UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

IMPLEMENTACIÓN DE UN SISTEMA DE PREVENCIÓN DE INTRUSIONES (IPS) EN UN MODELO DE RED INDUSTRIAL

Adrián Soucase Iranzo

Tutor: Víctor Miguel Sempere Paya

Cotutor: José Ramón Cano Sáez

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2021-22

Valencia, 7 de diciembre de 2021

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/ CAMPUS VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS Gestión ambiental verificada REG.NO. ES-CV- 000030

Resumen

Actualmente existen numerosos dispositivos y sistemas mediante los cuales se puede aumentar la seguridad de una red. Entre ellos se encuentran los IPS o sistemas de prevención de intrusiones, que tienen como función principal la prevención activa de posibles ataques a un sistema o red.
Con lo anterior en cuenta, lo que se pretende con este trabajo es, diseñar y configurar una red en el laboratorio que sirva como un pequeño modelo de red de control industrial, incorporando para ello, varios elementos que forman parte de una red de estas características. El elemento clave de la red y el foco central de este estudio es el IPS, mediante el que cual se van a detectar y detener distintos ataques lanzados por un intruso en la red. Además, para tener un mayor control sobre el tráfico detectado por el IPS y que este sea más sencillo de monitorizar resulta conveniente hacer uso de una herramienta con capacidades para ello. La solución que se propone es utilizar una herramienta software que a través de su interfaz web permita al usuario visualizar las alertas generadas por el IPS, realizar búsquedas concretas utilizando filtros o generar gráficas y tablas que proporcionen información relevante.

Abstract

Currently there are numerous devices and systems through which the security of a network can be improves. Among them are the IPS or intrusion prevention systems, whose main purpose is the active prevention of possible attacks on a system or network. With the above in mind, what is intended with this work is to design and configure a network in a laboratory that serves as a small model of an industrial control network, incorporating several elements that are part of a network of these characteristics. The key element of the network and the central focus of this study is the IPS, by which different attacks launched by an intruder on the network will be detected and stopped. In addition, to have greater control over the traffic detected by the IPS and for it to be easier to monitor, it is convenient to use a tool that is capable of this. The proposed solution is to make use of a software tool that allows the user through its web interface to visualize the alerts generated by the IPS, carry out searches using filters or generate graphs and tables that provide relevant information.

Índice de contenido

Índice de Figuras

Índice de Tablas

1. Introducción

1.1 Motivación

En los últimos años se han publicado cantidad de informes y artículos relacionados a los ataques a sistemas industriales que revelaban una situación creciente de estos. En el año 2012, la empresa McAfee afirmaba que los "atacantes suelen elegir sistemas que pueden ser fácilmente comprometidos y los SCI (sistemas de control industrial) han demostrado ser un entorno rico en posibles vulnerabilidades" a través de sus informes de *Threats Predictions*, que siguen, en la actualidad, recogiendo amenazas potenciales para la industria en sucesivos informes anuales [1]. Verizon y Cisco, son dos empresas que, actualmente, también publican informes anuales relacionados con las vulnerabilidades de estos sistemas y los ataques que sufren [2][3], presentando resultados que siguen la misma de lo mostrado en la Figura 1.

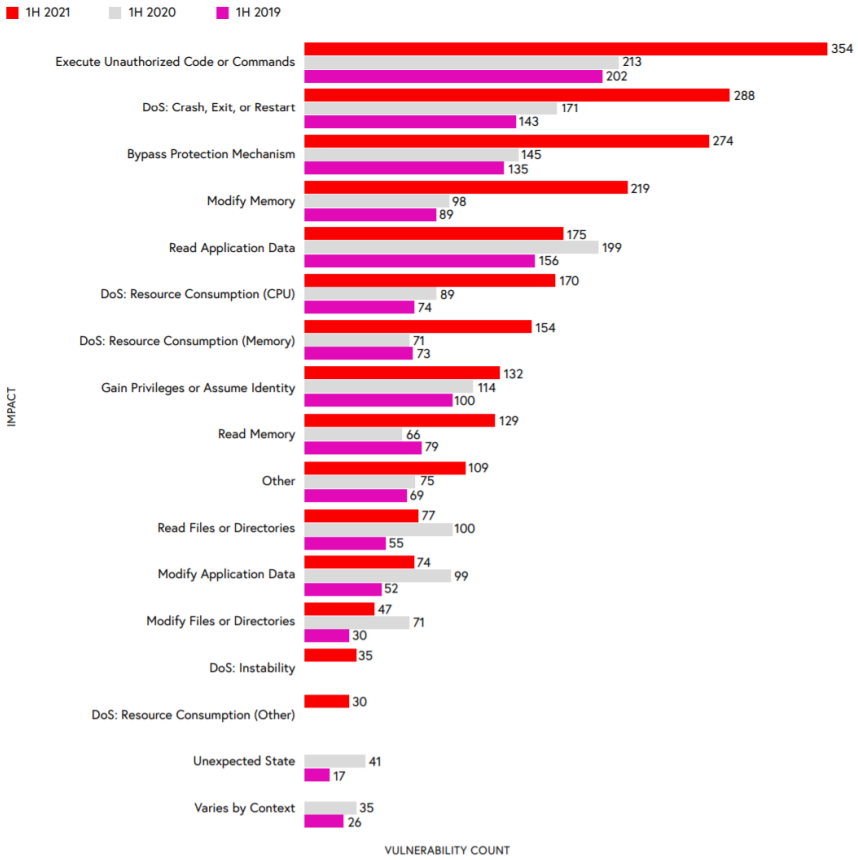


Figura 1. Número de vulnerabilidades contabilizadas por tipo de impacto en el primer semestre de los años 2019, 2020 y 2021. Fuente: [4]

Artículos muchos más recientes [4] presentan datos que muestran la misma tendencia que los de las publicaciones mencionadas previamente, pero con un matiz a destacar; un cambio en el paradigma industrial. Este cambio, acuñado por muchos como la Industria 4.0 implica la conexión de dispositivos a Internet para crear una convergencia entre la tecnología operativa (OT) y la tecnología informática (IT) dentro de la empresa. Este cambio implica que los activos de la empresa queden más expuestos al estar "online" y con ello todas sus imperfecciones: vulnerabilidades sin parches, credenciales inseguras, configuraciones inadecuadas y el uso de protocolos industriales obsoletos.

La capacidad de impacto que puede suponer un ciberataque puede variar mucho, pero cuando estos van dirigidos a grandes entidades los daños pueden ser catastróficos si el ataque resulta exitoso. Escenarios así ya se han vivido a lo largo de la historia como, por ejemplo, el ataque reciente a la compañía Colonial Pipeline, que se vio obligada a detener sus operaciones durante 5 días, lo que les supuso grandes pérdidas económicas [5].

Para prevenir situaciones como esta, las organizaciones deben comenzar a usar técnicas que permitan gestionar proactivamente el riesgo y contribuyan a minimizar o mitigar las amenazas. Ya que las empresas que no dispongan de estas medidas estarán más expuestas, lo que puede ocasionar interrupción en las operaciones de fabricación, interrupciones no planificadas y hasta pérdidas de cientos de millones de euros en ingresos y daños. Entre las técnicas más comunes se encuentran la segmentación de redes, seguridad de acceso remoto, protección contra spam y fishing, restricción de tráfico en la red y técnicas de detección de amenazas [4].

De entre todas las anteriores, este trabajo se centra en las últimas dos, restricción de tráfico y técnicas de detección de amenazas. Estas técnicas se pueden aplicar en varios puntos de la arquitectura de red de una empresa, generalmente haciendo uso de los denominados sistemas de detección y prevención de intrusos, que se verán en detalle a lo largo de este trabajo.

1.2 Objetivos

Los objetivos que se pretenden alcanzar con este trabajo son los detallados a continuación:

- Comprender en el concepto de Industria 4.0 y los cambios que supone respecto al modelo industrial más tradicional.
- Reconocer las vulnerabilidades que existen en el entorno de la Industria 4.0, las amenazas que pueden surgir y como se puede aplicar la ciberseguridad en estos entornos para minimizar la exposición a estos riesgos.
- Evaluar distintas alternativas y seleccionar una herramienta que funcione como un IDS e IPS.
- Conocer el funcionamiento de la herramienta de detección y prevención de intrusos seleccionada para poder implementarla en una red local de pruebas.
- Montar una red local en el laboratorio, conformada tanto por máquinas físicas como virtuales, que sirva como base para realizar pruebas de detección de intrusos.
- Llevar a cabo pruebas de detección, penetración y ataques a los sistemas de la red local configurada para comprobar el funcionamiento del IDS/IPS.
- Integrar la herramienta IDS/IPS con una interfaz gráfica que permita una visualización y análisis más exhaustivo de las alertas generadas.

1.3 Estructura de la memoria

En esta sección se resumen brevemente los contenidos de los capítulos que componen este trabajo.

1. **Introducción:** en este capítulo inicial se exponen las razones por las que resulta interesante desarrollar este trabajo y cuáles son los objetivos que se pretenden alcanzar al concluir el mismo.
2. **Estado del arte:** recopilación de la información más relevante del contexto sobre el que se basa el trabajo.
3. **Consideraciones previas:** introducción de las cuestiones y configuraciones iniciales a tener en cuenta antes de la parte experimental del trabajo.
4. **Configuración del IDS/IPS ante amenazas:** dividido en dos secciones, en este capítulo se recogen los tipos de ataques que se van a ejecutar por parte del intruso y en contra parte, el conjunto de reglas que se configuran en el IDS/IPS para alertar y mitigar los mismos.
5. **Recogida y análisis de eventos:** en este capítulo se explica y configura el sistema de recogida de eventos que permite monitorizar las alertas generadas por el IDS/IPS.
6. **Resultados:** análisis de los eventos recogidos por una interfaz gráfica de manera más exhaustiva, prestando especial atención a los contenidos de los paquetes que hacen saltar las alertas.
7. **Conclusiones y líneas futuras:** en este capítulo se comentan las ideas principales extraídas de los resultados obtenidos, se evalúan los objetivos iniciales, y se proponen mejoras y desarrollos futuros en la línea del presente estudio.
8. **Referencias:** un listado ordenado de las referencias bibliográficas utilizadas para conformar el marco teórico y práctico del trabajo.
9. **Anexos:** documentación complementaria que se considera relevante para poder seguir algunos aspectos de la parte práctica del trabajo y profundizar en algunos puntos teóricos.

2. Estado del arte

2.1 La Industria 4.0

A lo largo de las últimas décadas hemos vivido etapas o revoluciones que han marcado un antes y un después en la sociedad. Desde la máquina de vapor, hasta la línea de ensamblaje, a la industria automatizada y recientemente a la industria ciber física, actualmente denominada por muchos la Industria 4.0. Este término fue acuñado originalmente en Alemania y se definió por el Gobierno Federal Alemán como una estructura emergente en la que la fabricación y los sistemas logísticos en forma de Sistemas de Producción Ciber físicos (CPPS), hacen uso intensivo de las redes de información y comunicación disponibles para un intercambio de información ampliamente automatizado, en el que los procesos de producción y de negocio convergen [6].

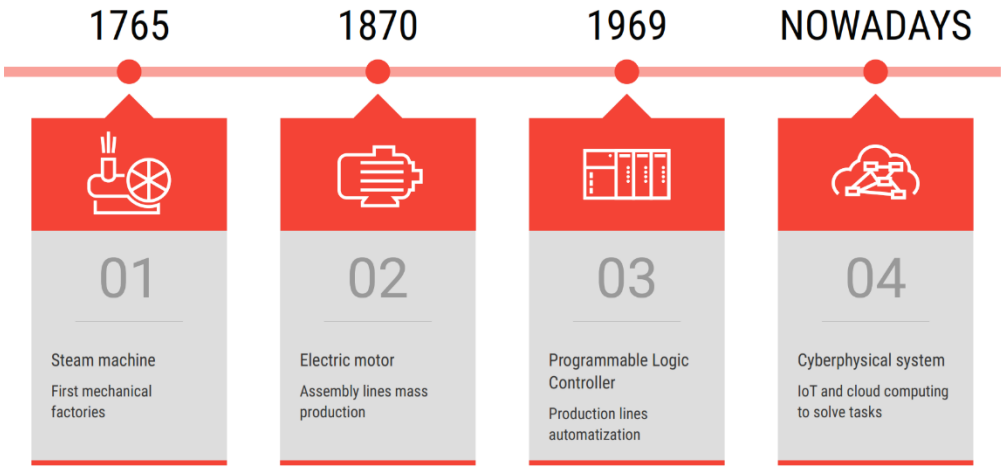


Figura 2. Las 4 revoluciones industriales a lo largo de la historia. Fuente: [7]

Esta última revolución viene marcada por los grandes avances tecnológicos que han traído consigo nuevas tecnologías como la robótica, la inteligencia artificial (IA) y el Machine Learning, las arquitecturas en la nube o el internet de las cosas (IoT), entre otras [8].

La implementación de estas tecnologías en las empresas ha dado paso a nueva forma de operar, que es precisamente la esencia de la Industria 4.0. Este proceso se puede caracterizar como un ciclo donde el acceso en tiempo real a los datos e información, impulsado por un flujo continuo de datos y acciones entre el mundo físico y el digital. Este flujo se puede modelar a través de una serie iterativa de 3 fases: Físico a digital, digital a digital y digital a físico [9].

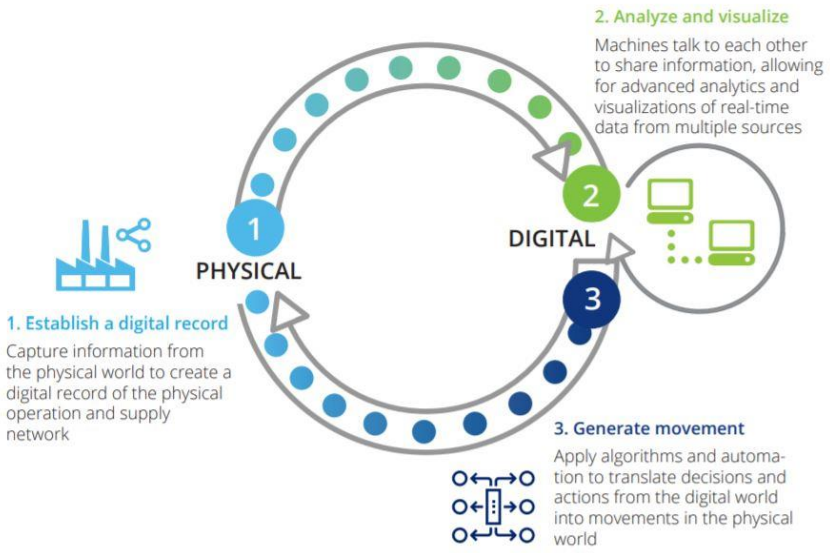


Figura 3. Ciclo físico-digital-físico en la Industria 4.0. Fuente: [9]

La primera fase consiste en capturar información relevante a través de elementos físicos y crear información digital de datos físicos. En la segunda fase, digital a digital, se comparte y analiza la información digital utilizando diferentes tecnologías, como las previamente mencionadas. La última fase, consta de aplicar algoritmos y automatización para trasladar las acciones y decisiones que se han de tomar en función de los anteriores análisis a acciones en el mundo físico. Es esta última fase en la que la habilidad de actuar en base a los datos ya procesados y analizados, la que representa la esencia de la Industria 4.0 [10].

2.1.1 Características de la Industria 4.0

La mayoría de los artículos relacionados con la Industria 4.0 proponen que esta surge a partir de dos tecnologías catalizadoras (IoT y los sistemas ciber físicos), se aplica a un contexto industrial concreto (la cadena de producción) y da pie a mejoras en la intercomunicación (las redes de datos). En consecuencia, combinando los conceptos anteriores se puede entender que la Industria 4.0 está directamente ligada al IoT y a los CPS, empleados en el ámbito de la fabricación y manufacturación, donde existen redes de datos que permiten compartir información obtenida a través de máquinas, productos y personas, o mediante la interconexión de dispositivos inteligentes. La relación entre IoT e Industria 4.0 es tal, que se han llegado a definir términos como el Internet Industrial, o conocido también como IIoT (Industrial Internet of Things) [11].

Así pues, la idea general que viene a proponer la Industria 4.0 gira en torno a la comunicación entre los distintos sistemas que coexisten en una empresa, tanto físicos como digitales. Esta comunicación debe funcionar desde el principio de la cadena de suministro hasta el final, de manera que se produzca ese proceso de realimentación de información descrito en la Figura 3.

Bajo estas premisas existen 4 características que demuestran la enorme capacidad de cambio que tienen la industria y la fabricación tradicional, acercándolas al concepto de Industria 4.0 [12]:

- La primera característica es la integración vertical de los sistemas de producción inteligentes en las fábricas. Para estas conexiones se requiere el uso de sistemas ciber-físicos de producción de manera que, como comentado anteriormente, exista un flujo de comunicación entre los distintos niveles del proceso de fabricación para, por ejemplo, poder reaccionar a variaciones en la demanda o cantidad de stock.

- La segunda característica que define la Industria 4.0 es la integración horizontal a través de nuevas redes de cadenas de valor. Lo interesante de estas nuevas redes es que trabajan en tiempo real, lo que permite una tener una transparencia integrada, ofrecer altos niveles de flexibilidad para responder a situaciones inesperadas y la facilidad de una optimización global.

- La tercera característica principal es la ingeniería intersectorial a lo largo de la cadena de valor y el ciclo de vida del producto. A través de la integración de datos e información asociados a los sistemas de producción, a los productos y a los ciclos de vida de estos, se pueden generar nuevas sinergias entre el desarrollo del producto y los sistemas de producción.

- La cuarta característica es la capacidad de aceleración gracias a las nuevas tecnologías, como la inteligencia artificial, la robótica avanzada o los sensores inteligentes. Mediante estas tecnologías se pueden alcanzar soluciones más flexibles, individualizadas y eficientes.

2.1.2 Automatización en el entorno industrial

Un modelo actualmente muy extendido que recoge las tecnologías presentes en la automatización y gestión de procesos productivos, categorizados en distintos niveles o redes, es el conocido como la pirámide de la automatización, representado en la Figura 4. Este modelo es previo a la revolución de la Industria 4.0, pero siendo la automatización un pilar fundamental en esta nueva era industrial, se puede considerar que la Industria 4.0 engloba la pirámide de la automatización [13].

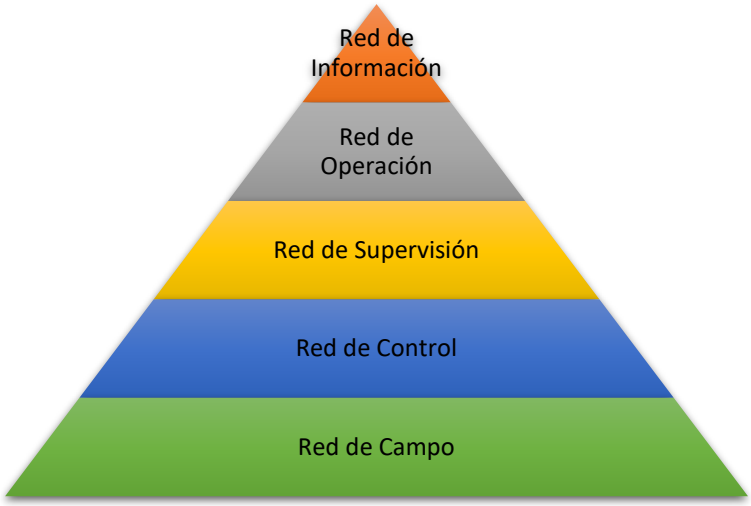


Figura 4. Pirámide de la automatización. Fuente: Elaboración propia.

Además, comprender la estructura y los elementos que componen los distintos niveles de este modelo es fundamental para poder establecer medidas de seguridad en los entornos de automatización y control industrial. Por ello, se detallan a continuación cada uno de los 5 niveles o redes que componen la pirámide.

Nivel 1: Red de campo

La red de campo es el primer nivel de la pirámide de la automatización. Este nivel está compuesto por dispositivos de entrada y salida que se encargan de generar, enviar y recibir información, generalmente bits o bytes, en una planta industrial. Estos dispositivos están comunicados con la red de control, mediante la cual se intercambia información a través de un bus de campo. Esto último con el objetivo de informar a los dispositivos de control sobre los eventos que ocurren en la planta (vaciado de un tanque, temperatura límite superada, maquina detenida, etc.) y para actuar en caso de recibir una orden. Entre estos dispositivos se pueden encontrar sensores, actuadores, interruptores, válvulas, motores eléctricos, entre otros.

Nivel 2: Red de control

La red de control se encarga de recoger la información obtenida por los dispositivos de entrada de la red de campo, procesarla y enviar órdenes de vuelta para que se ejecuten acciones. El nivel 2 transmite información tanto hacia el nivel 1 (red de campo) como al nivel 3, donde se encuentra la red de supervisión. El nivel 2 suele disponer de una capacidad considerable de almacenamiento de información, así pues, es transmitida al nivel 3, y a continuación se decide dónde almacenarla, si en ese nivel o en niveles superiores [14].

Generalmente, este nivel se compone de elementos de control como los PLC o (Controlador Lógico Programable) que son dispositivos a los cuales se les implementa lógica para que generen ciertos datos de salida en función de los que reciben a la entrada.

Nivel 3: Red de supervisión

La supervisión y el control de los niveles anteriores se puede llevar a cabo desde esta red, donde lo más común es disponer de un sistema SCADA (Sistema de control, supervisión y adquisición de datos) que permita monitorizar de manera más sencilla la información relacionada a un proceso al usuario encargado de ello. Para la visualización de los datos se suele hacer uso de HMIs (interfaz hombre-máquina), pequeños paneles que se pueden encontrar distribuidos en la red de control junto a los PLC y un monitor central desde donde se pueda manejar toda la información de una planta.

Nivel 4: Red de operación

En este nivel se emplean generalmente dos sistemas para poder gestionar las operaciones de una planta, estos sistemas son los historizadores y los MES (Sistema de ejecución de fabricación). Los primeros se utilizan para almacenar información de procesos e infraestructura, teniendo la capacidad de gestionar grandes cantidades de datos tanto analógicos como digitales [15].
Por otro parte los MES tienen la función de controlar y monitorizar los procesos de producción en tiempo real, manejando datos relacionados a la trazabilidad del producto, control de calidad, rendimiento de la maquinaria, recursos empleados, entre otros.

Nivel 5: Red de información

El nivel más alto de la pirámide es donde lleva a cabo la planificación y la gestión de recursos a nivel global en la empresa, para ello se usan las plataformas ERP (Planificación de recursos del negocio) donde se maneja información relacionada con clientes, proveedores, costes, gestión de proyectos, etc.

2.1.3 Ciberseguridad en la Industria 4.0

No cabe duda de que la aparición de la Industria 4.0 ha permitido un importante avance en los procesos industriales, pero esta a su vez ha traído consigo riesgos intrínsecos muy a tener en cuenta para las empresas que optan por adaptarse al modelo de interconexión de procesos y sistemas que propone esta nueva era industrial.

Los riesgos cibernéticos suponen una gran amenaza sobre todo porque muchos de ellos son complicados de entender y de prever. El aumento de estos tipos de ataques ha venido favorecido por la alta conectividad que requiere la Industria 4.0. Hoy en día lo más común es encontrarse con sistemas de control industrial donde la conectividad esté asentada sobre TCP/IP y Ethernet o el uso de sistemas inalámbricos estandarizados [1]. Pero a pesar de que estos protocolos, entre otros, ofrecen cierto de valor de madurez y fiabilidad, pueden ser explotados por ciberdelincuentes.

Dado que tradicionalmente los sistemas de control industrial funcionaban de forma aislada sin depender de otras infraestructuras como la TI, las vulnerabilidades que presentaban estos sistemas eran complejos de explotar al no existir un acceso directo, salvo el físico, a los elementos y dispositivos. La adopción de la Industria 4.0 supone una conexión de los sistemas de control industrial, tanto con los sistemas corporativos como con Internet, lo que hace que aumente la superficie de exposición de los sistemas a posibles ciber atacantes [16].

Según un informe publicado por Claroty [4], destacan que las principales vulnerabilidades de los dispositivos y componentes, tanto en la red campo como en las de control y operaciones, se deben a fallos en el software o firmware de estos. Esto ocurre en muchos casos por que en los sistemas de control industrial la actualización de los componentes software conlleva varias dificultades, entre las que se encuentran: un prolongado ciclo de vida de los componentes que no les permite ser actualizados con las últimas tecnologías de seguridad, y una interdependencia entre los dispositivos y componentes que implica llevar a cabo extensas pruebas para garantizar un correcto funcionamiento [16].

Con lo anterior en cuenta las mejores soluciones prácticas de ciberseguridad pueden ser aquellas que implican la mitigación de ciberataques, entre las más recomendadas [4] se encuentran:

- Segmentación de redes: es una forma importante de control y seguridad, ya que el tradicional “air-gap” industrial está desapareciendo a medida que las empresas van moviendo su información, infraestructura y servicios a la nube.

- Seguridad de acceso remoto: mejorar este aspecto prevendrá accesos indeseados a través de conexiones no locales.

- Protección contra ransomware, phishing y spam: el uso de aplicaciones y software contra estos tipos de ciberincidentes puede evitar posibles daños mayores.

- Monitorización y restricción de tráfico: la capacidad de inspeccionar tráfico es crucial para defenderse contra comportamientos anómalos.

2.2 Ciberincidentes

Un incidente se puede considera como un abanico ilimitado de posibles eventos de seguridad, impredecibles y no programados, con alta probabilidad de compromiso o interrupción del negocio y amenaza a la seguridad corporativa. En la misma línea, un ciberincidente se refiere a un incidente relacionado a una infraestructura tecnológica en la que interactúan personas, procesos, datos y sistemas de información.

2.2.1 Clasificación

Ya que no todos los Ciberincidentes tienen las mismas características e implicaciones, la comunidad de CSIRT (centros de respuesta frente a incidentes) consideró conveniente agruparlos, creando así una taxonomía de ciberincidentes. En la Tabla 1 se enumeran los Ciberincidentes reconocidos hasta la fecha clasificados por grupos.

Clasificación	**Tipo de incidente**
Contenido abusivo	-Spam -Delito de odio -Contenido sexual o violento inadecuado
Contenido dañino	-Sistema infectado -Servidor C & C -Distribución de malware -Configuración de malware
Obtención de información	-Escaneo de redes (*scanning*) -Análisis de paquetes -Ingeniería social
Intento de intrusión	-Explotación de vulnerabilidades conocidas -Intento de acceso con vulneración de credenciales -Ataque desconocido
Intrusión	-Compromiso de cuentas con privilegios -Compromiso de cuenta sin privilegios -Compromiso de aplicaciones -Robo
Disponibilidad	-DoS (Denegación de servicio) -DDoS (DoS distribuido) -Mala configuración -Sabotaje
Compromiso de la información	-Acceso no autorizado a información -Modificación no autorizada de información -Pérdida de datos
Fraude	-Uso no autorizado de recursos -Derechos de autor -Suplantación -Phishing
Vulnerable	-Criptografía débil -Amplificador DDoS -Servicios con acceso potencial no deseado -Revelación de información -Sistema vulnerable
Otros	-Otros -APT (Amenaza avanzada persistente)

Tabla 1. Clasificación de los Ciberincidentes. Fuente:[17] y elaboración propia.

2.2.2 Procedimiento de un ciberataque

Una de las claves para poder detectar y detener un ciberataque es comprender cuál es su ciclo de vida, es decir, cuál es su proceso de desarrollo, para poder establecer una serie de pasos y medidas que garanticen cierto nivel de seguridad en caso de ser víctima de un ciberataque. Este proceso creado por analistas de la compañía Lockheed Martin se denomina la *Cyber-Kill Chain* [18]. En la Tabla 2 se pueden apreciar las distintas etapas y como detectar y prevenir el ataque en cada fase.

Nº fase	**Nombre fase**	**Detección**	**Prevención**
1.	Reconocimiento	Analítica de web	Firewall ACL
2.	Armamento	NIDS	NIPS
3.	Entrega	Vigilancia de usuario	Filtro proxy
4.	Explotación	HIDS	Parche
5.	Instalación	HIDS	Cárcel chroot
6.	Comando y control	NIDS	Firewall ACL/NIPS
7.	Acciones sobre objetivos	Registro de auditoría	-

Tabla 2. Fases de la Cyber-Kill Chain y sus soluciones de detección y prevención. Fuente [18] y elaboración propia.

Es cierto que este cuando se publicó este modelo estaba orientado hacia el entorno corporativo, debido a la naturaleza de los sistemas y los ataques que iban dirigidos a ellos, por eso no se puede aplicar directamente el modelo a los sistemas de control industrial. A pesar de eso el instituto SANS publicó un informe [19] donde adaptaban la Cyber-Kill Chain al entorno Industrial, denominado Industrial Control System Cyber-Kill Chain.
La ICS Cyber-Kill Chain está dividida en dos fases, la primera es muy similar a la representada en la Tabla 2 y está representada en la Figura 5. Esta fase se puede categorizar como un proceso de espionaje o adquisición de inteligencia y ataque a los propios sistemas IT, ya que la misión principal es adquirir información acerca del ICS para conocer mejor el sistema y proporcionar mecanismos para conseguir acceso al entorno de producción.

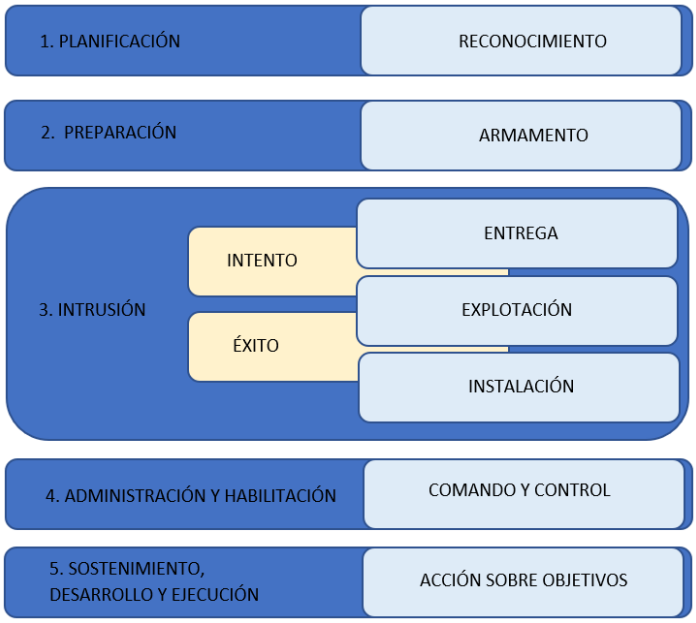


Figura 5. Fase 1 ICS Cyber-Kill Chain. Fuente [19] y elaboración propia.

Es en la segunda fase, donde mediante la información obtenida a lo largo de la fase 1 se desarrolla, pone a prueba y se utiliza alguna herramienta capaz de atacar el ICS. Este proceso se puede ver representado en la Figura 6.

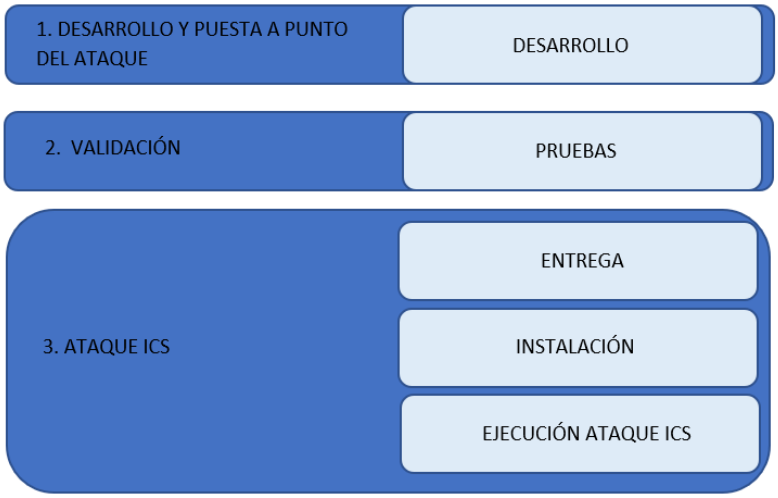


Figura 6. Fase 2 ICS Cyber-Kill Chain. Fuente [19] y elaboración propia.

2.3 Sistemas de detección y prevención de intrusos

En la actualidad existen multitud de herramientas, aplicaciones y sistemas que permiten mejorar la seguridad de los dispositivos y las subredes que componen una red industrial, como, por ejemplo, Firewalls, DMZs, Antivirus y Antimalware, VPNs, etc. En este trabajo los sistemas bajo estudio son los IDS e IPS, es por ello por lo que en esta sección se presentan descripciones pormenorizadas de ambos.

2.3.1 IDS

Un sistema de detección de intrusos o IDS (Intrusion Detection System en inglés) es una aplicación que se encarga de la monitorización del tráfico que circula por una red para poder detectar actividades sospechosas que posiblemente supongan una amenaza.

Los IDS se pueden clasificar en distintos grupos que se particularizan por las siguientes características: tipo de detección, ubicación, la estructura y el comportamiento [20].

Tipo de detección

- **Detección de anomalías**: estos sistemas se caracterizan por utilizar metodologías estadísticas para detectar comportamientos anómalos. Los sistemas más avanzados de esta categoría se basan en el aprendizaje automático que permita la diferenciación entre un compartimento normal y anómalo. En un punto intermedio estarían aquellos sistemas que, en función de un perfil de actividad, definido por el comportamiento de un usuario, usa métricas y estadísticas para valorar anomalías. El sistema más básico de este tipo se basa simplemente en las violaciones detectadas mediante el uso de reglas ya definidas.

- **Detección por firmas o reglas**: en este caso los sistemas de detección se basan en usos incorrectos, es decir, se monitorizan los eventos que ocurren en un sistema y se comparan con bases de datos que contienen registros de firmas o reglas de ataques ya conocidos.

Ubicación

- **HIDS (*Host-based IDS*):** Estos tipos de IDS funcionan para un único host, es decir, solo se monitoriza el tráfico se detectan intrusiones para una máquina en concreto.

- **NIDS (*Network-based IDS*):** Totalmente opuesto al HIDS, este IDS se instala de manera que se realice una monitorización de todo el tráfico que circula por una red o subred.

Estructura

- **DIDS (*Decentralized IDS*):** Se basa en la distribución de varios IDS en una misma red, los cuales se pueden comunicar entre ellos. A través de un nodo central. Esto facilita la monitorización global de la red y permite gestionar las posibles amenazas con mayor fiabilidad.

- **CIDS (*Centralized IDS*):** En este caso los IDS que están distribuidos por la red no se comunican entre ellos y simplemente trasmiten información a un servidor central que se encarga de la gestión de eventos, alertas, etc.

Comportamiento

- **Pasivo:** Esta es la característica que define al IDS tradicional, ya que el sistema solo tiene la función de analizar tráfico y generar alertas en caso de que detecte una posible amenaza o intrusión.

- **Activo:** En el caso de estar ante un IDS que presente un comportamiento activo, se está hablando de un IPS. A diferencia del IDS, estos sistemas van un paso más allá a la hora de lidiar con un ataque. Lo que permiten los IDS es establecer reglas que detengan una intrusión, por ejemplo, bloqueando o rechazando paquetes que contengan contenido malicioso.

2.3.2 IPS

Como se ha mencionado antes, los IPS pueden ser considerados como una extensión de los sistemas de detección de intrusos (IDS), con la capacidad adicional de bloquear tráfico, similar a la tarea que realiza un cortafuegos. Por lo que un IPS se puede definir como un dispositivo o software que permite detectar y actuar de manera proactiva contra tráfico malicioso que presenta una amenaza para un equipo o red.

Al igual que los IDS los IPS se pueden clasificar en distintos grupos en base a ciertas características [20], concretamente en función de:

Ubicación

- **HIPS (*Host-based IPS*):** El sistema IPS ofrece sus funcionalidades para un único equipo.

- **NIPS (*Network-based IPS*):** Este sistema monitoriza la red o subred para prevenir posibles ataques.

- **WIPS (*Wireless-network-based IPS*):** Similar a las NIPS, pero utilizado para redes inalámbricas.

- **NBA (*Network Behaviour Analysis*):** funciona a nivel de red como NIPS, pero con la particularidad que se basa en parámetros del tráfico como, paquetes por segundo, número de conexiones por host, etc., para detectar anomalías

Funcionalidad

- NIDS en línea
- Switches a nivel de capa de aplicación
- Cortafuegos/IDS de Aplicación
- Switches Híbridos
- Aplicaciones engañosas

Comparativamente, los IPS son por una parte superiores a los IDS por la ventaja que le otorga el hecho de poder detectar una intrusión y detenerla, mediante el uso de reglas habilitadas para ello. Debido a esto el número de alertas de posibles alertas de amenazas será más reducido. Otro punto a favor que va de la mano de esto último es que el tiempo de reacción frente a una intrusión será considerablemente menor en el caso de usar un IPS. Por otro lado, la capacidad de reacción proactiva de los IPS tiene una desventaja que se debe tener en cuenta y es que el IPS reaccione a falsos positivos. Esto es algo a tener presente a la hora de la configuración de reglas y filtros que bloquean del tráfico ya que si esta casuística ocurre en exceso puede llevar a la denegación de servicios e inoperatividad de los sistemas que ocupen la red.

También es interesante destacar las diferencias que existen entre los IPS y los cortafuegos, ya que a priori se tiende a pensar que la manera de controlar el tráfico es la misma, pero no es el caso. Los cortafuegos (firewall) son sistemas de seguridad compuestos o bien de programas (software) o de dispositivos hardware que tienen el objetivo de permitir y limitar, el flujo de tráfico entre los diferentes ámbitos que protege sobre la base de un conjunto de normas y otros criterios [21]. Hasta aquí parecen sistemas con una funcionalidad similar, pero la diferencia radica en la información de la que se sirve un cortafuegos para filtrar el tráfico. La información que tiene en cuenta el firewall a la hora de permitir el flujo de información se basa en los contenidos de la cabecera de un paquete (dirección de origen, dirección de destino, protocolo, puerto de origen y puerto de destino), capa de red y capa de transporte [22]. Por otro lado, los IPS, además de tener en cuenta los parámetros anteriores también tienen la posibilidad de analizar el contenido de datos del paquete, otorgándole una mayor fiabilidad a la ahora de bloquear y rechazar paquetes.

2.3.3 SIEM

El acrónimo SIEM proviene de *Security Information and Event management* o traducido, Información de Seguridad y Gestión de Eventos. Para definir de una más correcta este término, lo primero es conocer que SIEM es una combinación de las categorías SIM o gestión de información de seguridad y SEM o gestor de eventos de seguridad. El primero está orientado a la monitorización en tiempo real, correlación de eventos y notificaciones de alertas, mientras que el segundo se encarga de recopilar y organizar datos para que se puedan generar variedad de informes que proporcionen información valiosa. Mediante el funcionamiento de ambos en conjunto se conforma el SIEM, obteniendo como resultado una solución centralizada que permite una actuación rápida y eficaz frente a multitud de amenazas.

Estos sistemas resultan muy interesantes a la hora de utilizarlos en empresas o fábricas ya que permite la integración de múltiples fuentes de recopilación de datos informáticos, como por ejemplo varios sistemas IDS o IPS ubicados en distintos puntos de una red o incluso en redes distintas dentro de una organización.

2.4 Herramientas IDS/IPS

Actualmente existen multitud de herramientas de código abierto para la detección y prevención de intrusos, aunque cada una cuenta con sus propias funcionalidades y características. Es por ello, que en esta sección se describen en primer lugar 4 de ellas. Posteriormente se detalla más a fondo el funcionamiento de Snort 3, ya que es con el que se va a trabajar en la parte práctica. La decisión de utilizar Snort 3 se fundamenta en las posibilidades que ofrece funcionando tanto como IDS como IPS, la gran comunidad que posee, la frecuencia de mejoras y actualizaciones, las aplicaciones de terceros con las que se puede integrar y la posibilidad de usar reglas (ya definidas) actualizadas regularmente para detectar los ataques conocidos más recientes.

2.4.1 Comparación de las alternativas

Snort

Snort es uno de los IDS/IPS de código abierto más reconocidos en la actualidad, mantenido por la empresa Cisco Talos. Como es de esperar, Snort es capaz de analizar datos en tiempo real y dejar un registro de los paquetes capturados. Así pues, con Snort se puede detectar y neutralizar tráfico potencialmente malicioso dirigido a una red antes de que el ataque alcance cualquier máquina. Esto lo hace desechando tráfico malicioso fuera de la red antes de que llegue a su destino.

Actualmente Snort cuenta con dos versiones bastante diferenciadas, Snort 2 y Snort 3. La más actual, Snort 3, y no por ello la más utilizada, posee ciertas ventajas sobre su predecesora, entre las que se pueden destacar:

- La capacidad de procesamiento de paquetes multihilo, permitiendo la mejor utilización de recursos en cuento a memoria RAM.
- Soporte para Hyperscan, que permite la detección de patrones a una mayor velocidad.
- El fichero de configuración se ha simplificado, pudiendo configurar Snort de manera más eficiente.
- La sintaxis de escritura de reglas es más sencilla e intuitiva.

Suricata

Suricata es un motor de detección de código abierto, que puede actuar tanto de IDS como de IPS. En esencia es muy similar a Snort, ya que permite detectar y bloquear paquetes que supongan una amenaza para una red. El proyecto de Suricata y el propio código lo mantiene la fundación OISF, por lo que es una opción muy fiable ya que cuenta con el respaldo de esta y de una amplia comunidad. Prueba de ello es la frecuencia con la que se realizan mejoras y actualizaciones al código, la última de ellas siendo Suricata 6.0.4.

Zeek

Zeek, anteriormente conocida como Bro, es una plataforma de código abierto de monitorización de seguridad de redes. Esta plataforma se distingue de los IDS/IPS vistos anteriormente ya que los usuarios de Zeek únicamente tienen la posibilidad de monitorizar la red, es decir, de funcionar como IDS, sin tener la posibilidad de bloquear y detener paquetes como haría un IPS. Una de las ventajas que tiene Zeek es que ofrece la posibilidad de probar el software a través de su página web sin necesidad de realizar la instalación y configuración completa.

OSSEC

OSSEC es otra alternativa de software de código abierto capaz de monitorizar tráfico a nivel de host, caracterizándolo pues como un HIDS. Esto quiere decir que con OSSEC no es posible monitorizar una red mediante un solo despliegue si no que es necesario configurarlo en varios puntos de la red. A pesar de que con OSSEC no se tiene directamente la capacidad de bloquear tráfico, es posible integrarla con aplicación de terceros que ofrezcan esta opción como por ejemplo Cloudflare.

2.4.2 Snort 3

Snort soporta varias configuraciones que lo hacen funcionar en distintas modalidades:

- **Modo *sniffer***: en este modo, Snort captura los paquetes en tiempo real y se muestran de manera continua a través de la consola.
- **Modo registro de paquetes**: los paquetes capturados por Snort se guardan en un fichero.
- **Modo NIPS:** en este modo los paquetes capturados se comparan con las reglas o patrones de detección que se tengan establecidos, mostrando por la consola alertas en caso de que se produzca alguna coincidencia. Este es el modo más potente a nivel de configuración, pero a su vez el más complejo de manejar.

Los paquetes capturados por Snort siguen todos un proceso, pasando por varios módulos que permiten descomponer el paquete y extraer la información para poder tomar la decisión de dejar pasar un paquete o no. El ciclo de vida de un paquete que se recibe en Snort viene representado en la Figura 7.

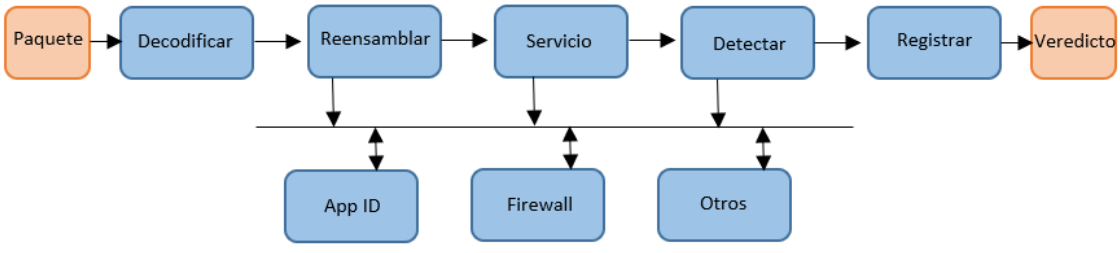


Figura 7. Módulos por los que pasa un paquete al ser procesado por Snort 3. Fuente: [23] y elaboración propia.

El proceso comienza cuando un paquete entra en la red y se recibe en una máquina que tenga Snort instalada. El paquete se decodifica para determinar las caracteristicas básicas de la red como el origen y destino de las direcciones IP y los puertos.. Además, también se examinan las cabecaras de los protocolos (Ethernet, IP, TCP y HTTP) que encapsulan el contenido de los paquetes durante la decodificación.

Después de pasar por los decodificadores el paquete pasa por un preprocesado, esto se hace con la intenciónde organizar los datos de tal manera que sean manejables para el bloque de detección. El preprocesado puede implicar un reensamblaje de fragmentos IP y de la secuencia TCP, y un servicio de análisis y normalización de los datos para que puedan ser usandos más adelante Este proceso se realiza mediante la ayuda de plugins (de configuración opcional) como OpenApp ID.

Una vez preprocesado, el paquete pasa al modulo de detección, donde se compara con las reglas que estén configuradas. Mediante esto, Snort marca el tráfico que puede ser malicioso y decide si rechazar/bloquear el tráfico para que la maquina de destino no lo reciba o lo deja pasar sin realizar ninguna modificación.

El último bloque sería la salida de estos eventos que bien se pueden mostrar por consola o registrar en un fichero para posteriormente revisar los eventos de intrusión y los paquetes asociados.

2.4.3 Reglas de Snort 3

Las reglas en Snort son la forma de detectar si el contenido de un paquete puede ser malicioso, es por esto por lo que la configuración de estas es de vital importancia para el correcto funcionamiento del IDS/IPS. Para poder definir reglas propias es fundamental conocer su sintaxis que se define a continuación:

```
[acción][protocolo][IP origen][puerto origen] -> [IP destino][puerto destino]
([Opciones de regla])
```

La cabecera de la regla contiene la acción de la regla, protocolo, IP de origen y de destino, y puertos de origen y destino. Algunos de los ítems tienen una lista de valores a elegir ya preestablecidos como, por ejemplo:

- **Acción**: este valor indica que debe hacer Snort cuando un paquete coincida con el criterio de una regla. Las posibles opciones son: alert, log, pass, drop, reject y sdrop.
- **Protocolo**: el tipo de protocolo para el que se analizan anomalías. Existen cuatro opciones de protocolos: TCP, UDP, ICMP e IP.
- **IP origen**: dirección(es) de origen del paquete (puede configurarse como *any*).
- **Puerto origen:** puerto(s) o puertos de origen (puede configurarse como *any*).
- **Operador de dirección:** Indica la dirección del tráfico al que se aplica la regla. Las dos opciones son: `->` o `<-`
- **IP destino:** dirección(es) de destino del paquete (puede configurarse como *any*).
- **Puerto destino:** puerto(s) de destino del paquete (puede configurarse como *any*).

El ítem que queda al final de la regla son las opciones y forman la base del motor de detección de Snort. Se pueden incluir varias opciones en este ítem separándolas por una coma. Las opciones se dividen en cuatro grandes categorías, las genérales, *payload*, *non-payload* y las *post-detection*. El primer grupo cuenta con opciones que proporcionan información sobre la regla, pero no afectan a la hora de la detección. Las segundas, se utilizan para buscar datos dentro del contenido del paquete, mientras que las *non-payload* no buscan datos dentro del contenido sino otros parámetros como el tiempo de vida o el tamaño del contenido del propio paquete. El último grupo recoge

opciones relacionadas a desencadenantes específicos que ocurren tras la activación de una alerta. En las tablas a continuación se pueden ver descritas algunas de las opciones más útiles de cada grupo.

Palabra clave	Descripción
msg	Se usa para definir brevemente la regla mediante una cadena de texto.
reference	Se usa para referenciar fuentes de información externas que puedan resultar útiles.
classtype	Cadena de texto que indica el posible efecto si el ataque o intrusión resultase exitoso.
sid	Abreviación de Snort id, un identificador único para cada regla.

Tabla 3. Algunas opciones de la categoría general. Fuente: [24] y elaboración propia.

Palabra clave	Descripción
content	Permite al usuario configurar reglas que busquen información específica dentro del contenido del paquete.
distance/offset	Sirve para indicar donde empezar a buscar relativo al punto de comienzo del contenido del paquete o de la coincidencia de contenido
within/depth	Sirve para lo mismo que distance/offset, pero empezando desde el final del contenido del paquete.
pcre	Habilita la escritura de reglas usando expresiones compatibles con *perl* para realizar búsquedas más complejas y concretas.
byte_test	Permite que la regla realice una prueba de varios bytes contra un valor en binario.

Tabla 4. Algunas opciones de la categoría *payload*. Fuente: [24] y elaboración propia.

Palabra clave	Descripción
ttl	Revisa el contenido del campo *Time to live* en la cabecera de IP
dsize	Testea el tamaño del contenido del paquete
ack	Comprueba el valor concreto de un ACK en TCP
icmp_id	Revisa el valor concreto del ID de ICMP

Tabla 5. Opciones en la categoría *non-payload*. Fuente: [24].

Palabra clave	Descripción
logto	Registra los paquetes que generan una alerta con esta opción en un fichero especial
react	Posibilita al usuario a reaccionar a tráfico que coincide con una regla cerrando la conexión y generando una notificación
replace	Reemplaza el contenido “malicioso” que coincide con la regla y lo reemplaza por una la cadena proporciona.

Tabla 6. Opciones en la categoría *post-detection*. Fuente: [24] y elaboración propia.

Un ejemplo de una regla bastante sencilla de Snort puede ser la siguiente:

```
alert icmp any any -> 192.168.1.100 any ( msg: “Prueba detección ping”; classtype:
“Es solo un Ping” ; icmp_id=8; sid 100002);
```

Esta regla tiene activada la acción de *alert*, por lo que en caso de coincidencia generará una alerta. Además, detectará paquetes ICMP que se dirijan a la dirección IP privada 192.168.1.100 desde cualquier origen y puerto (ambos configurados como *any*). En las opciones se especifica un número para el id del ICMP, de manera que solo se generará la alerta si esta coincide con el valor 8 (correspondiente a un *echo request*). También se añaden varias opciones generales como son *msg*, *classtype* y *sid*, donde el *sid* es el identificador único de la regla, *msg* y *classtype* son cadenas de texto descriptivas que proporcionan información sobre la alerta.

3. Consideraciones previas

Entrando ya en la parte práctica de este trabajo, lo primero de todo es introducir el contexto y la situación que se pretende recrear. Como se ha visto en el estudio teórico, las redes ubicadas en los niveles inferiores de una fábrica, como son las redes de campo y las de control, son ahora más propensas a recibir ataques e intrusiones debido a la conectividad global de los sistemas en la fábrica. Es por esto, que para incrementar la seguridad de la red es esencial tener cierto control sobre el tráfico de datos. Con todo lo anterior en cuenta la situación que se propone es que un atacante ha conseguido penetrar hasta una de las varias redes de control industrial que conforman el nivel 2 de una fábrica donde va a llevar a cabo acciones maliciosas.

Para simular este escenario, lo primero que se lleva a cabo es la configuración de una pequeña red local que pueda estar embebida en cualquier sistema de control industrial en el nuevo contexto que propone la Industria 4.0. Además, que pueda servir como base para trabajos futuros y de mayor complejidad. Todo esto para después llevar a cabo un estudio del tráfico generado por la máquina atacante, con la intención de poder detectar sus movimientos, monitorizando el tráfico de paquetes mediante un IPS/IDS (Snort 3) y analizando las alertas e información generada por este último a través de la interfaz web que proporciona la herramienta Splunk.

En la primera sección de este capítulo se describe como está configurada la red local, los elementos que la componen y algunos datos relevantes de cada uno, como por ejemplo el sistema operativo y la IP que tienen asignada. En la siguiente sección, se entra mucho más en detalle en los cambios que se han realizado en las configuraciones de cada dispositivo, así como cualquier instalación de software necesaria.

3.1 Arquitectura de la red modelo

En el laboratorio se realiza el montaje de la red mostrada en la Figura 8, que pretende modelar a grandes rasgos una red de control industrial de nivel bajo, donde los dispositivos más característicos que se pueden apreciar son un par de PLC acoplados en un mismo puesto de control, el cual también dispone de un HMI. Estos están conectados entre sí mediante un switch, que a su vez estará conectado a un switch principal donde también se tendrá conectado un PC.

Snort 3, se instala en una máquina virtual desde donde se monitorizará la actividad del intruso y se analizará a través de una interfaz gráfica. Para simular el equipo atacante se habilita también otra máquina virtual en el mismo equipo que el IDS. Haciendo esto y realizando algunas modificaciones en las configuraciones de red de ambos, se consigue que todo el tráfico generado por la maquina atacante sea capturado por Snort.

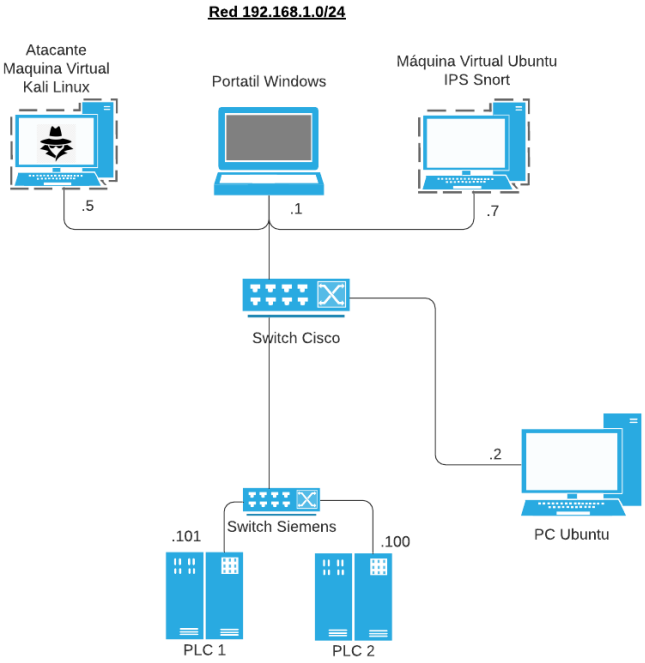


Figura 8. Modelo de red de control industrial de nivel bajo montada en el laboratorio. Fuente: Elaboración propia.

3.2 Elementos de la red

3.2.1 Información de los equipos

Conociendo ya la arquitectura de la red con la que se va a trabajar, a continuación, se detallan las máquinas y dispositivos que la componen:

Puesto de control

- PLC Siemens S7-1500 (CPU 1516F-3 PN/DP)
- PLC Siemens S7-1500 (CPU 1511-1 PN)
- Switch Siemens Scalance X208

Ordenador Sobremesa

- DellEMC PowerEdge T40 con sistema operativo Ubuntu 20.04

Ordenador portátil

- Huawei Matebook 14 con sistema operativo Windows 10 Home
- Máquina virtual (VirtualBox) con sistema operativo Ubuntu 20.04
- Máquina virtual (VirtualBox) con sistema operativo Kali Linux 2021.3

Switch

- Cisco Catalyst 3560 v2 series

3.2.2 Configuración de los equipos

Puesto de control

Por simplicidad, la configuración de los dispositivos del puesto de control es mínima, es decir, únicamente se realiza la configuración de red de estos. Para hacer esto se hace uso del software proporcionado por Siemens, TIA Portal V16, instalado en un ordenador y conectado mediante ethernet al switch Scalance X208. En la Figura 9 se puede observar cómo queda la conexión de los elementos del puesto de control a través del TIA Portal. Prestando atención a la IP de cada elemento se puede apreciar que todas forman parte de la misma red, en concreto la 192.168.1.0/24. El resto de los sistemas, como se verá a continuación, también forman parte de la misma red.

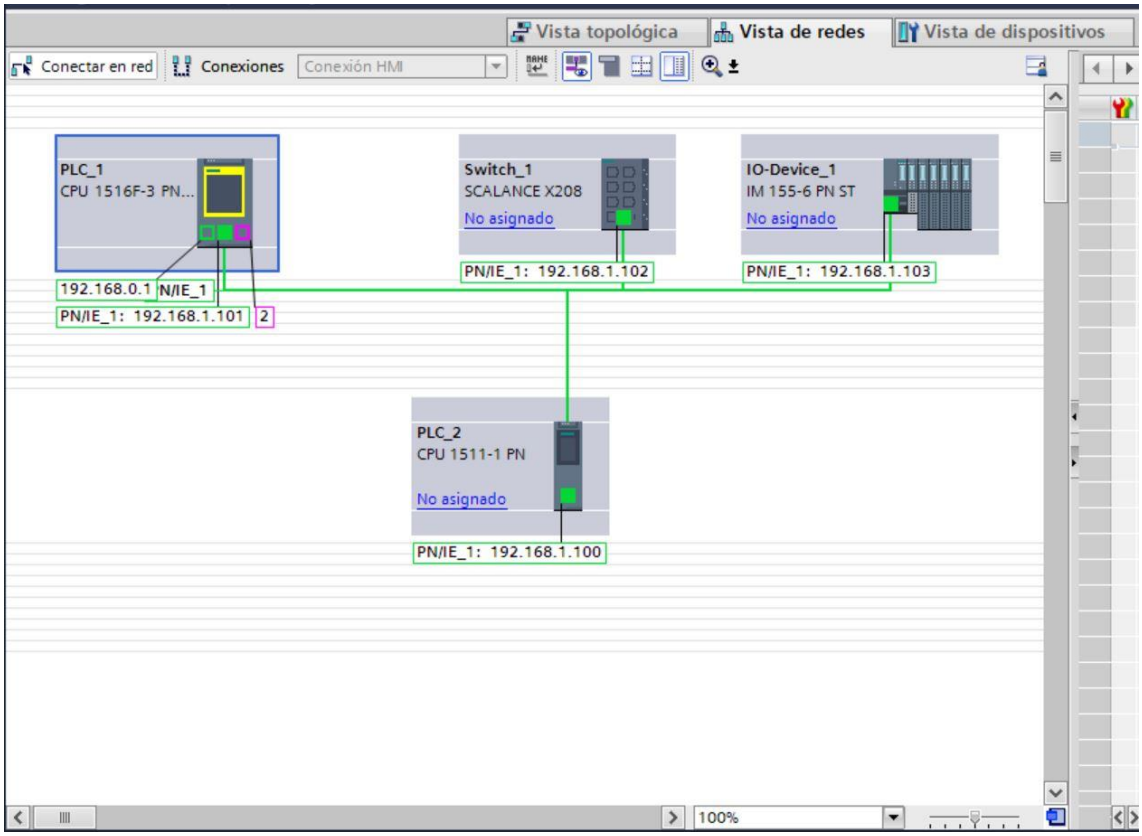


Figura 9. Vista de red de control desde TIA Portal. Fuente: Elaboración propia.

Ordenador de Sobremesa

En esta máquina la única configuración que se lleva a cabo es la asignación de una IP estática de manera que el equipo forme parte de la red 192.168.1.0/24. Para ello, desde el terminal se ejecuta lo siguiente:

```
cd /etc/netplan
sudo nano 00-installer-config.yaml
```

Se edita el fichero *00-installer-config.yaml* de manera que contenga lo siguiente:

```
network:
  ethernets:
    eno1:
      addresses: [192.168.1.2/24]
  version: 2
```

Para aplicar la configuración:

```
sudo netplan apply
```

Ordenador Portátil

Máquina virtual atacante:

A la máquina virtual atacante se le instala el sistema operativo Kali Linux, ya que este SO está diseñado principalmente para la auditoria y seguridad de redes informáticas, poniendo a disposición del usuario diversas herramientas para ello. Mediante dichas herramientas o software, se generará tráfico malicioso por la red bajo estudio con la intención de detectarlo mediante el IPS/IDS. Principalmente se van a utilizar los programas de código abierto Nmap y Metasploit, los cuales se mencionan con más detalle en el capítulo 4. Para instalarlos simplemente hay que ejecutar los siguientes comandos:

```
sudo apt-get install nmap

sudo apt install metasploit-framework
```

Para que esta máquina virtual se pueda comunicar con los elementos físicos de la red es preciso establecer el modo adaptador puente en la propia configuración de VirtualBox, esto se puede ver en la Figura mostrada a continuación.

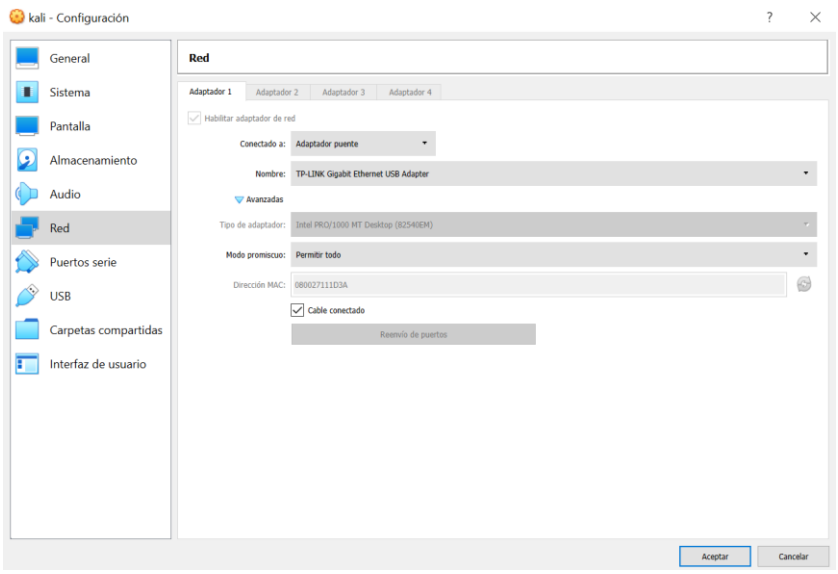


Figura 10. Adaptador de red configurado en modo adaptador puente. Fuente: Elaboración propia

Al igual que con el PC, se asigna una IP estática a esta máquina virtual, en este caso se hace a través de la GUI que ofrece este sistema operativo para la configuración de redes. A continuación, se muestra la IP que se asigna a la máquina atacante.

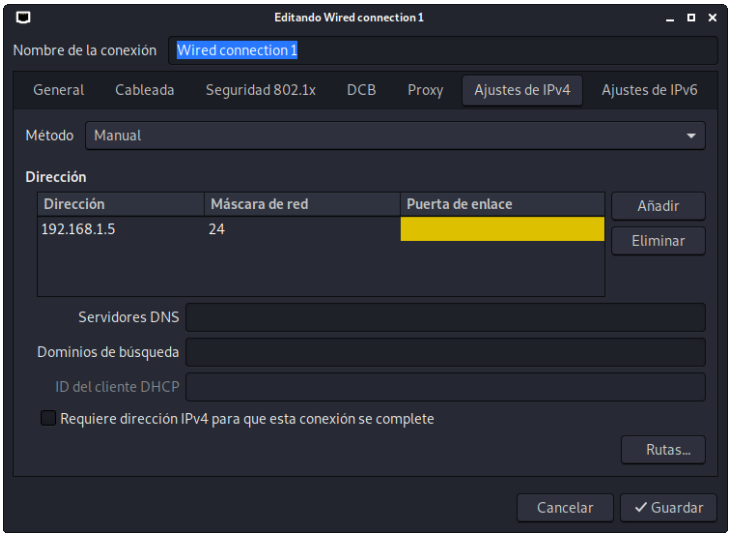


Figura 11. Asignación IP estática MV atacante. Fuente: Elaboración propia.

Máquina virtual IDS/IPS:

En primer lugar, hay que realizar varios ajustes en la configuración de red de la máquina virtual, para que forme parte de la red local y para que Snort funcione adecuadamente. Al igual que para la máquina virtual atacante, se configura el adaptador de red de la máquina en modo puente, se selecciona el adaptador ethernet y se activa el modo promiscuo. Dentro de la propia máquina virtual se le asigna una IP estática como al resto de dispositivos.

Figura 12. Asignación IP estática MV Snort. Fuente: Elaboración propia.

En esta MV es donde se va a llevar a cabo la monitorización del tráfico generado por la MV Kali Linux, así pues, es aquí donde se instala y configura Snort. Para que Snort 3 pueda funcionar de manera correcta es necesario instalar una serie de paquetes y librerías previo a su puesta en funcionamiento. Siguiendo la documentación proporcionada en la página web de Snort [25], se llevan a cabo los pasos que se muestran a continuación:

Primero se comprueba que el sistema esté actualizado y tenga las listas de paquetes más recientes.

```
sudo apt-get update && sudo apt-get dist-upgrade -y
```

A continuación, se crea un nuevo directorio donde se van a descargar varios archivos .tar entre otros.

```
mkdir ~/snort_src
cd ~/snort_src
```

Se instalan los prerrequisitos de Snort 3.

```
sudo apt-get install -y build-essential autotools-dev libdumbnet-dev \
     libluajit-5.1-dev libpcap-dev  zlib1g-dev pkg-config libhwloc-dev cmake \
     liblzma-dev openssl libssl-dev cpputest libsqlite3-dev libtool uuid-dev \
     git autoconf bison flex libcmocka-dev libnetfilter-queue-dev libunwind-dev \
     libmnl-dev ethtool
```

Descarga e instalación de la librería *safec* como extensión de la librería *libc*.

```
cd ~/snort_src wget https://github.com/rurban/safeclib/releases/download/v02092020/ \
   libsafec-02092020.tar.gz tar -xzvf libsafec-02092020.tar.gz
cd libsafec-02092020.0-g6d921f
./configure
make
sudo make install
```

Instalación de *Hyperscan*, librería que permite a Snort 3 detectar coincidencias a partir de múltiples expresiones regulares. *Hyperscan* cuenta con varios requisitos, entre ellos la instalación de las librerías PCRE, *gperftools*, *rage*l y *Boost*.

Instalación *Perl compatible regular expressions* (PCRE).

```
cd ~/snort_src/
wget https://ftp.pcre.org/pub/pcre/pcre-8.45.tar.gz
tar -xzvf pcre-8.45.tar.gz
cd pcre-8.45
./configure
make
sudo make install
```

Instalación *gperftools*.

```
cd ~/snort_src
wget https://github.com/gperftools/gperftools/releases/download/gperftools-
2.9.1/gperftools-2.9.1.tar.gz
tar xzvf gperftools-2.9.1.tar.gz
cd gperftools-2.9.1
./configure
make
sudo make install
```

Instalación *Ragel*.

```
cd ~/snort_src
wget http://www.colm.net/files/ragel/ragel-6.10.tar.gz
tar -xzvf ragel-6.10.tar.gz
cd ragel-6.10
./configure
make
sudo make install
```

Descarga de librerías *Boost*.

```
cd ~/snort_src
wget
https://boostorg.jfrog.io/artifactory/main/release/1.76.0/source/boost_1_76_0.tar.gz
tar -xvzf boost_1_76_0.tar.gz
```

Ahora sí, se descarga e instala la última versión de *Hyperscan* disponible, la 5.4.0.

```
cd cd ~/snort_src
wget https://github.com/intel/hyperscan/archive/refs/tags/v5.4.0.tar.gz
tar -xvzf v5.4.0.tar.gz

mkdir ~/snort_src/hyperscan-5.4.0-build
cd hyperscan-5.4.0-build/
cmake -DCMAKE_INSTALL_PREFIX=/usr/local \
      -DBOOST_ROOT=~/snort_src/boost_1_76_0/ ../hyperscan-5.4.0
make
sudo make install
```

Instalación de *flatbuffers*, librería serialización de datos.

```
cd ~/snort_src
wget https://github.com/google/flatbuffers/archive/refs/tags/v2.0.0.tar.gz \
     -O flatbuffers-v2.0.0.tar.gz
tar -xzvf flatbuffers-v2.0.0.tar.gz
mkdir flatbuffers-build
cd flatbuffers-build
cmake ../flatbuffers-2.0.0
make
sudo make install
```

Como último prerrequisito de Snort 3, se instala la última versión de librería de adquisición de datos, *libDAQ*.

```
cd ~/snort_src
wget https://github.com/snort3/libdaq/archive/refs/tags/v3.0.5.tar.gz \
     -O libdaq-3.0.5.tar.gz
tar -xzvf libdaq-3.0.5.tar.gz
cd libdaq-3.0.5
./bootstrap
./configure
make
sudo make install
```

Tras todos estos pasos, se procede a descargar e instalar la última versión de Snort 3, la 3.1.14.0 (es posible que esta versión ya no sea la más actual, ya que salen nuevas actualizaciones con frecuencia) que vendrá con la configuración por defecto.

```
cd ~/snort_src
wget https://github.com/snort3/snort3/archive/refs/tags/3.1.14.0.tar.gz \
     -O snort3-3.1.14.0.tar.gz
tar -xzvf snort3-3.1.14.0.tar.gz
cd snort3-3.1.14.0
./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
cd build
make
sudo make install
```

A modo de comprobación de que la instalación y la configuración por defecto son correctas, se ejecutan los siguientes comandos:

```
/usr/local/bin/snort -V

snort -c /usr/local/etc/snort/snort.lua
```

4. Configuración del IDS ante amenazas

4.1 Pruebas de penetración y ataques

En esta sección se detallan los tipos de escaneo de puertos y ataques realizados por la máquina atacante. Para cada tipo de escaneo de puertos y ataque se describen las principales características y objetivos, se muestran los comandos utilizados para su ejecución y el tráfico de paquetes que se genera utilizando WireShark.

4.1.1 Escaneo de puertos con NMAP

Los escaneos de puertos son una parte fundamental del proceso de reconocimiento de cualquier ciberataque. Mediante estos, un atacante puede recopilar información importante a cerca de los equipos a los que se pretende atacar. Lo que se consigue en general con las tácticas de escaneos de puertos es averiguar los puertos que están abiertos y los servicios que usan estos puertos, con la intención de explotar cualquier vulnerabilidad que presente estos. Dicho esto, se puede considerar que cada puerto abierto puede ser una posible puerta de entrada a un equipo.

Escaneo TCP sigiloso

Con este escaneo, lo que se pretende es obtener un listado de los puertos TCP abiertos que estén escuchando algún servicio. Hay varias alternativas a la hora de ejecutarlo, por ejemplo, el primer comando del bloque mostrado a continuación ejecuta un escaneo sigiloso (-sS) a un nivel de velocidad 4 (de 1 a 5, siendo 5 la más rápida) dirigida a todos los puertos (-p-) de todos los equipos en la red 192.168.1.0/24. En la Figura 13 se muestra el resultado tras la ejecución.

Por otro lado, el segundo comando solo se dirige al puerto 102 (-p102) del equipo con IP 192.168.1.100. Este equipo concretamente es uno de los PLCs que se encuentra en la red y la razón por la que se escanea el puerto 102 en concreto, es debido a que través de este puerto se comunican los equipos Siemens mediante un protocolo que funciona sobre TCP y se conoce como *ISO Transport Service on top of the TCP*.

```
nmap -sS -T4 -p- 192.168.1.0/24

nmap -sS -p102 192.168.1.100
```

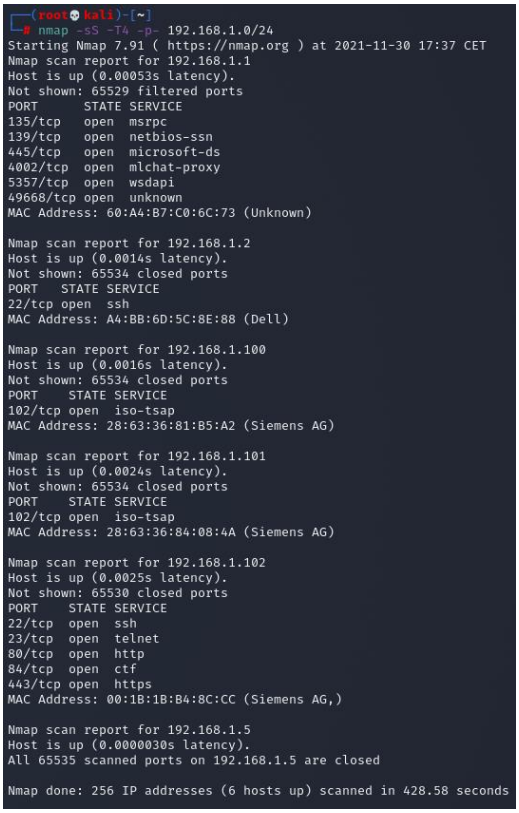
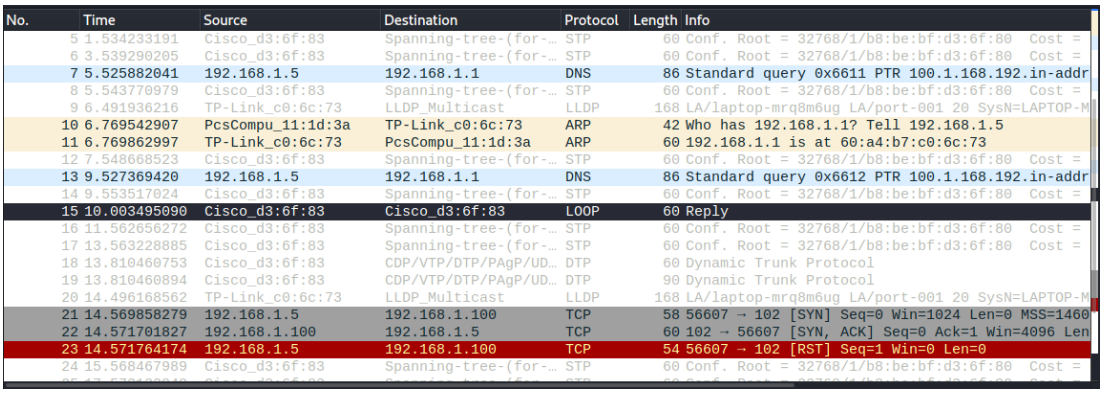


Figura 13. Listado de los puertos TCP abiertos de cada máquina tras un escaneo TCP sigiloso. Fuente: Elaboración propia.

Al ejecutar el segundo comando del bloque de código anterior y observando los paquetes que se generan a través de Wireshark, se puede ver como en primer lugar, la máquina atacante envía paquetes ARP en modo broadcast para poder asociar cada IP existente en la red a la dirección MAC correspondientes de cada máquina (No. 10 y 11). A continuación, la máquina atacante envía un paquete TCP con la *flag* SYN activa (No. 21) al puerto 102 del PLC con IP 192.168.1.100 para establecer una nueva conexión TCP. De acuerdo con el *three-way handshake protocol*, el PLC devuelve un SYN ACK (No. 22) indicando a la IP del atacante que se puede establecer una conexión. Lo interesante viene a continuación, y es que en vez de mandar un ACK, el atacante reinicia la conexión enviando un paquete con RST activo (No. 23). Al no completarse la conexión TCP, hace que este tipo de escaneo de puertos sea más rápido y que a su vez pase desapercibido por algunos sistemas de seguridad.



No.	Time	Source	Destination	Protocol	Length	Info
5	1.534233191	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
6	3.539290205	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
7	5.525882041	192.168.1.5	192.168.1.1	DNS	86	Standard query 0x6611 PTR 100.1.168.192.in-addr
8	5.543770979	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
9	6.491936216	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-M
10	6.769542907	PcsCompu_11:1d:3a	TP-Link_c0:6c:73	ARP	42	Who has 192.168.1.1? Tell 192.168.1.5
11	6.769862997	TP-Link_c0:6c:73	PcsCompu_11:1d:3a	ARP	60	192.168.1.1 is at 60:a4:b7:c0:6c:73
12	7.548668523	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
13	9.527369420	192.168.1.5	192.168.1.1	DNS	86	Standard query 0x6612 PTR 100.1.168.192.in-addr
14	9.553517024	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
15	10.003495090	Cisco_d3:6f:83	Cisco_d3:6f:83	LOOP	60	Reply
16	11.562656272	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
17	13.563228885	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
18	13.810460753	Cisco_d3:6f:83	CDP/VTP/DTP/PAgP/UD…	DTP	60	Dynamic Trunk Protocol
19	13.810460894	Cisco_d3:6f:83	CDP/VTP/DTP/PAgP/UD…	DTP	90	Dynamic Trunk Protocol
20	14.496168562	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-M
21	14.569858279	192.168.1.5	192.168.1.100	TCP	58	56607 → 102 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
22	14.571701827	192.168.1.100	192.168.1.5	TCP	60	102 → 56607 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len
23	14.571764174	192.168.1.5	192.168.1.100	TCP	54	56607 → 102 [RST] Seq=1 Win=0 Len=0
24	15.568467989	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =

Figura 14. Paquetes en Wireshark tras ejecutar un escaneo SYN sigiloso al puerto 102 del PLC2. Fuente: Elaboración propia.

Escaneo ACK

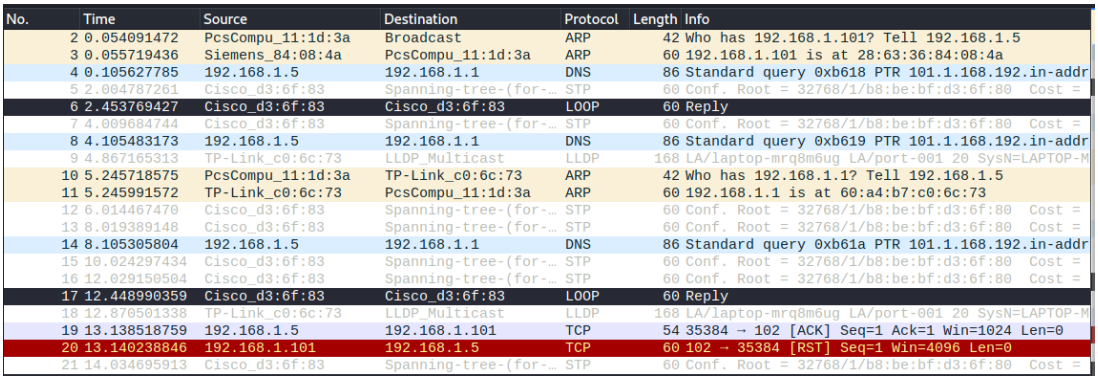
Con este otro tipo de escaneo de puertos, se pretende conocer que puertos TCP no están bloqueados o filtrados, por un firewall, por ejemplo; para ejecutarlo se usa el parámetro -sA. Concretamente el primer comando ejecuta este tipo de escaneo a velocidad 4 y dirigido a todos los puertos de la red local configurada. En cambio, el segundo comando solo va dirigido al puerto 102 del PLC con IP 192.168.1.101. En la Figura 15 se puede comprobar el resultado de ejecutar la primera línea de código mencionada.

```
nmap -sA -T4 -p- 192.168.1.0/24

nmap -sA -p102 192.168.1.101
```

```
┌──(root💀kali)-[~]
└─# nmap -sA -T4 -p- 192.168.1.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-30 17:54 CET
Nmap scan report for 192.168.1.1
Host is up (0.00063s latency).
All 65535 scanned ports on 192.168.1.1 are filtered
MAC Address: 60:A4:B7:C0:6C:73 (Unknown)

Nmap scan report for 192.168.1.2
Host is up (0.0012s latency).
All 65535 scanned ports on 192.168.1.2 are unfiltered
MAC Address: A4:BB:6D:5C:8E:88 (Dell)

Nmap scan report for 192.168.1.100
Host is up (0.0016s latency).
All 65535 scanned ports on 192.168.1.100 are unfiltered
MAC Address: 28:63:36:81:B5:A2 (Siemens AG)

Nmap scan report for 192.168.1.101
Host is up (0.0015s latency).
All 65535 scanned ports on 192.168.1.101 are unfiltered
MAC Address: 28:63:36:84:08:4A (Siemens AG)

Nmap scan report for 192.168.1.102
Host is up (0.0024s latency).
All 65535 scanned ports on 192.168.1.102 are unfiltered
MAC Address: 00:1B:1B:B4:8C:CC (Siemens AG,)

Nmap scan report for 192.168.1.5
Host is up (0.0000040s latency).
All 65535 scanned ports on 192.168.1.5 are unfiltered

Nmap done: 256 IP addresses (6 hosts up) scanned in 426.75 seconds
```

Figura 15. Listado de puertos TCP filtrados tras escaneo ACK. Fuente: Elaboración propia.

En la siguiente Figura se puede observar el tráfico generado tras ejecutar el segundo comando del bloque de código anterior. Se puede comprobar que este escaneo es muy sencillo ya simplemente se lanza un paquete TCP ACK (No. 19) esperando recibir un paquete TCP RST (No. 20), que confirma que el puerto no está filtrado.



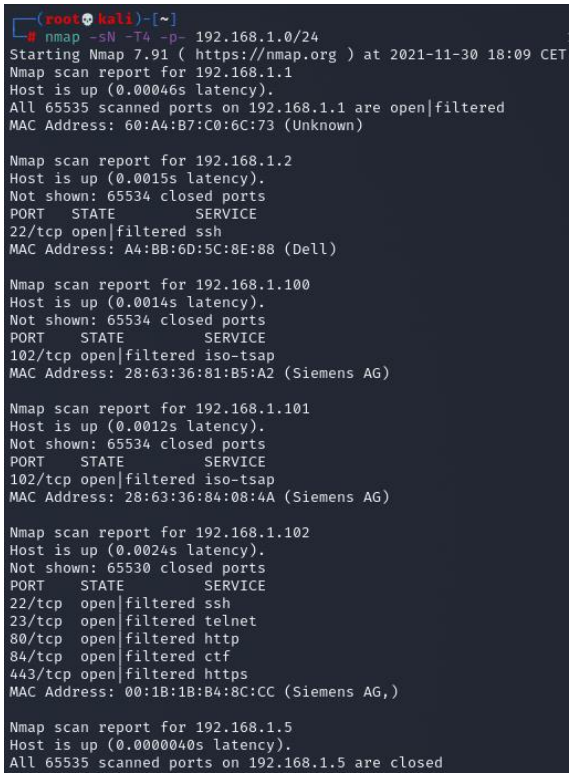
No.	Time	Source	Destination	Protocol	Length	Info
2	0.054091472	PcsCompu_11:1d:3a	Broadcast	ARP	42	Who has 192.168.1.101? Tell 192.168.1.5
3	0.055719436	Siemens_84:08:4a	PcsCompu_11:1d:3a	ARP	60	192.168.1.101 is at 28:63:36:84:08:4a
4	0.105627785	192.168.1.5	192.168.1.1	DNS	86	Standard query 0xb618 PTR 101.1.168.192.in-addr
5	2.004787261	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
6	2.453769427	Cisco_d3:6f:83	Cisco_d3:6f:83	LOOP	60	Reply
7	4.009684744	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
8	4.105483173	192.168.1.5	192.168.1.1	DNS	86	Standard query 0xb619 PTR 101.1.168.192.in-addr
9	4.867165313	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-M
10	5.245718575	PcsCompu_11:1d:3a	TP-Link_c0:6c:73	ARP	42	Who has 192.168.1.1? Tell 192.168.1.5
11	5.245991572	TP-Link_c0:6c:73	PcsCompu_11:1d:3a	ARP	60	192.168.1.1 is at 60:a4:b7:c0:6c:73
12	6.014467470	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
13	8.019389148	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
14	8.105305804	192.168.1.5	192.168.1.1	DNS	86	Standard query 0xb61a PTR 101.1.168.192.in-addr
15	10.024297434	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
16	12.029150504	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
17	12.448990359	Cisco_d3:6f:83	Cisco_d3:6f:83	LOOP	60	Reply
18	12.870501338	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-M
19	13.138518759	192.168.1.5	192.168.1.101	TCP	54	35384 → 102 [ACK] Seq=1 Ack=1 Win=1024 Len=0
20	13.140238846	192.168.1.101	192.168.1.5	TCP	60	102 → 35384 [RST] Seq=1 Win=4096 Len=0
21	14.034695913	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =

Figura 16. Paquetes en Wireshark tras escaneo ACK al puerto 102 del PLC1. Fuente: Elaboración propia.

Escaneo TCP NULL

Este tipo de escaneo es una forma alternativa a los dos anteriores que permite obtener un listado de los puertos TCP que están escuchando un servicio y los categoriza como filtrados, cerrados y abiertos o sin filtrar. En la Figura 17 se puede apreciar el resultado tras lanzar la primera línea de código, mientras que en la Figura 18 se presentan el tráfico generado al ejecutar la segunda línea.

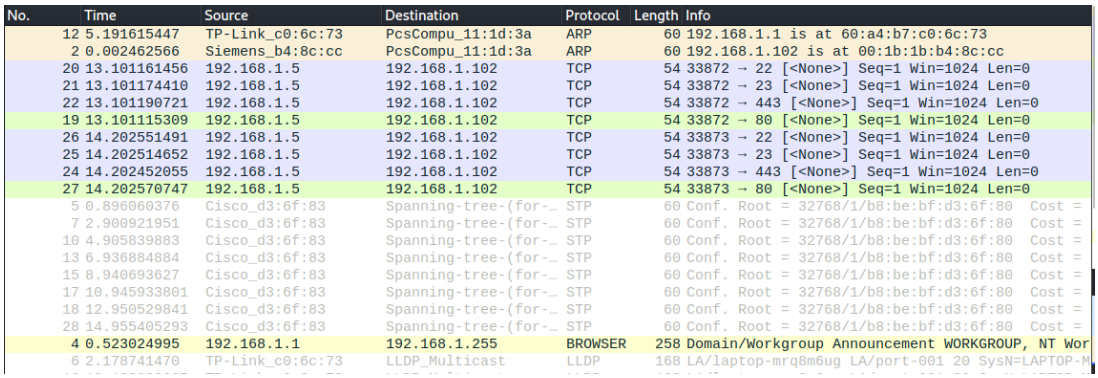
```
nmap -sN -T4 -p- 192.168.1.0/24

nmap -sN -p22,23,80,443 192.168.1.102
```



```
┌──(root💀kali)-[~]
└─# nmap -sN -T4 -p- 192.168.1.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-30 18:09 CET
Nmap scan report for 192.168.1.1
Host is up (0.00046s latency).
All 65535 scanned ports on 192.168.1.1 are open|filtered
MAC Address: 60:A4:B7:C0:6C:73 (Unknown)

Nmap scan report for 192.168.1.2
Host is up (0.0015s latency).
Not shown: 65534 closed ports
PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: A4:BB:6D:5C:8E:88 (Dell)

Nmap scan report for 192.168.1.100
Host is up (0.0014s latency).
Not shown: 65534 closed ports
PORT    STATE         SERVICE
102/tcp open|filtered iso-tsap
MAC Address: 28:63:36:81:B5:A2 (Siemens AG)

Nmap scan report for 192.168.1.101
Host is up (0.0012s latency).
Not shown: 65534 closed ports
PORT    STATE         SERVICE
102/tcp open|filtered iso-tsap
MAC Address: 28:63:36:84:08:4A (Siemens AG)

Nmap scan report for 192.168.1.102
Host is up (0.0024s latency).
Not shown: 65530 closed ports
PORT    STATE         SERVICE
22/tcp  open|filtered ssh
23/tcp  open|filtered telnet
80/tcp  open|filtered http
84/tcp  open|filtered ctf
443/tcp open|filtered https
MAC Address: 00:1B:1B:B4:8C:CC (Siemens AG,)

Nmap scan report for 192.168.1.5
Host is up (0.0000040s latency).
All 65535 scanned ports on 192.168.1.5 are closed
```

Figura 17. Listado de puertos y su estado al ejecutar un escaneo NULL sobre la red. Fuente: Elaboración propia.

Los paquetes TCP enviados por la máquina Kali no llevan ninguna *flag*, como se puede observar en la Figura 18. En el caso que un puerto este cerrado, la víctima devolverá un paquete RST ACK, por el contrario, si no se devuelve ningún paquete por parte de la víctima esto quiere decir que el puerto está abierto o filtrado.



No.	Time	Source	Destination	Protocol	Length	Info
12	5.191615447	TP-Link_c0:6c:73	PcsCompu_11:1d:3a	ARP	60	192.168.1.1 is at 60:a4:b7:c0:6c:73
2	0.002462566	Siemens_b4:8c:cc	PcsCompu_11:1d:3a	ARP	60	192.168.1.102 is at 00:1b:1b:b4:8c:cc
20	13.101161456	192.168.1.5	192.168.1.102	TCP	54	33872 → 22 [<None>] Seq=1 Win=1024 Len=0
21	13.101174410	192.168.1.5	192.168.1.102	TCP	54	33872 → 23 [<None>] Seq=1 Win=1024 Len=0
22	13.101190721	192.168.1.5	192.168.1.102	TCP	54	33872 → 443 [<None>] Seq=1 Win=1024 Len=0
19	13.101115309	192.168.1.5	192.168.1.102	TCP	54	33872 → 80 [<None>] Seq=1 Win=1024 Len=0
26	14.202551491	192.168.1.5	192.168.1.102	TCP	54	33873 → 22 [<None>] Seq=1 Win=1024 Len=0
25	14.202514652	192.168.1.5	192.168.1.102	TCP	54	33873 → 23 [<None>] Seq=1 Win=1024 Len=0
24	14.202452055	192.168.1.5	192.168.1.102	TCP	54	33873 → 443 [<None>] Seq=1 Win=1024 Len=0
27	14.202570747	192.168.1.5	192.168.1.102	TCP	54	33873 → 80 [<None>] Seq=1 Win=1024 Len=0
5	0.896060376	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
7	2.900921951	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
10	4.905839883	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
13	6.936884884	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
15	8.940693627	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
17	10.945933801	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
18	12.950529841	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
28	14.955405293	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost =
4	0.523024995	192.168.1.1	192.168.1.255	BROWSER	258	Domain/Workgroup Announcement WORKGROUP, NT Wor
6	2.178741470	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-M

Figura 18. Paquetes en Wireshark tras escaneo NULL hacia varios puertos de Switch Scalance. Fuente: Elaboración propia.

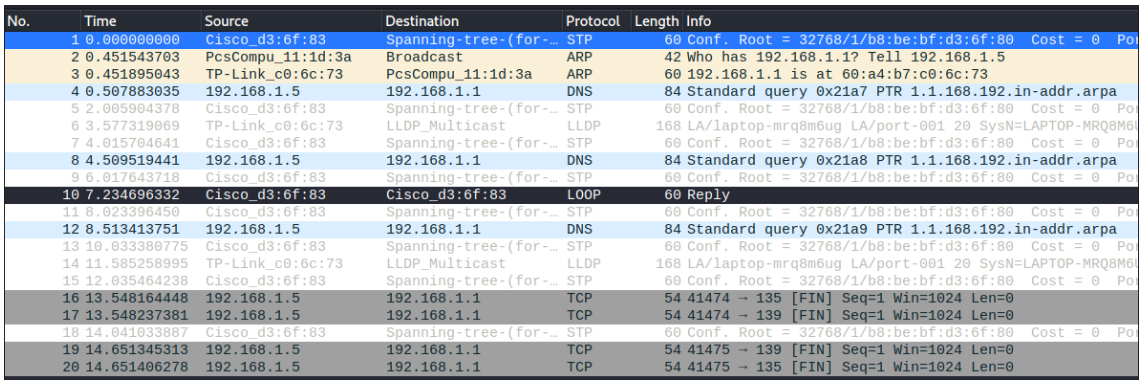
Escaneo FIN y XMAS

Estos dos tipos de escaneos de puertos tienen la misma funcionalidad que el de tipo NULL, por lo que el resultado mostrado en la consola es el mismo. Se ejecutan usando el parámetro -sF y -sX como se muestra a continuación.

Para el escaneo FIN:

```
nmap -sF -p135,139 192.168.1.1
```

En este caso se envían paquetes TCP a los puertos 135 y 139 del portátil con IP 192.168.1.1. Se puede ver en la Figura 19 como los paquetes TCP llevan activa la bandera FIN.



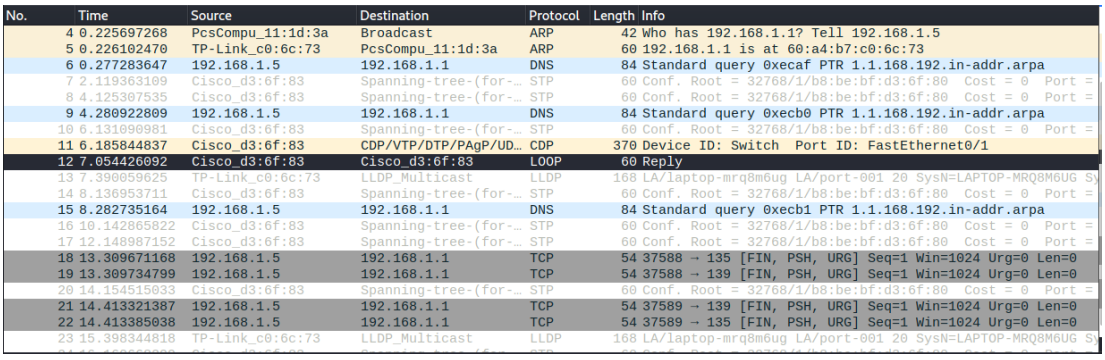
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
2	0.451543703	PcsCompu_11:1d:3a	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.5
3	0.451895043	TP-Link_c0:6c:73	PcsCompu_11:1d:3a	ARP	60	192.168.1.1 is at 60:a4:b7:c0:6c:73
4	0.507883035	192.168.1.5	192.168.1.1	DNS	84	Standard query 0x21a7 PTR 1.1.168.192.in-addr.arpa
5	2.005904378	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
6	3.577319069	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-MRQ8M6
7	4.015704641	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
8	4.509519441	192.168.1.5	192.168.1.1	DNS	84	Standard query 0x21a8 PTR 1.1.168.192.in-addr.arpa
9	6.017643718	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
10	7.234696332	Cisco_d3:6f:83	Cisco_d3:6f:83	LOOP	60	Reply
11	8.023396450	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
12	8.513413751	192.168.1.5	192.168.1.1	DNS	84	Standard query 0x21a9 PTR 1.1.168.192.in-addr.arpa
13	10.033380775	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
14	11.585258995	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-MRQ8M6
15	12.035464238	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
16	13.548164448	192.168.1.5	192.168.1.1	TCP	54	41474 → 135 [FIN] Seq=1 Win=1024 Len=0
17	13.548237381	192.168.1.5	192.168.1.1	TCP	54	41474 → 139 [FIN] Seq=1 Win=1024 Len=0
18	14.041033887	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Po
19	14.651345313	192.168.1.5	192.168.1.1	TCP	54	41475 → 139 [FIN] Seq=1 Win=1024 Len=0
20	14.651406278	192.168.1.5	192.168.1.1	TCP	54	41475 → 135 [FIN] Seq=1 Win=1024 Len=0

Figura 19. Paquetes en Wireshark tras escaneo FIN hacia varios puertos del portátil Windows. Fuente Elaboración propia.

Para el escaneo Xmas:

```
nmap -sX -p135,139 192.168.1.1
```

En este caso los paquetes TCP tiene activa la bandera FIN, PUSH y URGENT, como se puede ver en los paquetes 18, 19, 21 y 22 de la Figura 20.



No.	Time	Source	Destination	Protocol	Length	Info
4	0.225697268	PcsCompu_11:1d:3a	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.5
5	0.226102470	TP-Link_c0:6c:73	PcsCompu_11:1d:3a	ARP	60	192.168.1.1 is at 60:a4:b7:c0:6c:73
6	0.277283647	192.168.1.5	192.168.1.1	DNS	84	Standard query 0xecaf PTR 1.1.168.192.in-addr.arpa
7	2.119363109	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
8	4.125307535	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
9	4.280922809	192.168.1.5	192.168.1.1	DNS	84	Standard query 0xecb0 PTR 1.1.168.192.in-addr.arpa
10	6.131090981	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
11	6.185844837	Cisco_d3:6f:83	CDP/VTP/DTP/PAgP/UD…	CDP	370	Device ID: Switch Port ID: FastEthernet0/1
12	7.054426092	Cisco_d3:6f:83	Cisco_d3:6f:83	LOOP	60	Reply
13	7.390059625	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-MRQ8M6UG Sy
14	8.136953711	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
15	8.282735164	192.168.1.5	192.168.1.1	DNS	84	Standard query 0xecb1 PTR 1.1.168.192.in-addr.arpa
16	10.142865822	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
17	12.148987152	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
18	13.309671168	192.168.1.5	192.168.1.1	TCP	54	37588 → 135 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
19	13.309734799	192.168.1.5	192.168.1.1	TCP	54	37588 → 139 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
20	14.154515033	Cisco_d3:6f:83	Spanning-tree-(for-…	STP	60	Conf. Root = 32768/1/b8:be:bf:d3:6f:80 Cost = 0 Port =
21	14.413321387	192.168.1.5	192.168.1.1	TCP	54	37589 → 139 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
22	14.413385038	192.168.1.5	192.168.1.1	TCP	54	37589 → 135 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
23	15.398344818	TP-Link_c0:6c:73	LLDP_Multicast	LLDP	168	LA/laptop-mrq8m6ug LA/port-001 20 SysN=LAPTOP-MRQ8M6UG Sy

Figura 20. Paquetes en Wireshark tras escaneo Xmas hacia varios puertos del portátil Windows. Fuente: Elaboración propia.

4.1.2 Ataques

Ataque DoS

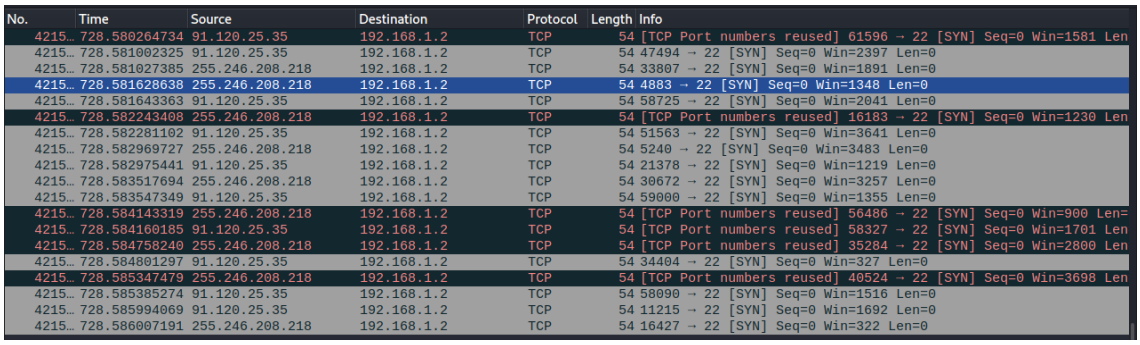
El primer ataque que se va a ejecutar es un ataque de denegación de servicio, o más conocido como DoS. Lo que se pretende con este tipo de ataque es saturar un servidor, una red o una máquina para que un servicio o recurso quede inoperativo e inaccesible a usuarios legítimos.

En este caso se ejecuta el *exploit* a través de varias terminales abiertas en la máquina virtual Kali, esto se hace para incrementar la carga de paquetes generados y, por ende, recibidos por la víctima. Ya que los recursos en este caso son limitados, no se va a generar tal cantidad de tráfico como para inhabilitar por completo a la víctima, simplemente generar lo suficiente para simular un ataque de estas características. Existen varias posibilidades a la hora ejecutar este ataque, en este trabajo se verán dos, que son el desbordamiento mediante paquetes TCP SYN y mediante paquetes ICMP, comúnmente conocido como *ping flood*.

Primero se ejecuta un ataque DoS con paquetes TCP SYN usando Metasploit. El ataque DoS se dirige al PC con IP 192.168.1.2 por el puerto 22, de tal forma que se pasan estos parámetros a las variables RHOSTS y RPORT respectivamente.

```
msfconsole
use auxiliary/dos/tcp/synflood
set RHOSTS 192.168.1.2
set RPORT 22
```

Se puede observar tanto a través de la consola como de Wireshark que los paquetes se van generando sin interrupción y con mucha frecuencia.



No.	Time	Source	Destination	Protocol	Length	Info
4215…	728.580264734	91.120.25.35	192.168.1.2	TCP	54	[TCP Port numbers reused] 61596 → 22 [SYN] Seq=0 Win=1581 Len
4215…	728.581002325	91.120.25.35	192.168.1.2	TCP	54	47494 → 22 [SYN] Seq=0 Win=2397 Len=0
4215…	728.581027385	255.246.208.218	192.168.1.2	TCP	54	33807 → 22 [SYN] Seq=0 Win=1891 Len=0
4215…	728.581628638	255.246.208.218	192.168.1.2	TCP	54	4883 → 22 [SYN] Seq=0 Win=1348 Len=0
4215…	728.581643363	91.120.25.35	192.168.1.2	TCP	54	58725 → 22 [SYN] Seq=0 Win=2041 Len=0
4215…	728.582243408	255.246.208.218	192.168.1.2	TCP	54	[TCP Port numbers reused] 16183 → 22 [SYN] Seq=0 Win=1230 Len
4215…	728.582281102	91.120.25.35	192.168.1.2	TCP	54	51563 → 22 [SYN] Seq=0 Win=3641 Len=0
4215…	728.582969727	255.246.208.218	192.168.1.2	TCP	54	5240 → 22 [SYN] Seq=0 Win=3483 Len=0
4215…	728.582975441	91.120.25.35	192.168.1.2	TCP	54	21378 → 22 [SYN] Seq=0 Win=1219 Len=0
4215…	728.583517694	255.246.208.218	192.168.1.2	TCP	54	30672 → 22 [SYN] Seq=0 Win=3257 Len=0
4215…	728.583547349	91.120.25.35	192.168.1.2	TCP	54	59000 → 22 [SYN] Seq=0 Win=1355 Len=0
4215…	728.584143319	255.246.208.218	192.168.1.2	TCP	54	[TCP Port numbers reused] 56486 → 22 [SYN] Seq=0 Win=900 Len=
4215…	728.584160185	91.120.25.35	192.168.1.2	TCP	54	[TCP Port numbers reused] 58327 → 22 [SYN] Seq=0 Win=1701 Len
4215…	728.584758240	255.246.208.218	192.168.1.2	TCP	54	[TCP Port numbers reused] 35284 → 22 [SYN] Seq=0 Win=2800 Len
4215…	728.584801297	91.120.25.35	192.168.1.2	TCP	54	34404 → 22 [SYN] Seq=0 Win=327 Len=0
4215…	728.585347479	255.246.208.218	192.168.1.2	TCP	54	[TCP Port numbers reused] 40524 → 22 [SYN] Seq=0 Win=3698 Len
4215…	728.585385274	91.120.25.35	192.168.1.2	TCP	54	58090 → 22 [SYN] Seq=0 Win=1516 Len=0
4215…	728.585994069	91.120.25.35	192.168.1.2	TCP	54	11215 → 22 [SYN] Seq=0 Win=1692 Len=0
4215…	728.586007191	255.246.208.218	192.168.1.2	TCP	54	16427 → 22 [SYN] Seq=0 Win=322 Len=0

Figura 21. Paquetes en Wireshark tras iniciar un ataque DoS con paquetes TCP SYN al PC. Fuente: Elaboración propia.

Abriendo el panel de monitorización de rendimiento del PC se puede apreciar como la cantidad de datos recibidos por el PC dan un salto importante al ser víctima del ataque. Esto se muestra en la Figura 22.

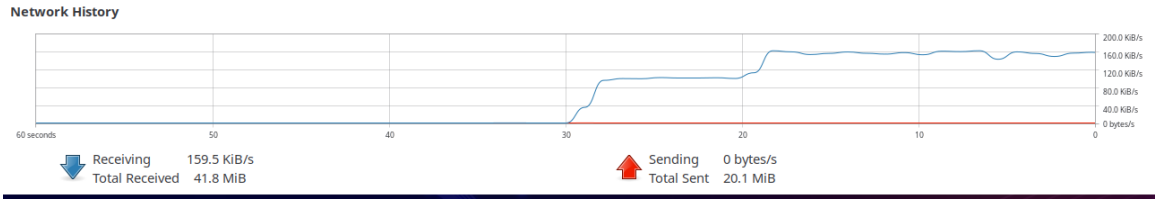
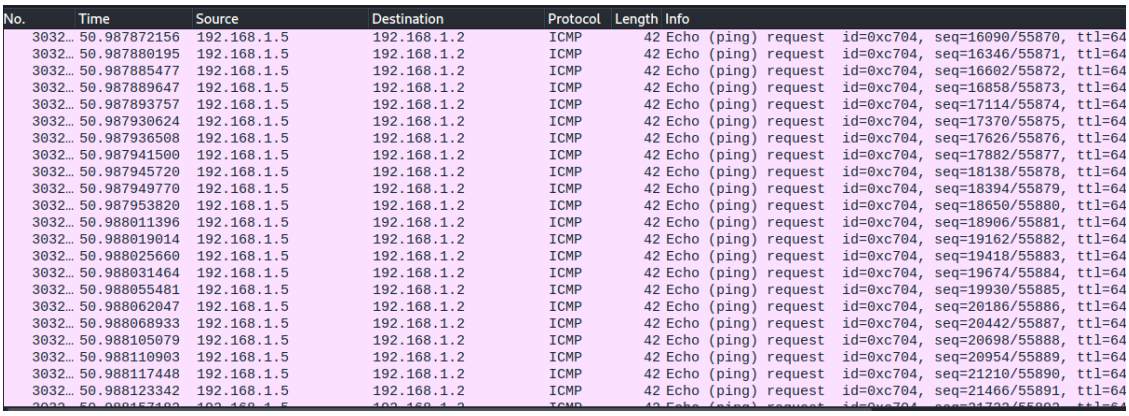


Figura 22. Datos recibidos en el PC mientras se ejecuta un ataque DoS con paquetes TCP. Fuente: Elaboración propia.

La segunda opción es ejecutar el mismo ataque, pero utilizando paquetes ICMP, esto se hace mediante HPING3.

```
sudo hping3 -1 –flood 192.168.1.2
```

Los paquetes generados en este caso son del protocolo ICMP como muestra en la siguiente figura.



No.	Time	Source	Destination	Protocol	Length	Info
3032…	50.987872156	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=16090/55870, ttl=64
3032…	50.987880195	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=16346/55871, ttl=64
3032…	50.987885477	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=16602/55872, ttl=64
3032…	50.987889647	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=16858/55873, ttl=64
3032…	50.987893757	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=17114/55874, ttl=64
3032…	50.987930624	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=17370/55875, ttl=64
3032…	50.987936508	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=17626/55876, ttl=64
3032…	50.987941500	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=17882/55877, ttl=64
3032…	50.987945720	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=18138/55878, ttl=64
3032…	50.987949770	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=18394/55879, ttl=64
3032…	50.987953820	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=18650/55880, ttl=64
3032…	50.988011396	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=18906/55881, ttl=64
3032…	50.988019014	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=19162/55882, ttl=64
3032…	50.988025660	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=19418/55883, ttl=64
3032…	50.988031464	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=19674/55884, ttl=64
3032…	50.988055481	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=19930/55885, ttl=64
3032…	50.988062047	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=20186/55886, ttl=64
3032…	50.988068933	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=20442/55887, ttl=64
3032…	50.988105079	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=20698/55888, ttl=64
3032…	50.988110903	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=20954/55889, ttl=64
3032…	50.988117448	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=21210/55890, ttl=64
3032…	50.988123342	192.168.1.5	192.168.1.2	ICMP	42	Echo (ping) request id=0xc704, seq=21466/55891, ttl=64

Figura 23. Paquetes en Wireshark tras iniciar un ataque DoS con paquetes ICMP al PC. Fuente: Elaboración propia.

Para este caso también se abre el panel de rendimiento del PC donde también se aprecia un incremento de los paquetes recibidos.

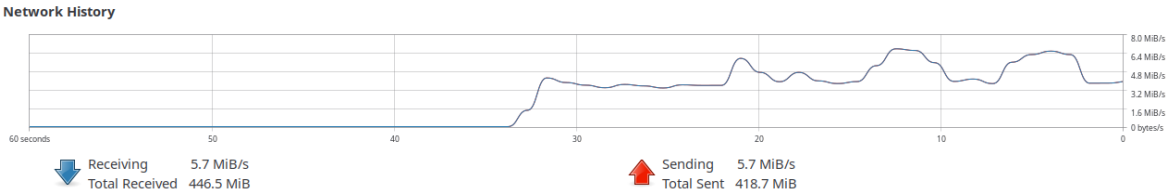


Figura 24. Datos recibidos en el PC tras ejecutar ataque DoS usando pings. Fuente: Elaboración propia.

Ataque fuerza bruta

Se considera un ataque de fuerza bruta a aquel con el que se pretende conseguir acceso a un sistema mediante prueba y error, es decir probando todas las combinaciones posibles de credenciales hasta dar con la correcta.

Este ataque se va a ejecutar a través del puerto 22, el que generalmente se utiliza para conexiones SSH, que justamente tiene abierto el PC ubicado en la red local. Metasploit cuenta con un módulo para llevar a cabo este ataque y precisa de dos archivos de texto que contengan listadas las posibles credenciales. Para ello se crean dos archivos de texto, USER.txt, que contiene un listado de posibles nombres de usuario y PASS.txt, el cual contiene un listado con las posibles contraseñas. Ambos ficheros se rellenan con 10 elementos para que la ejecución no demore demasiado tiempo.

Los comandos para la ejecución de este ataque y su resultado se muestran a continuación:

```
msfconsole
use auxiliary/scanner/ssh/ssh_login
set RHOSTS 192.168.1.2
set STOP_ON_SUCCESS true
set USER_FILE /home/adri/Escritorio/USER
set PASS_FILE /home/adri/Escritorio/PASS
set VERBOSE true
```

Si se realizan varios intentos de acceso a través de SSH y no se introducen las credenciales correctas la conexión se cerrará y el atacante deberá volver a abrir otra conexión ssh a través del puerto 22. Esto se puede ver en la Figura siguiente.

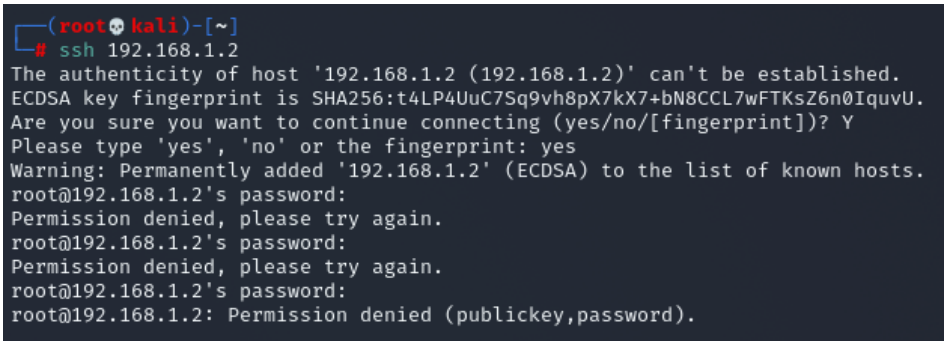


Figura 25. Intento fallido de acceso a través de SSH. Fuente: Elaboración propia.

4.2 Reglas en Snort

El objetivo es crear varias reglas que permitan la detección y la prevención de los ataques descritos anteriormente. Para ello, primero es conveniente crear una carpeta que contenga los ficheros donde se van a colocar las reglas. También se crea un directorio donde posteriormente se almacenarán las alertas producidas por Snort.

```
sudo mkdir /usr/local/etc/rules
sudo touch /usr/local/etc/rules/local.rules
sudo mkdir /var/log/snort
```

En el fichero *local.rules* se han creado reglas para detectar los escaneos de puertos, los ataques DoS y el de fuerza bruta. En el primer bloque se encuentran las correspondientes a los escaneos de puertos que son las siguientes:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "Posible escaneo TCP_SYN"; flags:S;
dsize:0; sid:10000001; priority:5;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "Posible escaneo ACK"; flags:A;
dsize:0; sid:10000002; priority:4;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "Posible escaneo NULL"; flags:0;
dsize:0; sid:1000003; priority:4;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "Posible escaneo FIN"; flags:F;
dsize:0; sid:1000004; priority:4;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "Posible escaneo Xmas"; flags:FPU;
dsize:0; sid:1000005; priority:4;)
```

La estructura de las alertas creadas para detectar un escaneo de puertos es similar para los distintos tipos de escaneos. Todas comparten la acción (alerta), el protocolo tcp, la dirección del origen y destino, y los puertos. Cada regla tiene la opción flag con un valor distinto, esto se debe a la composición de los paquetes vistos a través de Wireshark en la sección anterior. Cabe puntualizar en el parámetro dsize que se le pasa a la primera regla, que se hace para distinguir entre un paquete TCP legitimo y los que se envían a través de Nmap, estando el contenido del paquete vacío en los últimos.

Después se encuentran las que corresponden a los ataques DoS.

```
alert tcp any any -> $HOME_NET any (msg:“Posible ataque DoS SYN”; detection_filter:track
by_src, count 50, seconds 2; sid:1000006; priority:2;)

alert icmp any any -> $HOME_NET any (msg:“Posible ataque DoS ICMP”;
detection_filter:track by_src, count 1000, seconds 2; sid:1000007; priority:2;)
```

Las reglas para detectar los ataques DoS son similares a las de los escaneos de puertos, pero con la condición de que solo saltará la alerta si en un espacio de tiempo de 2 segundos se han enviado al menos 50 paquetes por parte de una dirección de origen.

La última regla corresponde al ataque de fuerza bruta.

```
alert tcp $HOME_NET 22 -> $EXTERNAL_NET 22 (msg:“Error de autenticación SSH”; content:“
sid:1000008; priority:1;)
```

Como se ha visto en la sección anterior, cada vez que se pasaban las credenciales de manera errónea, el servidor (la víctima en este caso) devolvía el mensaje que se pasa al parámetro *content*, de manera que cuando se lance este mensaje se generará esta alerta.

5. Recogida y análisis de eventos

A la hora de visualizar las alertas generadas por Snort se pueden tomar varios caminos que tienen sus ventajas y sus desventajas, en este capítulo se comentan dos de ellas. La primera es la visualización a través de la terminal y la segunda es a través de una interfaz gráfica de un software o aplicación.

La primera opción es la que se ha usado para desarrollar este trabajo y la que se utilizaría en cualquier entorno de pruebas y desarrollo. La principal ventaja que tiene esta forma de ver las alertas es que permite una mayor agilidad a la hora de realizar cambios y modificaciones a los ficheros de reglas o al de configuración de Snort. En cambio, a la hora de analizar redes donde haya un tráfico elevado de paquetes y se generen muchas alertas, esta alternativa no es adecuada ya que no va a ser posible llevar el control de todas y cada una de las alertas generadas. Para realizar este tipo de monitorización se hace uso del comando mostrado a continuación, obteniendo resultados como los mostrados en la Figura 26. En el comando se pasan como parámetros el archivo de configuración *snort.lua*, la interfaz de red donde se van a capturar los paquetes, el plugin de visualización rápida por pantalla, el *snapshot length* o tamaño de datos máximo por captura y la opción de ignorar cheksums erróneos.

```
sudo snort -c /usr/local/etc/snort/snort.lua -i enp0s3 -A alert_fast -s 65535 -k none
```

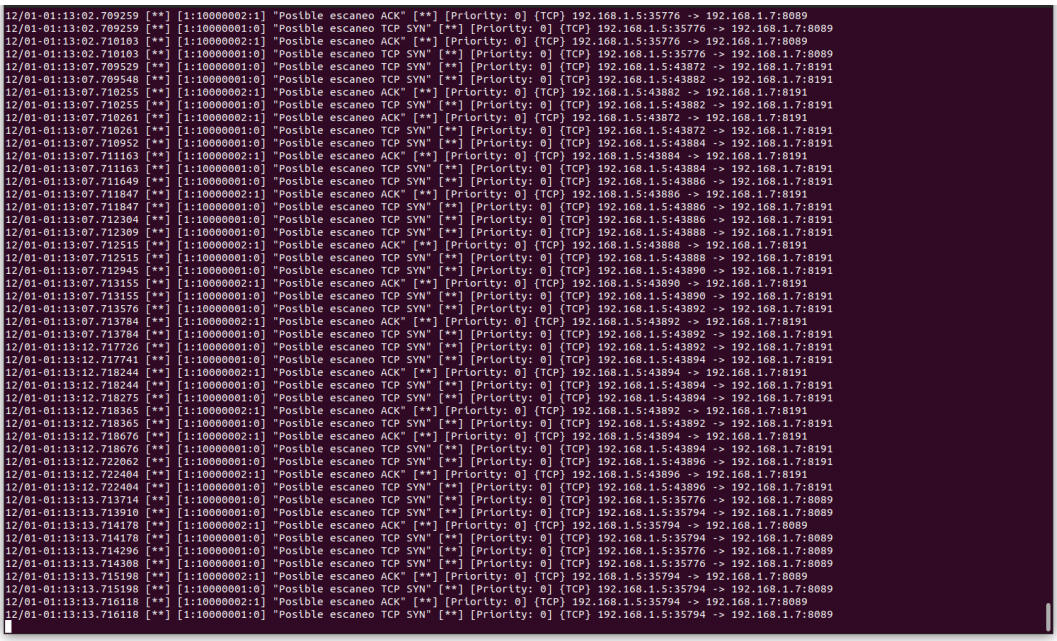


Figura 26. Alertas generadas por Snort visualizadas en el terminal de Ubuntu. Fuente: Elaboración propia.

La segunda opción que resulta más interesante en situaciones donde prima la gestión y análisis de grandes cantidades de alertas, es la de usar una aplicación o software que disponga de una interfaz gráfica y la capacidad de gestionar datos en ficheros de texto o json, como puede ser Splunk. De esta manera es como se pretende utilizar Splunk en este trabajo. Antes de comenzar con la configuración de Splunk y como se integra con Snort, cabe de mencionar que Splunk es un software que permite recolectar, monitorizar y analizar multitud de datos a través de una interfaz web.

La manera en la que Splunk accede a las alertas generadas por Snort no es demasiado compleja. En primer lugar, las alertas generadas por Snort se deben guardar en algún fichero, en este caso se van a guardar en el directorio por defecto /var/log/snort. Para que Snort pueda hacer esto es necesario habilitar el plugin *alert_json* ubicado dentro del archivo de configuración *snort.lua*.

Se abre el fichero de configuración de Snort para editarlo.

```
sudo nano /usr/local/etc/snort/snort.lua
```

Donde se encuentra ubicado el plugin alert_json se rellena con las siguientes opciones

```
alert_json =
{
file = true,
limit = 150,
fields = 'seconds action class b64_data dir dst_addr dst_ap dst_port eth_dst eth_len
eth_src eth_type gid icmp_code icmp_id icmp_seq icmp_type iface ip_id ip_len msg
pkt_gen pkt_len pkt_num priority proto service sid src_addr src_ap src_port
target tcp_ack tcp_flags tcp_len tcp_seq tcp_win timestamp',
}
```

Dentro del plugin se configuran 3 opciones:

1. *File*: se pone a true para que al ejecutar Snort se guarden las alertas en un fichero json
2. *Limit*: indica el tamaño límite del archivo json donde se guardan las alertas. En caso de superar el límite establecido, en este caso de 150 MB, se generará un nuevo fichero json.
3. *Fields*: se incluyen los distintos campos de la alerta que se van a incluir en el fichero json.

A la hora de ejecutar Snort para que las alertas se almacenen en /var/log/Snort en vez de mostrarse por pantalla, el comando que se usa es el siguiente:

```
sudo /usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -s 65535 -k none -l
/var/log/snort -i eth0 -m 0x1b
```

Con todo lo anterior configurado, se procede a la instalación y configuración de Splunk, donde se van a recoger los ficheros json generados a partir de las alertas. En primer lugar, se accede a la página web de Splunk donde es necesario registrarse para acceder a la versión gratuita del software. Tras hacer esto se descarga el archivo .tar y se instala usando los siguientes comandos a través del terminal.

```
sudo dpkg -i splunk-8.2.3.-*.deb
sudo chown -R splunk:splunk /opt/splunk
```

Se inicia Splunk y ya se puede acceder a través del servidor local http://localhost:8000.

```
 sudo /opt/splunk/bin/splunk start
```

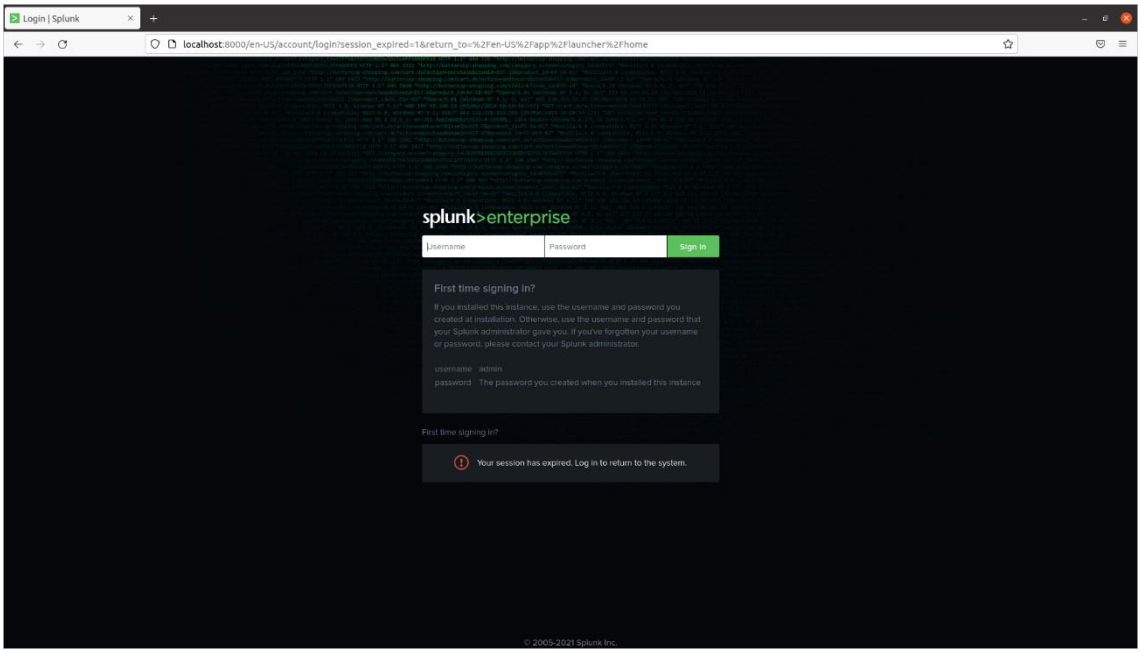


Figura 27. Acceso a Splunk a través de localhost. Fuente: Elaboración propia.

Una vez se accede a Splunk es necesario instalar un plugin que permiten a Splunk recoger y normalizar las alertas generados por Snort. El plugin o Add-on a instalar es Snort 3 JSON Alerts, pudiendo hacerlo desde la pestaña "+ Find More Apps".

Tras realizar la instalación del plugin se debe configurar para que este le diga a Splunk donde están ubicados los ficheros donde se almacenan las reglas generadas por Snort. Esto se hace creando un archivo que va a servir como archivo de configuración del plugin JSON Alerts. Para ello se usan los siguientes comandos a través del terminal.

```
sudo mkdir /opt/splunk/etc/apps/TA_Snort3_json/local
sudo touch /opt/splunk/etc/apps/TA_Snort3_json/local/inputs.conf
sudo nano /opt/splunk/etc/apps/TA_Snort3_json/local/inputs.conf
```

Se añaden dos líneas de código al archivo inputs.conf. la primera indica los archivos json desde donde se van a extraer las alertas y en la segunda se le asigna un nombre a esta fuente de información.

```
[monitor:///var/log/snort/*alert_json.txt*]
sourcetype = snort3:alert:json
```

Tras reiniciar Splunk, este ya deberá monitorizar el directorio /var/log/Snort, detectando cambios en los archivos alert_json de la ruta especificada en el fichero de configuración.

```
sudo service splunkd restart
```

Finalmente, para comprobar que se recoge algún fichero json, se introduce en la barra de búsqueda el *sourcetype* que se ha definido, *snort3:alert:json*.

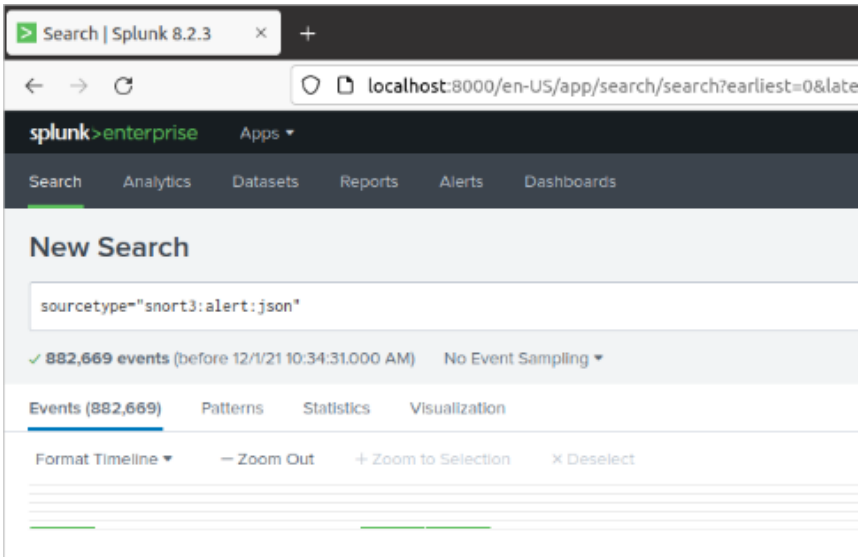


Figura 28. Panel de búsqueda en Splunk tras cargar datos. Fuente Elaboración propia.

6. Resultados

6.1 Características de los resultados obtenidos

En esta sección se muestran los resultados que se han obtenido a través de la terminal, a la hora de detectar los ataques descritos en el capítulo anterior. En primer lugar, las alertas generadas a causa de los escaneos de puertos han funcionado según lo previsto, es decir, al enviar un paquete TCP que coincide con alguna de las reglas creadas se genera una sola alerta. En las Figuras mostradas a continuación se puede observar como por cada paquete TCP enviado a un puerto de cualquier máquina se genera la alerta correspondiente.

En la Figura 29 se puede apreciar una alerta generada debido a una coincidencia con la regla con sid 10000001, que corresponde a un posible escaneo de tipo TCP sigiloso, al puerto 102 de la máquina con IP 192.168.1.100, correspondiente al PLC2.

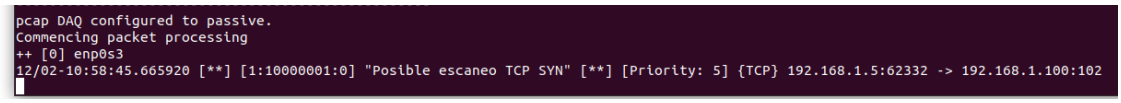


Figura 29. Alerta generada al escanear el puerto 102 del PLC2 mediante un escaneo TCP sigiloso. Fuente: Elaboración propia.

En la Figura mostrada a continuación se presenta una alerta generada por una coincidencia con la regla 10000002, correspondiente a un escaneo de puertos del tipo ACK dirigido al puerto 102 del autómata con IP 192.168.1.101.

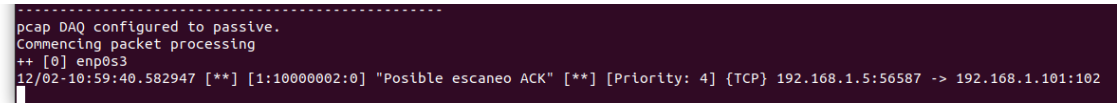


Figura 30. Alerta generada al escanear el puerto 102 del PLC1 mediante un escaneo tipo ACK. Fuente: Elaboración propia.

La Figura 31 muestra varias alertas, generadas por una coincidencia con la regla que pretende detectar un escaneo de puertos de tipo NULL. En este caso el escaneo iba dirigido a los puertos 80 y 21 del ordenador portátil.

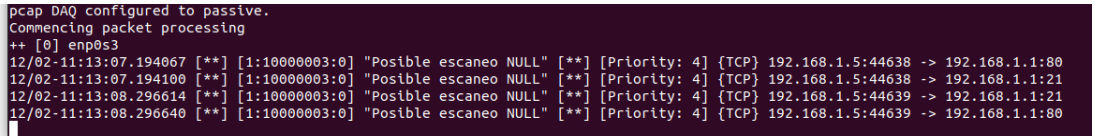


Figura 31. Alertas generadas al escanear los puertos 20 y 80 del portátil Windows mediante un escaneo tipo NULL. Fuente: Elaboración propia.

En el caso de las alertas generadas debido a los ataques de denegación de servicio, los resultados no resultan ser óptimos si unicamente se hace uso de las reglas creadas para detectar estos ataques. Esto se debe a que la cantidad de alertas que genera un ataque de este tipo es muy grande y no es algo que resulte óptimo. En la Figura mostrada a continuación se puede ver que las alertas se generan muy rápido, aproximadamente cada 1,3 ms.

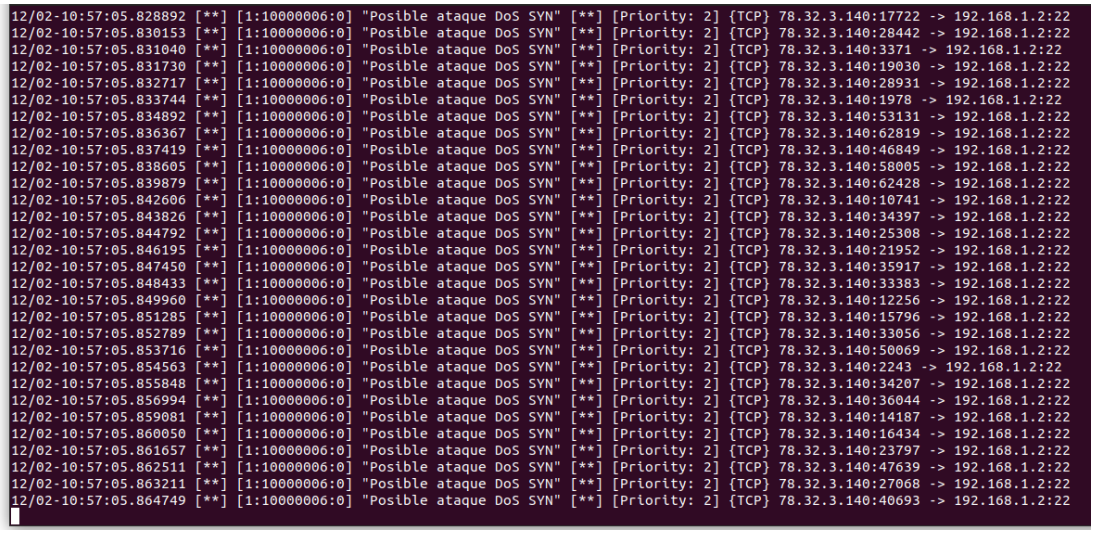


Figura 32. Alertas generadas al ejecutar un ataque DoS TCP al PC a través del puerto 22. Fuente: Elaboración propia.

Para solucionar esto se accede al archivo de configuración de Snort y se activa el bloque denominado como filtro de eventos. En este bloque se pueden añadir distintas instrucciones que filtren o ignoren alertas concretas. De tal forma, se añaden dos líneas de código al bloque que tienen como propósito limitar a que solo se genere una alerta cada 30 segundos si en un periodo de 30 segundos se generan 180 coincidencias o más. Las líneas de código se muestran a continuación.

```
{gid=1, sid=1 10000006, type='both', track='by_src', count=180, seconds=30}
{gid=1, sid=1 10000007, type='both', track='by_src', count=180, seconds=30}
```

El resultado de aplicar este filtro con unos valores de *count* y *seconds*, puestos a 30 y 5 correspondientemente, se pueden ver en la Figura 33.

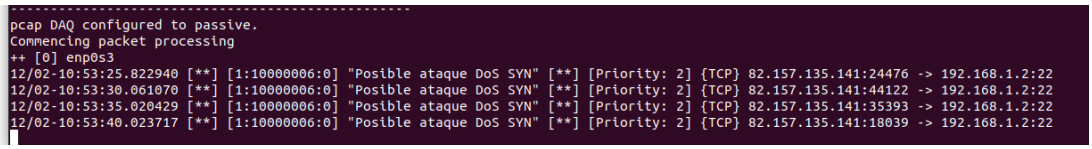


Figura 33. Alertas generadas al ejecutar un ataque DoS TCP al PC a través del puerto 22 con un filtrado de eventos aplicado. Fuente: Elaboración propia.

Finalmente se muestran en la Figura 34 las alertas generadas al lanzar un ataque de fuerza bruta con Metasploit. En este caso se ejecutan varias reglas, ya que al ejecutar este ataque se crean varios paquetes TCP que generan coincidencia con las reglas creadas para alertar sobre escaneos TCP sigilosos y ACK. La solución en este caso no se puede conseguir aplicando filtros al igual que se ha hecho con los ataques DoS. Es por ello por lo que esto se tendrá en cuenta a la hora de pasar estas alertas a Splunk y para realizar los filtros pertinentes allí.

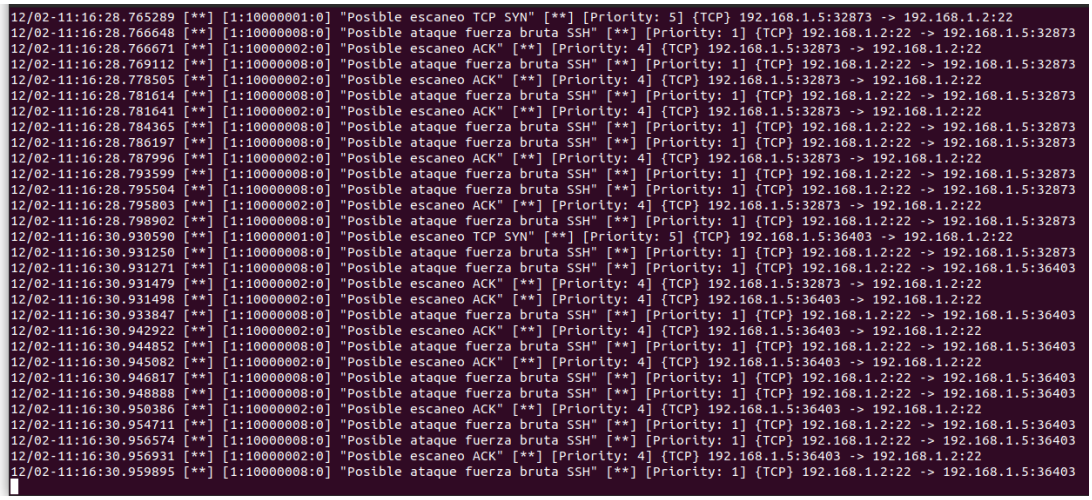


Figura 34. Alertas generadas al ejecutar un ataque de fuerza bruta al PC a través del puerto 22. Fuente: Elaboración propia

6.2 Visualización de resultados

A la hora de analizar los resultados con Splunk, lo primero a tener en cuenta es que en este software se trabaja con eventos, que realmente equivalen a las alertas generadas por Snort. Desde la ventana de búsqueda de Splunk se pueden llevar a cabo varias tareas que permiten visualizar y analizar los eventos que están actualmente cargados. La Figura 35 muestra la ventana de búsqueda de Splunk, desde donde se puede ver el número actual de eventos cargados, un gráfico de barras temporal de los eventos, y una tabla que muestra algunos campos de varios eventos. Cabe de mencionar que mientras Snort genera alertas, estas se pasan a Splunk y se generan los correspondientes eventos que se van representando en el gráfico de barras, aunque es cierto que existe un pequeño retraso de aproximadamente un par minutos entre que se genera la alerta y aparece en Splunk. Concretamente para los escenarios que se han visto, el gráfico de barras viene muy bien a la hora de detectar ataques DoS y escaneos a múltiples puertos, ya que se generan picos importantes en el gráfico.

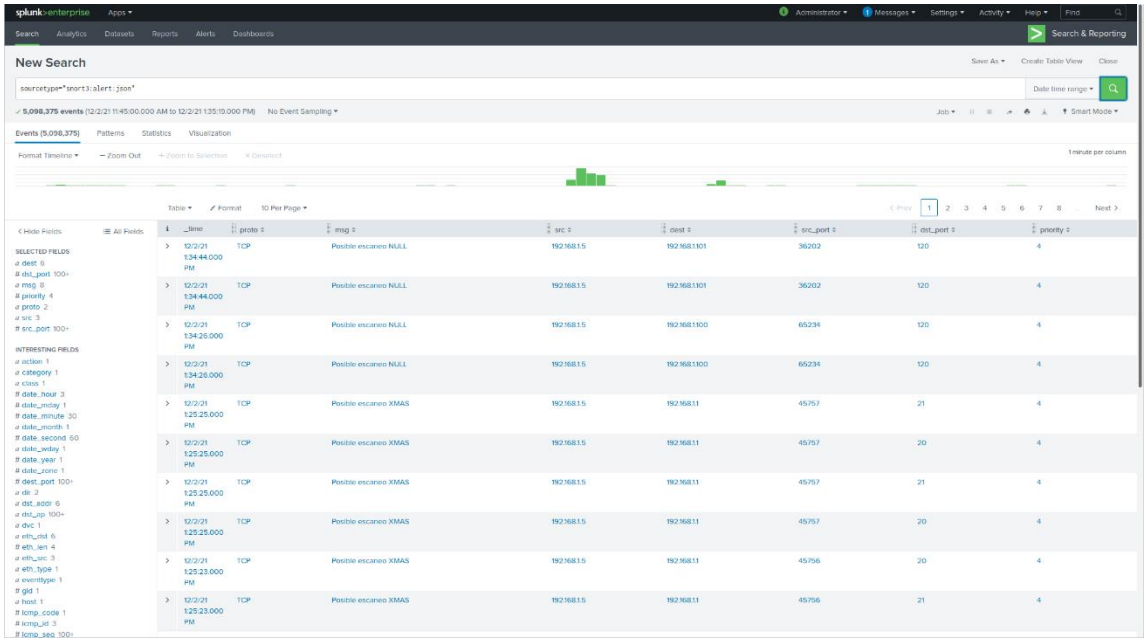


Figura 35. Ventana de búsqueda de Splunk. Fuente: Elaboración propia.

La barra de búsqueda de Splunk es una herramiente muy potente, ya que permite realizar filtrados en función de campos, selección de franjas temporales y varias funciones que más adelante se van a utilizar. La elección del periodo temporal se puede hacer a través del desplegable junto al botón verde de busqueda. Se proporcionan varias opciones por defecto pero también existe la opción de seleccionar un franja temporal concreta, especificando las fechas en modo dd/mm/aa hh:mm:ss.s. En la Figura 35, se usa ajusta la visualización de eventos para una franja temporal de aproximadamente una hora.

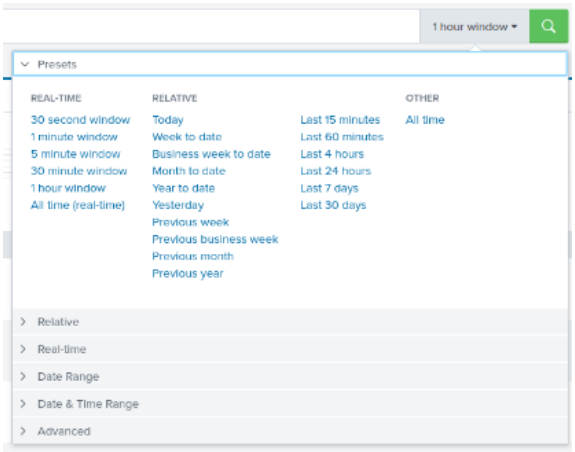


Figura 36. Pestaña de filtrado temporal de eventos. Fuente: Elaboración propia.

Un ejemplo del uso que se le puede dar a la barra de búsqueda es el filtrado de eventos según sus campos. Por ejemplo, para generar la tabla que aparece en la Figura 37, correspondiente a las alertas generadas por ataques DoS usando paquetes TCP y con el PC como destinatario del ataque se utiliza expresión:

```
sourcetype="snort3:alert:json" AND sid=10000006 dest=”192.168.1.2”
```

_time	proto	msg	src	dest	src_port	dst_port	priority
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38651	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2
12/2/21 1:14:47.000 PM	TCP	Posible ataque DoS SYN	192.168.1.5	192.168.1.2	38479	22	2

Figura 37. Eventos correspondientes a ataques DoS SYN con el PC como objetivo. Fuente: Elaboración propia.

Otro ejemplo de búsqueda utilizando filtrado de campos es el que tiene como resultado el mostrado en la Figura 38, utilizando para ello la siguiente expresión.

```
sourcetype="snort3:alert:json" AND sid=10000003 AND dest IN (192.168.1.100, 192.168.1.101)
```

_time	proto	msg	src	dest	src_port	dst_port	priority
12/2/21 1:34:44.000 PM	TCP	Posible escaneo NULL	192.168.1.5	192.168.1.101	36202	120	4
12/2/21 1:34:44.000 PM	TCP	Posible escaneo NULL	192.168.1.5	192.168.1.101	36202	120	4
12/2/21 1:34:26.000 PM	TCP	Posible escaneo NULL	192.168.1.5	192.168.1.100	65234	120	4
12/2/21 1:34:26.000 PM	TCP	Posible escaneo NULL	192.168.1.5	192.168.1.100	65234	120	4

Figura 38. Eventos correspondietes a escaneos tipo NULL con los dos PLC como objetivos. Fuente: Elaboración propia.

Otra de las funcionalidades que tiene la barra de búsqueda es que desde la misma se puede realizar insertar una expresión que combine un filtrado por campos y una función de visualización de eventos. En este primer ejemplo, mostrado en la Figura 39, se opta por generar un gráfico de barras que represente el número total de alertas generadas en función de la IP destinataria. Lo que se pretende expresar con esta gráfica es de forma aproximada el volumen de tráfico malintencionado que se ha lanzado hacia cada equipo, concretamente en el ejemplo las máquinas a las que potencialmente se les ha dirigido un mayor volumen de paquetes son en PLC2 (192.168.1.100) y la máquina virtual que aloja a Snort (192.168.1.7). Este gráfico se ha generado usando la expresión:

```
sourcetype="snort3:alert:json" dest !=192.168.1.5| chart count(sid) by dest | rename count(sid) AS "Numero de alertas" | rename dest AS " IP destino "
```

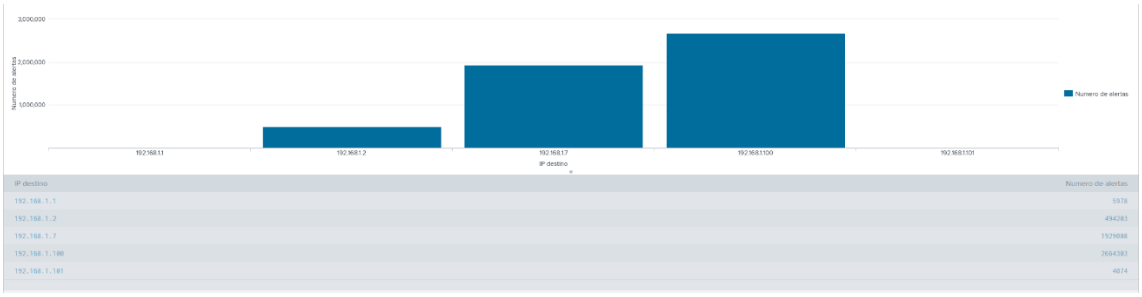


Figura 39. Número total de eventos generados en función de la IP de destino. Fuente: Elaboración propia.

Otra alternativa de visualizar datos es mediante un gráfico de tarta, tal como la figura 40. En este caso la información que proporciona es el número de alertas generadas debido a escaneos de puertos en el PLC2. Esto es una forma muy visual de ver que tipo de escaneos recibe un equipo en concreto. La expresión utilizada para generar este gráfico es:

```
sourcetype="snort3:alert:json" dest=192.168.1.100 sid IN (10000001, 10000002, 10000003, 10000004, 10000005)| chart count(dest) by msg | rename count(dest) AS "Numero de alertas"
```

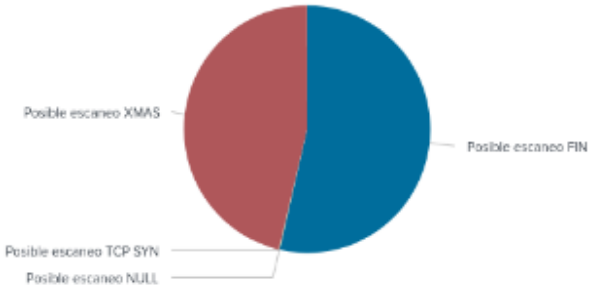


Figura 40. Número de eventos correspondientes a escaneos de puertos al PLC2 en función de cada tipo. Fuente: Elaboración propia.

Como último ejemplo se ha creado un gráfico de aspecto muy similar al que se muestra en la pantalla de búsqueda de Splunk. Así pues el gráfico mostrado en la Figura 41 se corresponde a un gráfico de barras temporal, donde se muestran los eventos generados por ataques DoS, diferenciando en los que usan paquetes TCP e ICMP. Esta gráfica es de gran utilidad a la hora de detectar cuando un equipo esta siendo atacado ya que los picos de eventos son muy pronunciados.

Figura 41. Número de eventos generados en el tiempo debido a los dos tipos de ataques DoS. Fuente: Elaboración propia.

Para poder visualizar todos los gráficos y tablas que se configuran, Splunk ofrece la posibilidad de agregarlos todos en un mismo Dashboard. Una caracterítcia muy interesante es que los Dashboards son dínamicos, es decir, que las gráficas y las tablas van ajustandose según Splunk

va generando eventos al ir recogiendo nuevas alertas generadas por Snort. En la Figura 40 se muestra el dashboard creado a partir de las gráficas mostradas anteriormente.

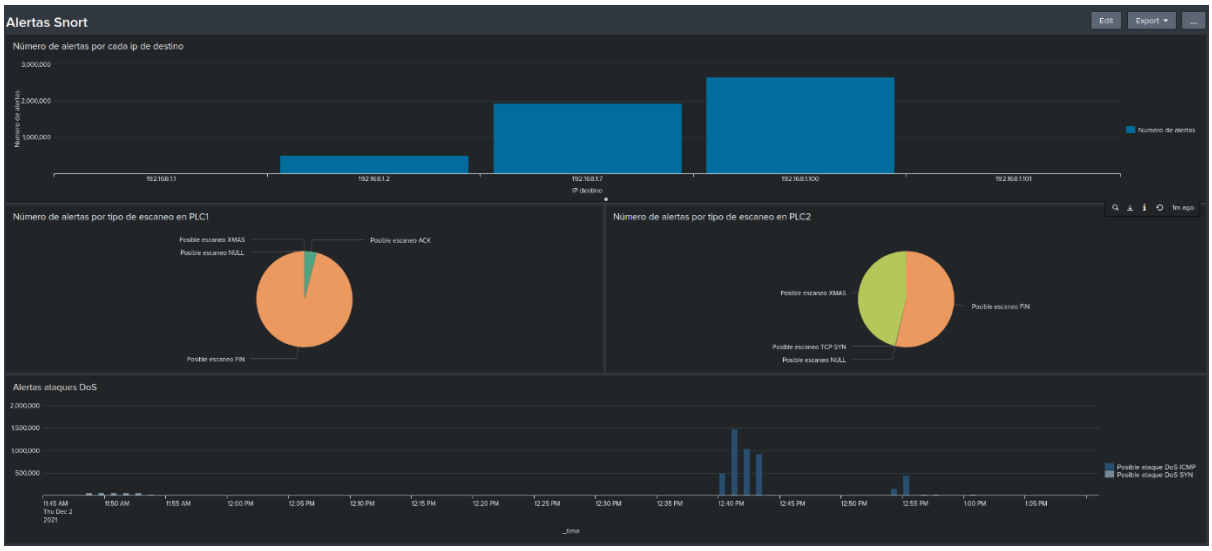


Figura 42. Gráficos mostrados en un Dashboard. Fuente: Elaboración propia.

7. Conclusiones y líneas futuras

Llegado a este último capítulo del trabajo, lo que se pretende es destacar aquellos puntos más interesantes y relevantes que se han podido aprender a lo largo del desarrollo del mismo.

En primer lugar, es innegable que en la actualidad existe una necesidad de establecer una estructura robusta de seguridad a lo largo de toda la cadena industrial, incluyendo tanto las redes IT como las OT. Esta necesidad no es algo nuevo, pero con la última revolución industrial que estamos viviendo, donde los todos los sistemas de una empresa se integran unos con otros con el propósito de agilizar procesos e incrementar la productividad, esta necesidad es mayor.

También se ha visto como los sistemas utilizados, tanto en las redes IT como en las OT siguen siendo vulnerables por fallos en su configuración, software y firmware propio. Todo esto añadido a que la interconexión de los sistemas supone más facilidades a la hora de explotar estas vulnerabilidades. Por otro lado, se han visto que existen varios métodos, sistemas y herramientas para mejorar la seguridad de red y los elementos que la componen. Entre todos estos sistemas se han visto en detalle los IDS e IPS.

Mediante la implementación de Snort, un software de código abierto IDS/IPS, se ha comprobado que una herramienta de estas características requiere cierto nivel de conocimiento en el ámbito de la arquitectura de redes y de programación para poder implementarlo y configurarlo de manera correcta. Tras llevar a cabo este proceso se ha intentado simular distantes escenarios de un ataque por medio de un intruso, los cuales se han conseguido detectar a través de las alertas que genera Snort y la visualización de las mismas a través de una interfaz web.

Con todo esto se ha podido comprobar el valor que aporta a la seguridad de la red el disponer de un sistema IDS/IPS, que es la capacidad de controlar en tiempo real lo que está ocurriendo en la red. Esto último puede permite responder de manera rápida a un ataque y denegarlo o en el peor de los casos actuar con rapidez para mitigar cualquier daño ya producido.

Como punto final a este trabajo y teniendo en cuenta las conclusiones que se han sacado al haberlo realizado, se plantean varias líneas de trabajo futuro que pueden resultar de gran interés en el ámbito de la detección y prevención de intrusos en el entorno de la Industria 4.0:

- Realizar un estudio de estas características sobre una red más compleja y que se asemeje en mayor grado a lo que uno se puede encontrar en una planta industrial. Por ejemplo, sería muy interesante configurar de forma completa la red de control integrándola con la red de campo, de tal forma que haya un flujo de información completo.
- Añadir a la red sobre la que se basa el estudio, un dispositivo capaz de actuar como un verdadero IPS, es decir, que disponga de varias tarjetas físicas y que se pueda colocar entre dos puntos de la red para bloquear tráfico potencialmente malicioso. Un producto bastante reciente que está diseñado exclusivamente para esto es el NGIPS (Next generation IPS) Firepower serie 1000, producto de la empresa Cisco.
- Comparar distintos despliegues de seguridad frente a intrusiones, por ejemplo, se podría realizar una configuración de IDS a nivel de host (en vez de a nivel de red) y de esta

manera poder comparar que tipo de configuración resulta más adecuada para distintos tipos de ataques y penetraciones.

Referencias

[1] INCIBE, “La Ciberseguridad en la Industria 4.0”, incibe-cert.es. https://www.incibe-cert.es/blog/ciberseguridad-industria-4-0 (accedido: 2 Septiembre, 2021).

[2] DBIR Team, “DBIR 2021 Data breach Investigation Report”, Verizon, 2021. Accedido: 2 Septiembre 2021. [En línea]. Disponible: https://www.verizon.com/business/resources/reports/2021-data-breach-investigations-report.pdfx

[3] CISCO, “Reporte Anual de Ciberseguridad”, Cisco, 2018. Accedido: 20 Agosto, 2021. [En línea]. Disponible: https://www.cisco.com/c/dam/global/es_mx/solutions/pdf/reporte-anual-cisco-2018-espan.pdf

[4] Claroty Team82, “Claroty bianual ICS risk & vulnerability report: 1H 2021”, Claroty, UK, 2021. Accedido: 1 Noviembre, 2021. [En línea]. Disponible: https://claroty.com/wp-content/uploads/2021/08/Claroty_Biannual_ICS_Risk_Vulnerability_Report_1H_2021.pdf

[5] J. Cavestany, “La pandemia de ciberseguridad que causó el desastre de Colonial Pipeline”, cincodias.elpais.com. https://cincodias.elpais.com/cincodias/2021/06/09/opinion/1623237705_377596.html (accedido: 5 Septiembre, 2021).

[6] M. Bahrin, M. Othman, N. Nor, y M. Azli, “Industry 4.0: A Review on Industrial Automation and Robotic”, *Jurnal Teknologi*, Junio 2016, DOI: 10.11113/jt.v78.9285.

[7] C.Torres, “Las cuatro revoluciones industriales”, power-mi.com. https://power-mi.com/es/content/las-cuatro-revoluciones-industriales (accedido: 7 Septiembre, 2021).

[8] IBM, “What technologies are driving Industry 4.0?”, ibm.com. https://www.ibm.com/topics/industry-4-0 (accedido: 7 Septiembre, 2021).

[9] M. Cotteleer, y B. Sniderman, “Forces of change: Industry 4.0”, Deloitte Insights, 2017. Accedido: 12 Septiembre, 2021. [En línea]. Disponible: https://www2.deloitte.com/content/dam/insights/us/articles/4323_Forces-of-change/4323_Forces-of-change_Ind4-0.pdf

[10] B. Sniderman, M. Mahto, y M. Cotteleer, “Industry 4.0 and manufacturing ecosystems: Exploring the world of connected enterprises”, Deloitte University Press, 2016. Accedido: 13 Septiembre, 2021. [En línea]. Disponible: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/consumer-industrial-products/Deloitte-Industry-4-0-and-manufacturing-ecosystems.pdf

[11] C. Parris, “What is the Industrial Internet of Things (IIoT)?”, ge.com. https://www.ge.com/digital/blog/ (accedido: 15 Septiembre, 2021).

[12] R. Schlaepfer, y M. Koch, “Industry 4.0: Challenges and solutions for the digital transformation and use of exponential technologies”, Deloitte, 2015. Accedido: 20 Septiembre, 2021. [En línea]. Disponible: https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/manufacturing/ch-en-manufacturing-industry-4-0-24102014.pdf

[13] F. Sevillano, et al., *Ciberseguridad Industrial e Infraestructuras Críticas*, 1ª ed. Bogotá, Colombia: Ra-ma, 2021.

[14] E. Echave, "Pirámide de Automatización Industrial", enredadoconredes.com https://enredandoconredes.com/2021/05/12/piramide-de-automatizacion-industrial-rev-12-05-21/ (accedido: 3 Noviembre, 2021).

[15] S. Mantravadi, y C. Møller, "An Overview of next generation manufacturing Execution Systems: How important is MES for Industry 4.0?", *Procedia Manufacturing*, vol. 30, pp. 588-595, 2019, DOI: 10.1016/j.promfg.2019.02.083.

[16] A. Ayerbe, "La ciberseguridad de la industria 4.0: Un medio para la continuidad del negocio", *Economía Industrial*, no. 410, pp. 37-36, 2018. [En línea]. Disponible: https://www.mincotur.gob.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/410/ANA%20I%20AYERBE.pdf

[17] *Esquema Nacional de Seguridad. Gestión de ciberincidentes*, CCN-STIC 817, CCN-CERT, Abril 2020. [En línea]. Disponible: https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/988-ccn-stic-817-gestion-de-ciberincidentes/file.html

[18] E. M. Hutchins, M. J. Cloppert, y R. M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains", Lockheed martin Corp., 2011. Accedido: 11 Noviembre, 2021. [En línea]. Disponible: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

[19] M. J. Assante, y R. M. Lee, "The Industrial Control System Cyber Kill Chain", SANS, 2015. Accedido: 11 Noviembre, 2021. [En línea]. Disponible: https://sansorg.egnyte.com/dl/HHa9fCekmc

[20] INCIBE, "Diseño y Con figuración de IPS, IDS y SIEM en Sistemas de Control Industrial", CERTSI, 2017. Accedido: 9 Septiembre, 2021. [En línea]. Disponible: https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi_diseno_configuracion_ips_ids_siem_en_sci.pdf

[21] INCIBE, "Glosario de términos de ciberseguridad: una guía de aproximación para el empresario", 2021. Accedido: 27 Septiembre, 2021. [En línea]. Disponible: https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_glosario_ciberseguridad_2021.pdf

[22] *Seguridad perimetral (cortafuegos)*, CCN-STIC 408, CCN-CERT, Marzo 2010. [En línea]. Disponible: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/74-ccn-stic-408-seguridad-perimetral-cortafuegos/file.html

[23] The Snort Team. *Snort 3 User Manual*. (2021). Accedido: 5 Noviembre, 2021. [En línea]. Disponible: https://github.com/snort3/snort3/releases/download/3.1.16.0/snort_user.pdf

[24] The Snort Project. *Snort Users Manual 2.9.16*. (2020). Accedido: 5 Noviembre, 2021. [En línea]. Disponible: https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf

[25] N. Dietrich. *Snort 3.1.6.0 on Ubuntu 18 & 20*. (2021). Accedido: 2 Noviembre, 2021. [En línea]. Disponible: https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/008/108/original/Snort_3_on_Ubuntu_18_and_2

Anexos

Anexo 1- Formato de una alerta del archivo json


```
{ "seconds" : 1638448584, "action" : "allow", "class" : "none", "dir" : "C2S", "dst_addr" : "192.168.1.2", "dst_ap" : "192.168.1.2:9666", "dst_port" : 9666, "eth_dst" : "A4:BB:6D:5C:8E:88", "eth_len" : 60, "eth_src" : "08:00:27:11:1D:3A", "eth_type" : "0x800", "gid" : 1, "iface" : "enp0s3", "ip_id" : 20417, "ip_len" : 20, "msg" : "Posible escaneo NULL", "pkt_gen" : "raw", "pkt_len" : 40, "pkt_num" : 732, "priority" : 4, "proto" : "TCP", "service" : "unknown", "sid" : 10000003, "src_addr" : "192.168.1.5", "src_ap" : "192.168.1.5:48654", "src_port" : 48654, "tcp_ack" : 0, "tcp_flags" : "********", "tcp_len" : 20, "tcp_seq" : 4012496164, "tcp_win" : 1024, "timestamp" : "12/02-13:36:24.419869" }
```