofrece conectividad punto a punto con validación jerárquica de usuarios y host conectados remotamente. Ofrece una combinación de seguridad, facilidad de uso y riqueza de características. Es una herramienta multiplataforma que ha permitido simplificar la configuración de VPNs frente a otras más antiguas y difíciles de configurar como Ipsec y haciéndola más accesible para gente inexperta a este tipo de tecnología.

2.6.1.22 COTP

Connection-Oriented Transport Protocol es un protocolo de transporte perteneciente a *IsoProtocolFamiliy.* COTP se explica comparándolo con el TCP, ya que ambos se utilizan para transportar los datos de los usuarios de manera confiable a través de una red. Este protocolo transporta paquetes de datos de un usuario a otro, de manera que el receptor obtendrá exactamente los mismos datos transmitidos por el emisor. Por otro lado, el TCP transporta una secuencia continua de datos al receptor.

2.6.1.23 ESP

Encapsulating Security Payload es uno de los protocolos utilizados por IPSec. El protocolo ESP proporciona confidencialidad en los datos mediante cifrado y autenticación (integridad de datos, autenticación de origen de estos y protección en su *replay*). También se puede usar solo con confidencialidad, solo autenticación o confidencialidad y autenticación. El mecanismo de autentificación ESP autentica sólo la porción del datagrama IP del paquete correspondiente.

2.6.1.24 DNP3

Distributed Network Protocol v3[61] es un protocolo industrial para comunicaciones entre equipos inteligentes y estaciones controladoras, ambos componentes de sistemas SCADA. Aunque el protocolo fue desarrollado sobre la base de una gran fiabilidad, no fue así respecto a los aspectos de seguridad, lo cual supuso un importante error de diseño. Debido a que el protocolo puede utilizar los niveles de transporte y enlace proporcionados por el sistema TCP/IP, gran parte del problema de añadir un sistema de autenticación seguro quedaba resuelto.

2.6.1.25 Modbus/TCP

Es un protocolo de comunicaciones situado en el nivel de aplicación. Está basado en la arquitectura cliente/servidor y se utiliza principalmente como protocolo industrial, es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales ya que permite el control de una red de dispositivos. Se trata de una variante de Modbus utilizada para comunicaciones a través de redes TCP/IP, conectándose a través del puerto TCP 502. No requiere un cálculo de la suma de verificación (*checksum*), ya que las capas inferiores ya proporcionan la protección de ésta.

2.7 AMENAZAS

A lo largo de esta sección del capítulo se van a desmembrar los ataques y amenazas que se han obtenido a lo largo del análisis forense del conjunto de datos. Para ello se seguirá un orden de las posibles amenazas, clasificándolos por protocolo, 1) Su nombre o cómo son conocidos, 2) En qué consisten, cuál es su propósito, 3) Cómo se llevan a cabo y 4) Cómo identificarlas con el analizador de redes Wireshark. El último punto es importante ya que, aunque no se hayan producido, sí se sabría cómo detectarlos con el análisis de los registros.

2.7.1 TCP

2.7.1.1 Ack scanning

- Información General

Se usa para determinar si el objetivo del ataque está protegido por algún tipo de sistema de filtrado. Para ello se atacan puertos determinados.

- Cómo se lleva a cabo

El atacante envía un paquete con la bandera ACK y con un número de secuencia aleatorio, si no se recibe respuesta alguna indica que el puerto está filtrado (protegido por un *firewall*) si, por lo contrario, el objetivo devuelve una respuesta RST, implica que el puerto está cerrado. Esto se muestra en la Figura 5 a continuación:

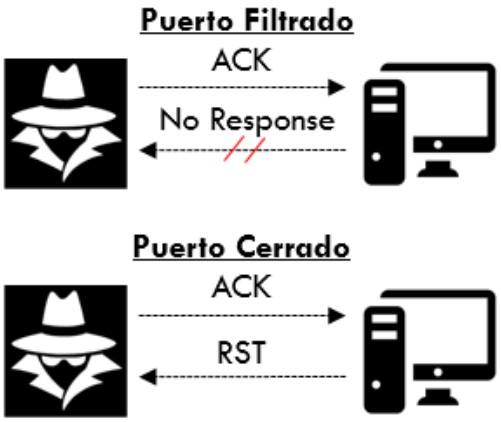


Figura 5 Ack Scanning

- Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.ack == 1$$

Y se sigue la transmisión para observar la respuesta del servidor.

2.7.1.2 Xmas scan

- Información General

Se utiliza para descubrir si un determinado puerto se encuentra cerrado.

- Cómo se lleva a cabo

El atacante envía un determinado paquete con las siguientes banderas del protocolo TCP activadas: FIN, URG y PSH. Si no se recibe una respuesta del objetivo entonces indica que el puerto está abierto o filtrado, si por lo contrario la respuesta contiene las banderas RST y ACK entonces es que está cerrado. Se muestra en la Figura 6:

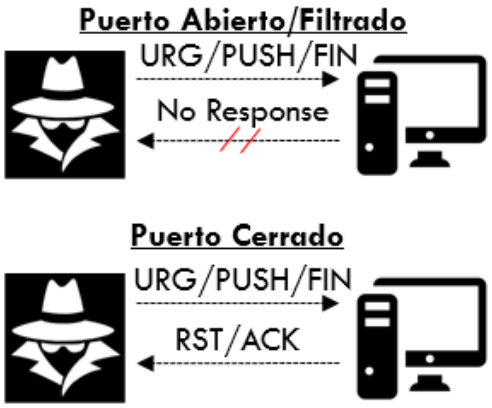


Figura 6 Xmas Scan

- Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.urg == 1 \;\&\&\; tcp.flags.fin == 1 \;\&\&\; tcp.flags.push == 1$$

Y se sigue la transmisión para observar la respuesta del servidor.

2.7.1.3 Fin scan

- Información General

Se utiliza para descubrir si un determinado puerto está cerrado.

- Cómo se lleva a cabo

El atacante envía un determinado paquete con la bandera FIN activada. Si no recibe una respuesta el puerto puede estar abierto o filtrado (por un *firewall*), si contesta con un paquete RST y ACK entonces implica que está cerrado. Se muestra en la Figura 7:

Figura 7 Fin scan

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.fin == 1$$

Y se sigue la transmisión para observar la respuesta del servidor.

2.7.1.4 Fin-Ack scanning

➢ Información General

Este escaneo se realiza con la intención de saturar la red o cerrar determinados puertos TCP.

➢ Cómo se lleva a cabo

El atacante envía de manera masiva una gran cantidad de paquetes con las banderas FIN y ACK activas a múltiples puertos. Al saturar los puertos con conexiones ininterrumpidas se provoca una ralentización del sistema y pérdida de las conexiones existentes. Se puede observar en la Figura 8:

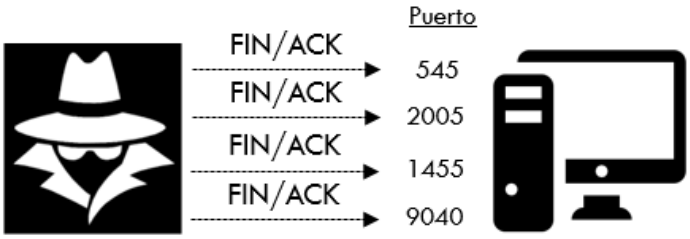


Figura 8 Fin Ack scanning

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.fin == 1 \;\&\&\; tcp.flags.ack == 1$$

Y se observa el número de peticiones desde el mismo puerto o IP a otros.

2.7.1.5 Syn scan / Stealth scan / Half open attack

➢ Información General

Es uno de los principales ataques de escaneo de puertos porque se puede realizar de forma rápida, escaneando miles de puertos por segundo a redes que no tengan *firewalls* muy restrictivos. Es bastante típico ya que no llega a completar la conexión TCP.

➢ Cómo se lleva a cabo

El atacante envía paquetes con la bandera SYN activada. Si se recibe una respuesta con las banderas SYN y ACK entonces implica que el puerto está abierto (también se considera abierto si sólo se recibe SYN). Es importante responder al servidor, después de obtener el paquete con SYN y ACK, con un nuevo paquete RST para terminar la conexión.

Si en vez de tener esa respuesta lo que se obtiene es RST y ACK el puerto se encuentra cerrado y termina la conexión. A continuación, se puede observar en la Figura 9:

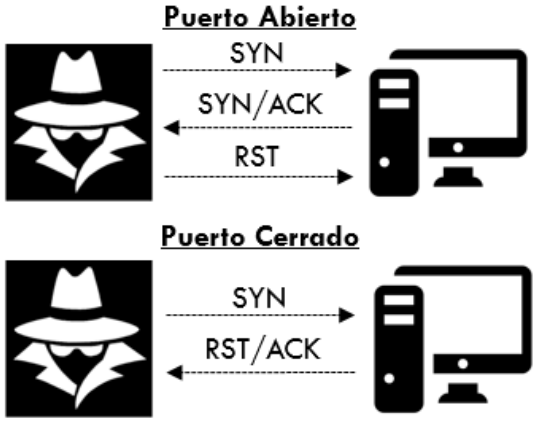


Figura 9 Syn scan / Stealth scan

➢ Identificación con Wireshark

Se utilizan los filtros:

$$tcp.flags.syn == 1 \; or \; (tcp.flags.syn == 1 \;\&\&\; tcp.flags.ack == 1)$$

Y

$$tcp.flags.reset == 1 \; or \; (tcp.flags.reset == 1 \;\&\&\; tcp.flags.ack == 1)$$

De manera que se obtengan las transmisiones completas.

2.7.1.6 Rst attack

➢ Información General

Su principal uso es la denegación de servicios (DDoS). Está basado en la comunicación del protocolo TCP, se aprovecha de que la bandera RST sirve para terminar una sesión TCP. De manera que realizando masivas peticiones de este tipo se saturan los puertos y las sesiones del servidor objetivo. El objetivo es, por tanto, agotar los recursos de éste a medida que el servidor intenta procesar solicitudes no válidas.

➢ Cómo se lleva a cabo

El atacante envía masivas peticiones TCP con la bandera RST a múltiples puertos y en periodos muy corto de tiempo. No se espera ninguna respuesta tal y como se puede ver en la Figura 10:

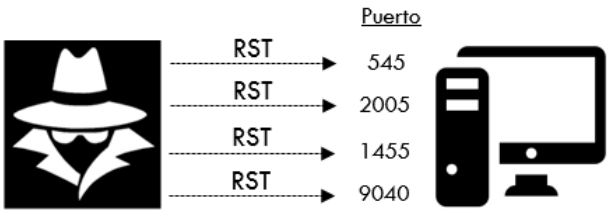


Figura 10 Rst attack

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.reset == 1$$

Y se observa el número de peticiones desde el mismo puerto o IP a otros.

2.7.1.7 Syn flood attack

➢ Información General

Este tipo de ataque lo que hace es empezar un número masivo de inicios de conexión con el servidor objetivo que nunca son finalizados, dejando al servidor en la espera del mensaje con la bandera ACK final. De esta manera se queda consumiendo recursos de forma desproporcionada. También se utiliza para saber si el puerto está abierto.

➢ Cómo se lleva a cabo

El atacante envía mensajes con la bandera SYN activada, el servidor los recibe y responde con un SYN y ACK. El atacante, ahí, no respondería, dejando al servidor en espera. El hecho de que el servidor conteste implica también que el puerto está abierto. El esquema de un caso puede verse a continuación en la Figura 11, aunque en la práctica se enviaría de manera masiva a múltiples puertos.

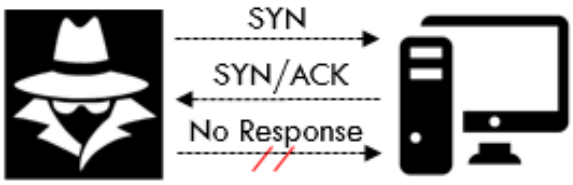


Figura 11 Syn flood attack

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.syn == 1$$

Y se observa el número de peticiones en el tiempo desde el mismo puerto o IP a otros.

2.7.1.8 Tcp ping sweep

➢ Información General

Se utiliza para descubrir qué IPs están "vivas" dentro del rango de una red. Esto se justifica porque detrás de cada IP "viva" existe un servidor o estación de trabajo que se encarga de contestar las peticiones dentro de la red. Una vez que se conocen qué IPs están disponibles, realizar ataques de penetración se vuelve una tarea mucho más sencilla.

➢ Cómo se lleva a cabo

Se realiza de la misma manera que el *Syn flood attack* solo que se envían peticiones a puertos característicos como 80 y 443. Si el servidor responde con SYN y ACK entonces es que la IP está "viva", tal y como se muestra en la Figura 12:

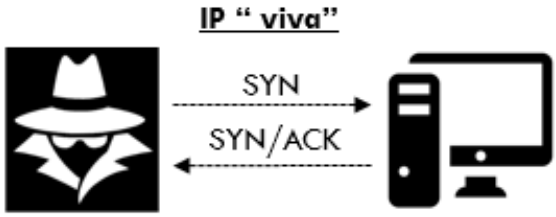


Figura 12 Tcp ping sweep

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.syn == 1 \;\&\&\; (tcp.port == 80 \; or \; tcp.port == 443)$$

Aquellas transmisiones con respuestas por parte del servidor de SYN y ACK marcarán las IP "vivas".

2.7.1.9 Null scan

- Información General

Su principal característica es que no fija ningún bit, es decir, todas las banderas del paquete están a cero. Se utiliza para averiguar si un determinado puerto está cerrado.

- Cómo se lleva a cabo

Se envía un paquete con todas las banderas del protocolo TCP fijadas a cero (ninguna activa). Si el servidor objetivo responde con la bandera RST entonces el puerto en concreto está cerrado. Si no se obtiene respuesta alguna puede estar abierto o filtrado por un *firewall.* La Figura 13 lo muestra a continuación:

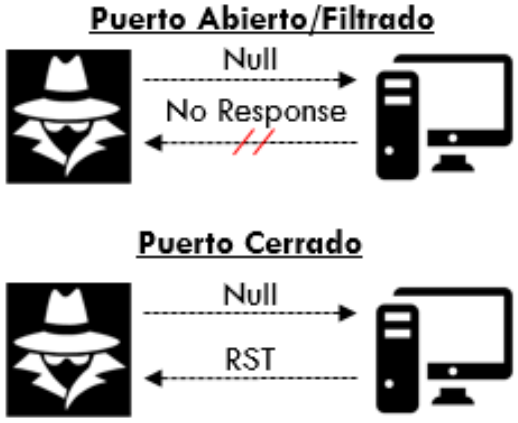


Figura 13 Null scan

- Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags == 0x000$$

Y se sigue la transmisión para observar la respuesta del servidor.

2.7.1.10 Psh-Ack flood

- Información General

Es un tipo de ataque de denegación de servicio (DDoS), consiste en obligar al servidor objetivo a vaciar su *buffer*. Cuando se realiza de manera masiva se puede inundar al servidor con muchas de estas solicitudes, abrumando el búfer correspondiente al protocolo TCP, de manera que no pueda procesar solicitudes ni procesarlas.

➢ Cómo se lleva a cabo

Se realizan, de manera masiva, peticiones TCP con las banderas PSH y ACK al mismo puerto de la misma dirección IP hasta saturarlo. Obviamente cambiando el puerto de inicio o incluso la IP para evitar ser bloqueados por el sistema. Un pequeño esquema se muestra a continuación en la Figura 14.

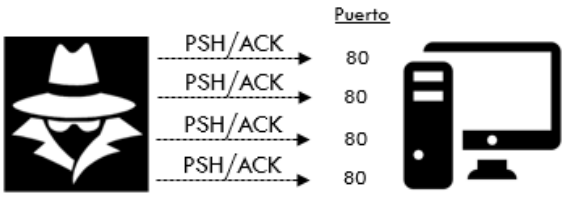


Figura 14 Psh-Ack flood

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.flags.psh == 1 \;\&\&\; tcp.flags.ack == 1$$

Y se observa el volumen de las transmisiones emitidas a los puertos.

2.7.1.11 Tcp-Dup-Ack attack / Duplicate ack spoofing

➢ Información General

Es un tipo de ataque de denegación de servicio en el que el atacante suplanta peticiones con la bandera ACK pasadas y las vuelve a emitir. Esto interfiere directamente en el rendimiento y provoca que el servidor considere que el paquete está corrupto y, por tanto, no se ha recibido correctamente, provocando la denegación de servicio buscada.

➢ Cómo se lleva a cabo

Se envían 2 paquetes exactamente iguales, al mismo tiempo, al mismo puerto de la dirección IP correspondiente. Para realizar este ataque se ha de configurar el parámetro len = 0, y que seq, ack y win coincidan. Un breve ejemplo se muestra a continuación en la Figura 15:

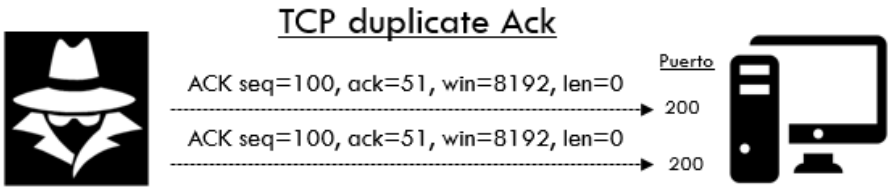


Figura 15 Ack Scanning

➢ Identificación con Wireshark

Se utiliza el filtro:

$$tcp.analysis.duplicate\_ack$$

Y se observa el volumen de peticiones duplicadas al puerto correspondiente.

2.7.2 ICMP

2.7.2.1 Icmp scan / ping sweep

➢ Información General

Es un tipo de técnica de escaneo de redes utilizada para determinar el rango de direcciones IP válidas dentro de ésta. Mientas que un único ping indicaría si existe un determinado ordenador *host* dentro de la red, un *ping sweep* consiste en múltiples peticiones ICMP *echo* enviadas a múltiples *hosts*, conociendo finalmente si son IP válidas.

➢ Cómo se lleva a cabo

Se envían paquetes ICMP como *echo request* a múltiples direcciones IP dentro de la red. Si el servidor objetivo es válido entonces devolverá una ICMP *echo reply*, tal y como se muestra en la Figura 16

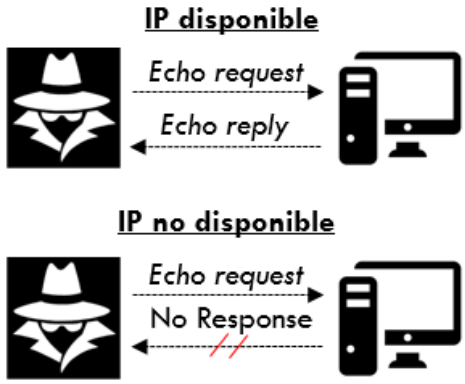


Figura 16 Icmp scan / ping sweep

➢ Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 8 \; or \; icmp.type == 0$$

Y se observa si el *echo ping request* tiene su correspondiente *echo ping reply* o si no hay contestación por parte del servidor.

2.7.2.2 Icmp route direction / redirect

➢ Información General

Este ataque se centra en, como su nombre indica, redireccionar los paquetes. Es decir, si en condiciones normales un usuario envía una petición al servidor y éste le responde, el atacante se situaría en medio de ambos, siendo capaz de ver y modificar la información que se está transmitiendo. Se suele utilizar para los tipos de ataque denominados man-in-the-middle-attacks.

➢ Cómo se lleva a cabo

Se basa principalmente en cómo está conformado el protocolo ICMP (*Internet Control Message Protocol*). El ordenador del atacante le ofrece al servidor una ruta más rápida hacia el cliente que si se lo mandara directamente a él, el servidor supondrá que ambos pertenecen a la misma red y, por tanto, se producirá un redireccionamiento de los paquetes. Un pequeño esquema se muestra en la Figura 17 a continuación:

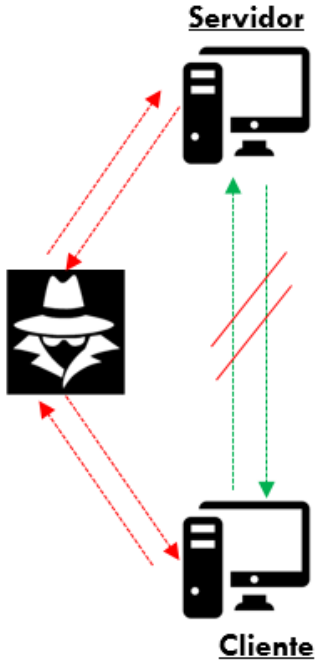


Figura 17 Icmp route direction / redirect

➢ Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 5$$

Después habría que analizar el código del error cuyo filtro es:

$$icmp.code == X$$

Y dónde cada valor implica un tipo de redirección diferentes (Tabla 2), todas susceptibles de ser un tipo de ataque.

Código	Descripción
0	Redirección del servidor
1	Redirección para el cliente
2	Redirección del tipo de servicio y servidor
3	Redirección para el tipo de servicio y el cliente

Tabla 2 Códigos de error tipo 5 ICMP

2.7.2.3 IP protocol scan

- Información General

Es ataque que resulta útil para descubrir protocolos que se pueden ejecutar sobre la IP objetivo. Esto permite asentar la base para planificar qué tipo de ataque se puede llevar a cabo.

- Cómo se lleva a cabo

El atacante se encarga de enviar paquetes distintos paquetes a una misma dirección IP para observar la respuesta del servidor. Si obtiene una respuesta ICMP de tipo 3 y código de error 2 (Tipo 3 → *Destination Unreachable*, Código 2 → *Protocol Unreachable*) significa que dicho protocolo no se está ejecutando en el sistema objetivo. Si no se tiene ninguna respuesta entonces implica que el protocolo está disponible o se encuentra filtrado.

- Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 3 \;\&\&\; icmp.code == 2$$

2.7.2.4 Smurf attack

- Información General

El ataque “pitufo” es un tipo de ataque de denegación de servicio (DDoS) que utiliza mensajes de *ping* transmitidos de forma masiva, que suelen provenir de un ordenador infectado (la víctima de este ataque), para saturar las transmisiones de éste.

➢ Cómo se lleva a cabo

El perpetrador envía grandes cantidades de paquetes ICMP de tipo *ping* a las direcciones IP de una red, todos ellos con la dirección de origen cambiada (la víctima). Por la forma de transmisión de estos paquetes, que es a la dirección de *broadcast*, el enrutamiento del tráfico será a través de la capa 2. La mayoría de los *host* que reciben la petición de *ping* ICMP *echo request* responderán, multiplicando el tráfico por cada *host* de la subred. Todas las respuestas, al volver a la IP origen, la IP de la víctima, provocan la denegación de servicio. Un pequeño esquema ilustra la configuración de este ataque en la Figura 18:

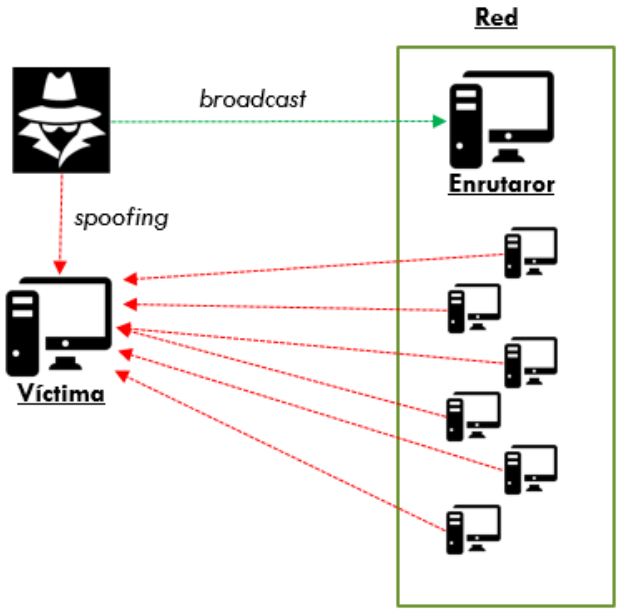


Figura 18 Smurf attack

➢ Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 8 \; or \; icmp.type == 0$$

Una vez se tiene el listado hay que observar que la dirección MAC (la física) que inicia el *echo ping request* no coincida con la dirección MAC del destinatario proveniente de la respuesta *echo ping reply* de la red. La dirección IP ha de ser la misma, ya que este es un caso de *spoofing.*

2.7.2.5 Ping of death

➢ Información General

El objetivo de este ataque consiste en producir una caída del sistema, es un ataque de denegación de servicio. Consiste en enviar un paquete ICMP cuyo tamaño total

supere el máximo autorizado por la configuración IP, éste será fragmentado y a la hora de ser reensamblado el sistema puede reiniciarse o colapsar.

- Cómo se lleva a cabo

El atacante envía un conjunto masivo de paquetes ICMP *echo ping request* deformados. Si un *ping* tiene un tamaño de 64 bytes normalmente, en este caso sería de más de 65.535 bytes, que es el máximo permitido por la configuración IP común. La forma de enviar un paquete tan grande es fragmentándolo, de manera que cuando el ordenador objetivo vuelva a montar el paquete ocurra una saturación del *buffer*, causando un fallo en el sistema.

- Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 8 \;\&\&\; frame.len > 65535$$

Y se observan el número de peticiones.

2.7.2.6 Icmp tunneling

- Información General

Es un tipo de ataque en el que se crea un canal secreto entre un ordenador y el objetivo cuyo flujo de información no está controlado por ningún mecanismo de seguridad. Es decir, este túnel obliga al supuesto *firewall* a no activar una alarma si los datos se envían a través de este protocolo. Se utiliza para inyectar datos aleatorios e incluso comunicaciones encriptadas.

- Cómo se lleva a cabo

Se inyectan datos arbitrarios en un paquete ICMP *echo ping request* y se envían al objetivo, éste inyecta una respuesta en otro paquete y lo envía de vuelta de la misma manera, con bytes extra en el campo de datos del ICMP. Esta comunicación pasa desapercibida para los *firewalls* de tipo proxy, y es posible por la propia estructura del protocolo. Los paquetes ICMP pueden tener cualquier longitud de datos arbitraria para el tipo 0 (*echo reply*) y el tipo 8 (*echo request*).

Este tipo de conexión está categorizada como canal de comunicación encriptado entre dos ordenadores.

- Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 8 \; or \; icmp.type == 0$$

Y entonces, teniendo en cuenta que la longitud normal de un paquete *ICMP echo request* debe rondar los 42 bytes, donde la longitud del campo *data* debería ser 0.

Se observan todas aquellas que muestren un número más elevado, sobre todo aquellas que tienen *echo ping request* y *echo ping reply*. Una vez identificadas se contempla lo que contiene el campo de datos del protocolo para descubrir si se están comunicando entre ellos.

2.7.2.7 Icmp get timestamp

- Información General

Es un procedimiento que explota ciertas vulnerabilidades del protocolo ICMP y que permite conocer el sistema operativo utilizado por el servidor objetivo dependiendo de cómo responda. Inicialmente utiliza el mismo procedimiento cuyo fin es conocer la latencia a lo largo de la red.

- Cómo se lleva a cabo

Una petición ICMP de tipo *Get Timestamp* permite a un ordenador preguntar a otro acerca de la hora actual. El atacante manda diferentes tipos de paquetes ICMP para detectar particularidades del sistema operativo.

- Identificación con Wireshark

Se utiliza el filtro:

$$icmp.type == 13 \; or \; icmp.type == 14$$

Y se observan el número de peticiones a diferentes direcciones IP así como la respuesta de las máquinas.

2.7.3 ARP

2.7.3.1 Arp poisoning / spoofing

- Información General

Es un tipo de ataque en el que se utiliza el protocolo ARP (*Address Resolution Protocol*) para vincular, de manera fraudulenta, una dirección MAC (la del atacante) con la dirección IP de un equipo legítimo en la red. De manera que se empezará a recibir cualquier dato que fuera a enviarse a la dirección IP en cuestión.

➢ Cómo se lleva a cabo

Cuando un sistema conectado a una red quiere comunicarse con otro, debe enviar primero un mensaje de difusión (*broadcast*) a toda la red usando este protocolo. El mensaje es enviado y recibido por todos los *hosts*, los cuales lo procesan y devuelven al interesado la dirección IP asociada a esa dirección MAC y se guarda en la tabla correspondiente al Caché ARP.

El *Arp poisoning* consiste justamente en modificar esa caché, de forma que el sistema objetivo recuerde una asociación falsa entre dirección IP y dirección MAC. Con esto logrado, toda comunicación hacia la "IP infectada" irá directa al ordenador del atacante, el cual ya guardará, modificará y reenviará a su destino original. Habiendo conseguido esto se pueden lograr los siguientes ataques:

- **Denegación de servicio**: si se enlazan múltiples direcciones IP de una red con la dirección MAC de un solo objetivo, el tráfico que originalmente está destinado a muchas IP se redirigirá concentrado a la MAC prestablecida, saturando al objetivo con el volumen de tráfico.
- ***Session hijacking**:* al poder monitorearse el tráfico de la IP infectada, se pueden robar identificadores de sesión, garantizando acceso a los sistemas privados y datos confidenciales.
- **Man-in-the-middle Attacks:** al estar en medio de dos dispositivos, se puede utilizer el envenenamiento por ARP para interceptor y/o modificar el tráfico entre dos víctimas.

A continuación, en la Figura 19, se muestra un pequeño esquema de este ataque:

Figura 19 Arp poisoning MITM attack

- Identificación con Wireshark

$$arp.duplicate-address-frame$$

A continuación, se examinan los resultados y se observa si una dirección IP de la red está siendo reclamada por otra dirección MAC, probablemente la del atacante.

2.7.3.2 Arp sweep / scan

- Información General

Es un tipo de ataque cuyo objetivo es realizar un escaneo rápido para identificar las IP de los *hosts* que se encuentran activas en la red. Presenta un beneficio y una dificultad. En primer lugar, mostrará todos los dispositivos, incluso los que tengan *firewalls* activos. Por otro lado, el protocolo ARP al ser no-enrutable, tiene que hacerse dentro de una red local o un segmento de red.

- Cómo se lleva a cabo

El atacante envía paquetes ARP a cualquier número de *hosts* como destinatarios, estos responderán con la dirección MAC correspondiente a la IP preguntada. De esta manera se sabe cuáles son los *hosts* activos.

- Identificación con Wireshark

$$arp$$

No hay forma particular de detectar este escaneo salvo por el hecho de que al mostrar todos los paquetes de este protocolo si una misma dirección MAC está haciendo peticiones a múltiples IP diferentes, se estará produciendo este tipo de escaneo.

2.7.4 UDP

2.7.4.1 Udp scan

- Información General

Este tipo de ataque se utiliza generalmente para descubrir si un determinado puerto está disponible. Este caso es interesante porque las peticiones se hacen en un protocolo y las respuestas en otro.

➢ Cómo se lleva a cabo

El atacante se dedica a enviar paquetes UDP (*User datagram protocol*) sin ningún tipo de data significativa a múltiples puertos. A estos paquetes, si el servidor objetivo responde con paquetes ICMP de tipo 3 y código 3 (*Destination Unreachable – Port unreachable*) implicará que el puerto está cerrado. Si, por lo contrario, responde con otro paquete UDP, el puerto estará abierto y si no hay respuesta, entonces indicaría que el puerto está abierto o filtrado por un *firewall*. La Figura 20 lo muestra a continuación:

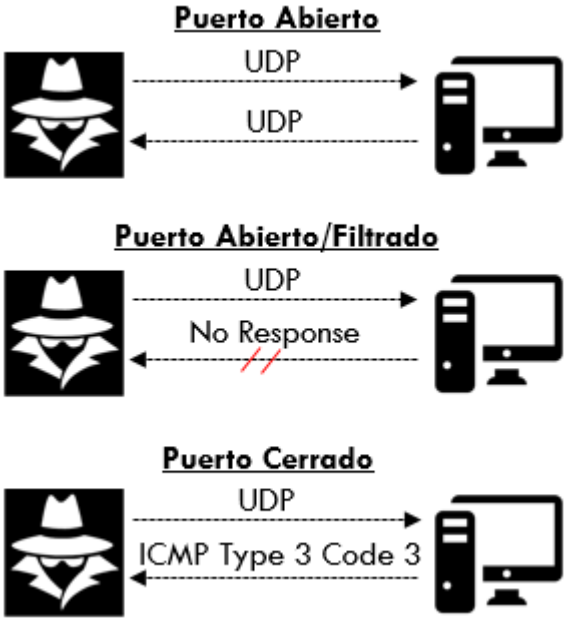


Figura 20 UDP scan

➢ Identificación con Wireshark

$$udp\ or\ (icmp.type == 3\ and\ icmp.code == 3)$$

Es conveniente cotejar las peticiones UDP con las respuestas ICMP pero un gran indicador de que se está realizando este ataque es simplemente el elevado número de paquetes ICMP de tipo 3 y código 3.

2.7.4.2 Udp flood attack

➢ Información General

Este tipo de ataque tiene como objetivo la denegación de servicio del sistema objetivo. Se realiza de la misma manera que el anterior, pero sin estar pendiente de la repuesta del sistema, mandando paquetes UDP de manera masiva a puertos, incluso repetidos.

➢ Cómo se lleva a cabo

El atacante se dedica a enviar paquetes UDP de manera masiva a puertos del objetivo a atacar. Da igual que respuesta dé el servidor porque lo que se busca es justo eso, que responda y realice las siguientes actividades para agotar sus recursos:

- Que compruebe la petición al puerto.
- Que vea que la comunicación no se ha completado.
- Que responda con un paquete *ICMP Type 3 Code 3*.

➢ Identificación con Wireshark

Se utiliza el filtro:

$$udp$$

Y se observa el número masivo de peticiones UDP por parte de una misma IP y MAC.

2.7.5 DNP3

2.7.5.1 Dnp3 warm reset

➢ Información General

Es un tipo de ataque de denegación de servicio para procesos industriales. Consiste en explotar una de las funcionalidades básicas de la configuración del protocolo DNP3, que es el permitir un reinicio parcial en la secuencia de las comunicaciones.

➢ Cómo se lleva a cabo

El atacante envía, de manera masiva, paquetes del protocolo DNP3 con los bits necesarios para realizar un reinicio parcial cuando se complete la comunicación de la secuencia. Es importante antes hacerle creer al sistema atacado que los mensajes los envía el sistema maestro o *master*. La vulnerabilidad se encuentra a la hora de enviar de manera masiva estos ataques ya que, al reconocer el sistema que proviene de un sistema maestro, comienza a realizar en bucle dicho reinicio parcial. De esta manera se produce la denegación de servicio buscada.

➢ Identificación con Wireshark

Se utiliza el filtro:

$$dnp3.al.func == 14$$

Y se observa el tráfico hacia la IP atacada.

2.7.5.2 Dnp3 cold start

- Información General

Es un tipo de ataque de denegación de servicio para procesos industriales. Al igual que el *warm start*, explota otra de las vulnerabilidades del protocolo. Esta vez provocará un reinicio completo.

- Cómo se lleva a cabo

En primer lugar, el atacante se hace pasar por el *master* para que el sistema objetivo pueda acatar las peticiones. A continuación, envía paquetes de este protocolo con los bits activos necesarios para provocar un reinicio completo al finalizar la secuencia de comunicación. Al enviar estos paquetes de manera masiva provoca la denegación de servicio buscada al entrar el sistema en un bucle de reinicio continuo.

- Identificación con Wireshark

Se utiliza el filtro:

$$dnp3.al.func == 13$$

Y se observa el tráfico hacia la IP atacada.

2.7.5.3 Dnp3 time change

- Información General

Es un tipo de ataque que inhabilita las operaciones del sistema, así como su supervisión a tiempo real. Se trata de una denegación de servicio

- Cómo se lleva a cabo

El atacante envía un tipo de paquete del protocolo DNP3 que provoca que el IED (entorno de desarrollo interno*)* y la RTU (unidad de terminar remota) cambien su reloj interno, lo que ocasiona que las órdenes recibidas no sean ejecutadas cuando se les pide y que los registros de los eventos se guarden en lugares distintos por lo que el operador es incapaz de ver lo que ocurre en tiempo real. De esta manera se consigue la denegación de servicio buscada.

- Identificación con Tcpdump

Se utiliza el filtro:

$$ip[52] = 2\ and\ ip[53] = 0x32\ and\ ip[54] = 1$$

Observándose, a continuación, qué está ocurriendo.

2.7.6 FTP

2.7.6.1 Brute force attack

- Información General

Este tipo de ataque se aprovecha de los comandos propios del protocolo FTP para pedirle al servidor datos como el usuario o contraseña. Básicamente consiste en hacer peticiones al servidor para enviarle de manera masiva el usuario y contraseña hasta acertar y conseguir acceso al servidor. Es un ensayo de prueba y error

- Cómo se lleva a cabo

Se realiza, de manera masiva, peticiones de usuario y contraseña al servidor por medio del protocolo FTP: peticiones de tipo USER y PASS. En función de lo que el atacante pruebe, se generan dos tipos de respuestas según el servidor, código 530 (para identificar que el *login* ha sido incorrecto) y código 200 (que identifica que el *login* ha sido un éxito).

- Identificación con Wireshark

Se utiliza el filtro:

$$ftp.request.command == USER \text{ or } ftp.request.command == PASS$$

Para ver el número de peticiones que se han llevado a cabo. También se puede combinar con:

$$ftp.response.code == 530$$

Para así tener en cuenta el número de intentos fallidos o incluso:

$$ftp.response.code == 200$$

Para ver si se ha conseguido penetrar, de darse este caso habría que observar los siguientes $ftp.request.command$ () para ver qué ha hecho el atacante después de conseguir el *login* exitoso:

LIST	Petición de una lista de los archivos en el directorio correspondiente.
STOR	Subir datos al servidor FTP.
RETR	Descargar datos desde el servidor.

2.7.6.2 Packet Capture / Sniffing

- Información General

Este ataque se centra en una de las principales vulnerabilidades del protocolo FTP. Ésta es que todo lo que se transmite a través de éste está en *clear text*, es decir, sin

codificar y visible para cualquiera que se encuentre monitoreando la red. Con esto se pueden obtener usuarios, contraseñas y todo tipo de documentación e información necesaria para acceder al servidor FTP o que se encuentre alojada ahí.

- Cómo se lleva a cabo

En primer lugar, es importante poner a capturar el tráfico que se está produciendo en una determinada red, para ello se utilizan los *packet sniffers*. Una vez que se está haciendo, se presta atención a las comunicaciones del protocolo TCP. Éstas se dan en texto plano y no están codificadas luego cualquier *TCP Retransmission* se puede capturar y observar, de manera que se descubre todo lo que está sucediendo en el servidor: usuarios y contraseñas, descarga de ficheros, disposición de contenido etc.

- Identificación con Wireshark

Se utiliza el filtro:

$$ftp$$

Para ver qué se está transmitiendo por este protocolo, lo cual se puede asumir que, de haber un *packet sniffer* actuando, lo guardará. En base a eso estar atentos para ver qué IP y dirección MAC solicita entrar con el usuario y contraseña previamente transmitido.

Para ello se volverá a usar el siguiente filtro:

$$ftp.request.command == USER\ or\ ftp.request.command == PASS$$

Capítulo 3 PROPUESTA DEL ESTÁNDAR EN FORMATO IMAGEN

3.1 INTRODUCCIÓN

Tras haber presentado las posibles amenazas detectadas en el dataset mediante el programa capturador de paquetes Wireshark, lo siguiente es exponer cómo se ha ido conformando la propuesta en formato imagen. Para ello a lo largo de este capítulo se repasarán las bases sobre las que se empezó el proceso iterativo para la conformación del estándar. Tras éstas se proseguirá con el proceso a seguir para la conformación de éstas y, finalmente, cada una de las principales iteraciones, así como por qué se fueron descartando hasta llegar a la propuesta final.

Es importante tener en cuenta que a medida que se iban profundizando en el análisis forense del dataset las imágenes iban adquiriendo también mayor profundidad para ser capaz de presentar la mayor cantidad posible de información en formato visible y además no tener pérdida de ésta para poder ser analizada por otras tecnologías de aprendizaje automático.

Al final de este capítulo se presentará la propuesta última, con todo lo que ello conlleva, como resultado del proceso iterativo.

3.2 BASES DEL FORMATO IMAGEN

En esta sección se pretenden establecer las bases que conforman el estándar en formato imagen. A partir de éstas luego se han ido realizando iteraciones a lo largo del proyecto hasta conformar la propuesta final. Es importante, por lo tanto, tener en cuenta que éstas son comunes a todas.

3.2.1 MATRIZ DE 128X128 PIXELES

Después de realizar un estudio del estado del arte respecto al tamaño de las imágenes que sirven como *inputs* para diferentes técnicas de *machine learning* y *deep learning*[62], [63], que se podría decir que un 220x220 es lo más común, se decidió a utilizar una matriz de 128x 128 pixeles ya que presentaba las siguientes ventajas:

- Para ser una imagen, presenta un peso pequeño (aprox 2,5KB) que, aunque resulte mayor que el almacenamiento de los *logs* en texto plano, permisible para poder acumular millones de éstas sin que ello represente un inconveniente.
- Al ser de 128x128, se pueden añadir hasta un máximo de 16384 elementos, lo cual da una gran libertad a la hora de estructurar la información dentro de las imágenes, más adelante esto se comentará.
- Se ha considerado que 128x128 cumple el compromiso de ser lo suficientemente pequeño para ocupar lo menos posible y lo suficientemente grande como para permitir un análisis visual por parte de personas formadas.

A continuación, en la Figura 21, se muestra lo que sería una de estas imágenes a escala 1:1.

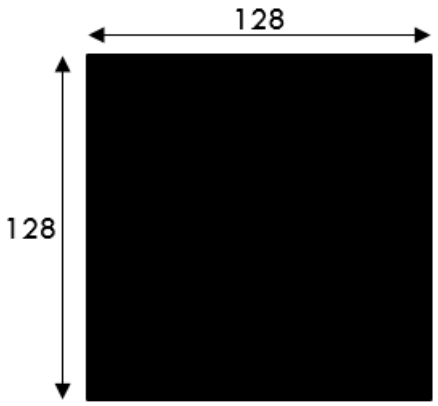


Figura 21 Matriz base de 128x128 pixeles

3.2.2 ESCALA DE GRISES UINT-8

Se plantearon diferentes formas para poder representar todos los caracteres posibles que tendrían que estar presentes en las imágenes. Se valoraron las siguientes posibilidades:

- Binario (Negro o Blanco)
- RGB (a color)
- **UINT-8 (Escala de grises) la finalmente escogida.**

3.2.2.1 Codificación Binaria

Esta, en apariencia simple y sencilla, planteaba un problema. Para ser capaz de representar todos los valores posibles que se pueden encontrar en los *logs* interpretados había que codificarlas al estilo código QR. Esto, teniendo en cuenta tenía que ser válido para un análisis visual por parte de personas, inhabilitada esta posibilidad.

3.2.2.2 Codificación RGB

La forma en la que se codificaría sería mediante conjunto de colores según los campos a la par que degradados según el valor dentro de estos. Era una opción con grandes posibilidades sobre todo en el ámbito de la inspección visual. El principal problema, y por lo que fue rechazada, es que para el tratamiento de imágenes una de las primeras cosas que se hace es pasarlas a escala de grises. Esto chocaba con la que en primer momento era una ventaja. Por otro lado, a nivel cuantitativo no es igual trabajar en dos dimensiones que en las correspondientes a un sistema RGB.

3.2.2.3 Codificación UINT-8, escala de grises

La escala de grises permite valores de entre 0 a 255. Un número de posibilidades lo bastante elevado para poder representar todos los posibles caracteres a encontrar en los registros. Por ello, además de que la escala de grises es relativamente sencilla de diferenciar por el ojo humano, sumado a que las técnicas de *machine learning* y *computer vision* trabajan con la matriz de píxeles de imágenes en escala de grises, fueron los motivos por los que al final se decantó este tipo de codificación.

3.2.3 DICCIONARIO DE CONVERSIÓN

Un diccionario de conversión ha sido el elemento a diseñar de cara a la codificación de los caracteres a encontrar dentro de los *logs* para pasarlos escala de grises. Se ha establecido una correlación entre los diferentes elementos a números de entre 0 a 255, a continuación se muestra esta correlación en la Tabla 3:

	0	**j**	10	**t**	20	**D**	83	**N**	93
a	1	**k**	11	**u**	21	**E**	84	**O**	94
b	2	**l**	12	**v**	22	**F**	85	**P**	95
c	3	**m**	13	**w**	23	**G**	86	**Q**	96
d	4	**n**	14	**x**	24	**H**	87	**R**	97
e	5	**o**	15	**y**	25	**I**	88	**S**	98
f	6	**p**	16	**z**	26	**J**	89	**T**	99
g	7	**q**	17	**A**	80	**K**	90	**U**	100
h	8	**r**	18	**B**	81	**L**	91	**V**	101
i	9	**s**	19	**C**	82	**M**	92	**W**	102
X	103	**%**	57	**´**	64	**;**	71	**~**	79
Y	104	**&**	58	**-**	65	<	72	^	194
Z	105	**'**	59	**_**	66	>	73	**`**	195
!	53	**(**	60	**.**	67	=	74	**Ç**	196
“	54	**)**	61	/	68	**?**	75	**ü**	197
#	55	*****	62	\	69	**{**	76	**é**	198
$	56	+	63	**:**	70	**}**	77	**â**	199
ä	200	**ï**	207	**Æ**	214	**Ö**	107	**á**	114
à	201	**î**	208	**ô**	215	**Ü**	108	**í**	115
å	202	**ì**	209	**ö**	216	**ø**	109	**ó**	170

ç	203	**À**	210	**ò**	217	**£**	110	**ú**	171
ê	204	**Ă**	211	**û**	218	**Ø**	111	**ñ**	172
ë	205	**É**	212	**ù**	219	**×**	112	**Ñ**	173
è	206	**æ**	213	**ÿ**	106	***ƒ***	113	**ª**	174
º	175	**»**	28	**¥**	35	**Ì**	42	**µ**	49
¿	176	**«**	29	**Ê**	36	**Ó**	43	**Ú**	50
®	177	**Á**	30	**Ë**	37	**ß**	44	**Û**	51
½	178	**Â**	31	**È**	38	**Ô**	45	**Ù**	52
¼	179	**À**	32	**Í**	39	**Ò**	46	**ý**	181
¬	180	**©**	33	**Î**	40	**Õ**	47	**Ý**	182
¡	27	**¢**	34	**Ï**	41	**Õ**	48	**¯**	183
≡	184	**§**	188	**²**	192	**1**	125	**5**	145
±	185	**÷**	189	**³**	193	**2**	130	**6**	150
¾	186	**·**	190	**@**	194	**3**	135	**7**	155
¶	187	**¹**	191	**0**	120	**4**	140	**8**	160
								9	165

Tabla 3 Diccionario de conversión

3.3 ITERACIONES PARA CONFORMAR LAS IMÁGENES

En este apartado se muestra, a continuación, como se han ido formando las imágenes. Se irá detallando paso a paso el proceso iterativo seguido para llegar a la solución final propuesta. Si lo que se busca es justo ésta última, puede ir a la siguiente referencia, Sección 3.3.6: Solución final.

3.3.1 PROCESO A SEGUIR

Para la formación de las imágenes el procedimiento a seguir es el siguiente:

1. Se carga el fichero .pcap en Wireshark y se escogen los campos que se consideren más representativos.
2. Se exportan dichos *logs* ya interpretados a un archivos .csv.
3. Se carga ese archivo a Matlab.
4. En Matlab se ejecuta un *script* que permite codificar todos los elementos a encontrar en el .csv siguiendo el diccionario de conversión y se separan por los campos escogidos del Wireshark.
5. Una vez que se tienen ya los datos separados y convertidos a UINT-8, de 0 a 255, cada uno de estos, dependiendo del campo al que pertenezcan, rellenará una matriz de pixeles. Cada campo será a su vez otra matriz de pixeles.
6. Finalmente, el código de Matlab redistribuye los nuevos datos codificados de forma estructurada dentro de la matriz de pixeles de 128x128 y se exporta en formato .tiff, que es un formato de imagen sin compresión.

De cara a distribuir las imágenes, se evaluó de manera aproximada el número de elementos que se podían llegar a encontrar en cada uno de los campos, ya interpretados, de Wireshark. Los resultados son los mostrados en la Tabla 4:

Campo	**Nº máximo de elementos**
No	Indefinido
Time	27
Source	17
Destination	17
Protocol	6
Length	3
SrcPort	5
DestPort	5
SrcMac	17

DestMac	17
Info	Indefinido

Tabla 4 Número máximo de elementos por campo

Estos números máximos serán de gran relevancia a la hora de redistribuirlos dentro de la estructura de cada una de las iteraciones de las imágenes propuestas.

Para ello, los campos se estructurarán como matrices. Luego, además, cada elemento, dependiendo del campo al que pertenezca, será otra matriz de píxeles de valor correspondiente a la codificación del elemento en cuestión. Un pequeño esquema se muestra en la Figura 22:

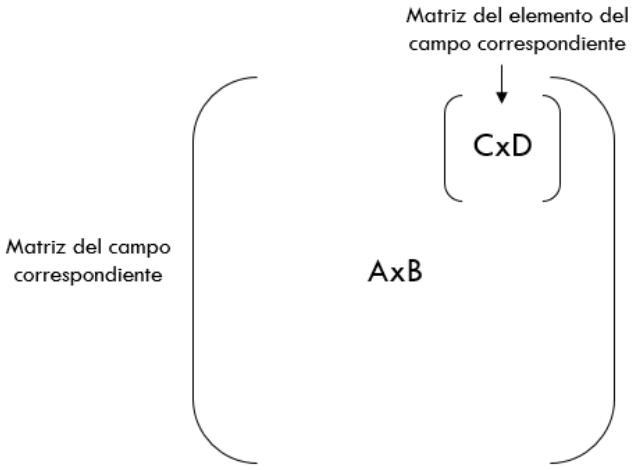


Figura 22 Organización de las matrices de píxeles

A continuación, se expondrán las iteraciones que se han llevado a cabo hasta la propuesta final del formato imagen. La estructura que seguirá la siguiente sección del capítulo será la siguiente:

- Campos elegidos en Wireshark.
- Matrices de cada elemento y disposición de los campos.
- Resultado propuesto.
- Observaciones y motivos por lo que se ha necesitado otra iteración.

3.3.2 PRIMERA ITERACIÓN

➢ Campos de Wireshark

Para la primera iteración se consideraron los campos de Wireshark propuestos a continuación en la Tabla 5:

No	Time	Source	Destination	Protocol	Length	SrcPort	DestPort	Info

Tabla 5 Campos de Wireshark primera iteración

Dónde:

- No: número del *log* en el que fue grabado.
- Time: fecha del evento, siguiendo el siguiente formato: "aaaa-mm-dd hh:mm:ss".
- Source: la dirección IP desde donde se realiza la petición.
- Destination: la dirección IP que recibe la petición.
- Protocol: protocolo de comunicaciones utilizado en el *log.*
- Length: longitud de la cadena del *log*, aquí se encuentra lo que ocupa y cuanto espacio está dedicado a cada una de las tramas del registro.
- SrcPort: puerto desde donde parte la petición.
- DestPort: puerto al que le llega la petición.
- Info: información general del *log*.

➢ Matrices de los pixeles

Con el objetivo de estructurar las imágenes, tal y como se ha comentado anteriormente, se constituye la Tabla 5 mostrando el valor de las matrices de cada campo, la correspondiente a sus elementos y el número máximo de elementos a representar.

Campo	**Matriz del campo**	**Matriz del elemento**	**Nº de elementos en el campo**
No	36x26	3x13	24
Time	42x26	3x13	28
Source	45x56	15x7	24
Destination	45x56	15x7	24

Protocol	30x36	15x9	8
Length	12x36	12x9	4
SrcPort	20x36	15x7	6
DestPort	20x36	15x7	6
Info	30x120	6x3	200

Tabla 6 Distribución de las matrices de la primera iteración

Como se puede observar, en todos los campos se cumple que el número posible de elementos a representar sea igual o mayor que el número máximo aproximado esperable.

Por otro lado, las matrices correspondientes a los campos han sido elegidas de manera que sean similares aquellas que se tienen que cotejar unas con otras, como es el caso de las direcciones IP y de los puertos. El resto, aunque puedan variar en poco, es lo suficientemente significativo para que la diferencia sea notable.

- Resultado de la imagen

Según las características expuestas en los anteriores apartados, la propuesta correspondiente a la primera iteración queda reflejada en la Figura 23:

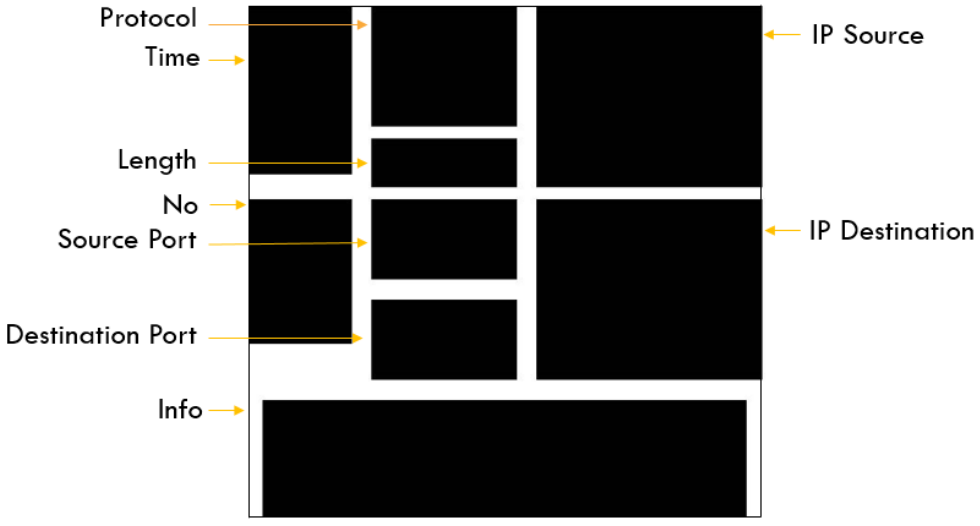


Figura 23 Propuesta de formato imagen primera iteración

Como se puede apreciar en la figura, la imagen queda distribuida de tal manera que una inspección visual permite un rápido análisis, se conserva toda la información del *log* y, por su distribución y formato, permite el uso de técnicas de procesamiento de imágenes.

- Observaciones

Esta primera iteración se realizó como un producto mínimo viable al comienzo del proyecto, cuando también se estaba empezando a realizar el análisis forense de datos. Debido a esto quedó obsoleta al poco tiempo cuando se observó que se podía añadir, a las propias imágenes, información ya tratada del análisis. Por ello la segunda iteración, muy parecida a la primera, tiene la misma estructura, pero añadiendo el autodenominado campo "Control Bits", que se encarga de representar los bits de control del protocolo TCP (las banderas activadas).

3.3.3 SEGUNDA ITERACIÓN

- Campos de Wireshark

Para la segunda iteración se consideraron los mismos campos de Wireshark que en la primera iteración, se vuelven a mostrar en la Tabla 7:

No	Time	Source	Destination	Protocol	Length	SrcPort	DestPort	Info

Tabla 7 Campos de Wireshark segunda iteración

Dónde:

- No: número del *log* en el que fue grabado.
- Time: fecha del evento, siguiendo el siguiente formato: "aaaa-mm-dd hh:mm:ss".
- Source: la dirección IP desde donde se realiza la petición.
- Destination: la dirección IP que recibe la petición.
- Protocol: protocolo de comunicaciones utilizado en el *log.*
- Length: longitud de la cadena del *log*, aquí se encuentra lo que ocupa y cuanto espacio está dedicado a cada una de las tramas del registro.
- SrcPort: puerto desde donde parte la petición.
- DestPort: puerto al que le llega la petición.
- Info: información general del *log*.

Y además se añade el siguiente campo, el cual no corresponde a Wireshark, sino que es información implícita que se encuentra dentro del *log*. Se encuentra representado en la Tabla 8:

TCP – Control Flags

Tabla 8 Campo extra para el log segunda iteración

Dónde:

- TCP – Control Flags: muestra las banderas correspondientes al protocolo TCP si están activas o si no, éstas pueden ser: **RST, ACK, SYN, FIN, PSH** y **URG.**

➢ Matrices de los pixeles

A continuación, la Tabla 9 muestra el valor de las matrices de cada campo, la correspondiente a sus elementos y el número máximo de elementos a representar.

Campo	**Matriz del campo**	**Matriz del elemento**	**Nº de elementos en el campo**
No	36x26	3x13	24
Time	42x26	3x13	28
Source	45x56	15x7	24
Destination	45x56	15x7	24
Protocol	20x36	20x4	9
Length	9x36	9x9	4
SrcPort	20x36	15x7	6
DestPort	20x36	15x7	6
Info	30x120	6x3	200
Control Bits	10x36	10x6	6

Tabla 9 Distribución de las matrices de la segunda iteración

Respecto a la anterior iteración, la matriz del campo correspondiente a Protocol cambia para poder añadir el nuevo campo: Control Bits. Por otro lado, se añade un elemento más para éste al encontrar, en el dataset, el protocolo ModbusTCP (9 elementos). Las dimensiones de Length también se ven disminuidas un poco, pero sin variar su número máximo de elementos. Destacar, por último, el nuevo campo Control Bits el cual tiene 6 elementos ya que corresponden a las 6 posibles banderas del protocolo TCP.

- Resultado de la imagen

Según las características expuestas en los anteriores apartados, la propuesta correspondiente a la segunda iteración queda reflejada en la Figura 24:

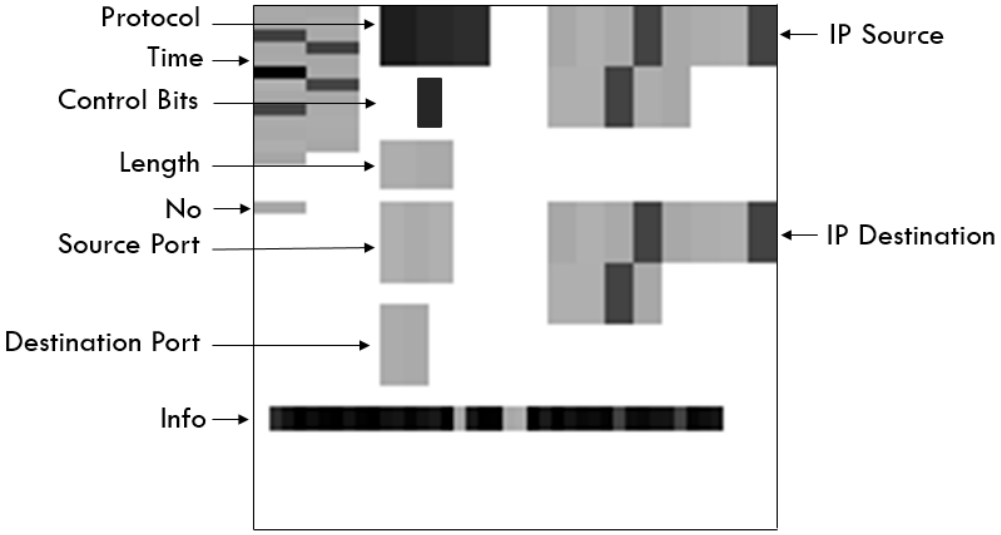


Figura 24 Propuesta de formato imagen segunda iteración

La disposición del nuevo campo ha sido abajo del de Protocol ya que está directamente ligado a éste. Se rellena únicamente si es un caso de protocolo TCP, y dependiendo de la bandera que lleve activa el elemento correspondiente de este campo se pintará de negro.

A diferencia de la primera iteración, esta imagen ya sí representa un *log* concreto.

- Observaciones

Esta imagen incluía mayor información que la anteriormente propuesta, lo cual era un avance significativo. Ahora era posible identificar ataques basados en el protocolo TCP simplemente observando qué recuadros estaban pintados de negro, además de significar que determinados pixeles de valor correspondiente a 0, implicaban determinadas características ya analizadas. Por otro lado, esta propuesta

fue también desechada pronto cuando, al seguir profundizando en el análisis forense se vio que era necesario que este estándar contuviera las direcciones físicas (dirección MAC) de los dispositivos encargados de enviar y recibir las peticiones de los *logs.* Debido a esto la siguiente iteración conllevó una pequeña restructuración en lo que a la disposición de la imagen se refiere para añadir tanto la dirección MAC desde dónde se enviaba la petición como la de destino.

3.3.4 TERCERA ITERACIÓN

- Campos de Wireshark

Para la tercera iteración se han considerado los campos anteriores de Wireshark más dos que antes no se habían utilizado. Se muestran en la siguiente tabla (Tabla 10):

No	Time	Source	Destination	Protocol	Length	SrcPort	DestPort	SrcMac	DestMac	Info

Tabla 10 Campos de Wireshark tercera iteración

Dónde:

- No: número del *log* en el que fue grabado.
- Time: fecha del evento, siguiendo el siguiente formato: "aaaa-mm-dd hh:mm:ss".
- Source: la dirección IP desde donde se realiza la petición.
- Destination: la dirección IP que recibe la petición.
- Protocol: protocolo de comunicaciones utilizado en el *log.*
- Length: longitud de la cadena del *log*, aquí se encuentra lo que ocupa y cuanto espacio está dedicado a cada una de las tramas del registro.
- SrcPort: puerto desde donde parte la petición.
- DestPort: puerto al que le llega la petición.
- SrcMac: dirección física del dispositivo desde donde se envía la petición.
- DestMac: dirección física del dispositivo que recibe la petición.
- Info: información general del *log*.

El campo extra, con información tratada que se encuentra de forma implícita en el *log,* está representado en la Tabla 11:

TCP – Control Flags

Tabla 11 Campo extra para el log tercera iteración

Dónde:

- Control Bits: muestra las banderas correspondientes al protocolo TCP si están activas o si no, éstas pueden ser: **RST, ACK, SYN, FIN, PSH** y **URG.**

➢ Matrices de los pixeles

A continuación, la Tabla 12 muestra el valor de las matrices de cada campo, la correspondiente a sus elementos y el número máximo de elementos a representar.

Campo	**Matriz del campo**	**Matriz del elemento**	**Nº de elementos en el campo**
No	36x26	3x13	24
Time	42x26	3x13	28
Source	21x56	7x7	24
Destination	21x56	7x7	24
Protocol	20x36	20x4	9
Length	9x36	9x9	4
SrcPort	20x36	15x7	6
DestPort	20x36	15x7	6
SrcMac	21x56	7x7	24
DestMac	21x56	7x7	24
Info	30x120	6x3	200
Control Bits	10x36	10x6	6

Tabla 12 Distribución de las matrices de la tercera iteración

Respecto a la anterior iteración se han añadido dos nuevos campos, esto ha provocado una restructuración en los elementos ya situados en la imagen. Se ha decidido que las dimensiones sean las mismas a IP Source e IP Destination ya que sirven para identificar las fuentes de emisión y recepción. Unas las direcciones asignadas por la red y otra por el dispositivo en cuestión que las realiza. También

se restructuraron las matrices de los elementos de estos dos campos para adecuarlas al nuevo tamaño.

- Resultado de la imagen

Según las características expuestas en los anteriores apartados, la propuesta correspondiente a la tercera iteración queda reflejada en la Figura 25:

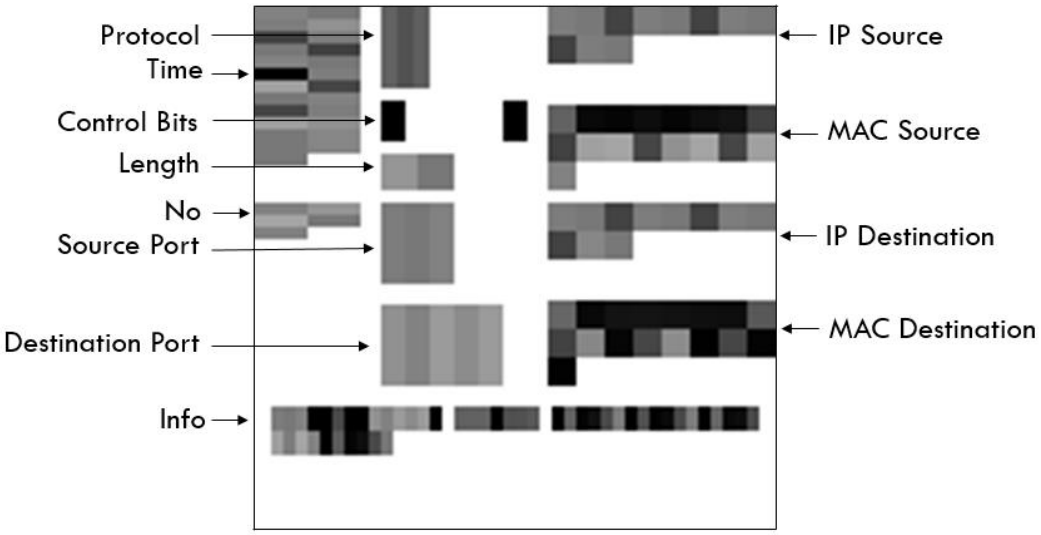


Figura 25 Propuesta de formato imagen tercera iteración

Como se puede observar, respecto a la iteración anterior, se ha reducido el tamaño correspondiente a los campos IP Source y IP Destination de manera que debajo se coloquen los relativos a las direcciones MAC. Se han agrupado por campos que emiten el registro y campos que lo reciben.

- Observaciones

Ya se tienen todos los campos representativos y necesario para un análisis de los registros (tanto normal como forense) estructurados dentro de la imagen. Con la adición de las direcciones MAC se podían ahora identificar otros como el ya mencionado en Arp poisoning / spoofing. En ese aspecto quedaban ya todos los elementos dispuestos para la conservación de la información y el análisis a través de este estándar. Lo que ocasionó la siguiente iteración fue la restructuración espacial de los elementos, para que de una manera más intuitiva el ojo humano pudiera cotejar y percatarse de las diferencias, principalmente en la disposición de los campos correspondientes a las direcciones IP y MAC.

3.3.5 CUARTA ITERACIÓN

- Campos de Wireshark

Para la cuarta iteración se han considerado los mismos campos correspondientes a la tercera de Wireshark. Se representan, a continuación, en la Tabla 13:

No	Time	Source	Destination	Protocol	Length	SrcPort	DestPort	SrcMac	DestMac	Info

Tabla 13 Campos de Wireshark cuarta iteración

Dónde:

- No: número del *log* en el que fue grabado.
- Time: fecha del evento, siguiendo el siguiente formato: "aaaa-mm-dd hh:mm:ss".
- Source: la dirección IP desde donde se realiza la petición.
- Destination: la dirección IP que recibe la petición.
- Protocol: protocolo de comunicaciones utilizado en el *log.*
- Length: longitud de la cadena del *log*, aquí se encuentra lo que ocupa y cuanto espacio está dedicado a cada una de las tramas del registro.
- SrcPort: puerto desde donde parte la petición.
- DestPort: puerto al que le llega la petición.
- SrcMac: dirección física del dispositivo desde donde se envía la petición.
- DestMac: dirección física del dispositivo que recibe la petición.
- Info: información general del *log*.

Además del campo extra que ya se tenía, se añade un segundo, que corresponde a más información, que se encuentra en el *log* de forma implícita, tratada para facilitar su análisis. Los dos campos se muestran representados en la Tabla 14:

TCP – Control Flags	TCP – Control Info

Tabla 14 Campos extra para el log cuarta iteración

Dónde:

- TCP – Control Flags: muestra las banderas correspondientes al protocolo TCP si están activas o si no, éstas pueden ser: **RST, ACK, SYN, FIN, PSH** y **URG.**
- TCP – Control Info: donde se representan ciertas condiciones que, al ser evaluadas, dependiendo de las banderas activadas y la longitud de la trama del *log* así como el contenido del campo Info, permiten un análisis

más rápido y preciso de otros campos dentro del protocolo TCP. Los parámetros que aquí se consideran son:

- **C→ Control:** Que adoptará el color negro si alguno de los tres siguientes está representado como negro o gris.
- **A→ Ack:** si ack = 0, adoptará el color negro, si ack = 1, entonces será gris. Para cualquier otro valor de ack estará en blanco.
- **S→ Seq:** si seq = 0, adoptará el color negro, si seq = 1, entonces será gris. Para cualquier otro valor de seq estará en blanco.
- **L→ Len:** si len = 0, adoptará el color negro, si len = 1, entonces será gris. Para cualquier otro valor de len estará en blanco.
- **T→ TCP:** si "TCP" aparece en la trama entonces adoptará el color negro, si no, será blanco.
- **D→ Dup:** si "Dup" aparece en la trama entonces adoptará el color negro, si no, será blanco.

➢ Matrices de los pixeles

A continuación, la Tabla 15 muestra el valor de las matrices de cada campo, la correspondiente a sus elementos y el número máximo de elementos a representar.

Campo	Matriz del campo	Matriz del elemento	Nº de elementos en el campo
No	36x26	3x13	24
Time	42x26	3x13	28
Source	21x56	7x7	24
Destination	21x56	7x7	24
Protocol	20x36	20x4	9
Length	9x36	9x9	4
SrcPort	14x36	14x6	6
DestPort	14x36	14x6	6
SrcMac	21x56	7x7	24
DestMac	21x56	7x7	24

Info	30x120	6x3	200
TCP – Control Flags	10x36	10x6	6
TCP – Control Info	8x36	8x6	6

Tabla 15 Distribución de las matrices de la cuarta iteración

Se pueden apreciarlos cambios más característicos respecto a la anterior iteración. Al añadirse otro campo de elementos analizados para el protocolo TCP, se cambió el nombre de Control Bits a TCP – Control Flags. Por otro lado, al querer añadir el nuevo campo de TCP – Control Infor, fue necesario disminuir las dimensiones del resto de campos correspondientes a esa parte de la estructura de la imagen. Ahí los afectados fueron los SrcPort y DestPort.

- Resultado de la imagen

Según las características expuestas en los anteriores apartados, la propuesta correspondiente a la cuarta iteración es la representada en la Figura 26:

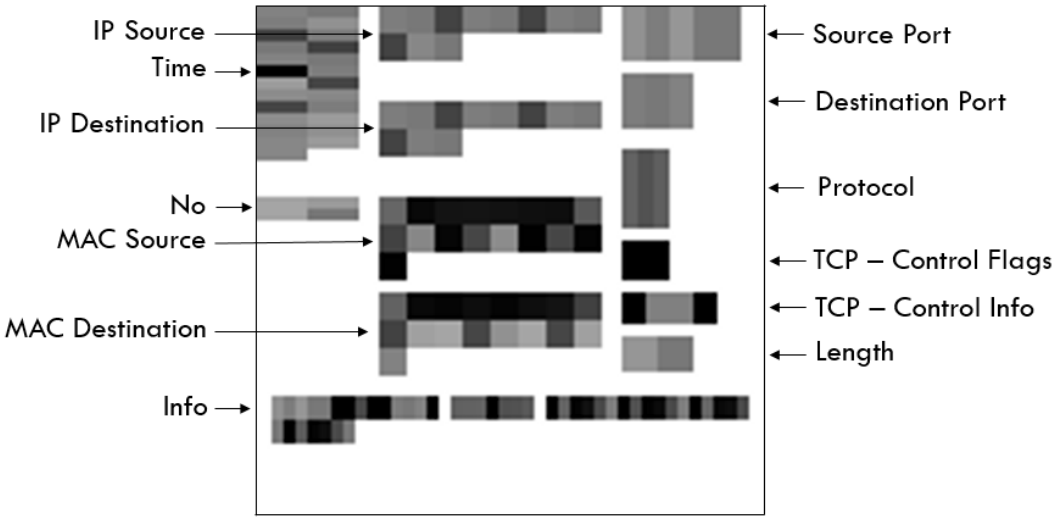


Figura 26 Propuesta de formato imagen tercera iteración

A nivel estructurar, se pueden observar cambios bastante representativos. Por un lado, se ha añadido el nuevo campo TCP – Control Info el cual se ha situado debajo del otro orientado al protocolo TCP. Aparte, la disposición de los campos principales se cambió, las direcciones (tanto IP como MAC) que estaban a la derecha se pasaron a la izquierda y en vez de estar IP Source seguida de MAC Source se han agrupado las IP por un lado y las MAC por otro.

➢ Observaciones

El haber cambiado la disposición de los elementos se ha conseguido que sea más sencillo el análisis visual. Se tienen, en el centro, las cuatro direcciones principales. Éstas además se encuentran agrupadas, por un lado las direcciones IP (una encima de otra) y por otro las direcciones MAC. A la derecha quedaría ahora el resto de información necesaria, los puertos, protocolo, los campos correspondientes el análisis del TCP y la longitud de la trama del *log.* El motivo que propició a realizar la última iteración fue doble. Se consideró que los elementos de los campos de IP y MAC no podían estar formados por las mismas matrices y que con el protocolo ICMP (otro de los principales protocolos mediante los cuales se realizan ataques) se podían sacar campos similares a los presentados para el TCP.

3.3.6 SOLUCIÓN FINAL

➢ Campos de Wireshark

Para la última iteración se han considerado los campos de Wireshark establecidos para la cuarta iteración, estos, además, se representan en la Tabla 16:

No	Time	Source	Destination	Protocol	Length	SrcPort	DestPort	SrcMac	DestMac	Info

Tabla 16 Campos de Wireshark Tercera iteración

Dónde:

- No: número del *log* en el que fue grabado.
- Time: fecha del evento, siguiendo el siguiente formato: “aaaa-mm-dd hh:mm:ss”.
- Source: la dirección IP desde donde se realiza la petición.
- Destination: la dirección IP que recibe la petición.
- Protocol: protocolo de comunicaciones utilizado en el *log.*
- Length: longitud de la cadena del *log*, aquí se encuentra lo que ocupa y cuanto espacio está dedicado a cada una de las tramas del registro.
- SrcPort: puerto desde donde parte la petición.
- DestPort: puerto al que le llega la petición.
- SrcMac: dirección física del dispositivo desde donde se envía la petición.
- DestMac: dirección física del dispositivo que recibe la petición.
- Info: información general del *log*.

A los campos extra que ya se tenían sobre el protocolo TCP con información implícita se le añaden tres campos correspondientes a información tratada (que se

encontraba en la trama del *log* de forma implícita también) sobre el protocolo ICMP. Los cinco campos correspondientes se muestran en la Tabla 17:

TCP – Control Flags	TCP – Control Info	ICMP – Control	ICMP – id	ICMP – seq

Tabla 17 Campos extra para el log solución final

Dónde:

- TCP – Control Flags: muestra las banderas correspondientes al protocolo TCP si están activas o si no, éstas pueden ser: **RST, ACK, SYN, FIN, PSH** y **URG.**
- TCP – Control Info: donde se representan ciertas condiciones que, al ser evaluadas, dependiendo de las banderas activadas y la longitud de la trama del *log* así como el contenido del campo Info, permiten un análisis más rápido y preciso de otros campos dentro del protocolo TCP. Los parámetros que aquí se consideran son:
  - **C→ Control:** Que adoptará el color negro si alguno de los tres siguientes está representado como negro o gris.
  - **A→ Ack:** si ack = 0, adoptará el color negro, si ack = 1, entonces será gris. Para cualquier otro valor de ack estará en blanco.
  - **S→ Seq:** si seq = 0, adoptará el color negro, si seq = 1, entonces será gris. Para cualquier otro valor de seq estará en blanco.
  - **L→ Len:** si len = 0, adoptará el color negro, si len = 1, entonces será gris. Para cualquier otro valor de len estará en blanco.
  - **T→ TCP:** si "TCP" aparece en la trama entonces adoptará el color negro, si no, será blanco.
  - **D→ Dup:** si "Dup" aparece en la trama entonces adoptará el color negro, si no, será blanco.
- ICMP – Control: campo que corresponde a determinados códigos de error y tipos del protocolo ICMP.
  - **Echo (ping) request →** ICMP Type 8
  - **Echo (ping) reply →** ICMP Type 0
  - **Destination unreachable →** ICMP Type 3
  - **No response found →** dentro del ICMP Type 8 cuando no se registra ninguna respuesta
- ICMP – id: se encarga de representar el valor del "id" correspondiente a este protocolo.
- ICMP – seq: se encarga de representar el valor de "seq" correspondiente a este protocolo.

➢ Matrices de los pixeles

A continuación, en la Tabla 18 se muestran los valores definitivos de las matrices de cada campo, las correspondiente a sus elementos y el número máximo de elementos a representar.

Campo	**Matriz del campo**	**Matriz del elemento**	**Nº de elementos en el campo**
No	36x26	3x13	24
Time	42x26	3x13	28
Source	20x50	10x5	20
Destination	20x50	10x5	20
Protocol	20x36	20x4	9
Length	9x36	9x9	4
SrcPort	14x36	14x6	6
DestPort	14x36	14x6	6
SrcMac	21x56	7x7	24
DestMac	21x56	7x7	24
Info	30x84	6x3	140
TCP – Control Flags	10x36	10x6	6
TCP – Control Info	8x36	8x6	6
ICMP – Control	16x36	4x36	4
ICMP – id	4x36	4x6	6
ICMP – seq	4x36	4x4	9

Tabla 18 Distribución de las matrices de la solución final

Aparte de la adición de los nuevos campos correspondientes al análisis de protocolo ICMP, cabe destacar que se han cambiado las dimensiones de los campos

de las direcciones IP, tanto Source como Destination. De esta manera ahora resultan diferentes a los de las direcciones MAC. Por otro lado, al añadir los nuevos y, teniendo en cuenta que el campo Info nunca llenaba los 200 elementos posibles, se ha restructurado éste para albergar 140.

- Resultado de la imagen

Según las características expuestas en los anteriores apartados, la propuesta correspondiente a la última iteración es la representada en la Figura 27:

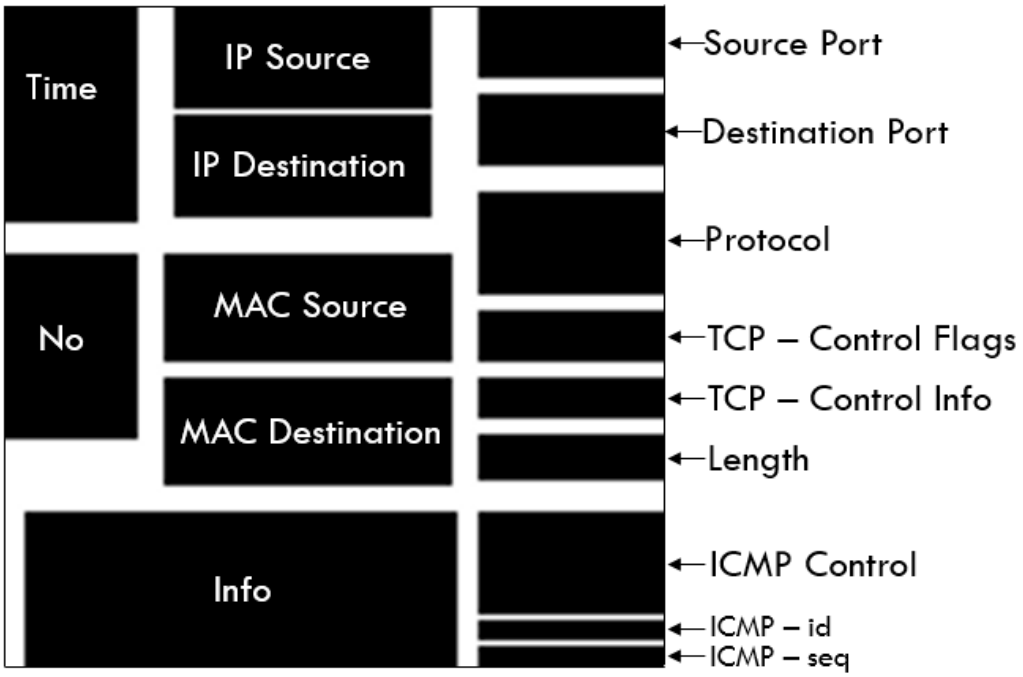


Figura 27 Propuesta de formato imagen solución final

Como ahora las imágenes representan información diferente dependiendo del protocolo que se esté evaluando, a continuación, se presentan dos imágenes pertenecientes al protocolo TCP y al ICMP. Figura 28 y Figura 29 respectivamente:

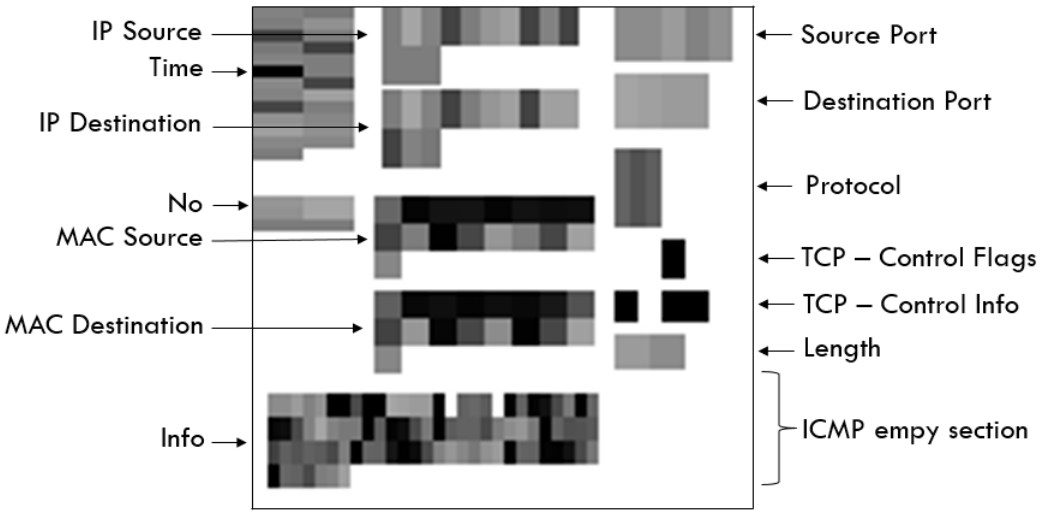


Figura 28 Solución final protocolo TCP

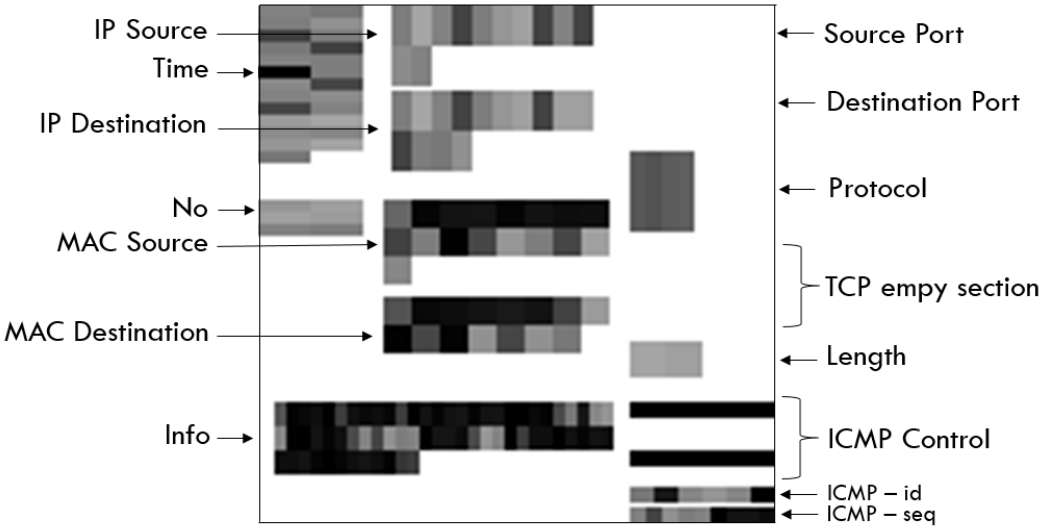


Figura 29 Solución final protocolo ICMP

La nueva ubicación de los campos correspondientes al análisis del protocolo ICMP queda a la derecha del campo de Info, el cual ve reducido su tamaño. Por otro lado ahora se distingue visualmente entre direcciones IP y direcciones MAC. A diferencia de lo que ocurre con el protocolo TCP, los campos del ICMP – Control están en vertical, por otro lado los de ICMP – id y ICMP – seq muestran su valor numérico en las imágenes.

- Observaciones

Con la incorporación de los campos de análisis del protocolo ICMP se abren las puertas a un mejor análisis tanto a nivel visual como a cotejar por parte de una máquina. Se incluyeron los ICMP – id y ICMP – seq porque si se observan de forma masiva (coincidiendo su valor) se estaría ante un posible ataque de denegación de servicio. ICMP – Control muestra también determinados tipos de errores dentro de este protocolo que al estar representados en las imágenes facilita mucho su análisis. El cambio dentro de las dimensiones de los campos de las direcciones IP (Source y Destination) también es relevante ya que dejan de tener el mismo tamaño que los correspondientes a las direcciones MAC, esto también permite facilitar el análisis tanto visual como operativo.

Capítulo 4 RESULTADOS Y ANÁLISIS DE LAS IMÁGENES

4.1 INTRODUCCIÓN

Después de explicar todo lo relacionado con la formación de las imágenes, y el proceso iterativo llevado hasta conformar la propuesta final del estándar, es necesario evaluar cómo este estándar gráfico sirve para analizar, tanto a nivel forense como normal, los diferentes registros que se pueden encontrar en el mundo de la ciberseguridad.

Para ello se seguirá el siguiente orden. En primer lugar, se mostrarán cada uno de los diferentes campos y se explicará cuál es su relevancia y su función, sobre todo cómo hacer efectivo el análisis visual con las imágenes. Poniendo especial atención en los protocolos ICMP y TCP ya que son los que tienen campos extras con información implícita ya analizada.

Después, en segundo lugar, se mostrarán los resultados de las imágenes, *logs* convertidos al nuevo formato estándar de distintos protocolos, para observar y detallar las características y peculiaridades de cada uno.

Tras estos, en el último apartado de este capítulo, se mostrarán cómo se pueden detectar diferentes tipos de ataques a partir de las secuencias de imágenes correspondientes. Aquellas que dan la alerta de un posible ataque o preparación de uno.

4.2 ANÁLISIS DE LAS IMÁGENES

A lo largo de esta sección se analizarán cada uno de los campos correspondientes a la propuesta en formato imagen. Se procederá primero mostrando la sección correspondiente al campo total (la matriz de píxeles que abarca el campo) y contrastando dos ejemplos de imágenes. El análisis más detallado corresponderá al protocolo TCP debido a que representan un 92% del volumen total de los *logs*. Luego, para poder analizar los campos que pertenecen al protocolo ICMP se cogerán dos imágenes y se contrastarán de igual manera.

4.2.1 IMÁGENES BASE

En la Figura 30 se muestra la distribución base del estándar propuesto. Es importante para poder contrastar con el par de ejemplos de *logs* ya convertidos en imágenes. Los dos registros convertidos en imágenes se muestran, a continuación, en la Figura 31 y corresponden a una transmisión.

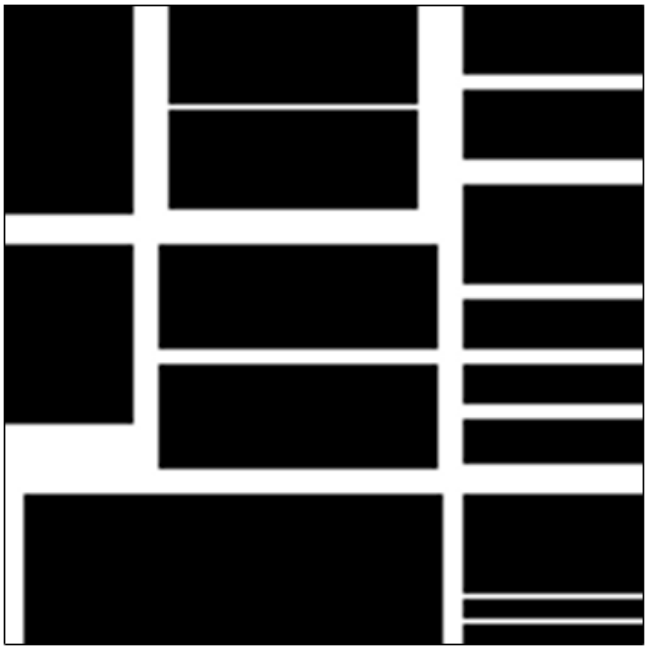
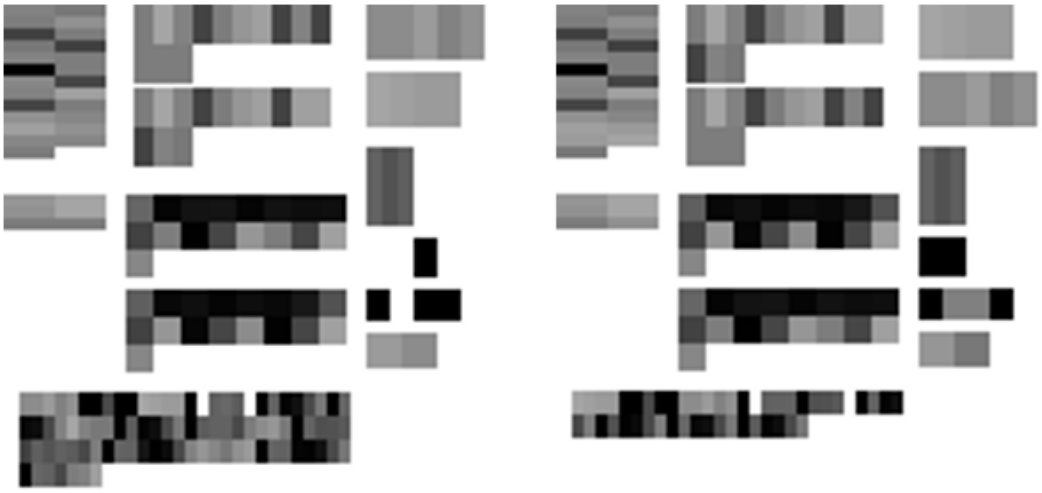
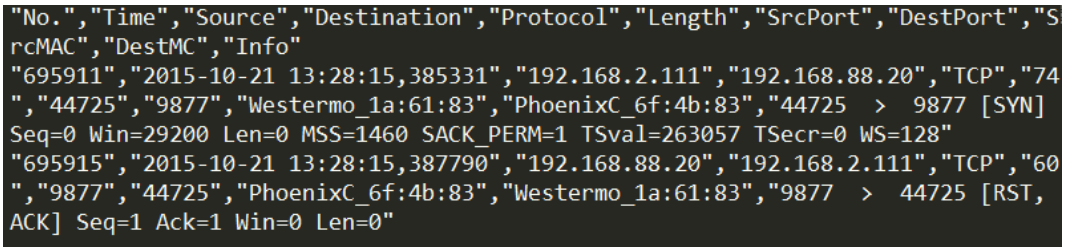


Figura 30 Imagen base del estándar





```
"No.","Time","Source","Destination","Protocol","Length","SrcPort","DestPort","S
rcMAC","DestMC","Info"
"695911","2015-10-21 13:28:15,385331","192.168.2.111","192.168.88.20","TCP","74
","44725","9877","Westermo_1a:61:83","PhoenixC_6f:4b:83","44725  >  9877 [SYN]
Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=263057 TSecr=0 WS=128"
"695915","2015-10-21 13:28:15,387790","192.168.88.20","192.168.2.111","TCP","60
","9877","44725","PhoenixC_6f:4b:83","Westermo_1a:61:83","9877  >  44725 [RST,
ACK] Seq=1 Ack=1 Win=0 Len=0"
```

Figura 31 Transmisión TCP convertida a formato imagen

Que las dos imágenes correspondan a una trasmisión quiere decir que se trata de dos registros correspondientes a una petición y a su respectiva respuesta. En este caso corresponde a una transmisión cuyo objetivo es realizar un escaneo de puertos, del tipo explicado en Syn scan / Stealth scan / Half open attack.

En los siguientes apartados se irán analizando cada uno de los campos, explicando su relevancia, para qué se usa y qué permite.

4.2.2 NÚMERO DEL *LOG* Y TIEMPO

A continuación, se evaluarán los No y Time, resaltados en la Figura 32, correspondientes al número del registro y a la fecha en la que fue grabado. Se contrastarán con los ejemplos prácticos de la Figura 33.

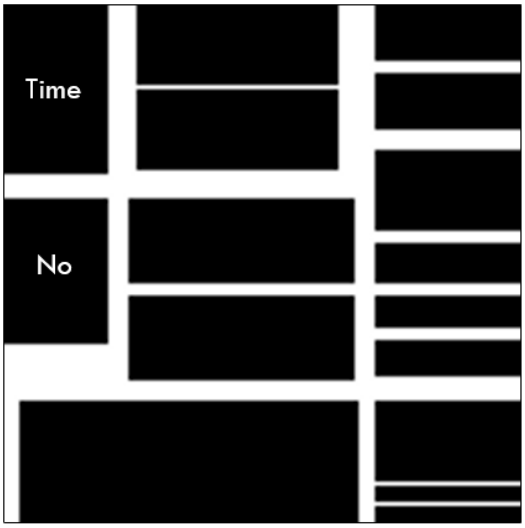


Figura 32 Imagen base de No y Time

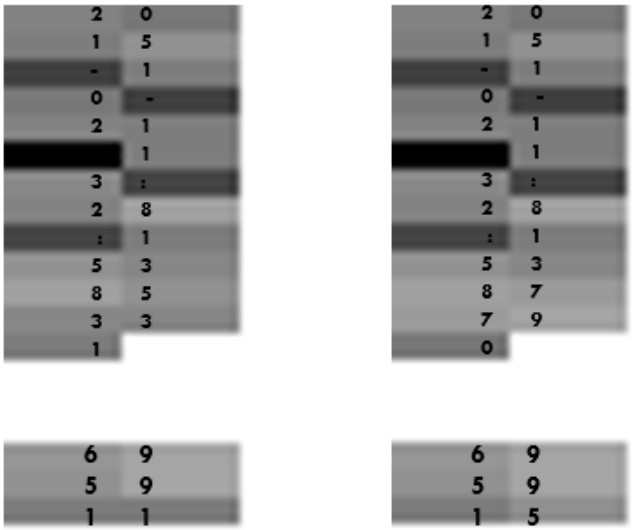


Figura 33 Campo de Num y Time de los dos registros

4.2.2.1 Relevancia del número del registro y del tiempo

La relevancia es la propia que indica el nombre. No indica el orden en el que han sido guardados los registros dentro del dataset. Esto es de gran importancia porque no es lo mismo que un *log* se dé antes o después, principalmente porque en las transmisiones, según el orden espacio temporal en el que se den los distintos eventos, importa.

Por otro lado, el campo Time ayuda a saber cuándo sucedió el evento. Sirve tanto para saber la diferencia de tiempos entre petición y respuesta como para localizarlo en el tiempo debido a un conjunto más global de sucesos.

4.2.2.2 Cómo analizarlas en las imágenes

Estos dos campos no tiene sentido estudiarlos por separado, es decir, sin contrastar con al menos otro registro. Ambos están situando el *log* en la escala de tiempos, tanto real (fecha, el campo Time), como relativa (el campo No con la numeración del *log* respecto al dataset).

Son importantes, pero sirven para permitir conocer el orden de los acontecimientos, son datos necesarios para ordenar y permitir empezar el análisis del dataset aunque visualmente la información es difícil de interpretar. A niveles prácticos, a mayor No y mayor Time los píxeles mostrados tienen un valor mayor luego si una máquina tuviera que ordenarlas, lo haría de la forma correcta e idéntica a como se encontraría en el dataset original. Esto es fundamental ya que, como se ha comentado antes, el orden en el que se suceden los eventos dentro de las transmisiones es vital para el análisis y la identificación de ataques y posibles amenazas.

4.2.3 DIRECCIÓN IP

A continuación, se evaluarán los campos IP Source y IP Destination, resaltados en la Figura 34, y se contrastarán con los ejemplos prácticos de la Figura 35.

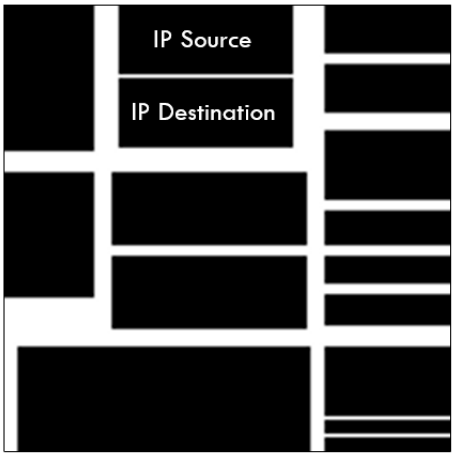


Figura 34 Imagen base direcciones IP

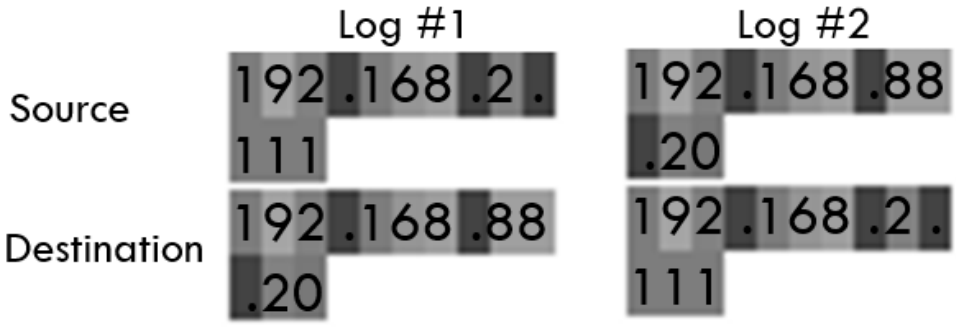


Figura 35 Campo de las direcciones IP de los dos registros

4.2.3.1 Relevancia de las direcciones IP

Una dirección IP es un número, éste identifica de manera lógica y jerárquica, a una interfaz en la red de un dispositivo. Es decir, es un identificador que otorga la red a los dispositivos conectados a ésta. Por ello la dirección IP puede cambiar debido a cambios en la red. según el número de direcciones máximas posibles que pueda albergar la red, se estará ante direcciones de clase A, B o C.

Se pueden expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal de cada octeto está comprendido en el intervalo de 0 a 255.

Básicamente con este campo se está identificando quién realiza la petición, y su objetivo, y quién responde, y su objetivo. Todo esto dentro del dominio de la red.

4.2.3.2 Cómo analizarlas en las imágenes

Como se puede observar en la Figura 35, se tienen las partes correspondientes a las direcciones IP de los dos registros de una transmisión. En primer lugar, comentar que en cada imagen quedan representadas la dirección IP de la fuente y la dirección IP de destino. Las direcciones se ven codificadas según el diccionario de conversión para representar en escala de grises los elementos correspondientes.

Otro punto a destacar es que se ve una correlación directa entre la IP Source del *log#1* y la IP Destination del *log#2*, así como las otras dos. Esto corrobora que, efectivamente, se trata de una transmisión en la cual el ordenador con IP 192.168.2.111 realiza una petición al servidor de IP 192.168.88.20 y luego éste último le responde.

Está claro que con sólo la IP no se puede identificar de forma absoluta que se trata de una transmisión cerrada (hacen falta más campos), pero sí sirve para saber qué dispositivos se están comunicando.

4.2.4 PUERTOS Y PROTOCOLO

A continuación, se evaluarán los campos Source Port, Destination Port y Protocol (puertos de enlace y el protocolo correspondiente), resaltados en la Figura 36Figura 34, y se contrastarán entre sí con los ejemplos prácticos de la Figura 37.

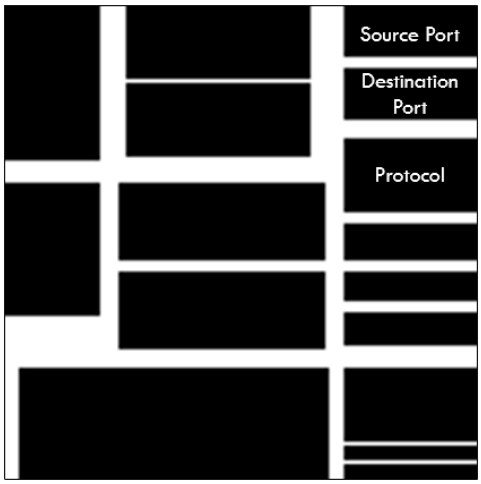


Figura 36 Imagen base Puertos y Protocolo

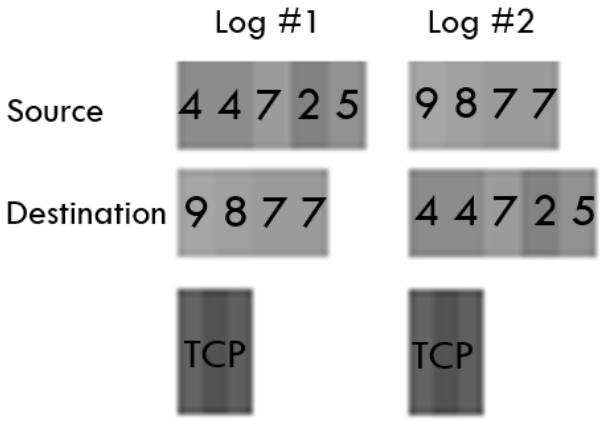


Figura 37 Campo de los puertos y el protocolo de los dos registros

4.2.4.1 Relevancia de los puertos y del protocolo

Cuando se habla de protocolo se hace referencia a la "familia de protocolos de internet", que es el conjunto de protocolos de red en los que se basa internet y que permiten la transmisión de datos entre ordenadores. Lo más importantes son el TCP (el de transmisión) y el IP (el de internet). Dependiendo de la capa del modelo OSI en la que se encuentre, se dan unos protocolos u otros. Algunos incluso están en varios niveles al mismo tiempo. En definitiva, el protocolo usado está definido por los límites para los que se diseñó dicho protocolo. Según ante que protocolo se esté,

se podrán esperar unas acciones u otras proviniendo de quién realiza la petición o de quien responde.

Por otro lado, los puertos de red son la interfaz a través de la cual se pueden enviar y recibir los diferentes tipos de datos. Al ser de red indican, del número 0 al 65535 (una palabra de 16 bits) el puerto en cuestión. Hay tres principales categorías relacionadas con el uso dado a cada conjunto de puertos.

- Aquellos inferiores al 1024 son puertos reservados para el sistema operativo y usados por protocolos comunes.
- Los comprendidos entre 1024 y 49151 son los denominados puertos registrados ya que pueden ser usados por cualquier aplicación. Existe una lista pública dónde se puede ver qué protocolo usa cada uno de ellos.
- De entre 49152 al 65535 están los puertos dinámicos o privados, estos normalmente se asignan en forma dinámica a las aplicaciones de clientes al iniciarse la sesión. Se usan en conexiones *peer to peer* (P2P).

4.2.4.2 Cómo analizarlos en las imágenes

La Figura 37 muestra, al igual que ocurría con las direcciones IP, cuatro campos que se relacionan dos a dos (la parte de los puertos). Por supuesto, la información está convertida a escala de grises, tal y como se puede apreciar. Por un lado, se tiene el puerto desde donde surge la petición y el puerto a donde se envía. En el *log#2* ocurre igual. El hecho de que los puertos coincidan dos a dos lleva a la conclusión de que se está ante una transmisión. La primera petición se realiza desde un puerto a otro, y la respuesta provendrá desde el que recibió la petición y la enviará al que la solicitó.

El análisis de puertos es de gran importancia porque no todos los puertos se utilizan para lo mismo. El analista experto sabe identificar cuando se realizan peticiones desde determinados puertos y con qué propósitos. Cuando se tienen un conjunto de peticiones, provenientes de la misma IP o incluso del mismo puerto, probablemente se esté ante un posible ataque de denegación de servicio. Al igual, si se tienen múltiples peticiones a distintos puertos, provenientes del mismo puerto e IP, se estará ante un escaneo de puertos.

Respecto al protocolo es de gran importancia su análisis por lo que se ha explicado anteriormente. Según el protocolo la configuración de las peticiones y las respuestas variará y por tanto los riesgos también cambiarán. Según el protocolo, se usarán unos puertos u otros lo cual, de cara a su análisis, es de gran importancia.

4.2.5 DIRECCIÓN MAC

A continuación, se evaluarán los campos MAC Source y MAC Destination, resaltados en la Figura 38, y se contrastarán con los ejemplos prácticos de la Figura 39.

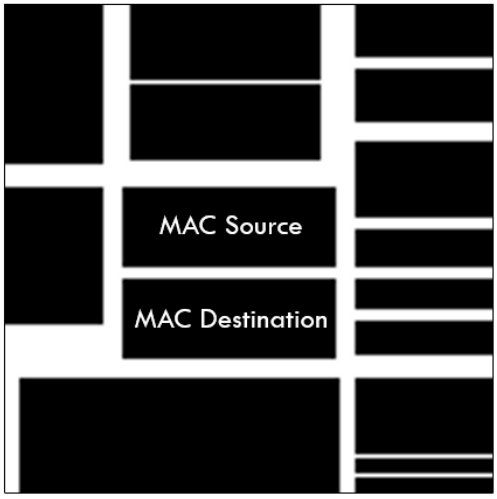


Figura 38 Imagen base direcciones MAC

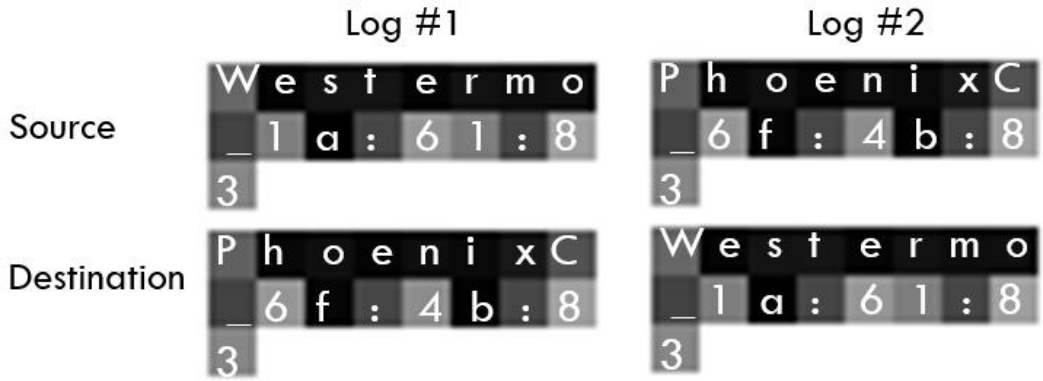


Figura 39 Campo de las direcciones MAC de los dos registros

4.2.5.1 Relevancia de las direcciones MAC

En las redes, la dirección MAC es un identificador, de 6 bloques de dos caracteres hexadecimales, que corresponde de forma única a una tarjeta o dispositivo de red. También suele denominarse como dirección física y es única para cada dispositivo. Es importante destacar que no todos los protocolos de comunicación usan direcciones MAC y no todos los protocolos requieren identificadores únicos.

Son únicas a nivel mundial, ya que son escritas directamente, en forma binaria, en el *hardware* en su momento de fabricación. En la mayoría de los casos no es

necesario conocer la dirección MAC, usándose ésta sólo a niveles internos de la red.

A pesar de que cada dispositivo de red tiene una dirección MAC única globalmente que lo identifica y lo hace único, es la capa del sistema operativo la que gestiona y distribuye la red. Debido a esto es posible modificar la dirección MAC que identifica la interfaz de red.

4.2.5.2 Cómo analizarlas en las imágenes

En la Figura 39 se puede observar una distribución muy similar a la que se trató con las direcciones IP. Se tiene una dirección MAC desde donde surge la petición y otra de destino. Ambas han de ser diferentes si se trata de una comunicación entre dos dispositivos diferentes. Al tratarse de una conversación de nuevo surge esa correlación dos a dos. Como se puede ver, ya no tienen por qué ser números, puesto que la codificación es en hexadecimal (lo cual abre la vía a identificadores únicos como marcas, etc).

El análisis de las direcciones MAC es de vital importancia ya que permite descubrir muchos casos de IP *spoofing*. En los que un determinado dispositivo adopta IPs que no le corresponden. Aquí se detectaría fácilmente pues todas las peticiones se estarían haciendo desde la misma dirección MAC. También cuando se está ante el protocolo ARP, se sabe que hay que prestar especial atención a este campo por lo mismo, o incluso con el ICMP (y el anterior) para detectar *man-in-the-middle-attacks*. Puesto que la dirección física no se puede modificar una vez que se está conectado a la red, cosa que con la dirección IP sí es posible.

Detectar que, de manera fiable, se están recibiendo múltiples o masivas peticiones por parte del mismo dispositivo, es un indicativo claro de que algo está sucediendo. Además, de conocerse la dirección física del atacante, ésta se puede bloquear, expulsándolo de la red.

4.2.6 LONGITUD DE LA TRAMA Y CAMPOS DE CONTROL TCP

A continuación, se evaluarán los campos correspondientes al análisis del protocolo TCP (TCP – Control Flags y TCP – Control Info) y a la longitud de la trama (Length), resaltados en la Figura 40, y se contrastarán con los ejemplos prácticos de la Figura 41.

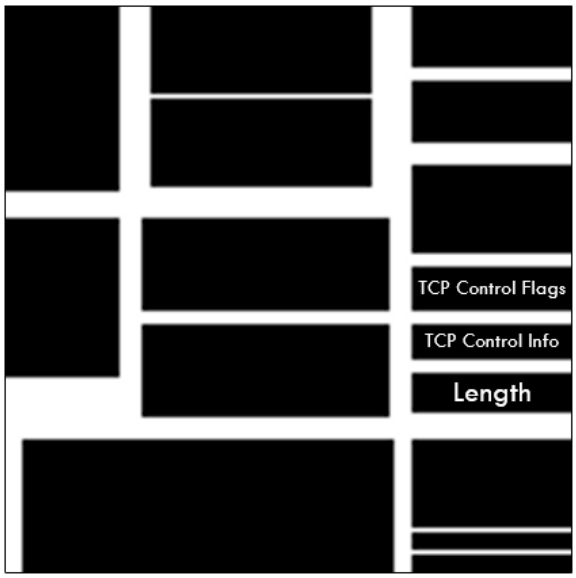


Figura 40 Imagen base Length y TCP – Control

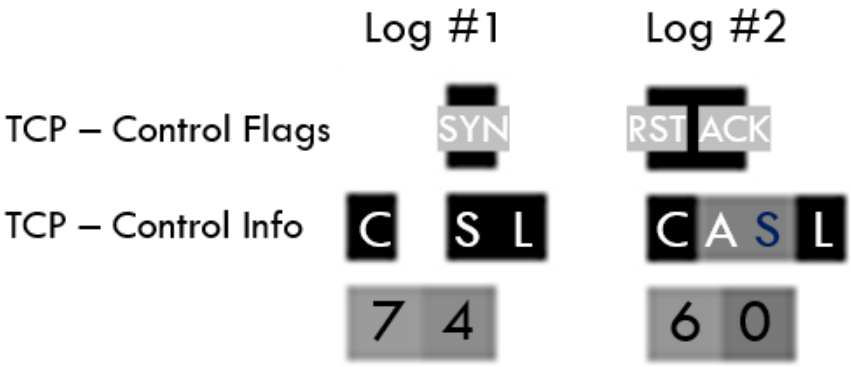


Figura 41 Campos de Length y TCP – Control

4.2.6.1 Relevancia de la longitud de la trama y de los campos de control TCP

La longitud de la trama es un número, que está en bytes, y representa lo que ocupa la trama en cuestión. Según el protocolo la trama o *frame* se rellena según unos parámetros que ocupan unos determinados bits y bytes. La Tabla 19 muestra como estaría distribuida una trama genérica del protocolo ICMP:

Ethernet Header (bytes)	IP Header (bytes)	ICMP Header (bytes)	ICMP payload size (bytes)	MTU (1514) (bytes)
14	20	8	1472 (max)	14+20+8+1472 = 1514

Tabla 19 Trama de un paquete ICMP

Por otro lado, los campos de control TCP, al ser información implícita ya analizada y representada directamente en las imágenes cobran gran importancia de cara a la inspección visual para este protocolo. A continuación, se repasan los contenidos que muestran estos dos campos:

- TCP – Control Flags: muestra las banderas correspondientes al protocolo TCP si están activas o si no, éstas pueden ser las representadas en la Tabla 20, conservando la disposición ahí descrita**.**

TCP – Control Flags	RST	ACK	SYN	FIN	PSH	URG

Tabla 20 Distribución del campo TCP – Control Flags

- TCP – Control Info: donde se representan ciertas condiciones que, al ser evaluadas, dependiendo de las banderas activadas y la longitud de la trama del *log* así como el contenido del campo Info, permiten un análisis más rápido y preciso de otros campos dentro del protocolo TCP. Los parámetros que aquí se consideran son:
  - **C→ Control:** Que adoptará el color negro si alguno de los tres siguientes está representado como negro o gris.
  - **A→ Ack:** si ack = 0, adoptará el color negro, si ack = 1, entonces será gris. Para cualquier otro valor de ack estará en blanco.
    Ack hace referencia a *Acknowledgement* y el número que adopta dependerá de la secuencia asignada del paquete enviado
  - **S→ Seq:** si seq = 0, adoptará el color negro, si seq = 1, entonces será gris. Para cualquier otro valor de seq estará en blanco.
    Aeq hace referencia a *Sequence number*, y viene a indicar cuando "comienza" la secuencia (de manera relativa) para la transmisión.
  - **L→ Len:** si len = 0, adoptará el color negro, si len = 1, entonces será gris. Para cualquier otro valor de len estará en blanco.
    Len hace referencia a *length* y es la longitud de esta parte correspondiente de la trama.
  - **T→ TCP:** si "TCP" aparece en la trama entonces adoptará el color negro, si no, será blanco.

TCP hace referencia a si se está produciendo una retransmisión TCP ya sea por un intento de sesión o por una ya establecida.

- **D→ Dup:** si "Dup" aparece en la trama entonces adoptará el color negro, si no, será blanco.
  Dup lo que quiere señalar es que se están dando duplicados dentro de la trama, Dup de *duplicate*. Suelen ser un indicativo de paquetes fragmentados y mal reensamblados.

La Tabla 21 muestra cómo están representados cada uno de los campos en la estructura de las imágenes:

TCP – Control Info	Ctrl	ack	len	seq	TCP	Dup

Tabla 21 Distribución del campo TCP - Control Info

4.2.6.2 Cómo analizarlas en las imágenes

La longitud de la trama se analiza de la siguiente manera: en primer lugar, para detectar paquetes que han sido enviados con el mismo propósito. Es decir, si se tienen una gran cantidad de peticiones desde la misma dirección IP o MAC y todos con la misma longitud de trama, se tratará de algún tipo de ataque Enel que los paquetes se habrán generado artificialmente.

Otro caso, por ejemplo, sería el tratado en la Sección 2.7.2.5, el *Ping of Death*, el cual consiste en sobrecargar la longitud de la trama para provocar una denegación de servicio ya que supera la longitud máxima posible para el protocolo de comunicaciones. Además de esto, si en determinadas acciones como *echo ping request* y *echo ping reply* se observa que su trama correspondiente ocupa más de lo que debería, pudiera estar ocurriendo un Icmp tunneling y encubriendo una comunicación no autorizada. Por estas cosas, además de la relación directa que tienen el campo Length con el protocolo y las direcciones, es un punto importante para asegurar la normalidad de la comunicación.

En el caso que se muestra en la Figura 41, las longitudes de las tramas son coherentes con la información que se transmite, tanto en la petición como en la respuesta. Se trata de un escaneo de puertos así que contiene la longitud necesaria para intentar establecer la conexión con el servidor.

Analizando los campos correspondientes al protocolo TCP en primer lugar se miran las banderas activas de éste. En el log#1 se tiene **SYN**, lo cual ya da indicios que pudiera ser un ejemplo de un Syn scan / Stealth scan / Half open attack. Al observar las banderas de la respuesta del servidor: **RST, ACK**, no cabe ninguna duda que se trata justamente de este tipo de escaneo de puerto, en el cual le está diciendo, de forma implícita, que está cerrado.

El análisis de este campo es de gran importancia ya que con un breve vistazo el ojo humano puede detectar fácilmente si se dan la combinación apropiada para distinguir entre un tipo de ataque u otro. Por otro lado, si se reciben de manera masiva peticiones desde la misma IP/MAC con la bandera **SYN** (como es el ejemplo aquí descrito), se tratará de un Syn flood attack.

El segundo campo, el correspondiente al TCP – Control Info, sirve principalmente para dar información extra de lo que está ocurriendo en la transmisión. En la petición por parte del atacante, en su trama, el campo correspondiente a "seq" y "len" valen:

$$Seq = 0, \qquad Len = 0$$

Esto es importante porque en el protocolo TCP cuando se quiere iniciar una sesión con el servidor en cuestión, han de valer cero. Si el puerto no hubiera estado cerrado, y le hubiera respondido con las banderas **SYN y ACK,** los campos hubieran valido lo siguiente:

$$Seq = 0, \qquad Ack = 1, \qquad Len = 0$$

Puesto que la conexión hubiera sido establecida. Al responder con **RST, ACK** los valores adoptados son los siguientes:

$$Seq = 1, \qquad Ack = 1, \qquad Len = 0$$

El valor de la secuencia relativa aumenta en 1, ya que es la contestación (refiriéndose a que no se ha establecido la conexión) respecto a la petición original, ack también adopta 1 ya que ha recibido el mensaje y ha de ir acorde al valor de la secuencia y len sigue siendo cero ya que sigue sin transmitirse nada.

Por estas cosas hay que destacar que ambos campos de control para el protocolo TCP ayudan a distinguir de forma rápida y efectiva la situación que se está produciendo dentro de las comunicaciones de este protocolo. Tanto a nivel de los bits de control con las banderas como con la información implícita expuesta de forma clara en la representación de las imágenes.

4.2.7 CAMPOS DE CONTROL DEL PROTOCOLO ICMP

Para poder evaluar este campo es necesario estar frente a un registro cuya comunicación fuera mediante el protocolo ICMP. Por ello se mostrarán los pertenecientes a otra imagen, que es la mostrada en la Figura 42.

Figura 42 Log perteneciente al protocolo ICMP

A continuación, se evaluarán los campos pertenecientes al análisis del protocolo ICMP, resaltados en la Figura 43, y se contrastarán con el ejemplo prácticos de la Figura 44.

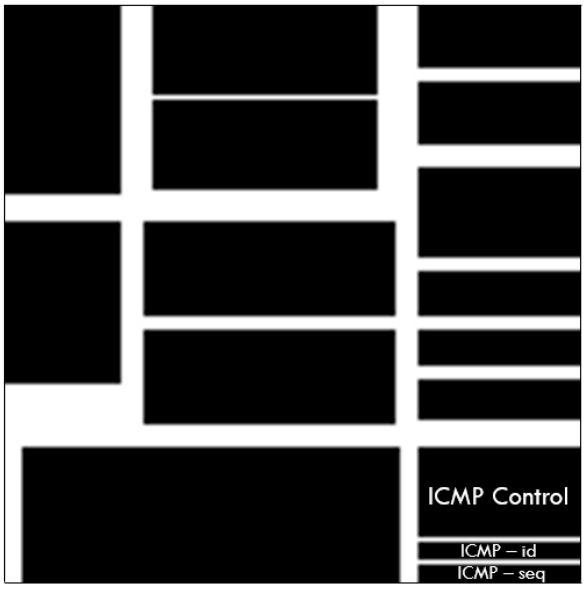


Figura 43 Imagen base direcciones MAC

Figura 44 Campos de análisis ICMP de un registro

4.2.7.1 Relevancia del campo de análisis ICMP

Este campo, al haberse añadido tras estudiar y, en cierto modo, diseccionar el propio protocolo, contiene información de gran utilidad para detectar y analizar sus registros. La información que queda representada se mostrará a continuación:

- ICMP – Control: campo que corresponde a determinados códigos de error y tipos del protocolo ICMP.
  - **Echo (ping) request** → ICMP Type 8
  - **Echo (ping) reply** → ICMP Type 0
  - **Destination unreachable** → ICMP Type 3
  - **No response found** → dentro del ICMP Type 8 cuando no se registra ninguna respuesta
- ICMP – id: se encarga de representar el valor del "id" correspondiente a este protocolo.
  Es el número de la secuencia de identificación perteneciente al paquete ICMP *echo*, tanto *request* como *reply.* Viene representado por dos números *little endian* y *big endian byte order*, LE/BE.
- ICMP – seq: se encarga de representar el valor de "seq" correspondiente a este protocolo.
  Es el número de la secuencia que el paquete ICMP *echo* tiene, viene representado por dos números separados por una "/" estos son *little endian* y *big endian byte order.* Es decir: LE/BE. Hacen referencia al número de secuencia en el que se almacenan en la memoria.

En la siguiente Tabla 22 se muestra cómo son representados estos dentro de las imágenes:

ICMP – Control	Echo (ping) request
	Echo (ping) reply
	Destination unreachable
	No response found
ICMP – id	LE / BE
ICMP – seq	LE / BE

Tabla 22 Distribución de los campos de análisis ICMP

4.2.7.2 Cómo analizarlas en las imágenes

Si se observa la Figura 44 junto con la Figura 42 se puede ver cómo se ha realizado un ICMP *echo ping request*, el cual también lleva el concepto de *Destination unreachable*, por las líneas en negro dentro del campo de análisis del protocolo ICMP. Esto, puede ser indicativo de paquetes duplicados. Al haber contestado ya el primer paquete el servidor no encuentra respuesta posible para el resto. Esto se puede observar si se tienen múltiples peticiones de ICMP *echo ping request* con el mismo seq e id, lo cual estaría significando que se está perpetrando un ataque de denegación de servicio queriendo agotar los recursos del sistema objetivo.

Los campos pertenecientes para analizar este protocolo están escogidos para representar de manera rápida y visual los tipos de ICMP y códigos de error que sirven de mayor indicación a la hora de detectar ataques. Los más conocidos ya se han comentado en la Sección 2.7.2. Los dos representantes a los números de la secuencia e identificación son importantes para tener información representativa perteneciente a este protocolo, así como para identificar otros.

4.3 ATAQUES VISUALIZADOS CON LAS IMÁGENES

A continuación, se mostrarán algunos ejemplos de ataques visualizados con el estándar de las imágenes. Se han elegido los siguientes, un escaneo de direcciones IP, un escaneo de puertos, una preparación de *IP spoofing* o *man-in-the-middle* y una denegación de servicios.

4.3.1 ESCANEO DE DIRECCIONES IP

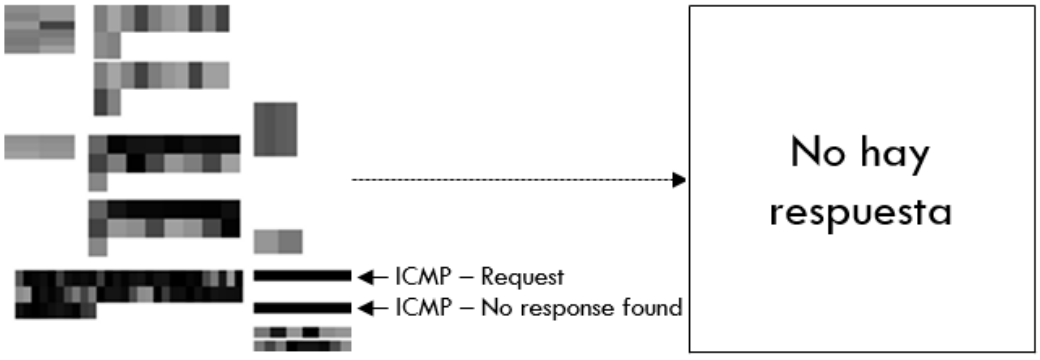


Figura 45 Log número 546586

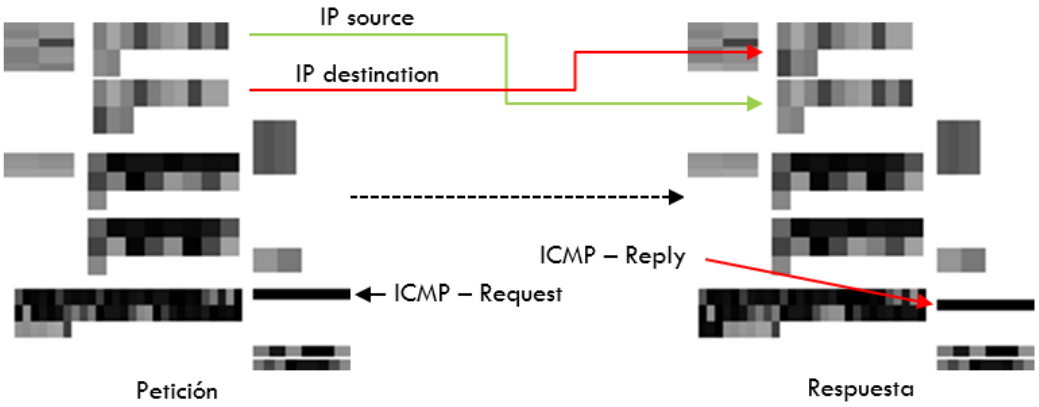


Figura 46 Logs números 546588 y 564589 respectivamente

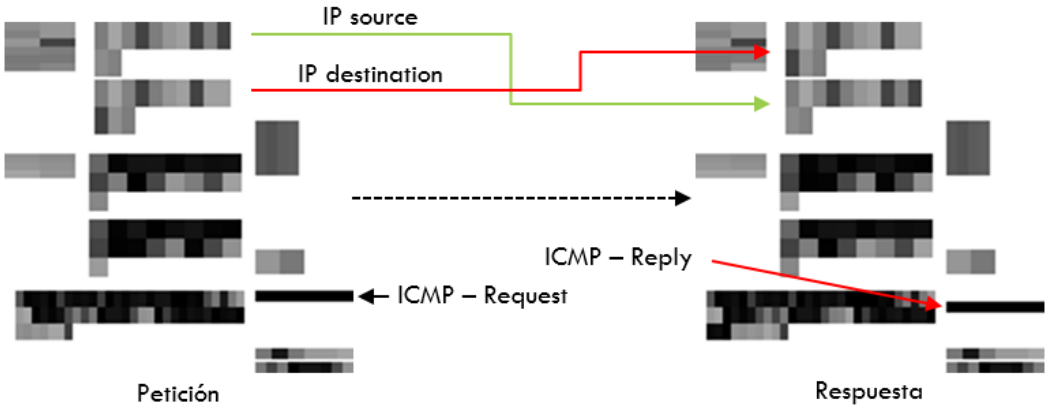


Figura 47 Logs números 546596 y 564599 respectivamente

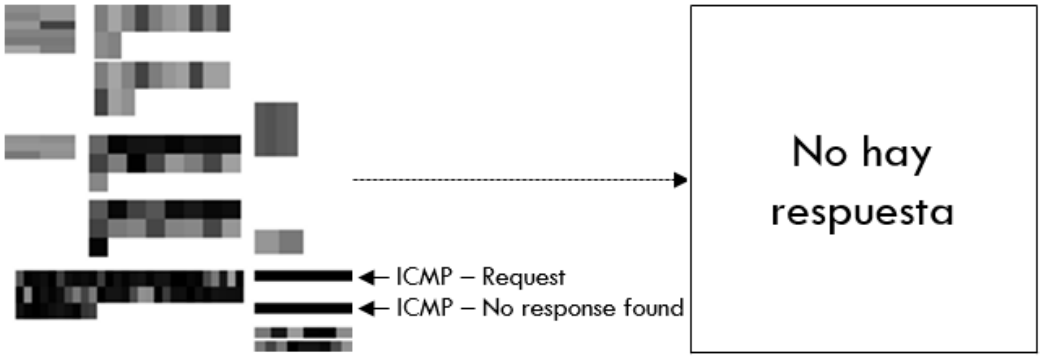


Figura 48 Log número 546605

Como se puede observar, desde las Figura 45-Figura 48, se está ante un caso de ICMP Scan. En este tipo de ataque lo que se busca es encontrar las IP disponibles dentro de una red. Por ello, tal y como se ha explicado en la Sección 2.7.2.1, es de gran importancia la respuesta del servidor. Si responde significará que está disponible y si no lo hace es que entonces no es válida.

4.3.2 ESCANEO DE PUERTOS

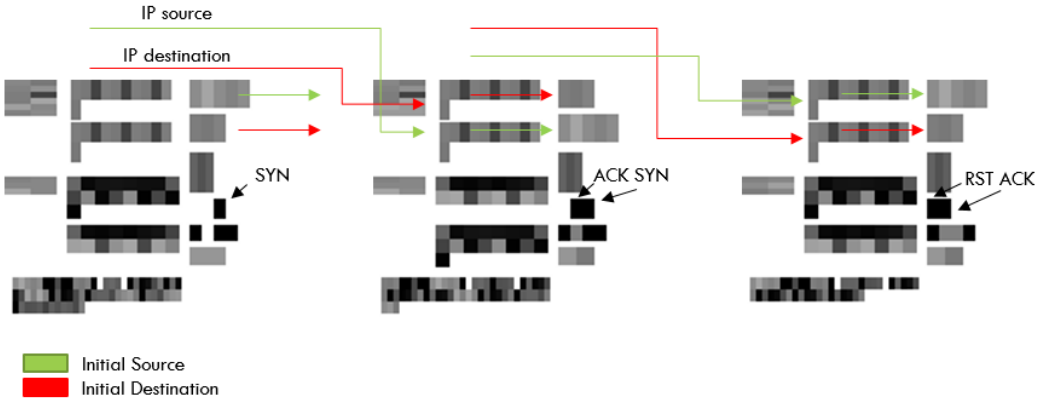


Figura 49 Logs números 233393, 233394 y 234813 respectivamente

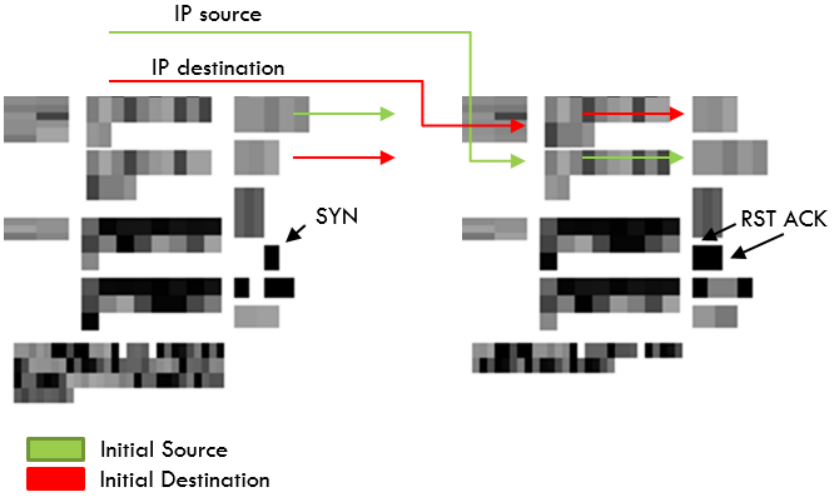


Figura 50 Logs números 458503 y 458508 respectivamente

Las Figura 49 y Figura 50 muestran el ataque ya estudiado de Syn scan / Stealth scan / Half open attack, en la Sección 2.7.1.5. Se trata de un caso de escaneo de puertos, en la primera imagen se muestra la transmisión completa que indicaría que el puerto correspondiente está abierto. La transmisión transcurre siguiendo las flechas verdes (la fuente que origina la primera petición) y rojas (la fuente de destino del primer evento), esto ocurre tanto para las direcciones IP y puertos de origen y destino. Al tratarse del protocolo TCP lo siguiente sería ver qué bits de control están activados, en la primera imagen se tiene **SYN**, en la segunda **ACK-SYN** y en la tercera **RST-ACK**; completando así este tipo de escaneo de puertos conociendo que éste está abierto.

Por otro lado, en Figura 50 lo que se obtiene es que el puerto está cerrado. A la petición de **SYN** el servidor responde con **RST-ACK**; terminando la transmisión.

4.3.3 PREPARACIÓN DE IP SPOOFING O *MAN-IN-THE-MIDDLE*

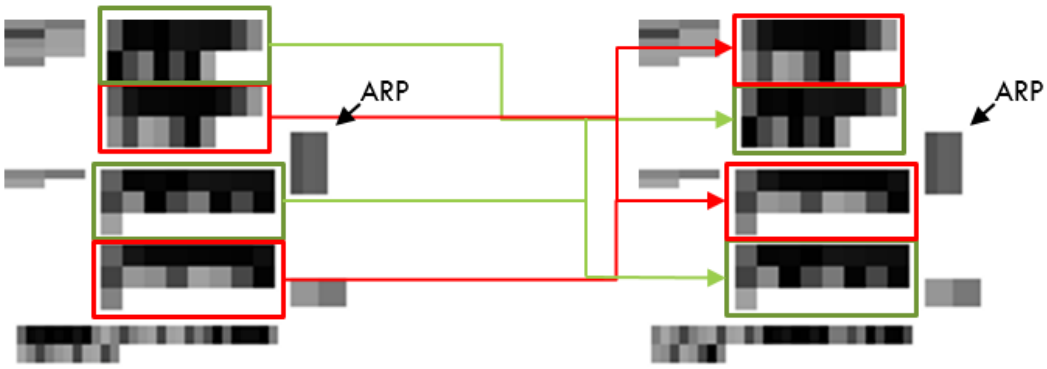


Figura 51 Logs números 308 y 309 respectivamente

Las dos imágenes que se pueden observar en la Figura 51 muestran una petición con su respectiva respuesta del protocolo ARP. En ésta lo que hace es preguntarle al servidor la IP de uno de los dispositivos. Esto se puede reconocer en las imágenes observando que las direcciones IP y direcciones MAC de origen y destino coinciden entre ellas (los recuadros verdes por un lado, y los recuadros rojos por otro). Otro indicador de que se trata de una preparación para obtener la IP de un dispositivo es que en estas transmisiones se ve que no se han utilizado puertos. Que la dirección IP de origen coincida con la dirección MAC de origen no es algo común salvo en estos casos.

4.3.4 DENEGACIÓN DE SERVICIO

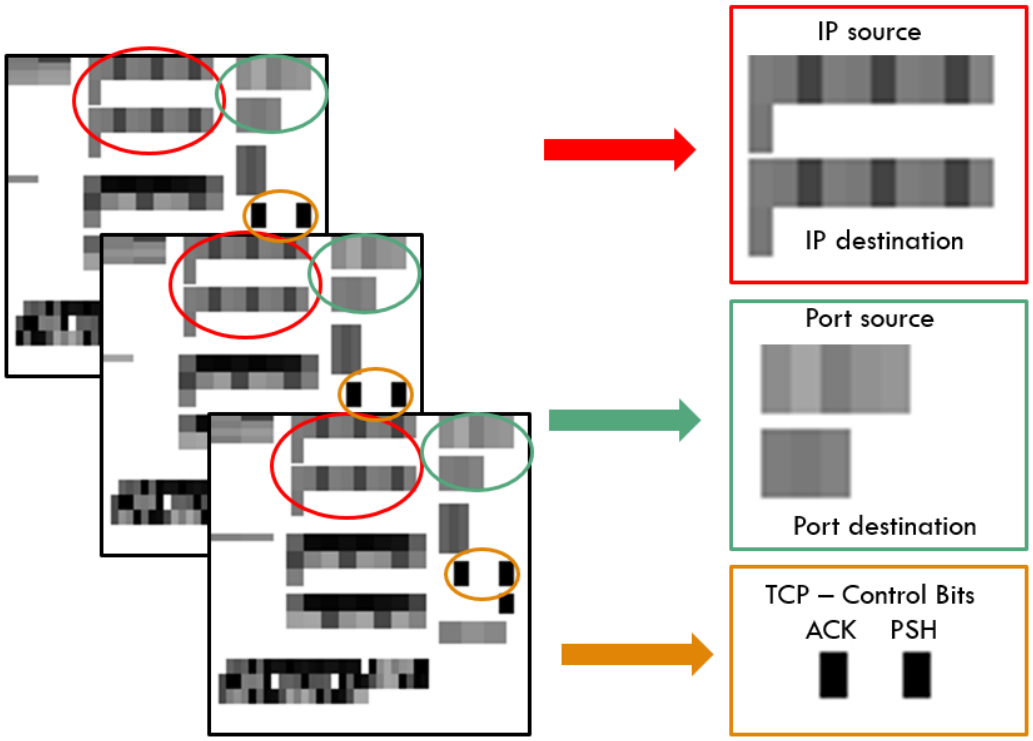


Figura 52 Logs números 8,9 y 10 respectivamente

La Figura 52 muestra las imágenes correspondientes a tres registros que, junto con otros más, forman un ataque de denegación de servicio de tipo Psh-Ack flood. Los indicadores para saber que se está dando este tipo son los siguientes: misma dirección IP enviando múltiples transmisiones en cortos periodos de tiempo, estas transmisiones se realizarán al mismo puerto (pueden originarse del mismo también o no, dependiendo del atacante) y, por último, las banderas **ACK-PSH** han de estar activadas. Tal y como se ha representado en la figura, en los tres registros se envían peticiones desde la misma dirección IP a la misma dirección IP, desde el mismo puerto de origen hasta el mismo puerto de destino, y todas con las dos banderas de control representadas en las imágenes **ACK-PSH.** El objetivo de este ataque es saturar el puerto atacado para así realizar la denegación de servicio.

Capítulo 5 CONCLUSIONES

5.1 INTRODUCCIÓN

Dentro de este capítulo resulta necesario subdividir dos grandes grupos de conclusiones debido a la dualidad de este proyecto.

- El primero engloba las conclusiones referidas al análisis del dataset. Tanto referidas al análisis normal como al análisis forense del conjunto de registros de carácter industrial para la identificación de posibles amenazas y ataques. Aquí se englobará una valoración crítica de lo que esto ha supuesto y las limitaciones que se han encontrado.

- El segundo contiene las conclusiones respecto al estándar en formato imagen propuesto, de carácter más técnico. Referidas siempre en torno a la motivación principal de este proyecto para poder realizar una valoración crítica acerca de lo obtenido. Por lo tanto, se nombrarán los detalles más fundamentales que afectarían de forma directa a estas imágenes y a su trabajo y desarrollo con ellas.

5.2 CONCLUSIONES SOBRE EL ANÁLISIS FORENSE

- El conjunto de datos del cual se ha hecho el análisis forense pertenece a los paquetes capturados durante la convención de 4SICS GeekLounge. El propósito de esta era perpetuar ataques a un conjunto de servidores y dispositivos industriales. Por esto probablemente se pueda inferir que en cierto modo estos estén sesgados. No se trata de una situación normal, sino de una en la cual personas formadas tenían como objetivo superar las barreras y penetrar en el sistema. Teniendo en cuenta el tipo de dispositivos que se utilizaban, mencionados al final de la Sección 2.3, los registros guardados de los protocolos utilizados deberían estar fuertemente influenciados por esto, habiendo dejado sin tratar otros posibles ataques si se tratara de un dataset de otra índole. Para sustentarse en esto simplemente se puede mencionar que el porcentaje de *logs* referidos a HTTP no llegaba al 3% de los totales.

- De los 25 protocolos diferentes encontrados en el conjunto de datos (Sección 2.3), se han encontrado ataques de 6: TCP, ICMP, ARP, UPD, DNP3 y FTP. De estos, los principales son TCP, con 11 tipos diferentes, e ICMP, con 7 tipos. Teniendo en cuenta que los *logs* correspondientes al protocolo TCP e ICMP suman el 93% del volumen total, se podría decir que a mayor volumen mayor número de amenazas y de ataques. De esta manera, si se hubiera dispuesto de un dataset con más muestras del resto, muy probablemente se hubieran identificado otros ataques pertenecientes a los demás protocolos.

- Aunque a lo largo de la realización de este proyecto se hayan tratado los protocolos por separado (agrupándolos por estos) en una situación real todos están interrelacionados ya que según el nivel de la capa del modelo OSI, se usan unos u otros. Al estudiarlos por separado, se pierde parte de la visión global, aunque se facilite su análisis. Es importante tener en cuenta que distintos protocolos pueden interactuar con diferentes niveles lo cual hace que entren en juego factores con contemplados si se estudia el conjunto de datos aislándolos por protocolos.

- El análisis se ha realizado utilizando el programa capturador de paquetes Wireshark, gracias a sus reglas de coloración y a posibilidades en cuanto

al filtrado, se han conseguido detectar una gran variedad de amenazas y distinguir situaciones que de haber sido con otro analizador de paquetes hubiera sido mucho más complicado o incluso imposible. Poder filtrar por direcciones IP de envío o puerto de destino permite detectar ataques de denegación de servicio o posibles escaneos de puertos. Filtrar por dirección MAC ha servido para descubrir ataques de *spoofing* como el Arp poisoning / spoofing de la Sección 2.7.3.1. Por otro lado, los *logs* al seguir transmisiones, es posible que algunas (debido al volumen de registros y su rápida sucesión) no fueran capturadas por el *packet sniffer* que las grababa y se perdieran. De manera que algunos registros, podrían estar incompletos.

5.3 CONCLUSIONES GENERALES SOBRE LAS IMÁGENES

- El estándar en formato imagen permite de forma rápida la distinción entre protocolos. Se puede detectar si ocurre algo extraño con una simple visualización. Tal y como se ha ido diseccionando cada campo en la Sección 4.2: Análisis de las imágenes. Con éstas se pueden revisar las direcciónes IP y MAC, así como el puerto de destino, peticiones de la misma a otras, e incluso una misma MAC, con distinta IP dirigiendo peticiones a múltiples puertos pertenecientes a la misma IP. Casos claros de amenazas ante escaneos de puertos, ataques de denegación de servicio, suplantación de identidad (*spoofing*) o *man-in-the-middle-attack.*

- Al haberse incluido campos especiales como los referentes a los protocolos TCP/ICMP, de forma visual se pueden sacar ahora conclusiones rápidas en poco tiempo, así como al estar incorporadas en la estructura de las imágenes también para su análisis a nivel numérico. Es importante destacar que se escogieron estos dos protocolos, y su respectiva información implícita en la trama de los *logs*, debido a su elevado volumen en el conjunto de datos. Si se hubiera dispuesto de un dataset con distintos tipos de registros y lo antes mencionado hubiera cambiado, probablemente se habrían desarrollado otros campos de análisis para representar en las imágenes.

- El formato de las imágenes, UINT-8 de 128x128 píxeles, es capaz de representar todo lo que se ha planteado sin presentar ningún tipo de limitación de espacio. A la hora de guardarlas y analizarlas, aunque su tamaño sea superior a si se trabajara con sus correspondientes en .csv, su media de 2.5KB permite su masivo tratamiento y almacenamiento, así como transporte.

- La conversión de los caracteres a escala de grises se realiza mediante un diccionario, debido a esto es fácilmente modificable según las nuevas especificaciones que se busquen. Esto permite cualquier tipo de modificación o incluso codificación ya sea por motivos de seguridad o de adaptabilidad ante lo que se persiga.

- Los campos dentro de las imágenes se encuentran estructurados, estos han sido creados de forma independiente y modular. El objetivo buscado con esto es facilitar su análisis tanto a nivel visual como numérico (técnicas de *machine learning*). Aparte de esto, al haber sido diseñado desde su concepción de forma modular, esto permite que cualquier tipo de redistribución y modificación será de fácil implementación.

- Al haberse elegido la escala de grises para la conversión de las imágenes (matrices de 2 dimensiones). Los valores están comprendidos entre 0 (negro) y 255 (blanco), esto puede resultar un problema a la hora de distinguir entre elementos que tengan valores cercanos en la conversión. Tanto a nivel visual como a la hora de utilizarlas como *inputs* en técnicas de *machine learning*. Respecto a la inspección visual, como lo importante es distinguir patrones y observar qué elementos son distintos no supone una dificultad en realidad, pero a nivel numérico (procesamiento de píxeles a nivel imagen) no se ha podido testear. Esto podría resultar en un inconveniente a la hora de clasificar y separar.

- Los *logs* utilizados para conformar las imágenes se han obtenido por medio de la exportación de los registros ya interpretados por el programa Wireshark. Se ha escogido qué información exportar de forma más detallada y el resto ha ido al campo Info. Esto conlleva una doble situación, de tenerse otro analizador de protocolos los registros interpretados podrían variar y quizás, por la forma de decodificar del propio analizador, parte de

la información podría perderse o no haberse exportado apropiadamente. La opción de coger los datos sin decodificar (en hexadecimal), no parece ser una opción viable actualmente.

- Los *logs* siguen transmisiones de comunicación, esto hace que un conjunto de 2-3-5 registros formen parte de la misma transmisión. Para poder analizar las amenazas y los posibles ataques es necesario estudiar la transmisión completa, ya que es ahí donde se detectan anomalías y puntos importante que sirven de indicador para un tipo u otro. Con esto en mente hay que tener en cuenta que el análisis de las imágenes no puede hacerse de forma individual, sino en conjunto, tal y como se mostró en la Sección 4.2: Análisis de las imágenes. Aunque en ese caso se contrastaban 2 imágenes.

Capítulo 6 FUTUROS DESARROLLOS

Este proyecto deja la puerta abierta a futuros desarrollos. Los principales se presentan a continuación:

- **Realizar las imágenes a color (RGB):** una posible vía de investigación, en primer lugar, porque fue una de las posibilidades que se barajaron en un principio. Siendo ahora conscientes de que, en dos dimensiones, a escala de grises, se ha podido representar todo de forma estructurada y clara, pasar de dos dimensiones a tres es un hecho factible teniendo en cuenta las posibles implicaciones que ello podría conllevar. Habría que estudiar si las ventajas obtenidas en el análisis visual al tener ahora una gama de colores compensarían las posibles complicaciones para poder analizar las imágenes como matrices de píxeles RGB.

- **Representar la información desde la *raw binary data* (en hexadecimal):** utilizar la información obtenida de manera bruta capturada por los *packet sniffers* para conformar las imágenes de forma que sea fácilmente interpretada por un análisis visual sería un gran avance. Actualmente es necesario pasar por el analizador de protocolos para interpretar la información y poder exportarla para así conformar las imágenes. Esta vía plantea una serie de problemas e inconvenientes bastante grandes ya que los *logs* en bruto están codificados en hexadecimal y a su vez contienen las tramas de los eventos, todo en binario. De ahí la gran importancia de los programas como Wireshark.

- **Construir un módulo de *machine learning* para analizar y clasificar las imágenes:** ahora que se tienen ya las imágenes estructuradas y preparadas para el análisis visual y numérico (por los píxeles), uno de los siguientes puntos debería ser construir un módulo de *machine learning* que utilice técnicas de análisis de imágenes o *deep learning* para poder aprender de las transmisiones y poder clasificar y detectar posibles amenazas y ataques. Es importante construir el módulo de tal forma que tenga "memoria" para poder ser capaz de analizar las transmisiones de 2-3-5 imágenes.

- **Ampliar el número de campos de información ya analizada de más protocolos:** al igual que se ha hecho con los protocolos TCP e ICMP se pueden desarrollar más campos con información implícita en la trama de los eventos. De esta manera se podrá facilitar el análisis de muchos más protocolos.

- **Estudiar la comunicación de un módulo de *machine learning* con un SIEM:** lo cual sería el objetivo último a nivel logístico y operativo. Ser capaz de poder cotejar a tiempo real las distintas amenazas con el módulo de *machine learning.* Para ello se tendría el módulo y el SIEM, el primero analizando las imágenes y el segundo los *logs* habituales. Incluso se podría estudiar una posible comunicación entre los dos dispositivos para que se pudieran desarrollar nuevas reglas de correlación gracias al módulo. De esta manera se podría conseguir también un SIEM constantemente actualizándose gracias a este módulo, mejorando su rendimiento.

BIBLIOGRAFÍA

[1] V. Roblek, M. Meško, y A. Krapež, «A Complex View of Industry 4.0», *SAGE Open*, vol. 6, n.º 2, pp. 2-9, abr. 2016.

[2] I. Lee y K. Lee, «The Internet of Things (IoT): Applications, investments, and challenges for enterprises», *Bus. Horiz.*, vol. 58, n.º 4, pp. 431-440, jul. 2015.

[3] R. Moreno-Vozmediano, R. S. Montero, y I. M. Llorente, «Key Challenges in Cloud Computing: Enabling the Future Internet of Services», *IEEE Internet Comput.*, vol. 17, n.º 4, pp. 18-25, jul. 2013.

[4] N. Jazdi, «Cyber physical systems in the context of Industry 4.0», en *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*, Cluj-Napoca, Romania, 2014, pp. 1–4.

[5] S. Dua y X. Du, *Data Mining and Machine Learning in Cybersecurity*, 1st ed. Boston, MA, USA: Auerbach Publications, 2011.

[6] Y. Zhang, L. Wang, Y. Xiang, y C.-W. Ten, «Power System Reliability Evaluation With SCADA Cybersecurity Considerations», *IEEE Trans. Smart Grid*, vol. 6, n.º 4, pp. 1707-1721, jul. 2015.

[7] S. McLaughlin *et al.*, «The Cybersecurity Landscape in Industrial Control Systems», *Proc. IEEE*, vol. 104, n.º 5, pp. 1039-1057, may 2016.

[8] C. C. Aggarwal y P. S. Yu, *Privacy-preserving data mining models and algorithms*. New York: Springer, 2008.

[9] CCN-CERT, «CCN-CERT_IA-16-17 Ciberamenzas y Tendencias», Resumen ejecutivo Edición 2017, 2017.

[10] N. Diakun-Thibault, «Defining Cybersecurity», *Technol. Innov. Manag. Rev.*, vol. 2014, pp. 13-22, oct. 2014.

[11] K. Fowler, «Cibersecurity», en *Enterprise Risk Management*, vol. 7, Butterworth-Heinemann, 2015, pp. 91-108.

[12] S. Bhatt, P. K. Manadhata, y L. Zomlot, «The Operational Role of Security Information and Event Management Systems», *IEEE Secur. Priv.*, vol. 12, n.º 5, pp. 35-41, sep. 2014.

[13] P. J. Moreano Jurado, «Técnicas de detección de ataques en un sistema SIEM (Security Information and Event Management)», B.S. thesis, Universidad San Francisco de Quito, 2015., Quito, 2015.

[14] I. G. Jason Creasey, «Cyber Security Incident Response Guide». CREST, 2014.

[15] S. Khan *et al.*, «Cloud Log Forensics: Foundations, State of the Art, and Future Directions», *ACM Comput. Surv.*, vol. 49, n.º 1, pp. 1-42, may 2016.

[16] G. G. Granadillo, M. El-Barbori, y H. Debar, «New Types of Alert Correlation for Security Information and Event Management Systems», en *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2016, pp. 1-7.

[17] J. D. Avella Coronado, L. F. Calderón Barrios, M. Díaz, y C. Andrés, «Guía metodológica para la gestión centralizada de registros de seguridad a través de un SIEM», B.S. thesis, Universidad Católica de Colombia, Bogotá D.C, 2015.

[18] A. Sohail, «Providing Information Security Using ArcSight SIEM in an Organization.», *Int. J. Eng. Comput. Sci.*, vol. 3, n.º 4, pp. 5364–5368, abr. 2014.

[19] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, y R. Atkinson, «Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey», *ArXiv170102145 Cs*, ene. 2017.

[20] A. L. Buczak y E. Guven, «A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection», *IEEE Commun. Surv. Tutor.*, vol. 18, n.º 2, pp. 1153-1176, 2016.

[21] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, y M. Rajarajan, «A survey of intrusion detection techniques in Cloud», *J. Netw. Comput. Appl.*, vol. 36, n.º 1, pp. 42-57, ene. 2013.

[22] Y. Chang, W. Li, y Z. Yang, «Network Intrusion Detection Based on Random Forest and Support Vector Machine», 2017, pp. 635-638.

[23] C. Deng y H. Qiao, «Network security intrusion detection system based on incremental improved convolutional neural network model», en *2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016, pp. 1-5.

[24] T. Gladkykh, T. Hnot, y V. Solskyy, «Fuzzy logic inference for unsupervised anomaly detection», en *2016 IEEE First International Conference on Data Stream Mining Processing (DSMP)*, 2016, pp. 42-47.

[25] B. Rossi, S. Chren, B. Buhnova, y T. Pitner, «Anomaly detection in Smart Grid data: An experience report», en *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, Budapest, Hungary, 2016, pp. 2313–2318.

[26] C. Liu *et al.*, «A Systematic Approach for Dynamic Security Assessment and the Corresponding Preventive Control Scheme Based on Decision Trees», *IEEE Trans. Power Syst.*, vol. 29, n.º 2, pp. 717-730, mar. 2014.

[27] J. Vanerio y P. Casas, «Ensemble-learning Approaches for Network Security and Anomaly Detection», 2017, pp. 1-6.

[28] B. Xue, M. Zhang, W. N. Browne, y X. Yao, «A Survey on Evolutionary Computation Approaches to Feature Selection», *IEEE Trans. Evol. Comput.*, vol. 20, n.º 4, pp. 606-626, ago. 2016.

[29] Y.-J. Hu, W.-Y. Liu, y W.-N. Wu, «Structured Machine Learning for Data Analytics and Modeling: Intelligent Security as an Example», 2015, pp. 325-332.

[30] G. Cervone, P. Franzese, y A. P. K. Keesee, «Algorithm quasi-optimal (AQ) learning», *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, n.º 2, pp. 218-236, mar. 2010.

[31] Y. Ma, S. Liang, X. Chen, y C. Jia, «The Approach to Detect Abnormal Access Behavior Based on Naive Bayes Algorithm», 2016, pp. 313-315.

[32] O. P. B. V. Y. Zybtseva, «Collection of lexical and grammatical assignments in English on computer science». 2018.

[33] B. Li, J. Springer, G. Bebis, y M. Hadi Gunes, «A survey of network flow applications», *J. Netw. Comput. Appl.*, vol. 36, n.º 2, pp. 567-581, mar. 2013.

[34] R. Hofstede *et al.*, «Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX», *IEEE Commun. Surv. Tutor.*, vol. 16, n.º 4, pp. 2037-2064, 2014.

[35] V. Ndatinya, Z. Xiao, V. Rao Manepalli, K. Meng, y Y. Xiao, «Network forensics analysis using Wireshark», *Int. J. Secur. Netw.*, vol. 10, p. 91, ene. 2015.

[36] D. Quick y K.-K. R. Choo, «Big forensic data management in heterogeneous distributed systems: quick analysis of multimedia forensic data: Big Forensic Data Management in Smart Cities», *Softw. Pract. Exp.*, 2016.

[37] L. Wang y C. A. Alexander, «Big Data in Distributed Analytics, Cybersecurity, Cyber Warfare and Digital Forensics», *Digit. Technol.*, vol. 1, n.º 1, pp. 22-27, 2015.

[38] S. Khan, A. Gani, A. W. A. Wahab, M. Shiraz, y I. Ahmad, «Network forensics: Review, taxonomy, and open challenges», *J. Netw. Comput. Appl.*, vol. 66, pp. 214-235, may 2016.

[39] H. Lalla, S. Flowerday, T. Sanyamahwe, y P. Tarwireyi, «A Log File Digital Forensic Model», en *Advances in Digital Forensics VIII*, vol. 383, G. Peterson y S. Shenoi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 247-259.

[40] O.-M. Latvala, J. Toivonen, A. Evesti, M. Sihvonen, y V. Jordan, «Security Risk Visualization with Semantic Risk Model», *Procedia Comput. Sci.*, vol. 83, pp. 1194-1199, 2016.

[41] D. Freet y R. Agrawal, «A virtual machine platform and methodology for network data analysis with IDS and security visualization», 2017, pp. 1-8.

[42] D. Staheli *et al.*, «Visualization evaluation for cyber security: trends and future directions», 2014, pp. 49-56.

[43] K. K. Murugiah Souppaya, «Guide to Computer Security Log Management». NIST - National Institute of Standards and Technology.

[44] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, y Y. Xiao, «Network forensics analysis using Wireshark», p. 16.

[45] R. Vaarandi y M. Pihelgas, «Using Security Logs for Collecting and Reporting Technical Security Metrics», 2014, pp. 294-299.

[46] A. Sapegin, D. Jaeger, A. Azodi, M. Gawron, F. Cheng, y C. Meinel, «Hierarchical object log format for normalisation of security events», en *2013 9th International Conference on Information Assurance and Security (IAS)*, 2013, pp. 25-30.

[47] N. M. Adams y N. Heard, *Dynamic networks and cyber-security*. New Jersey: World Scientific, 2016.

[48] C. Humphries, N. Prigent, C. Bidan, y F. Majorczyk, «ELVIS: Extensible log VISualization», en *ACM International Conference Proceeding Series*, 2013, pp. 9-16.

[49] Y. LeCun, Y. Bengio, y G. Hinton, «Deep learning», *Nature*, vol. 521, n.º 7553, pp. 436-444, may 2015.

[50] «4SICS with ICS PCAP Files», *Netresec*. [En línea]. Disponible en: https://www.netresec.com/?page=Blog&month=2015-11&post=From-4SICS-with-ICS-PCAP-Files.

[51] CCN-CERT, «CCN-CERT IA-09-18 Ciberamenazas y Tendencias», Informe Anual Sin Clasificar, may 2018.

[52] J. Bhusal, «Network analysis with open source packet analyzers : case wireshark», 2016.

[53] A. F. D. Mustapha Adamu Mohammed*, «Brute Force Attack Detection And Prevention On A Network Using Wireshark Analysis», jun. 2017.

[54] L. Chappell, *Wireshark 101: Essential Skills for Network Analysis - Second Edition Wireshark Solution Series*. USA: Laura Chappell University, 2017.

[55] S. Yuan, P. Liu, y E. Zhao, «Research on Security Protection of the Communication Network for Space TT&C Based on TCP/IP Protocol Vulnerabilities», en *Proceedings of the 28th Conference of Spacecraft TT&C Technology in China*, Springer, Singapore, 2016, pp. 557-566.

[56] R. Abramov y A. Herzberg, «TCP Ack Storm DoS Attacks», en *Future Challenges in Security and Privacy for Academia and Industry*, vol. 354, J. Camenisch, S. Fischer-Hübner, Y. Murayama, A. Portmann, y C. Rieder, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 29-40.

[57] A. K. Kaushik y R. C. Joshi, «Network Forensic System for ICMP Attacks», *Int. J. Comput. Appl.*, vol. 2, n.º 3, pp. 14-21, may 2010.

[58] H. A. Mangut, A. Al-Nemrat, C. Benzaïd, y A. R. H. Tawil, «ARP Cache Poisoning Mitigation and Forensics Investigation», en *2015 IEEE Trustcom/BigDataSE/ISPA*, 2015, vol. 1, pp. 1392-1397.

[59] A. Borgwart, S. Boukoros, H. Shulman, C. van Rooyen, y M. Waidner, «Detection and Forensics of Domains Hijacking», en *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1-6.

[60] A. Kleinmann y A. Wool, «Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics», *J. Digit. Forensics Secur. Law*, vol. 9, pp. 37-50, ene. 2014.

[61] T. Mander, R. Cheung, y F. Nabhani, «Power system DNP3 data object security using data sets», *Comput. Secur.*, vol. 29, n.º 4, pp. 487-500, jun. 2010.

[62] I. J. Goodfellow *et al.*, «Generative Adversarial Networks», *ArXiv14062661 Cs Stat*, jun. 2014.

[63] W. Liang, Z. Chen, X. Yan, X. Zheng, y P. Zhuo, «Multiscale Entropy-Based Weighted Hidden Markov Network Security Situation Prediction Model», 2017, pp. 97-104.

Parte II

ESTUDIO ECONÓMICO

Capítulo 1 ESTUDIO ECONÓMICO

Tal y como se ha podido observar a lo largo de la Parte I, la elaboración de este proyecto persigue un objetivo de gran ambición. Para la elaboración de este no se han necesitado herramientas específicas de gran coste puesto que el mayor trabajo se ha realizado con el programa Wireshark, el cual es *open source.* Teniendo en cuenta estos conceptos, a continuación se evalúan los siguientes aspectos relativos al estudio económico:

❖ Viabilidad

La viabilidad del proyecto ha quedado constatada por sí misma. Utilizando recursos disponibles a nivel global se ha realizado una labor de formación y de estudio que han permitido adquirir los conocimientos necesarios para realizar el análisis forense de un conjunto de datos industriales sin catalogar. Además, con los conocimientos que se iban adquiriendo, utilizando a la par el programa Matlab, se ha propuesto un estándar en formato gráfico para analizar los registros para la ciberseguridad. Se ha constatado que el proyecto ha sido viable.

❖ Rentabilidad

Teniendo en cuenta que el mayor porcentaje de los costes (95%) corresponden a la mano de obra, implica que es una inversión fija el formar a un profesional para realizar este trabajo. Además, teniendo en cuenta que el objetivo último de este proyecto es crear un puente entre el mundo del análisis de registros y el mundo del *machine learning*, es importante constatar que el ser capaz de automatizar procesos de análisis para clasificación o regresión conlleva una gran rentabilidad.

❖ Fiabilidad

Este proyecto proporciona unas herramientas y un *background* de gran importancia para este campo. Pocos estudios se encontrarán en los que se pueda conseguir la documentación aquí mostrada con respecto al análisis del dataset. Teniendo en cuenta la propuesta en estándar gráfico destacar que esto abre paso a un nuevo mundo de posibilidades, lo aquí propuesto, tal y como se puede observar en el capítulo de Futuros desarrollos, es algo dinámico listo y preparado para evolucionar. Respecto al tema económico, apostar por este tipo de cambio es hacia donde se está dirigiendo el mundo contemporáneo. El uso de redes neuronales, convolucionales y sus demás derivadas es cada vez más común y su uso intenta aplicarse en todos los campos de la vida. El habilitar dicho uso en un campo que hasta ahora no era viable es de gran aportación y fiabilidad.

❖ Interés económico del mismo

No hay mayor interés económico del mismo más allá que el servir como puente para dichos mundos. Significa un cambio que, de lograrse, permitirá muchos y nuevos avances para la ciberseguridad. Si en vez de tener que actualizar manualmente las reglas de los SIEM, se dispone de un módulo de aprendizaje que, con estas imágenes, es capaz de aprender, evolucionar y predecir ataques y nuevas estructuras de estos, las posibilidades son de gran relevancia.

Parte III

CÓDIGO FUENTE

Capítulo 1 ARCHIVO PARA IMPORTAR LOS DATOS DE WIRESHARK A MATLAB Y DEJARLOS PREPARADOS

```
% Nombre del archivo: ImportData.m
%
% Autor: Jorge Buil Garcí (jorgbg@gmail.com)
%
% Fecha: 17-01-18
%
% Descripción: Script que permita la importación de los datos exportados desde
% Wireshark con los registros. Además de importarlos también se encarga de
% estructurarlos de forma que cada campo quede en una celda y cada elemento de
% de cada campo sea un elemento de la lista de la celda.
```

1.1 INICIALIZACIÓN, IMPORTACIÓN Y PREPARACIÓN

```
%% Import Data
%% Los datos se importan manualmente (para que no pete todo)
clc
clear
close all

% fdata=fopen('fdata_prueba');
fdata=fopen('Ataqueplot');
data_out=[];
line=fgetl(fdata);
while line~=-1
    line(line==',')='';
    l=[];
    while numel(line)>0
        l=[l,{line(1:find(line=='"',1)-1)}];
        line=line(find(line=='"')+1:end);
    end
    data_out=[data_out;l];
    line=fgetl(fdata);

end
data_out = data_out(:,2:2:end);
fclose(fdata);
save('AtaquePlot.mat')
```

Capítulo 2 ARCHIVO PARA CONVERTIR LOS ELEMENTOS EN PÍXELES, AGRUPARLOS EN MATRICES Y ESTRUCTURAR LAS IMÁGENES

```
% Nombre del archivo: Data_Conversion.m
%
% Autor: Jorge Buil García (jorgbg@gmail.com)
%
% Fecha: 10-02-18
%
% Descripción: Este Script permite recoger los datos ya preparados de los logs
%importados a Matlab. Con estos realiza la conversión a escala de grises mediante
%la función get_dictionary.m y search.dictionary.m Se encarga de realizar los
%análisis necesarios para los campos TCP e ICMP y finalmente distribuye los
%elementos (ya en valores numéricos) entre sus matrices de píxeles dentro de las
%correspondientes a su campo con get_conversion.m Finalmente se encarga de
%estructurar las imágenes y de exportarlas. Todo esto en un bucle para realizarlo
%de forma automática cuántas veces sea necesarios según el conjunto de datos de
%los que se disponga.
```

2.1 *INICIALIZACIÓN DE LAS VARIABLES*

```
clc
clear

dic = get_dictionary('Dictionary.txt','|');

%% Inicialización de Variables
vars = {};
vec = {};
n = 2; % Número de la línea
cont_fail = 0;
cont_fails = {};
```

2.1.1 FUNCIÓN GET_DICTIONARY

Esta función se encarga de leer el diccionario de conversión, de manera que prepara los datos para luego ir convirtiendo los elementos al valor asignado.

```
function dic = get_dictionary(filename,delimiter)
%GET_DICTIONARY reads the specified file and treats it as a
dictionary with
%index and meaning separated by the specified delimiter. The
return value
%is a cell matrix with the first column the index word and the
second
%column the meaning.

fid=fopen(filename);
dic=[];
line=fgetl(fid);
count=1;
while line~=-1
    ind=strfind(line,delimiter);
    try
        ind=ind(1); %only first one is used
    catch
        error(['No delimiter found in line ',num2str(count),'.']);
    end
    dic=[dic;{line(1:ind-1)},{line(ind+length(delimiter):end)}];
    line=fgetl(fid);
    count=count+1;
end

fclose(fid);


 return
```

2.2 CREACIÓN DE UNA CARPETA DONDE GUARDAR LAS FUTURAS IMÁGENES

```
%% Creamos Folder con nombre igual a fecha
t = [datetime('now')];
fecha_numerica=datenum(t);
nombre_fecha = datestr(fecha_numerica,'yyyy_mmmm_dd_HH_MM_SS');
name_folder= strcat(nombre_fecha,'_Image_Set');


 mkdir (name_folder);
```

2.3 SELECCIÓN DEL CONJUNTO DE DATOS A CONVERTIR

2.3.1 CONVERTIR UN GRUPO ALEATORIO DE MUESTRAS

```
data_num={};
    num = 20;
    ran_vec = randi(size(data_out,1),1,num+1);
    for i=1:(num+1)
```

```
      for j=1:size(data_out,2)
            if i == 1
                data_num{i,j}=data_out{i,j};
            else
                data_num{i,j}= data_out{ran_vec(i),j};
            end
        end

     end
```

2.3.2 CONVERTIR UN DATO EN PARTICULAR

```
conversation = 1+[481119]; % El +1 es porque al importarlo hay que
sumarselo
num = size(conversation,2);
data_specific={};
for i=1:(num+1)
    for j=1:size(data_out,2)
        if i == 1
            data_specific{i,j}=data_out{i,j};
        else
            data_specific{i,j}= data_out{conversation(i-1),j};
        end
    end

 end
```

2.3.3 ELECCIÓN DE LOS ELEMENTOS A CONVERTIR Y SU CONVERSIÓN

```
aux = data_out; % Para convertir todo
% aux = data_num; % Para convertir un grupo
% aux = data_specific; % Para convertir una muestra
%% Comienza la conversion
    %% Conversión de Todos los campos
    for j=1:size(aux,2)
        for i=1:size(aux{k,j},2) % j especifica la columna
            [vec{i},fail]= search_dictionary(dic,aux{k,j}(i));
            if fail == 1
                if cont_fail > 0
                    cont_fail = cont_fail +1;
                else
                    cont_fail = 1;
                end
            end
        end
        cont_fails{j} = cont_fail;
        cont_fail = 0;
        vars{j}= vec;
        vec = {};

     end
```

Este Código comienza con un bucle **for** ya que se encarga de convertir todas las imágenes, para le representación del código se ha obviado para clarificar. Pero vendría a ser lo siguiente:

```
for k=2:(size(aux,1))
[todo el resto del código del script]
End
```

La k de la función search_dictionary es justo esa.

2.3.3.1 Función search_dictionay

Esta función se encarga de ir realizando la conversión. Para ello va leyendo cada uno de los elementos de cada celda de campo y los va cotejando con su valor correspondiente en el diccionario de conversión. Estos valores pueden verse en la Sección Parte I3.2.3.

```
function [meaning,fails] = search_dictionary(dic,word)
%READ_DICTIONARY searches for the specified word in the previously
read
%dictionary with GET_DICTIONARY and returns the meaning if found
and the
%original word if not.

% fails = 1 if the symbol has not been found
if size(dic,2)~=2
    error('Invalid dictionary variable format. Use function
get_dictionary().');
end
%TODO: más errores

matches=cellfun(@(x) strcmp(word,x), dic(:,1));

if any(matches)
    %use first match
    meaning=dic{find(matches,1),2};
    meaning = str2double(meaning);
    fails = 0;
else
    meaning='255';
    meaning= str2double(meaning);
    fails = 1;
end


 return
```

2.4 ANÁLISIS PARA LOS CAMPOS TCP – CONTROL

En esta parte del código se analizará los elementos necesarios para conformar los campos del TCP – Control.

2.4.1 EVALUACIÓN DE TCP – CONTROL FLAGS

```
control={'RST','ACK', 'SYN', 'FIN', 'PSH','URG'};
    control_bits=[];
    info = 11; % Posición de la cadena info
    for i=1:size(control,2)
        isack=cellfun(@(x)
isempty(strfind(x,control(i))),aux(k,info));

        if i == 2 && isack == 0 %1 porque es la posición de ACK
            issack=~cellfun(@(x)
isempty(strfind(x,'SACK')),aux(k,info));
            isack = ~isack()&issack();
        end
        control_bits(i) = double(isack); % Esto es para que 0 si
sí hay, 1 si no hay
    end

     vars{size(aux,2)+1}= num2cell(255*control_bits); % pixel = 255
si no hay (Blanco) y pixel = 0 si sí hay (Negro)
```

2.4.2 EVALUACIÓN DE TCP – CONTROL INFO

```
control2 = {'Ack=1 ','Ack=0', 'Seq=1 ', 'Seq=0
','Len=1','Len=0','TCP', 'Dup'};
    info_eval =[];
    cotnrol_info = [];
    info = 11; % Posición de la cadena info

    for i=1:size(control2,2)
        isdat2=cellfun(@(x)
isempty(strfind(x,control2(i))),aux(k,info));
        info_eval(i) = double(isdat2);
    end
    info_eval;
   control_info =
double([(info_eval(1)&&info_eval(2))&&(info_eval(3)&&info_eval(4))
&&...
                            (info_eval(5)&&info_eval(6))]);
    % Eval de los Ack
    if info_eval(1)==0
        control_info(2)= 0.5;
    end
    if info_eval(2)==0
        control_info(2)= 0;
```

```
    end
    if info_eval(1) && info_eval(2) == 1
        control_info(2) = 1;
    end
    % Eval de los Len
    if info_eval(3)==0
        control_info(3)= 0.5;
    end
    if info_eval(4)==0
        control_info(3)= 0;
    end
    if info_eval(3) && info_eval(4) == 1
        control_info(3) = 1;
    end
    % Eval de los Seq
    if info_eval(5)==0
        control_info(4)= 0.5;
    end
    if info_eval(6)==0
        control_info(4)= 0;
    end
    if info_eval(5) && info_eval(6) == 1
        control_info(4) = 1;
    end

    control_info(5)= info_eval(7);
    control_info(6) = info_eval(8);
    control_info;


     vars{size(aux,2)+2}= num2cell(255*control_info); % pixel = 255
si no hay (Blanco) y pixel = 0 si sí hay (Negro)
```

2.5 *ANÁLISIS PARA LOS CAMPOS DE ICMP*

En esta parte del código se analizará los elementos necesarios para conformar los campos del ICMP.

2.5.1 EVALUACIÓN DE ICMP – CONTROL

```
control3 = {'Echo (ping) request','Echo (ping) reply',
'Destination unreachable', '(no response found!)','id=','seq='};
    icmp_eval =[];
    control_icmp = [];
    info = 11; % Posición de la cadena info

    for i=1:size(control3,2)
        isdat3=cellfun(@(x)
isempty(strfind(x,control3(i))),aux(k,info));
        icmp_eval(i) = double(isdat3);
    end
```

2.5.2 EVALUACIÓN DE LOS NÚMEROS DE REFERENCIA ID Y SEQ

```
    if icmp_eval(5) == 0
        strid = extractAfter(aux(k,info),'id=');
        tokenid = char(strtok(strid,','));
    else
        tokenid =1;
    end
    if icmp_eval(6) == 0
        strseq = extractAfter(aux(k,info),'seq=');
        tokenseq = char(strtok(strseq,','));
    else
        tokenseq = 1;
    end
    token = {tokenid, tokenseq};

    for j=1:size(token,2)
        for i=1:size(token{1,j},2) % j especifica la columna
            [tokenvec{i},cont_failtoken]=
search_dictionary(dic,token{1,j}(i)); %k es la n de arriba
            if cont_failtoken == 1
                if cont_failtoken > 0
                    cont_failtoken = cont_failtoken +1;
                else
                    cont_failtoken = 1;
                end
            end
        end
        cont_failstoken{j} = cont_failtoken;
        cont_failtoken = 0;
        tokenvars{j}= tokenvec;
        tokenvec = {};

     end
```

2.5.3 CONFORMACIÓN FINAL DEL CAMPO ICMP

```
    control_icmp = [icmp_eval(1), icmp_eval(2), icmp_eval(3),
icmp_eval(4)];
    vars{size(aux,2)+3}= num2cell(255*control_icmp); % pixel = 255
si no hay (Blanco) y pixel = 0 si sí hay (Negro
    vars{size(aux,2)+4}= tokenvars{1};

     vars{size(aux,2)+5}= tokenvars{2};
```

2.6 ESTRUCTURACIÓN DE LAS MATRICES DE LOS CAMPOS

Donde se establecen las dimensiones de las matrices de píxeles, tanto de cada elemento como de cada campo, para su distribución. Se utiliza la función get_conversion para que se distribuyan adecuadamente.

```
    % Dim1 es de la matriz completa, Dim2 de las submatrices
    %% Parámetros de los campos
    % [No, Time, SrcIP, DestIP, Protocol, Length, ScrPort,
DestPort, SrcMac, DestMac, Info, ControlBits, ControlInfo,
ControlICMP, TokenICMP(1), TokenICMP(2)]
    % [ 1,    2,     3,      4,        5,      6,       7,
8,      9,      10,   11,          12,          13,          14,
15,           16]
    y=[36,42,20,20,20,9 ,14,14,21,21,30,10,8 ,36,4 ,4 ];
    x=[26,26,50,50,36,36,36,36,56,56,84,36,36,20,36,36];

    a=[3 ,3 ,10,10,20,9,14,14,7,7,6,10,8,36,4,4];
    b=[13,13,5 ,5 ,4 ,9,6 ,6 ,7,7,3,6 ,6,4 ,6,4];
    total_fallos = {};
    for i=1:size(vars,2)
        vec_conv{i}= get_conversion(y(i),x(i),a(i),b(i),vars{i});

     end
```

2.6.1 FUNCIÓN GET_CONVERSION

Sirve para distribuir adecuadamente las nuevas matrices de píxeles correspondientes a cada elemento en la disposición prestablecida para rellenar adecuadamente los campos.

```
function [mat,error_a,error_b] = get_conversion(x,y,a,b,vars)
%GET_CONVERSION sirve para cada uno de los 'caracteres' de vars en
una
%matriz de pixeles de dimension (a,b) con su valor
correspondiente. Ademas
%tambien se encarga de recopilar todas las nuevas matrices y
ordenarlas
%rellenandolas por filas en una nueva matriz (x,y). El resultado
que
%devuelve es una matriz mat de dimension (x,y) cuyos valores seran
255
%excepto en aquellos donde se hayan incorporado los valores de
vars.


%Comprobamos que la matriz de pixeles (x,y) es divisible en
submatrices
```

```
%(a,b) de devolver error habría que cambiar las entradas
if rem(x,a) == 0
    error_a = 0;
else
    error_a = 1;
end

if rem(y,b) == 0
    error_b = 0;
else
    error_b = 1;
end

% Creamos matriz llena de 255 que en UINT-8 son pixeles blancos
mat_pix = uint8(255*ones(x,y));

% El numero de divisiones que se forman de la matriz principal
para las
% submatrices dix en filas y diy en columnas
dix= x/a;
diy= y/b;

%Varibles para el recorrido de filas y columnas, 'a' corresponde a
filas y
%'b' a columnas:
% va1=[1,(n-1)*a+1];
% va2=[n*a];
% vb1=[1,(n-1)*b+1];
% vb2=[n*b];

%j es la variable que va a recorrer el vector
j=1;
for m=1:dix
   va1=[1,(m-1)*a+1];
   va2=m*a;
   if m==1
      for k=1:diy
          vb1=[1,(k-1)*b+1];
          vb2=k*b;
          %Requisito de salida
          if j > size(vars,2)
              break;
          end
          if k==1
              mat_pix(va1(1):va2, vb1(1):(vb2)) =
cell2mat(vars(j))*ones(a,b);
              j=j+1;
          else
              mat_pix(va1(1):va2, vb1(2):(vb2)) =
cell2mat(vars(j))*ones(a,b);
              j=j+1;
          end
      end
   else
       for k=1:diy
          vb1=[1,(k-1)*b+1];
          vb2=k*b;
```

```
            % Requisito de salida
            if j > size(vars,2)
                break;
            end
            if k==1
                mat_pix((va1(2)):va2, vb1(1):(vb2))=
cell2mat(vars(j))*ones(a,b);
                j=j+1;
            else
                if j > size(vars,2)
                    break;
                end
                mat_pix((va1(2)):va2, vb1(2):(vb2))=
cell2mat(vars(j))*ones(a,b);
                j=j+1;
            end
        end
    end
end

 mat = mat_pix;
```

2.7 DISTRIBUCIÓN DE LOS CAMPOS EN LAS IMÁGENES

La disposición final de cada uno de los campos a representar dentro de la matriz global de 128x128 píxeles

```
% Base en blanco para luego rellenar
    for i=1:size(x,2)
       conv{i}= uint8(255*ones(y(i),x(i)));

     end

 %% Estructuración de la matriz data

    data(1:42,1:26)=uint8(vec_conv{2});  %Time
    data(49:84,1:26)=uint8(vec_conv{1}); %No
    data(1:20,34:83)=uint8(vec_conv{3}); %SourceIP
    data(22:41,34:83)=uint8(vec_conv{4});%DestinationIP
    data(49:69,32:87)=uint8(vec_conv{9}); %SrcMac
    data(73:93,32:87)=uint8(vec_conv{10}); %DestMac
    data(1:14,93:128)=uint8(vec_conv{7}); %SrcPort
    data(18:31,93:128)=uint8(vec_conv{8}); %DestPort
    data(37:56,93:128)=uint8(vec_conv{5}); %Protocol
    data(60:69,93:128)=uint8(vec_conv{12}); %ControlBits
    data(73:80,93:128)=uint8(vec_conv{13}); % ControlInfo
    data(84:92,93:128)=uint8(vec_conv{6}); %Length
    data(99:128,5:88)=uint8(vec_conv{11}); %Info
    data(99:118,93:128)=uint8(vec_conv{14}'); % ContrlICMP ESTÁ
TRASPUESTO PORQUE ESTOS VAN EN VERTICAL
    data(120:123,93:128)=uint8(vec_conv{15});
    data(125:128,93:128)=uint8(vec_conv{16});

     imshow(data)
```

2.8 EXPORTACIÓN DE LAS IMÁGENES Y ALMACENAMIENTO

```
image = sprintf('image_%d.tif',str2num(data_out{k}));
imwrite(data,image , 'tif');
  figure;imshow(image)

 movefile  ('image*', name_folder)
```

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER INGENIERÍA INDUSTRIAL

ANÁLISIS FORENSE DE UN DATASET INDUSTRIAL Y PROPUESTA DE UN ESTÁNDAR GRÁFICO PARA LOS REGISTROS EN LA CIBERSEGURIDAD

PRESUPUESTO

Autor: Jorge Buil García
Director: Álvaro López López

Madrid
Junio 2018

Índice del presupuesto

Índice de tablas

Capítulo 1 RECURSOS EMPLEADOS

En este capítulo se calculará la cantidad de recursos que se han dedicado al proyecto, de quipo, software o humanos. Al ser un proyecto de análisis forense de un conjunto de dataset y de creación de un estándar a nivel informático, sólo se han tenido en cuenta los recursos como herramientas, software y mano de obra.

1.1 HERRAMIENTAS Y SOFTWARE

Componentes	Cantidad	Horas de proyecto	Horas de uso al año
Ordenador	1	480	720
Matlab	1	430	645
Wireshark	1	520	780
Office 365	1	50	75
	Total	1480	2220

Tabla 1 Recursos de herramientas y software

1.2 MANO DE OBRA

Actividad	Horas
Actualización de conocimientos	250
Análisis del dataset	500
Elaboración del Código de Matlab	280
Análisis de las imágenes	380
Redacción de Documentación	70
Total	1480

Tabla 2 Recursos de mano de obra

Capítulo 2 COSTES UNITARIOS

En esta sección se detallan los costes y precios de cada uno de los elementos previamente analizados con los que se ha llevado a cabo este proyecto.

2.1 HERRAMIENTAS Y SOFTWARE

Componentes	Precio (€/Ud)
Ordenador	800
Matlab	200
Wireshark*	-
Office 365	89

Tabla 3 Costes unitarios de herramientas y software

*Wireshark es un programa *open source.*

2.2 MANO DE OBRA

Actividad	Precio (€/hora)
Actualización de conocimientos	14
Análisis del dataset	25
Elaboración del Código de Matlab	23
Análisis de las imágenes	22
Redacción de Documentación	40

Tabla 4 Costes unitarios de mano de obra

Capítulo 3 SUMAS PARCIALES

En este capítulo se calcula el coste total de cada uno de los recursos empleados a partir de las mediciones y los precios unitarios. Se ha contemplado un horizonte de 4 años para amortizar los gastos necesarios en el equipo.

3.1 HERRAMIENTAS Y SOFTWARE

Componentes	Cantidad	Horas de Proyecto	Horas de uso al año	Precio (€/Ud)	Amortización	Coste(€)
Ordenador	1	480	720	800	25%	133,33
Matlab	1	430	645	200	25%	33,33
Wireshark*	1	520	780	-	25%	-
Office 365	1	50	75	89	25%	14,833
					Total	181,5

Tabla 5 Sumas parciales de herramientas y software

*Wireshark es un programa *open source.*

3.2 MANO DE OBRA

Actividad	Horas	Precio (€/h)	Precio total (€)
Actualización de conocimientos	250	14	3500
Análisis del dataset	500	25	12500
Elaboración del Código de Matlab	280	23	6440
Análisis de las imágenes	380	22	8360
Redacción de Documentación	70	40	2800
		Total	33600

Tabla 6 Sumas parciales de mano de obra

Capítulo 4 PRESUPUESTO GENERAL

Sumando todas las contribuciones, se el coste del proyecto asciende a:

Concepto	Coste(€)
Herramientas y Software	181,5
Mano de Obra	33600
Total	33781,5

Tabla 7 Costes del presupuesto general