Universitat Oberta de Catalunya

uoc.edu

Industrial control systems and IoT botnets

Sergio Soro Miranda
Master en seguridad de las tecnologías de la información y de las comunicaciones
Seguridad en la internet de las cosas

Carlos Hernández Gañán
Helena Rifà Pous

04/06/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Industrial control systems and IoT botnets*
Nombre del autor:	*Sergio Soro Miranda*
Nombre del consultor/a:	*Calos Hernández Gañán*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	Master en seguridad de las tecnologías de la información y de las comunicaciones
Área del Trabajo Final:	*Seguridad en la internet de la cosas*
Idioma del trabajo:	*Castellano*
Palabras clave	*ICS, SYSTEMS, HONEYPOT*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Hoy en día los sistemas de control industrial están más conectados que nunca dada su relación con los dispositivos IoT, por este motivo cada vez están más expuesto a posibles ataques, por este motivo este trabajo se centra en la implementación de un sistema de Honeypots que nos permita recoger y analizar información de ataques a sistemas ICS. Este sistema nos permitirá estudiar de donde vienen los ataques, que métodos utilizan y que buscan los atacantes.

Otro de los objetivos desarrollados en este trabajo es que el sistema implementado sea escalable, además de que su puesta en marcha sea rápida y fácil, permitiendo que el sistema pueda añadir nuevos Honeypots para estudiar ataques a otros dispositivos.

Este trabajo se ha realizado siguiendo la metodología de ciclo en cascada mejorada, que nos permite evolucionar el sistema obtenido en función de los resultados obtenidos de iteraciones previas, de esta forma se obtiene un producto final de mayor calidad.

Abstract (in English, 250 words or less):

Nowadays industrial control systems are more connected than ever given their relationship with IoT devices, for this reason they are increasingly exposed to possible attacks, this work is focused on the implementation of a Honeypot system that allow to collect and analyze information about attacks on ICS systems. This system will allow us to study where attacks come from, what methods are used and what are looking for this attacks.

Another objective developed in this work is that the implemented system is scalable, as well as making it fast and easy to deploy, allowing the system to

add new Honeypots to study attacks on other devices.

This work has been done following the improved cascade cycle methodology, which allows us to evolve the system obtained based on the results obtained from previous iterations, in this way we obtain a final product with higher quality.

Índice

1. Introducción 6
1.1 Contexto y justificación del Trabajo 6
1.2 Objetivos del Trabajo 6
1.3 Enfoque y método seguido 6
1.4 Planificación del Trabajo 7
1.4.1 Recursos necesarios 7
1.4.2 Hitos 7
1.4.2 Planificación temporal 8
1.5 Breve sumario de productos obtenidos 11
1.6 Breve descripción de los otros capítulos de la memoria 11
2. Selección de Honeypot 12
Conpot 12
Honeyd 12
Gaspot 12
Gridpot 12
La elección 12
3. Arquitectura del sistema 16
3.1. Primer diseño de arquitectura 16
3.2. Segundo diseño de arquitectura 18
4. Configuración del sistema 21
4.1 Configuración de contenedores Conpot 21
4.2 Configuración de contenedores del sistema de análisis 22
5. Analisis de pruebas al sistema 24
6. Conclusiones 28
6.1 Objetivos completados 29
6.2 Seguimiento y planificación 29
6.3 Trabajo futuro 30
7. Glosario 31
8. Bibliografía 32
9. Anexos 34
9.1. Configuración componentes del sistema 34
9.1.1 imágenes Docker Conpot 34
9.1.2 imágenes Docker ELK 62

Lista de figuras

Metodología 1 7
Diagrama Gantt 2 10
Esquema conpot [7] 3 13
Primer diseño de arquitectura 4 16
Segundo diseño de arquitectura 5 18
Esquema de red 6 19
Esquema de implementado para pruebas 7 21
Kibana Dashboard 1 8 24
Kibana Dashboard 2 9 25
Guardian AST 10 27

Lista de de tablas

Planificación temporal 1 9
Contenedores Conpot 2 17
Contenedores ELK 3 17
Protocolos atacados 4 25
Tipo de eventos 5 26
Número de ataques por puerto 6 26
Número de ataques según la reputación del atacante 7 26
Número de ataques por país 8 26
Conpot DockerFile 9 35
Conpot Docker-compose 10 37
Conpot config 11 38
S7-200 template 12 40
Guardian AST template 13 41
S7-300 template 14 53
IPMI template 15 53
Kamstrup 384 template 16 62
Logstash Dockerfile 17 63
Logstash config 18 65
Elasticsearch Dockerfile 19 65
Elasticsearch cluser config 20 66
Kibana Dockerfile 21 67
ELK Docker-compose 22 68

1. Introducción

1.1 Contexto y justificación del Trabajo

Las soluciones tecnológicas y los procesos que utilizan el internet de las cosas se están convirtiendo en un estándar en un gran número de organizaciones e instalaciones industriales. Los atacantes ven un beneficio a la hora de explotar las vulnerabilidades de estos dispositivos, uniéndolos en grandes redes Botnet, para obtener un beneficio. Desafortunadamente, estos atacantes están utilizando los conocimientos obtenidos con IoT para utilizarlos en los sistemas de control industriales (ICS). Estos dispositivos son muy sensibles para las compañías, dado que estas dependen de los mismos para llevar a cabo el control de sus procesos y de fabricación.

Es necesario aprender cómo se efectúan estos ataques, que es lo que buscan estos atacantes, y cómo defenderse de los mismos. En estos momentos, las compañías, usan diferentes técnicas para defenderse de ataques informáticos conocidos. El problema es, que muchos de estos ataques son nuevos y no se conocen, de ahí la necesidad de investigar cómo estos atacantes utilizan las vulnerabilidades de nuestros sistemas en su propio beneficio.

El objetivo principal de este trabajo es diseñar un Honeypot que permita capturar los ataques que van dirigidos a sistemas ICS. Esto nos servirá para investigar sobre las vulnerabilidades y la explotación de estos sistemas. Este Honeypot debe de ser escalable y fácil de mantener.

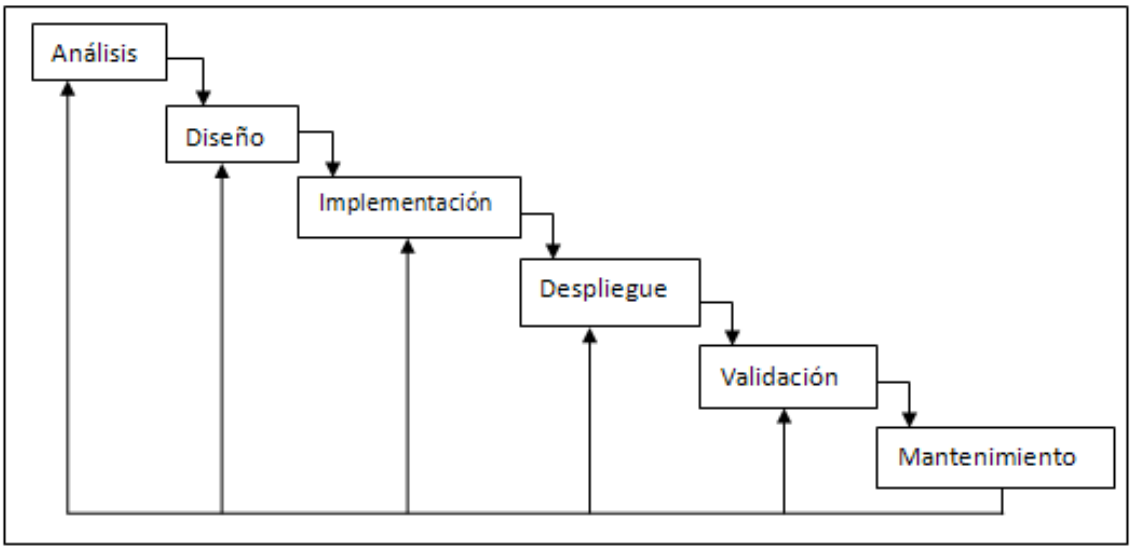
1.2 Objetivos del Trabajo

1. Seleccionar un Honeypot
2. Realizar la configuración del mismo
3. Realizar test que permitan verificar si correcto funcionamiento del sistema
4. Establecer unos criterios de escalabilidad y la configuración necesaria para este fin
5. Establecer un plan de mantenimiento adecuado y eficaz para el sistema
6. Análisis de vulnerabilidades que afectan a los sistemas ICS
7. Realizar la documentación del trabajo

1.3 Enfoque y método seguido

Se ha decidido llevar a cabo el proyecto adaptando unos de los Honeypot especializado en ICS que existen, el único condicionante para la selección del mismo es que disponga de una licencia de software libre.
La metodología que se va a utilizar para llevar a cabo una correcta gestión del proyecto, va a ser la un modelo de ciclo de desarrollo en cascada mejorado.

Esta metodología establece una organización en fases del proyecto y sus correspondientes hitos, facilitando su seguimiento, identificación de carencias y retroalimentación del desarrollo.
Las fases principales del proyecto corresponden a la fase de captura de requisitos y análisis del sistema, diseño de la solución, implementación y despliegue.
Al final de cada una de las fases se va a realizar una verificación de la misma proporcionando retroalimentación, tanto para la fase actual como las fases previas.
Se va a seguir un flujo de trabajo basado en las fases principales que se consideran en Ingeniería de sistemas, haciendo especial hincapié en aquellas relacionadas con la Ingeniería de Requisitos. Esto proporcionará como resultado un análisis profundo de las dependencias, características, necesidades y limitaciones de la solución.
La siguiente Figura muestra gráficamente el modelo de desarrollo en cascada mejorado.



Metodología 1

Esta metodología nos permitirá desarrollar cada una de las fases del proyecto con una validación y retroalimentación para las fases posteriores; de esta forma se conseguirán los objetivos marcados en el proyecto y que estos queden resueltos de la mejor forma.

1.4 Planificación del Trabajo

1.4.1 Recursos necesarios

Para llevar a cabo el proyecto es necesario disponer de un servidor con al menos 32 gigas de memoria ram, para vitalizar los sistemas necesarios que permitan realizar los test al servicio de Honeypot que vamos a implementar.

1.4.2 Hitos

El proyecto está dividido en 4 hitos parciales que permitirán evaluar los procesos conseguidos a los largo del mismo.

- **Hito1:** Definir un plan de trabajo que contenga el punto de partida del proyecto y el que queremos obtener con el mismo.
- **Hito2:** Obtener la configuración básica del Honeypot y realizar los primeros test que validen el mismo.
- **Hito3:** Obtener un sistema Honeypot escalable y validar su funcionamiento.
- **Hito4:** Definir un plan de mantenimiento, analizar vulnerabilidades en sistemas ICS con la ayuda de nuestro sistema y terminar la documentación del proyecto.

1.4.2 Planificación temporal

En la siguiente tabla se pueden ver la tareas que se van a realizar a lo largo del proyecto, el tiempo de dedicación de cada una de ellas y las dependencias entre las mismas.

Nombre de la tarea	Fecha de Inicio	Fecha final	Duración	Dependencias
TFM				
Entrega 1	**20/02/19**	**05/03/19**	**10d**	
Plan de trabajo	20/02/19	05/03/19	10d	
Hito 1	05/03/19	05/03/19	0d	
Entrega 2	**06/03/19**	**02/04/19**	**20d**	
Selección Honeypot	06/03/19	08/03/19	3d	Hito 1
Configuración	11/03/19	25/03/19	11d	Selección Honeypot
Primeros test	26/03/19	02/04/19	6d	Configuración
Hito 2	02/04/19	02/04/19	0d	Primeros test
Entrega 3	**03/04/19**	**30/04/19**	**20d**	
Criterios de escalabilidad	03/04/19	08/04/19	4d	Hito 2
Configuración sistema escalable	09/04/19	16/04/19	6d	Criterios de escalabilidad
Segunda ronda de test	17/04/19	30/04/19	10d	Configuración sistema escalable
Hito 3	30/04/19	30/04/19	0d	Segunda ronda de test
Entrega 4	**01/05/19**	**04/06/19**	**25d**	
Definición plan mantenimiento	01/05/19	14/05/19	10d	Hito 3
Análisis de vulnerabilidades	15/05/19	22/05/19	6d	Definición plan mantenimiento
Documentación del trabajo	23/05/19	03/06/19	8d	Análisis de vulnerabilidades
Hito 4	04/06/19	04/06/19	0d	Documentación del trabajo
Entrega 5	**05/06/19**	**11/06/19**	**5d**	
Presentación TFM	05/06/19	11/06/19	5d	Hito 4
Defensa TFM	**17/06/19**	**21/06/19**	**5d**	**Presentación TFM**

Planificación temporal 1

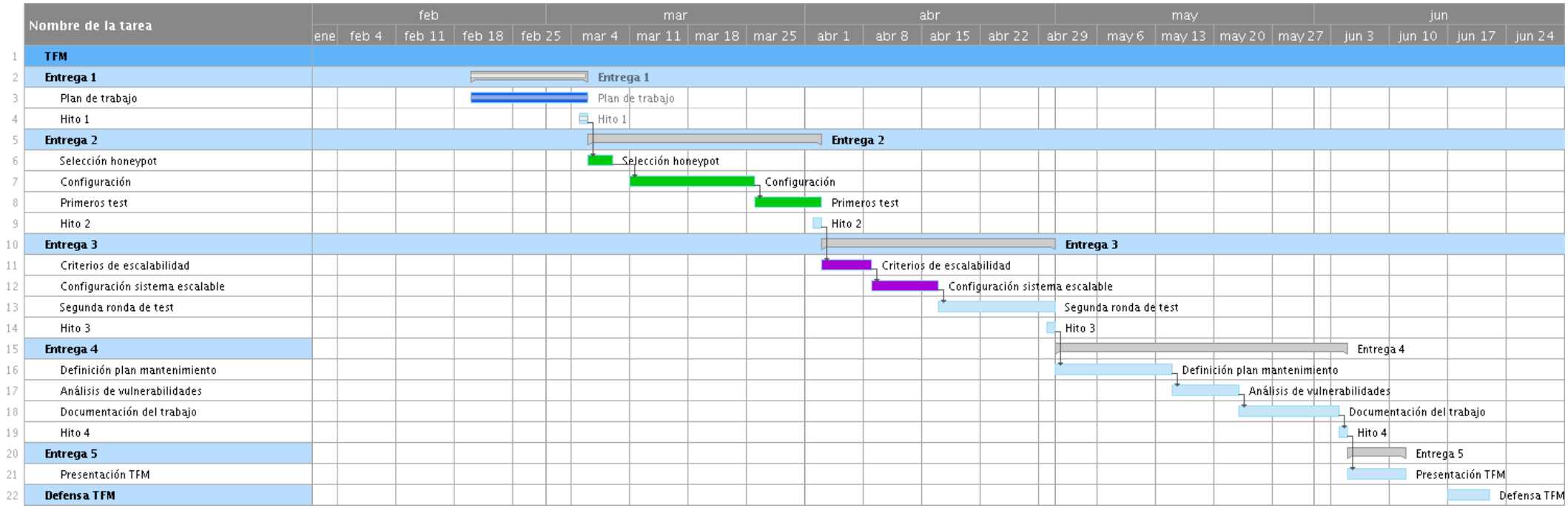


Diagrama Gantt 2

1.5 Breve sumario de productos obtenidos

Tras la realización de este proyecto el resultado ha sido la obtención de un sistema de despliegue de honeypots para entornos industriales (ICS) [17] y monitorización de los mismos. El sistema está compuesto por dos subsistemas:

1. El primer subsistema es el honeypot Conpot que se encarga de la simulación de los sistemas industriales, recoge información de los ataques recibidos para su posterior análisis. Puede haber varias instancias de este subsistema.

2. El segundo subsistema es el encargado recoger la información de los diferentes honeypot para permite analizar la misma, es un sistema ELK compuesto por tres componentes (Logstash, Elasticsearch y Kibana) [18].

1.6 Breve descripción de los otros capítulos de la memoria

2. **Selección de honeypot**
   En este capítulo se expone el análisis realizado para la selección del honeypot Conpot [5], el cual es utilizado en el sistema que se ha implementado.

3. **Arquitectura del sistema**
   En este capítulo se habla de la arquitectura del sistema implementado, su escalabilidad, mantenimiento y la posible integración con nuevos Conpot y herramientas de análisis.

4. **Configuración del sistema**
   En este capítulo se expone la configuración del sistema realizada.

5. **Análisis de pruebas al sistema**
   En este capítulo se habla de cómo explotar el sistema y se realizado un análisis de los ataques recibidos por el mismo en la fase de pruebas, esta pruebas se realizaron exponiendo nuestro sistema Conpot a internet.

6. **Conclusiones**
   En capitulo se exponen las conclusiones del sistema obtenido y del trabajo realizado durante el proyecto.

2. Selección de Honeypot

Honepots

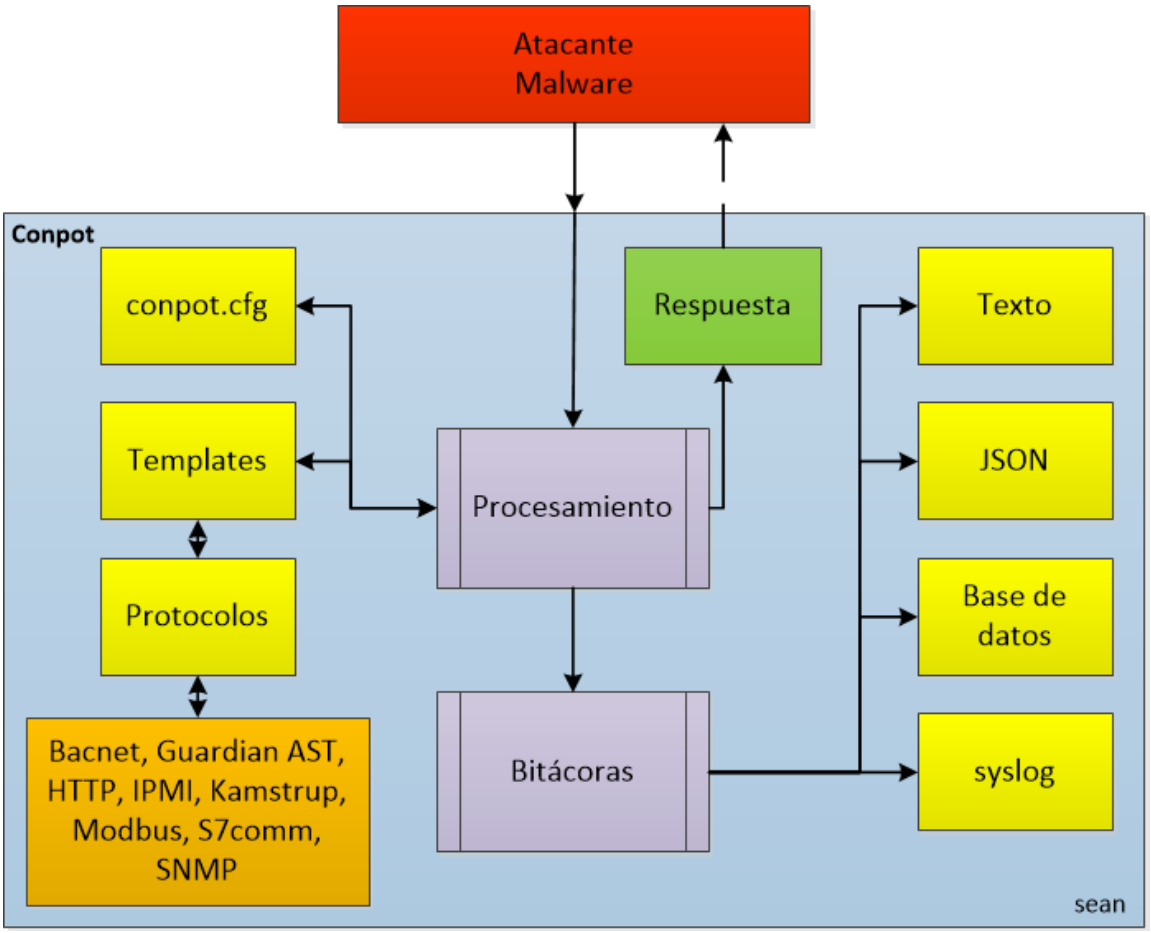
Conpot
Este Honeypot nos permite simular diferentes módulos con distintos protocolos de comunicación, Estos módulos son: Bacnet, Guardian AST, Modbus, IPMI, Kamstrup, HTTP, SNMP y s7comm. Nos permite simular el comportamiento de una CPU Siemens S7-200 (PlC industrial) Registra todas la interacciones realizadas con sus módulos, de esta forma después podemos estudiar el comportamiento de los atacantes. [5]

Honeyd
Permite simular varios dispositivos industriales basados en IP en un mismo anfitrión como por ejemplo: un servidor de Modbus/TCP en el puerto 502 y EtherNet/IP en los puertos 44818/2222. Y de esta forma recoger datos sobre los ataques que se producen a dichos dispositivos.
Para completar este honeypot necesitamos simular conexiones serie debido a que muchos dispositivos industriales utilizan RS-232/485, es posible, utilizando el modulo programado en python llamado pySerial. De esta forma presentamos un interfaz de protocolo a un atacante que se conecte por el puerto serie. [6]

Gaspot
Fue diseñado para simular un AST Veeder Root Gaurdian, los medidores de tanques comúnmente utilizados en la industria del petróleo y el gas para los tanques de gasolineras, para ayudar con el inventario de combustibles. GasPot fue diseñado para funcionar de forma totalmente aleatoria de modo que no hay dos honeypots con el mismo aspecto. [2]

Gridpot
Es un honeypot de código abierto que simula un SCADA de red eléctrica de forma realista. Gridpot es una combinación del honeypot Conpot y el simulador de redes eléctricas GridLAB-D. Esta combinación permite que este honeypot adquiera todas las ventajas de adquisición de datos de Conpot y un entorno de simulación con múltiples modelos que le aportan gran realismo gracias al simulador de redes eléctricas GridLAB-D. [3]

La elección
Finalmente se ha decidido utilizar Conpot, se ha tomado esta decisión debido que nos permite simular varios protocolos de comunicación industriales y es capaz de conectarse con otros sistema de análisis a través de sus logs, en el siguiente esquema podemos ver cómo funciona de Conpot. Cada instancia dispone de un modulo de procesamiento que se encarga de recibir la peticiones y responder a los atacantes, este procesamiento es configurable mediante la configuración general y una templates que representan el protocolo seleccionado; Después tiene un módulo de bitácora que se encarga de guardar las interacciones con los atacantes en diferentes formatos y sistemas.



Esquema conpot [7] 3

Protocolos

Modbus [8]

Es un protocolo de comunicaciones situado en los niveles 1, 2 y 7 del Modelo OSI, basado en la arquitectura maestro/esclavo (RTU) o cliente/servidor (TCP/IP), diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs). Convertido en un protocolo de comunicaciones estándar de facto en la industria, es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales.
Las principales razones por las cuales el uso de Modbus en el entorno industrial se ha impuesto a otros protocolos de comunicaciones son:

- Se diseñó teniendo en cuenta su uso para aplicaciones industriales
- Es público y gratuito
- Es fácil de implementar y requiere poco desarrollo
- Maneja bloques de datos sin suponer restricciones

Modbus TCP/IP o Modbus TCP — Se trata de una variante Modbus utilizada para comunicaciones a través de redes TCP/IP, conectándose a través del puerto 502.2 No requiere un cálculo de suma de verificación (checksum), ya que las capas inferiores ya proporcionan protección de checksum.

IPMI [9]
Es un set de especificaciones para interfaces de subsistemas autónomos que da la capacidad de manejar y monitorizar independientemente de los sistemas del host como la CPU, firmware o sistema operativo. Permite realizar operaciones de administración en sistemas informáticos como por ejemplo encender un host a través de la red, sin interactuar con el sistema operativo. El puerto utilizado por este servicio es el 623.

FTP [10]
El Protocolo de transferencia de archivos es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.
El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor y/o apropiarse de los archivos transferidos.

HTTP [11]
El protocolo de transferencia de hipertexto es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de sesión, y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado. Este protocolo usa el puerto 80.

S7comm [12]
Es un protocolo propietario de Siemens que se utiliza para comunicar diferentes PLCs de la familia de S7-300/400. Se utiliza para intercambiar información entre PLCs, para que los sistemas SCADA obtengan información de los PLC y también para controlarlos. El puerto utilizado para conectarse a los PLC es el 102.

SNMP [13]
El protocolo simple de administración de red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Los dispositivos que normalmente

soportan SNMP incluyen routers, switches, servidores, estaciones de trabajo, impresoras, bastidores de módem y muchos más. Permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento. El puerto utilizado es el 161.

Bacnet [14]
Es un protocolo que fue diseñado para comunicar los sistemas de automatización y control de edificios tales como, calefacción, ventilación, aire acondicionado, iluminación, control de acceso, detección de incendios, etc. Este protocolo permite intercambiar información entre sistemas informáticos y los dispositivos nombrados anteriormente. La variante utilizada es la que usa el protocolo UDP en el puerto 47808.

Guardian AST [15]
Es un sistema de control de tanques de almacenamiento, sirve para monitorizar el estado de los mismos y controlar su funcionamiento. El puerto utilizado para este Conpot es el 10001

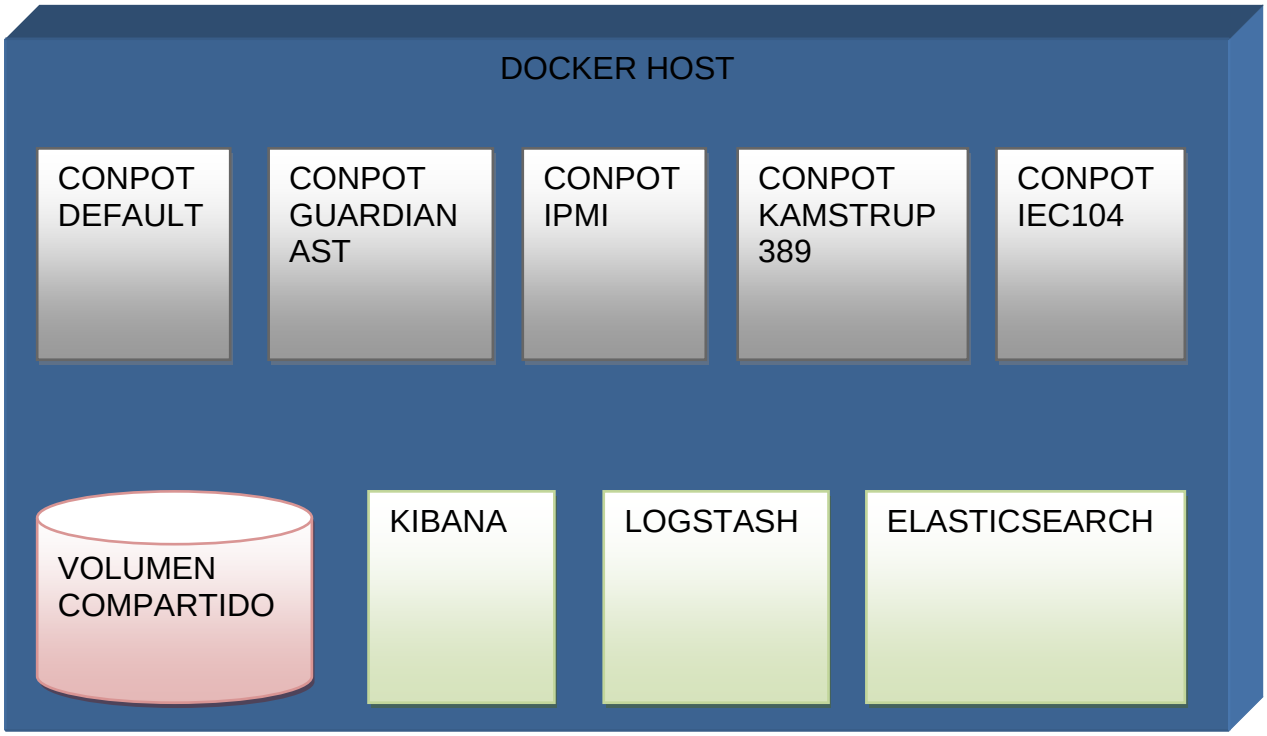
Kamstrup [16]
Kamstrup 382 es un contador de energía eléctrica de conexión directa. Los puertos utilizados para este Conpot son 1025 y 50100.

3. Arquitectura del sistema

Las tecnologías utilizadas para el diseño e implementación han sido VMware, Ubuntu 18.04 server, Docker, Docker-compose, Conpot, ELK y Filebeat.

3.1. Primer diseño de arquitectura

La arquitectura del sistema que se definió para las primeras pruebas se puede ver en la siguiente figura:



Primer diseño de arquitectura 4

Esta arquitectura se basa en un Docker hosts que corre en una máquina virtual con VMware, esta tiene el sistema operativo Ubuntu 18.04 server. En este Docker Host se despliega mediante la herramienta Docker-compose, ocho contenedores Docker que componen el sistema. Se trata de 5 contenedores honeypot de Conpot con diferentes configuraciones y otros 3 contendores para el sistema de análisis ELK que está compuesto por Logstash, Elasticsearch y Kibana. Todos estos contenedores comparten información por red y mediante un volumen compartido de disco que se usa para guardar los logs generados por los honeypots y que el sistema de análisis pueda obtenerlos mediante Logstash.

Características de los contenedores honeypot Conpot:

Nombre	Puertos	Descripción
Default	21, 69, 80, 102, 161, 502, 44818, 47808	Este contenedor simula el comportamiento de una CPU de una PLC Siemens S7-200 con dos esclavos.
Guardian AST	10001	Este contenedor simula un sistema de

		control de tanques.
IPMI	623	Este contenedor simula un dispositivo simple con una interfaz IPMI.
Kamstrup_382	1025, 50100	Este contendor simula el comportamiento de un medidor de electricidad tipo kamstrup_382.
IEC104	2404	Este contenedor simula el comportamiento de un PLC siemens S7-300 comunicándose con el protocolo IEC-104. [19]

Contenedores Conpot 2

Características de los contenedores ELK:

Nombre	**Puertos**	**Descripción**
Logstash	-	Logstash se encarga de recoger, parsear y filtrar los logs generados por los diferentes honeypot, para posteriormente pasárselos a Elasticsearch.
Elasticsearch	9200/9201	Se encarga de almacenar los datos que le envía Logstash y realizar búsquedas sobre los mismos.
Kibana	5601/5602	Se encarga de mostrarnos los datos en tiempo real que almacena Elasticsearch, podemos realizar búsquedas y crear paneles de visualización para nuestros datos..

Contenedores ELK 3

Se realizó la siguiente configuración en el firewall para exponer el sistema a internet y evaluar los ataques:

Nombre	**Puerto externo**	**Puerto Interno**	**Ip Interna**	**Protocolo**
Conpot-n	Para todos los puerto de Conpot	Para todos los puerto de Conpot	Ip de Docker host	TCP/UDP

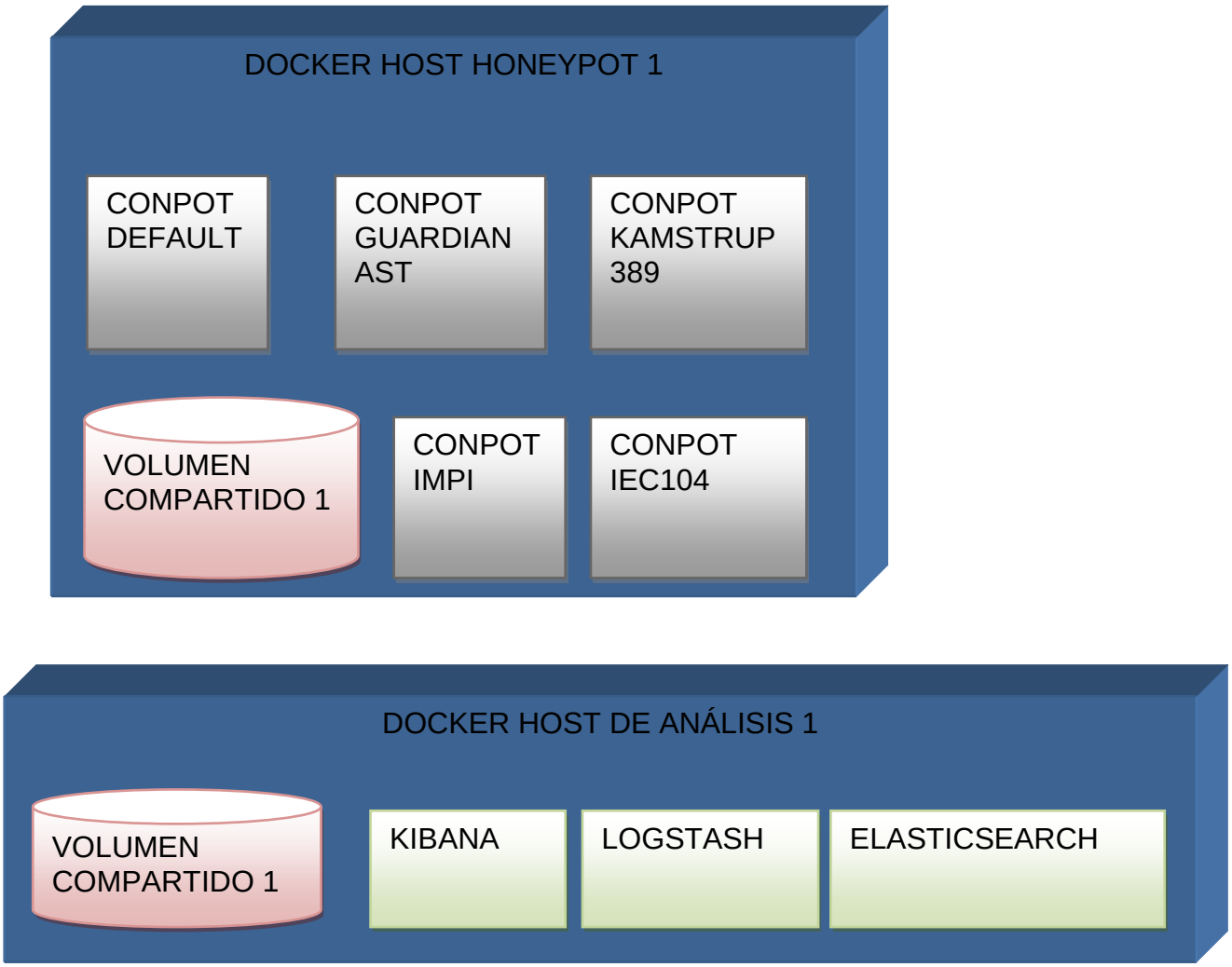
Los problemas que tiene este diseño son los siguientes:

1. No existe una separación entre los contenedores honeypot y los contenedores del ELK, esto puede exponer nuestro sistema de análisis a los ataques.
2. Esta arquitectura no es escalable, dado que solo disponemos de un Docker Host.
3. Cualquier fallo en la máquina virtual provoca que perdamos ambos sistemas.

3.2. Segundo diseño de arquitectura

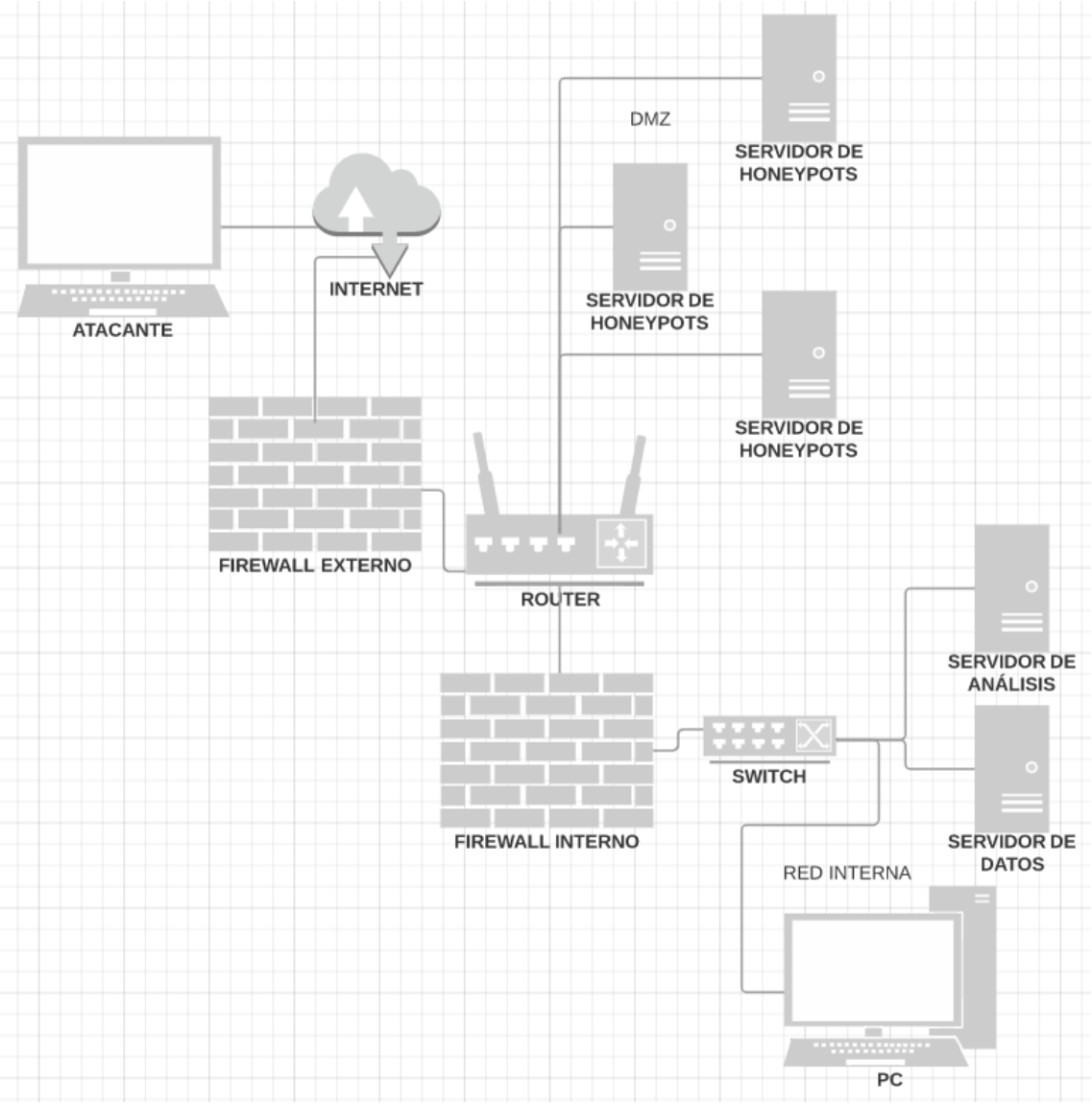
Este segundo diseño se ha realizado para que la arquitectura del sistema sea escalable, y que el sistema de análisis y de honeypots estén separados y de esta forma no se expongan ambos a los ataques, ni a posibles fallos de la máquina virtual.

En el siguiente esquema podemos ver los componentes del sistema:



Segundo diseño de arquitectura 5

Como se puede apreciar tenemos dos Docker hosts uno para los honeypots y otro para el sistema de análisis. Estos se encuentran aislados entre sí por medio de un Firewall interno que permite aislar el sistema de análisis de la subred de honeypots expuesta, en la siguiente figura podemos ver el esquema de red:



Esquema de red 6

En el diseño de red se decidió implementar una DMZ con Dual Firewall para aislar por completo la DMZ de la red interna. El único servicio queda accesible para la DMZ de la red interna es el servidor de datos necesario para guardar los logs generados por los honeypots.

Esta arquitectura nos permite escalar nuestro sistema fácilmente gracias a la tecnología de virtualización utilizada "Docker" una vez que tenemos nuestros Docker hosts, en nuestro servidores de honeypot, podemos gestionar desde nuestra red interna el despliegue de nuevos contenedores y realizar el mantenimiento de los mismos. Además en caso de que ocurra un incidente que grave podríamos recuperar nuestro sistema rápidamente, gracias a que tenemos las imágenes definidas mediante scripts de Dockerfile y Docker-compose, esto nos permite volver a recrear los contenedores rápidamente o incluso si tenemos un repositorio de imágenes no sería necesario ni recrearlas, solo descargar las mismas y poner en marcha los contenedores.

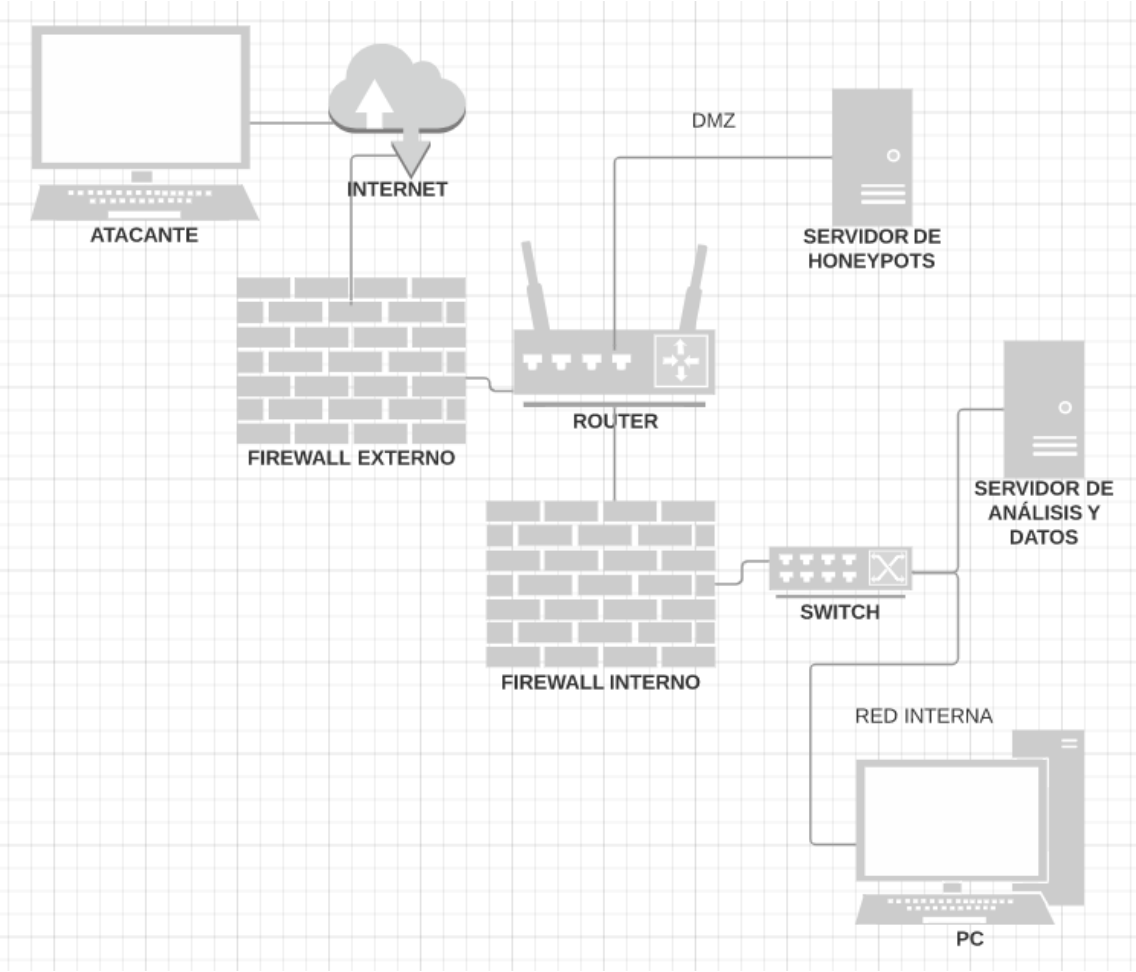
Este diseño de arquitectura soluciona los siguientes problemas del primer diseño:

1. Tenemos una separación entre los sistemas expuestos, en este caso los honeypots y los sistemas de análisis, los cuales se encuentran seguros en la red interna.
2. La arquitectura permite escalar el sistema y aumenta si disponibilidad.
3. Un fallo en una máquina virtual no afecta al resto de máquinas.

Una posible mejora a la arquitectura seria utilizar el sistema de Filebeat, que se encargaría de enviar los log directamente a Logstash y de esta forma evitaríamos compartir el servidor de datos con los honeypot.

4. Configuración del sistema

En este apartado se habla de cómo se configuro la arquitectura del entorno de pruebas, que fue el que se expuso a los ataques en internet, para intentar obtener datos de ataque a sistemas ICS.
La arquitectura implementada fue la expuesta en el segundo diseño de arquitectura, aunque solo se llegaron a generar dos Docker host, uno para los contenedores Conpot en la DMZ y otro para los contenedores de análisis de la red interna.
En el siguiente esquema podemos ver como quedo el sistema:



Esquema de implementado para pruebas 7

4.1 Configuración de contenedores Conpot

La configuración del contenedor de Conpot podemos verla en el en la tabla 9.1.1 del anexo 9. En esta tabla se defino el Dockerfile que crea nuestra imagen de Conpot. Esta imagen se crea con una serie de variables de entorno que nos permiten cambiar la configuración del contenedor cuando este se pone en marcha.
Estas variables de entorno son las siguientes:

- CONFIG: Define el archivo de configuración de Conpot “conpot.cfg”
- LOG: Define donde se guardan los logs de Conpot y el nombre de este fichero

- JSON_LOG: Define donde se guardan los logs en formato json de Conpot y el nombre de este fichero.
- TEMPLATE: Define el template que define el comportamiento del contendor Conpot.
- TMP= Define el directorio temporal de Conpot

Una vez creada la imagen del contenedor de Conpot se definio un script de Docker-compose para poner en marcha varios contenedores Conpot con distintas configuraciones para simular diversos sistemas ICS.
Esta configuración la podemos encontrar en la tabla conpot docker-compose del apartado 9.1.1 del anexo 9.
En este script se definen todos los contenedores de Conpot que se han utilizado en las pruebas, que son los especificados en el apartado de arquitectura. En la configuración se especifica que el contenedor se reinicie en caso de error, además se especifica "/data/conpot/log:/var/log/conpot:ro" el volumen compartido entre el contenedor y el Docker host para que se puedan persistir los log en el disco compartido entre los contenedores Conpot y los de análisis.
Para poner en marcha los contenedores solo debemos de ejecutar el comando "Docker-compose up" en el directorio donde se encuentra nuestro docker-compose.yml.

Podemos ver los template que definen el comportamiento de nuestros contenedores Conpot en:

Default	**S7-200 template 12**
Guardian AST	**Guardian AST template 13**
IEC104	**S7-300 template 14**
IPMI	**IPMI template 15**
Kamstrup_382	**Kamstrup 384 template 16**

4.2 Configuración de contenedores del sistema de análisis

La configuración de los contenedores que forman el sistema de análisis la podemos encontrar en la siguiente tabla:

Logstash	**Logstash Dockerfile 17**
Elasticsearch	**Elasticsearch Dockerfile 19**
Kibana	**Kibana Dockerfile 21**

En este caso tenemos tres imágenes Docker diferentes y con cada una de ellas podemos crear el contenedor específico del subsistema de análisis.
En el caso de Logstash además del Dockerfile podemos destacar su configuración, definida en "Logstash config 18", en esta se aprecia que la entrada de datos del mismo son los ficheros de log de los sistemas Conpot del

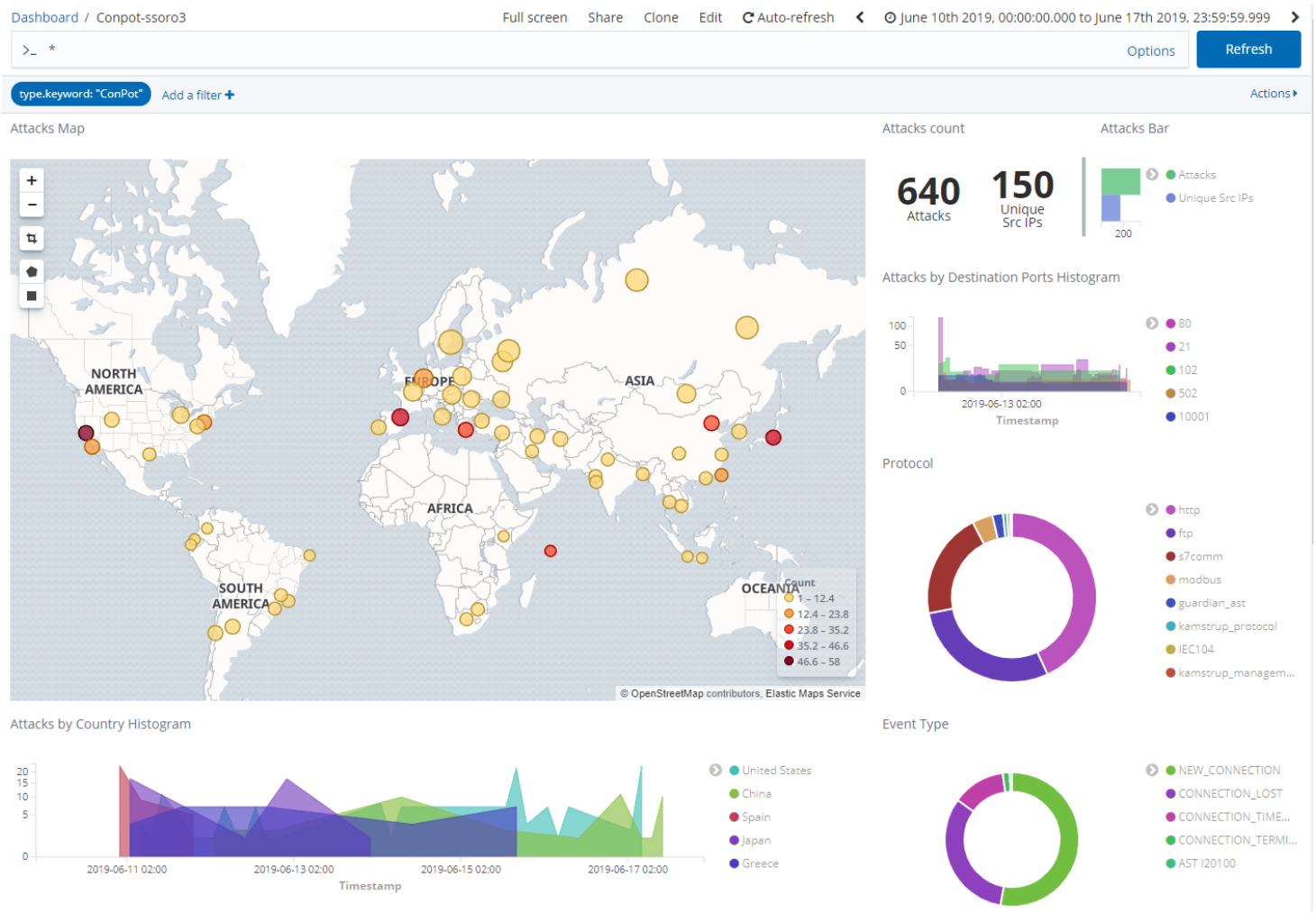
directorio "/data/conpot/log/*.json", y que envía la información después de pasar unos filtros a Elasticsearch.
El caso de Elasticsearch a aparte de su Dockerfile también tenemos la configuración del cluster de su fichero "elasticsearch.yml" que podemos ver en la tabla "Elasticsearch cluser config 20".
En el caso de Kibana se modifica el fichero de configuración durante la creación de la imagen Docker.

La puesta en marcha del sistema de análisis se realiza mediante el script de Docker-compose siguiente "ELK Docker-compose 22", en este se especifica que se debe de verificar el buen estado del cluster para que los contenedores funcionen correctamente.

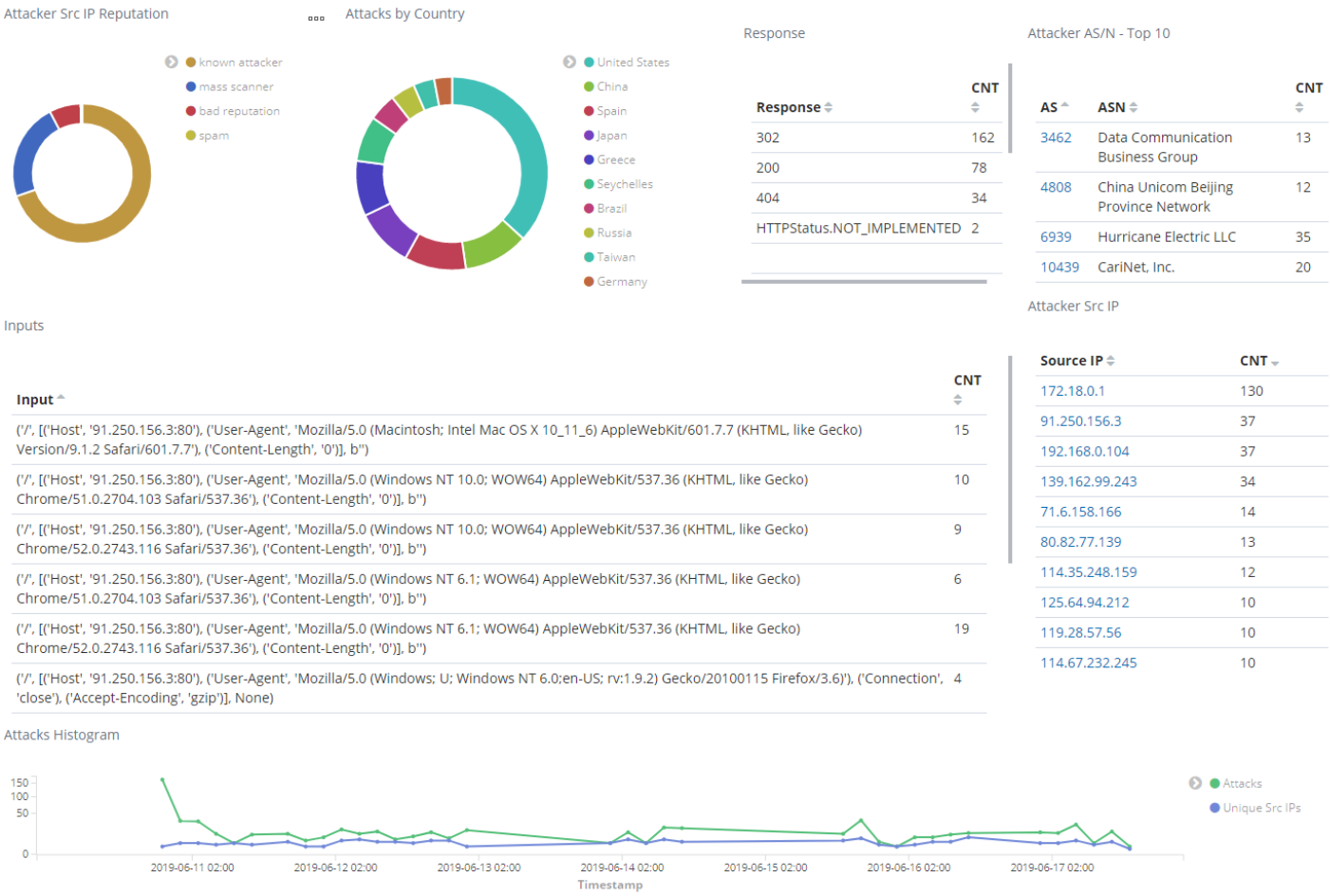
5. Analisis de pruebas al sistema

En este apartado se exponen los resultados obtenidos tras exponer nuestro sistema de honeypots Conpot a internet durante una semana desde el 10 de Junio hasta el 17 de Junio de 2019.
En las siguientes dos figuras podemos observar el Dashboard que se diseño con Kibana para monitorizar los ataques recibidos.
Este Dashboard está compuesto por las siguientes métricas o información:

- Un mapa en el que podemos ver de donde provienen los ataques.
- El número total de ataques recibidos.
- Los puertos que se atacan y el número de ataques recibidos por cada uno de ellos.
- Los protocolos que se han atacado y el número de veces que han sido atacados.
- El tipo de eventos que se han dado en cada ataque.
- La reputación de los atacantes.
- Las ips desde donde se reciben los ataques.
- El momento en el que se produjeron los ataques.
- Un descripción de los ultimos ataques.



Kibana Dashboard 1 8



Kibana Dashboard 2 9

Los resultados obtenidos los vemos en las siguientes tablas:

Como se puede observar los protocolos más que sufren más ataques o escaneos son los más comunes http y ftp. Dentro de los protocolos específicos de sistemas ICS S7comm es el más atacado.

Protocolo	Número de ataques
http	276
ftp	185
s7comm	130
modbus	25
guardian_ast	13
kamstrup_protocol	5
IEC104	4
kamstrup_management_protocol	2

Protocolos atacados 4

Los eventos detectados los podemos ver en la siguiente tabla, el único evento específicos que obtiene información de nuestro Conpot Guardian AST es el AST I20100, este se analiza más adelante.

Tipo de evento	Número de eventos
NEW_CONNECTION	95
CONNECTION_LOST	58
CONNECTION_TIMEOUT	23
CONNECTION_TERMINATED	3
AST I20100	1

Tipo de eventos 5

Los cinco puertos más atacados son los siguientes.

Puerto	Número de ataques
80	276
21	185
102	130
502	25
10001	13

Número de ataques por puerto 6

En esta tabla podemos ver la reputación que tiene las ip’s de los atacantes que realizan los ataques a nuestro honeypot "https://www.talosintelligence.com".

Reputación del atacante	Número de ataques
known attacker	182
mass scanner	59
bad reputation	19
spam	1

Número de ataques según la reputación del atacante 7

En esta tabla vemos desde que pais nos estan atacando.

País	Número de ataques
United States	135
China	40
Spain	38
Japan	36
Greece	34
Seychelles	28
Brazil	17
Russia	15
Taiwan	13
Germany	11

Número de ataques por país 8

Ataque más representativo

Ataque al puerto 10001 del Conpot de Guardian AST, se puede ver como el atacante obtiene información del estado del tanque. Esta información se definió en el template “Guardian AST template 13”.

June 12th 2019, 12:49:48.811 type: ConPot src_port: 52878 response: I20100 06/12/2019 10:49 AVIA IN-TANK INVENTORY TANK PRODUCT VOLUME TC VOLUME ULLAGE HEIGHT WATER TEMP 1 SUPER 4463 4555 5288 72.83 4.16 55.17 2 UNLEAD 3147 3235 9255 70.14 0.90 51.62 3 DIESEL 2334 2424 6440 38.29 4.75 58.71 4 ADBLUE 6275 6398 6440 39.49 7.51 59.10 t-pot_hostname: comparableemployer @timestamp: June 12th 2019, 12:49:48.811 event_type: AST I20100 public_ip: 91.250.156.3 path: /data/conpot/log/conpot_guardian_ast.json dest_ip: 172.20.0.2 @version: 1 t-pot_ip_int: 192.168.1.66 geoip.country_code2: SC geoip.longitude: 55.667 geoip.country_name: Seychelles geoip.timezone: Indian/Mahe geoip.asn: 202425 geoip.latitude: -4.583 geoip.ip: 80.82.77.139 geoip.continent_code: AF geoip.country_code3: SC geoip.as_org: IP Volume inc geoip.location: { "lat": -4.5833,

Table JSON View surrounding documents View single document

@timestamp	June 12th 2019, 12:49:48.811
@version	1
_id	e2VOS2sBPVy6GGLM4cPv
_index	logstash-2019.06.12
_score	-
_type	doc
data_type	guardian_ast
dest_ip	172.20.0.2
dest_port	10001
event_type	AST I20100
geoip.as_org	IP Volume inc
geoip.asn	202425
geoip.continent_code	AF
geoip.country_code2	SC
geoip.country_code3	SC
geoip.country_name	Seychelles
geoip.ip	80.82.77.139
geoip.latitude	-4.583
geoip.location	{ "lat": -4.5833, "lon": 55.6667 }
geoip.longitude	55.667
geoip.timezone	Indian/Mahe
host	fbed2c0ca2f6
id	f981efca-3624-449c-ae2b-782f0dcb74ae
ip_rep	known attacker
path	/data/conpot/log/conpot_guardian_ast.json
public_ip	91.250.156.3
request	-
response	I20100 06/12/2019 10:49 AVIA IN-TANK INVENTORY TANK PRODUCT VOLUME TC VOLUME ULLAGE HEIGHT WATER TEMP 1 SUPER 4463 4555 5288 72.83 4.16 55.17 2 UNLEAD 3147 3235 9255 70.14 0.90 51.62 3 DIESEL 2334 2424 6440 38.29 4.75 58.71 4 ADBLUE 6275 6398 6440 39.49 7.51 59.10
sensorid	conpot
src_ip	80.82.77.139
src_port	52878
t-pot_hostname	comparableemployer
t-pot_ip_ext	91.250.156.3
t-pot_ip_int	192.168.1.66
timestamp	June 12th 2019, 12:49:48.811
type	ConPot

Guardian AST 10

Tras analizar los resultados que se han expuesto en este apartado, se puede decir que el país desde el que más ataques se ha recibido es Estados Unidos seguido por china con menos de la mitad de ataques. La mayoría de estos ataques son realizados por atacantes ya conocidos, dado que están registrados en la base de datos de atacantes "https://www.talosintelligence.com". Estos ataques normalmente son intentos de conexión a nuestros sistemas, pero no parecen ser ataques con fines específicos salgo el ataque mencionado al Guardian AST, el cual si obtuvo información de nuestro honeypot.
Por otra parte los protocolos más atacados suelen ser http y ftp, los cuales no son protocolos específicos de sistemas industriales; En el caso de protocolos industriales el más atacado es S7comm.
Se concluye que los sistemas industriales sufren una amenaza real y que los atacantes realmente tienen interés en los mismos, dado que en solo una semana se han recibido más de 640 ataques, aunque solo uno obtuvo información específica de nuestro sistema.

6. Conclusiones

Tras la implementación del sistema, se puso este a prueba exponiéndolo a internet durante una semana, para obtener datos de ataques reales contra dispositivos ICS. En el posterior análisis de los resultados se pudo verificar el buen funcionamiento del sistema, los datos obtenidos indican que los sistemas ICS son atractivos para los atacantes dado que fueron víctima de varios intentos de conexión y en una ocasión se obtuvo información específica de uno de estos sistemas, la cual podría ser utilizada en contra de los interese del dueño del sistema atacado.

6.1 Objetivos completados

Estos fueron los objetivos planteados y los resultados obtenidos:

1. Seleccionar un Honeypot.

   Se realizó la selección de un Honeypot tipo ICS entre varias opciones.

2. Realizar la configuración del mismo.

   Se realizó una configuración del Honeypot Conpot que permitió simular varios sistemas ICS.

3. Realizar test que permitan verificar si correcto funcionamiento del sistema.

   Se puso a prueba el correcto funcionamiento del sistema.

4. Establecer unos criterios de escalabilidad y la configuración necesaria para este fin.

   Se rediseño la arquitectura del sistema para permitir que este escalara si fuera necesario.

5. Establecer un plan de mantenimiento adecuado y eficaz para el sistema.

   La arquitectura del sistema diseña hace que no requiera apenas mantenimiento.

6. Análisis de vulnerabilidades que afectan a los sistemas ICS.

   Se puso a prueba el sistema en un escenario real y se analizaron los ataques recibidos.

7. Realizar la documentación del trabajo.

   Se realizó la documentación del trabajo y los resultados obtenidos.

6.2 Seguimiento y planificación

No se puedo realizar el proyecto según la planificación establecida debido a problemas de agenda no previsto, por esta razón se tuvo que aplazar la entrega del mismo. Se constató que la metodología elegida para la realización

del proyecto fue la correcta dado que en el diseño del producto pasó por las fases que esta define que son:
Análisis, diseño, implementación, despliegue, validación y mantenimiento. Se realizó este proceso dos veces, en la primera se obtuvo como resultado la primera arquitectura que se define en el punto tres y en la segunda se obtuvo la arquitectura final.

6.3 Trabajo futuro

Como trabajo a futuro queda pendiente integrar el módulo de FileBeat para envió de log directos a Logstash desde las máquinas de honeypot y por otra parte integrar más honeypots para obtener más información en las pruebas que se realicen.

7. Glosario

- **Honeypot:** Herramienta utilizada en seguridad informática para publicar servicios vulnerables con el fin de atraer atacantes y estudiar los métodos utilizados por los mismos.
- **Docker:** Herramienta de virtualización de contenedores. [24]
- **Docker-compose:** Herramienta de despliegue y gestión de contenedores Docker. [25]
- **ICS:** Sistemas de control industrial.
- **ELK:** Sistema de análisis de monitorización de información basada en Elasticsearch, Logstash y Kibana. [18]
- **Elasticsearch:** Sistema de almacenamiento, indexación y gestión de información, utilizado para realizar búsquedas y análisis con la misma.
- **Logstash:** Herramienta de lectura, procesado y envió de logs a otros sistemas.
- **Kibana:** Frontend que muestra la información que se guarda en Elasticsearch y permite realizar búsquedas en la misma de forma fácil.
- **Filebeat:** Servicio que recopila y envía información de logs.
- **DMZ:** Zona desmilitarizada, es una red local que se ubica entre la red interna de la organización y la red externa. Las conexiones desde la red interna a la DMZ están permitidas, pero la DMZ generalmente no se puede conectar a elementos de la red interna. [26]
- **IoT:** Internet de las cosas.

8. Bibliografía

[1] https://www.incibe-cert.es/en/blog/industrial-honeypots
[Último acceso: Marzo de 2019]

[2] https://github.com/sjhilt/GasPot
[Último acceso: Marzo de 2019]

[3] https://github.com/sk4ld/gridpot
[Último acceso: Marzo de 2019]

[4] https://conpot.readthedocs.io/en/latest/
[Último acceso: Abril de 2019]

[5] http://conpot.org/
[Último acceso: Abril de 2019]

[6] https://www.gurudelainformatica.es/2016/08/seguridad-scada-honeypots-para-simular.html [Último acceso: Marzo de 2019]

[7] https://revista.seguridad.unam.mx/numero29/conpot-honeypot-de-sistemas-de-control-industrial [Último acceso: Mayo de 2019]

[8] https://es.wikipedia.org/wiki/Modbus
[Último acceso: Mayo de 2019]

[9] https://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface
[Último acceso: Mayo de 2019]

[10] https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_archivos
[Último acceso: Mayo de 2019]

[11] https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto
[Último acceso: Mayo de 2019]

[12] https://wiki.wireshark.org/S7comm
[Último acceso: Mayo de 2019]

[13] https://es.wikipedia.org/wiki/Protocolo_simple_de_administraci%C3%B3n_de_red [Último acceso: Mayo de 2019]

[14] https://wiki.wireshark.org/Protocols/bacnet
[Último acceso: Mayo de 2019]

[15] http://docs.veeder.com/gold/download.cfm?doc_id=4438
[Último acceso: Mayo de 2019]

[16] https://descargas.futurasmus-knxgroup.org/DOC/ES/Lingg&Janke/8223/5810-572-ES.pdf [Último acceso: Junio de 2019]

[17] https://en.wikipedia.org/wiki/Industrial_control_system
[Último acceso: Junio de 2019]

[18] https://www.elastic.co/es/elk-stack
[Último acceso: Junio de 2019]

[19] https://es.wikipedia.org/wiki/IEC_60870-5-101
[Último acceso: Junio de 2019]

[20] http://dtag-dev-sec.github.io/
[Último acceso: Mayo de 2019]

[21] https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones [Último acceso: Mayo de 2019]

[22] https://faculty.nps.edu/ncrowe/oldstudents/Hyun_Dahae_ICS_thesis.htm
[Último acceso: Mayo de 2019]

[23] https://github.com/mushorg/conpot
[Último acceso: Junio de 2019]

[24] https://www.docker.com/
[Último acceso: Junio de 2019]

[25] https://docs.docker.com/compose/overview/
[Último acceso: Mayo de 2019]

[26] https://es.wikipedia.org/wiki/Zona_desmilitarizada_(inform%C3%A1tica)
[Último acceso: Junio de 2019]

9. Anexos

9.1. Configuración componentes del sistema

9.1.1 imágenes Docker Conpot

```
FROM alpine

# add config files and templates
ADD config/ /root/config/

# install apps
RUN apk -U add wget python3-dev py-cryptography pkgconfig tcpdump build-base
git libcap libxslt \
             libxslt-dev mariadb-dev python3 python3-dev py-cffi file libev
libtool

# Install a config ConPot templates
RUN git clone --depth=1 https://github.com/mushorg/conpot /opt/conpot && \
    cd /opt/conpot/ && \
    # For accept ENV for MIB path
    sed -i "s/tmp_mib_dir = tempfile.mkdtemp()/tmp_mib_dir =
tempfile.mkdtemp(dir=os.environ['CONPOT_TMP'])/"
/opt/conpot/conpot/protocols/snmp/snmp_server.py && \
    # Modify template ports
    sed -i 's/port="2121"/port="21"/'
/opt/conpot/conpot/templates/default/ftp/ftp.xml && \
    sed -i 's/port="8800"/port="80"/'
/opt/conpot/conpot/templates/default/http/http.xml && \
    sed -i 's/port="6230"/port="623"/'
/opt/conpot/conpot/templates/default/ipmi/ipmi.xml && \
    sed -i 's/port="5020"/port="502"/'
/opt/conpot/conpot/templates/default/modbus/modbus.xml && \
    sed -i 's/port="10201"/port="102"/'
/opt/conpot/conpot/templates/default/s7comm/s7comm.xml && \
    sed -i 's/port="16100"/port="161"/'
/opt/conpot/conpot/templates/default/snmp/snmp.xml && \
    sed -i 's/port="6969"/port="69"/'
/opt/conpot/conpot/templates/default/tftp/tftp.xml && \
    sed -i 's/port="16100"/port="161"/'
/opt/conpot/conpot/templates/IEC104/snmp/snmp.xml && \
    sed -i 's/port="6230"/port="623"/'
/opt/conpot/conpot/templates/ipmi/ipmi/ipmi.xml && \
    pip3 install --no-cache-dir -U pip setuptools && \
    pip3 install --no-cache-dir . && \
    cd / && \
    rm -rf /opt/conpot /tmp/* /var/tmp/* && \
    setcap cap_net_bind_service=+ep /usr/bin/python3.6 && \

# Get user, groups, configs
    mkdir -p /etc/conpot /var/log/conpot && \
      addgroup -g 5000 conpot && \
    adduser -S -s /bin/bash -u 5000 -D -g 5000 conpot && \
    cp /root/config/conpot.cfg /etc/conpot/conpot.cfg && \
    cp -R /root/config/templates /usr/lib/python3.6/site-packages/conpot/ &&
\
```

```
# Remove apps
    apk del --purge build-base cython-dev file git libev libtool libxslt-dev
mariadb-dev \
            pkgconfig python3-dev py-cffi wget && \

# Remove temporal files
    rm -rf /root/* && \
    rm -rf /tmp/* && \
    rm -rf /var/cache/apk/*

# CMD
STOPSIGNAL SIGINT
USER conpot:conpot
CMD exec /usr/bin/conpot --config $CONFIG --logfile $LOG --template
$TEMPLATE --temp_dir $TMP
```

Conpot DockerFile 9

```
networks:
  default_network:
  guardian_ast_network:
  IEC104_network:
  ipmi_network:
  kamstrup_382_network:

services:

# Default service
  conpot_default:
    build: .
    container_name: conpot_default
    image: "conpot/ssoro"
    restart: always
    networks:
     - default_network
    ports:
     - "69:69"
     - "80:80"
     - "102:102"
     - "161:161"
     - "502:502"
     - "2121:21"
     - "44818:44818"
     - "47808:47808"
    environment:
     - CONFIG=/etc/conpot/conpot.cfg
     - LOG=/var/log/conpot/conpot_default.log
     - JSON_LOG=/var/log/conpot/conpot_default.json
     - TEMPLATE=default
     - TMP=/tmp/conpot
    tmpfs:
     - /tmp/conpot:uid=5000,gid=5000
    volumes:
     - /data/conpot/log:/var/log/conpot:ro

# Guardian_ast service
  conpot_guardian_ast:
    build: .
    container_name: conpot_guardian_ast
```

```
    image: "conpot/ssoro"
    restart: always
    networks:
     - guardian_ast_network
    ports:
     - "10001:10001"
    environment:
     - CONFIG=/etc/conpot/conpot.cfg
     - LOG=/var/log/conpot/conpot_guardian_ast.log
     - JSON_LOG=/var/log/conpot/conpot_guardian_ast.json
     - TEMPLATE=guardian_ast
     - TMP=/tmp/conpot
    tmpfs:
     - /tmp/conpot:uid=5000,gid=5000
    volumes:
     - /data/conpot/log:/var/log/conpot:ro

# IEC104 service
  conpot_IEC104:
    build: .
    container_name: conpot_IEC104
    image: "conpot/ssoro"
    restart: always
    networks:
     - IEC104_network
    ports:
     - "2404:2404"
    environment:
     - CONFIG=/etc/conpot/conpot.cfg
     - LOG=/var/log/conpot/conpot_IEC104.log
     - JSON_LOG=/var/log/conpot/conpot_IEC104.json
     - TEMPLATE=IEC104
     - TMP=/tmp/conpot
    tmpfs:
     - /tmp/conpot:uid=5000,gid=5000
    read_only: true
    volumes:
     - /data/conpot/log:/var/log/conpot

# Kamstrup_382
  conpot_kamstrup_382:
    build: .
    container_name: conpot_kamstrup_382
    image: "conpot/ssoro"
    restart: always
    networks:
     - kamstrup_382_network
    ports:
     - "1025:1025"
     - "50100:50100"
    environment:
     - CONFIG=/etc/conpot/conpot.cfg
     - LOG=/var/log/conpot/conpot_kamstrup_382.log
     - JSON_LOG=/var/log/conpot/conpot_kamstrup_382.json
     - TEMPLATE=kamstrup_382
     - TMP=/tmp/conpot
    tmpfs:
     - /tmp/conpot:uid=5000,gid=5000
    volumes:
```

```
     - /data/conpot/log:/var/log/conpot:ro

# Ipmi
  conpot_ipmi:
    build: .
    container_name: conpot_ipmi
    image: "conpot/ssoro"
    restart: always
    networks:
     - ipmi_network
    ports:
     - "623:623"
    environment:
     - CONFIG=/etc/conpot/conpot.cfg
     - LOG=/var/log/conpot/conpot_ipmi.log
     - JSON_LOG=/var/log/conpot/conpot_ipmi.json
     - TEMPLATE=ipmi
     - TMP=/tmp/conpot
    tmpfs:
     - /tmp/conpot:uid=5000,gid=5000
    volumes:
     - /data/conpot/log:/var/log/conpot:ro
```

Conpot Docker-compose 10

```
[common]
sensorid = conpot

[virtual_file_system]
data_fs_url = %(TMP)s
fs_url = tar:///usr/lib/python3.6/site-packages/conpot/data.tar

[session]
timeout = 30

[daemon]
user = conpot
group = conpot

[mysql]
enabled = False
device = /tmp/mysql.sock
host = localhost
port = 3306
db = conpot
username = conpot
passphrase = conpot
socket = tcp

[syslog]
enabled = False
device = /dev/log
host = localhost
port = 514
facility = local0
socket = dev

[hpfriends]
```

```
enabled = False
host = hpfriends.honeycloud.net
port = 20000
ident = 3Ykf9Znv
secret = 4nFRhpm44QkG9cvD
channels = ["conpot.events", ]

[taxii]
enabled = False
host = taxiitest.mitre.org
port = 80
inbox_path = /services/inbox/default/
use_https = False

[fetch_public_ip]
enabled = True
urls = ["http://whatismyip.akamai.com/", "http://wgetip.com/"]

[change_mac_addr]
enabled = False
iface = eth0
addr = 00:ed:12:ec:fe:00

[json]
enabled = True
filename = %(JSON_LOG)s

[sqlite]
enabled = False
```

Conpot config 11

```
<core>
    <template>
        <!-- General information about the template -->
        <entity name="unit">S7-200</entity>
        <entity name="vendor">Siemens</entity>
        <entity name="description">Rough simulation of a basic Siemens S7-
200 CPU with 2 slaves</entity>
        <entity name="protocols">HTTP, MODBUS, s7comm, SNMP</entity>
        <entity name="creator">the conpot team</entity>
    </template>
    <databus>
        <!-- Core value that can be retrieved from the databus by key -->
        <key_value_mappings>
            <key name="FacilityName">
                <value type="value">"DoE Water Service"</value>
            </key>
            <key name="SystemName">
                <value type="value">"Central Pump"</value>
            </key>
            <key name="SystemDescription">
                <value type="value">"Pump Control Unit"</value>
            </key>
            <key name="Uptime">
                <value
type="function">conpot.emulators.misc.uptime.Uptime</value>
            </key>
            <key name="sysObjectID">
```

```
                <value type="value">"0.0"</value>
            </key>
            <key name="sysContact">
                <value type="value">"DoE"</value>
            </key>
            <key name="sysName">
                <value type="value">"Pump Control Unit"</value>
            </key>
            <key name="sysLocation">
                <value type="value">"DoE"</value>
            </key>
            <key name="sysServices">
                <value type="value">"72"</value>
            </key>
            <key name="memoryModbusSlave0BlockA">
                <value type="value">[random.randint(0,1) for b in
range(0,128)]</value>
            </key>
            <key name="memoryModbusSlave0BlockB">
                <value type="value">[random.randint(0,1) for b in
range(0,32)]</value>
            </key>
            <key name="memoryModbusSlave255BlockA">
                <value type="value">[random.randint(0,1) for b in
range(0,128)]</value>
            </key>
            <key name="memoryModbusSlave255BlockB">
                <value type="value">[random.randint(0,1) for b in
range(0,32)]</value>
            </key>
            <key name="memoryModbusSlave1BlockA">
                <value type="value">[random.randint(0,1) for b in
range(0,128)]</value>
            </key>
            <key name="memoryModbusSlave1BlockB">
                <value type="value">[random.randint(0,1) for b in
range(0,32)]</value>
            </key>
            <key name="memoryModbusSlave2BlockC">
                <value type="value">[random.randint(0,1) for b in
range(0,8)]</value>
            </key>
            <key name="memoryModbusSlave2BlockD">
                <value type="value">[0 for b in range(0,32)]</value>
            </key>
            <key name="Copyright">
                <value type="value">"Original Siemens Equipment"</value>
            </key>
            <key name="s7_id">
                <value type="value">"88111222"</value>
            </key>
            <key name="s7_module_type">
                <value type="value">"IM151-8 PN/DP CPU"</value>
            </key>
            <key name="empty">
                <value type="value">""</value>
            </key>
        </key_value_mappings>
    </databus>
```

```
</core>
```

S7-200 template 12

```
<core>
    <template>
        <!-- General information about the template -->
        <entity name="unit">Guardian AST tank-monitoring system</entity>
        <entity name="vendor">Guardian</entity>
        <entity name="description">Guardian AST tank-monitoring
system</entity>
        <entity name="protocols">guardian_ast</entity>
        <entity name="creator">the conpot team</entity>
    </template>
    <databus>
        <!-- Core value that can be retrieved from the databus by key -->
        <key_value_mappings>
            <key name="product1">
                <value type="value">"SUPER"</value>
            </key>
            <key name="product2">
                <value type="value">"UNLEAD"</value>
            </key>
            <key name="product3">
                <value type="value">"DIESEL"</value>
            </key>
            <key name="product4">
                <value type="value">"ADBLUE"</value>
            </key>
            <key name="station_name">
                <value type="value">"AVIA"</value>
            </key>
            <key name="vol1">
                <value type="value">random.randint(1000, 9050)</value>
            </key>
            <key name="vol2">
                <value type="value">random.randint(1000, 9050)</value>
            </key>
            <key name="vol3">
                <value type="value">random.randint(1000, 9050)</value>
            </key>
            <key name="vol4">
                <value type="value">random.randint(1000, 9050)</value>
            </key>
            <key name="ullage1">
                <value type="value">random.randint(3000, 9999)</value>
            </key>
            <key name="ullage2">
                <value type="value">random.randint(3000, 9999)</value>
            </key>
            <key name="ullage3">
                <value type="value">random.randint(3000, 9999)</value>
            </key>
            <key name="ullage4">
                <value type="value">random.randint(3000, 9999)</value>
            </key>
            <key name="height1">
                <value type="value">round(random.uniform(25.00, 75.99),
2)</value>
```

```
            </key>
            <key name="height2">
                <value type="value">round(random.uniform(25.00, 75.99),
2)</value>
            </key>
            <key name="height3">
                <value type="value">round(random.uniform(25.00, 75.99),
2)</value>
            </key>
            <key name="height4">
                <value type="value">round(random.uniform(25.00, 75.99),
2)</value>
            </key>
            <key name="h2o1">
                <value type="value">round(random.uniform(0.0, 9.99),
2)</value>
            </key>
            <key name="h2o2">
                <value type="value">round(random.uniform(0.0, 9.99),
2)</value>
            </key>
            <key name="h2o3">
                <value type="value">round(random.uniform(0.0, 9.99),
2)</value>
            </key>
            <key name="h2o4">
                <value type="value">round(random.uniform(0.0, 9.99),
2)</value>
            </key>
            <key name="temp1">
                <value type="value">round(random.uniform(50.0, 59.99),
2)</value>
            </key>
            <key name="temp2">
                <value type="value">round(random.uniform(50.0, 59.99),
2)</value>
            </key>
            <key name="temp3">
                <value type="value">round(random.uniform(50.0, 59.99),
2)</value>
            </key>
            <key name="temp4">
                <value type="value">round(random.uniform(50.0, 59.99),
2)</value>
            </key>
            <key name="empty">
                <value type="value">""</value>
            </key>
        </key_value_mappings>
    </databus>
</core>
```

Guardian AST template 13

```
<!-- Copyright (C) 2017  Patrick Reichenberger (University of Passau)
<patrick.reichenberger@t-online.de>

 This program is free software; you can redistribute it and/or
 modify it under the terms of the GNU General Public License
 as published by the Free Software Foundation; either version 2
```

```
 of the License, or (at your option) any later version.

 This program is distributed in the hope that it will be useful,
 but WITHOUT ANY WARRANTY; without even the implied warranty of
 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 GNU General Public License for more details.

 You should have received a copy of the GNU General Public License
 along with this program; if not, write to the Free Software
 Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
-->

<core>
    <template>
        <!-- General information about the template -->
        <entity name="unit">S7-300</entity>
        <entity name="vendor">Siemens</entity>
        <entity name="description">Creates a simple device for IEC 60870-5-
104</entity>
        <entity name="protocols">IEC104, SNMP</entity>
        <entity name="creator">Patrick Reichenberger</entity>
    </template>
    <databus>
        <!-- Core value that can be retrieved from the databus by key -->
        <key_value_mappings>
            <!-- SNMPv2-MIB -->
            <key name="SystemDescription">
                <value type="value">"Siemens, SIMATIC, S7-300"</value>
            </key>
            <key name="sysObjectID">
                <value type="value">"0.0"</value>
            </key>
            <key name="Uptime">
                <value
type="function">conpot.emulators.misc.uptime.Uptime</value>
            </key>
            <key name="sysContact">
                <value type="value">"Corporate IT"</value>
            </key>
            <key name="sysName">
                <value type="value">"DE-BER01"</value>
            </key>
            <key name="sysLocation">
                <value type="value">"BER01, T2E"</value>
            </key>
            <key name="sysServices">
                <value type="value">"72"</value>
            </key>
            <!-- IF-MIB -->
            <key name="ifNumber">
                <value type="value">1</value>
            </key>
            <key name="ifIndex">
                <value type="value">1</value>
            </key>
            <key name="ifDescr">
                <value type="value">"Siemens, SIMATIC NET, CP 343-1 PN, 6GK7
343-1EX21-0XE0, HW: Version 2, FW: Version V1.2.3, Ethernet Port 1, Rack 0,
```

```
100Mbit"</value>
            </key>
            <key name="ifType">
                <value type="value">6</value>
            </key>
            <key name="ifMtu">
                <value type="value">1000</value>
            </key>
            <key name="ifSpeed">
                <value type="value">100000000</value>
            </key>
            <key name="ifPhysAddress">
                <value type="value">"\x00\x0e\x8c\x29\xc5\x1a"</value>
            </key>
            <key name="ifAdminStatus">
                <value type="value">1</value>
            </key>
            <key name="ifOperStatus">
                <value type="value">1</value>
            </key>
            <key name="ifLastChange">
                <value
type="function">conpot.emulators.misc.uptime.Uptime</value>
            </key>
            <key name="FacilityName">
                <value type="value">"Compagnie Generale des Eaux"</value>
            </key>
            <key name="0">
                <value type="value">0</value>
            </key>
            <key name="1">
                <value type="value">1</value>
            </key>
            <key name="ifInOctets">
                <value type="value">1618895</value>
            </key>
            <key name="ifInUcastPkts">
                <value type="value">7018</value>
            </key>
            <key name="ifInNUcastPkts">
                <value type="value">291</value>
            </key>
            <key name="ifOutOctets">
                <value type="value">455107</value>
            </key>
            <key name="ifOutUcastPkts">
                <value type="value">872264</value>
            </key>
            <key name="ifOutUNcastPkts">
                <value type="value">143</value>
            </key>

            <!-- IP-MIB -->
            <key name="ipForwarding">
                <value type="value">2</value>
            </key>
            <key name="ipDefaultTTL">
                <value type="value">60</value>
            </key>
```

```
            <key name="ipInReceives">
                <value type="value">31271</value>
            </key>
            <key name="ipInHdrErrors">
                <value type="value">0</value>
            </key>
            <key name="ipInAddrErrors">
                <value type="value">0</value>
            </key>
            <key name="ipForwDatagrams">
                <value type="value">0</value>
            </key>
            <key name="ipInUnknownProtos">
                <value type="value">0</value>
            </key>
            <key name="ipInDiscards">
                <value type="value">0</value>
            </key>
            <key name="ipInDelivers">
                <value type="value">31282</value>
            </key>
            <key name="ipOutRequests">
                <value type="value">69023</value>
            </key>
            <key name="ipOutDiscards">
                <value type="value">0</value>
            </key>
            <key name="ipOutNoRoutes">
                <value type="value">0</value>
            </key>
            <key name="ipReasmTimeout">
                <value type="value">60</value>
            </key>
            <key name="ipReasmReqds">
                <value type="value">7</value>
            </key>
            <key name="ipReasmOKs">
                <value type="value">3</value>
            </key>
            <key name="ipReasmFails">
                <value type="value">0</value>
            </key>
            <key name="ipFragOKs">
                <value type="value">0</value>
            </key>
            <key name="ipFragFails">
                <value type="value">0</value>
            </key>
            <key name="ipFragCreates">
                <value type="value">0</value>
            </key>
            <key name="ipAdEntAddr">
                <value type="value">"217.172.190.137"</value>
            </key>
            <key name="ipAdEntIfIndex">
                <value type="value">1</value>
            </key>
            <key name="ipAdEntNetMask">
                <value type="value">"255.255.255.255"</value>
```

```
            </key>
            <key name="ipAdEntBcastAddr">
                <value type="value">1</value>
            </key>
            <key name="ipAdEntReasmMaxSize">
                <value type="value">65528</value>
            </key>
            <key name="ipRoutingDiscards">
                <value type="value">0</value>
            </key>
            <key name="icmpInMsgs">
                <value type="value">4</value>
            </key>
            <key name="icmpInErrors">
                <value type="value">0</value>
            </key>
            <key name="icmpInDestUnreachs">
                <value type="value">1</value>
            </key>
            <key name="icmpInTimeExcds">
                <value type="value">0</value>
            </key>
            <key name="icmpInParmProbs">
                <value type="value">0</value>
            </key>
            <key name="icmpInSrcQuenchs">
                <value type="value">0</value>
            </key>
            <key name="icmpInRedirects">
                <value type="value">0</value>
            </key>
            <key name="icmpInEchos">
                <value type="value">0</value>
            </key>
            <key name="icmpInEchoReps">
                <value type="value">0</value>
            </key>
            <key name="icmpInTimestamps">
                <value type="value">0</value>
            </key>
            <key name="icmpInTimestampReps">
                <value type="value">0</value>
            </key>
            <key name="icmpInAddrMasks">
                <value type="value">0</value>
            </key>
            <key name="icmpInAddrMaskReps">
                <value type="value">0</value>
            </key>
            <key name="icmpOutMsgs">
                <value type="value">0</value>
            </key>
            <key name="icmpOutErrors">
                <value type="value">0</value>
            </key>
            <key name="icmpOutDestUnreachs">
                <value type="value">144</value>
            </key>
            <key name="icmpOutTimeExcds">
```

```
            <value type="value">0</value>
        </key>
        <key name="icmpOutParmProbs">
            <value type="value">0</value>
        </key>
        <key name="icmpOutSrcQuenchs">
            <value type="value">0</value>
        </key>
        <key name="icmpOutRedirects">
            <value type="value">0</value>
        </key>
        <key name="icmpOutEchos">
            <value type="value">0</value>
        </key>
        <key name="icmpOutEchoReps">
            <value type="value">0</value>
        </key>
        <key name="icmpOutTimestamps">
            <value type="value">0</value>
        </key>
        <key name="icmpOutTimestampReps">
            <value type="value">0</value>
        </key>
        <key name="icmpOutAddrMasks">
            <value type="value">0</value>
        </key>
        <key name="icmpOutAddrMaskReps">
            <value type="value">0</value>
        </key>

        <!-- TCP-MIB -->
        <key name="tcpRtoAlgorithm">
            <value type="value">2</value>
        </key>
        <key name="tcpRtoMin">
            <value type="value">0</value>
        </key>
        <key name="tcpRtoMax">
            <value type="value">100</value>
        </key>
        <key name="tcpMaxConn">
            <value type="value">-1</value>
        </key>
        <key name="tcpActiveOpens">
            <value type="value">0</value>
        </key>
        <key name="tcpPassiveOpens">
            <value type="value">101</value>
        </key>
        <key name="tcpAttemptFails">
            <value type="value">42</value>
        </key>
        <key name="tcpEstabResets">
            <value type="value">45</value>
        </key>
        <key name="tcpCurrEstab">
            <value type="value">0</value>
        </key>
        <key name="tcpInSegs">
```

```
            <value type="value">30321</value>
        </key>
        <key name="tcpOutSegs">
            <value type="value">67821</value>
        </key>
        <key name="tcpRetransSegs">
            <value type="value">2511</value>
        </key>
        <key name="tcpConnState">
            <value type="value">2</value>
        </key>
        <key name="tcpConnLocalAddress">
            <value type="value">"217.172.190.137"</value>
        </key>
        <key name="tcpConnLocalPort">
            <value type="value">2404</value>
        </key>
        <key name="tcpConnRemAddress">
            <value type="value">"0.0.0.0"</value>
        </key>
        <key name="tcpConnRemPort">
            <value type="value">0</value>
        </key>
        <key name="tcpInErrs">
            <value type="value">1</value>
        </key>
        <key name="tcpOutRsts">
            <value type="value">728</value>
        </key>
        <!-- UDP-MIB -->
        <key name="udpInDatagrams">
            <value type="value">1441</value>
        </key>
        <key name="udpNoPorts">
            <value type="value">1280</value>
        </key>
        <key name="udpInErrors">
            <value type="value">23</value>
        </key>
        <key name="udpOutDatagrams">
            <value type="value">47</value>
        </key>
        <key name="udpLocalAddress">
            <value type="value">"217.172.190.137"</value>
        </key>
        <key name="udpLocalPort">
            <value type="value">161</value>
        </key>
        <key name="SystemName">
            <value type="value">"CP 343-1 IT"</value>
        </key>


        <!-- IEC104 Protocol parameter -->
        <!-- Timeout of connection establishment -->
        <key name="T_0">
            <value type="value">30</value>
        </key>
        <!-- Timeout of send or test APDUs (Wartezeit auf Quittung) -->
```

```
            <key name="T_1">
                <value type="value">15</value>
            </key>
            <!-- Timeout for acknowledges in case of no data messages T_2 <
T_1 (Quittieren nach x sek) -->
            <key name="T_2">
                <value type="value">10</value>
            </key>
            <!-- Timeout for sending test frames in case of a long idle
state -->
            <key name="T_3">
                <value type="value">20</value>
            </key>
            <!-- Maximum difference receive sequence number to send state
variable (Max. Anzahl unquittierter Telegramme) -->
            <!-- not implemented yet -->
            <key name="k">
                <value type="value">12</value>
            </key>
            <!-- Latest acknowledge after receiving w I-format APDUs
(Quittieren nach w Telegrammen) -->
            <key name="w">
                <value type="value">8</value>
            </key>
            <!-- Maximum frame size (in bytes) -->
            <key name="MaxFrameSize">
                <value type="value">254</value>
            </key>

            <!-- Devices -->
            <!-- 13- -->
            <key name="13_20">
                <value type="value">1</value>
            </key>
            <key name="13_21">
                <value type="value">0</value>
            </key>
            <key name="13_22">
                <value type="value">0</value>
            </key>
            <key name="13_24">
                <value type="value">1</value>
            </key>
            <key name="13_25">
                <value type="value">1</value>
            </key>
            <key name="13_32">
                <value type="value">1</value>
            </key>
            <key name="13_33">
                <value type="value">1</value>
            </key>
            <key name="13_34">
                <value type="value">1</value>
            </key>
            <key name="13_35">
                <value type="value">1</value>
            </key>
            <key name="13_36">
```

```
            <value type="value">1</value>
        </key>
        <key name="13_37">
            <value type="value">1</value>
        </key>
        <key name="13_38">
            <value type="value">1</value>
        </key>
        <key name="13_39">
            <value type="value">1</value>
        </key>
        <key name="13_40">
            <value type="value">0</value>
        </key>
        <key name="13_41">
            <value type="value">1</value>
        </key>
        <key name="13_42">
            <value type="value">0</value>
        </key>

        <!-- 22- -->
        <key name="22_19">
            <value type="value">1</value>
        </key>
        <key name="22_20">
            <value type="value">1</value>
        </key>
        <key name="22_21">
            <value type="value">0</value>
        </key>
        <key name="22_22">
            <value type="value">0</value>
        </key>
        <key name="22_24">
            <value type="value">1</value>
        </key>
        <key name="22_25">
            <value type="value">1</value>
        </key>
        <key name="22_42">
            <value type="value">1</value>
        </key>
        <key name="22_43">
            <value type="value">1</value>
        </key>
        <key name="22_54">
            <value type="value">1</value>
        </key>

        <!-- 33- -->
        <key name="33_2">
            <value type="value">1</value>
        </key>
        <key name="33_3">
            <value type="value">2</value>
        </key>
        <key name="33_4">
            <value type="value">1</value>
```

```
            </key>
            <key name="33_5">
                <value type="value">2</value>
            </key>
            <key name="33_6">
                <value type="value">2</value>
            </key>
            <key name="33_7">
                <value type="value">1</value>
            </key>
            <key name="33_8">
                <value type="value">1</value>
            </key>
            <key name="33_9">
                <value type="value">1</value>
            </key>
            <key name="33_10">
                <value type="value">1</value>
            </key>
            <key name="33_11">
                <value type="value">1</value>
            </key>

            <!-- 60- -->
            <key name="60_6">
                <value type="value">2</value>
            </key>
            <key name="60_7">
                <value type="value">1</value>
            </key>
            <key name="60_8">
                <value type="value">1</value>
            </key>
            <key name="60_9">
                <value type="value">1</value>
            </key>
            <key name="60_20">
                <value type="value">1</value>
            </key>
            <key name="60_21">
                <value type="value">1</value>
            </key>
            <key name="60_32">
                <value type="value">1</value>
            </key>
            <key name="60_34">
                <value type="value">1</value>
            </key>
            <key name="60_35">
                <value type="value">1</value>
            </key>
            <key name="60_36">
                <value type="value">1</value>
            </key>

            <!-- 100- -->
            <key name="100_12">
                <value type="value">103</value>
            </key>
```

```
            <key name="100_13">
                <value type="value">31</value>
            </key>
            <key name="100_51">
                <value type="value">-49</value>
            </key>
            <key name="100_108">
                <value type="value">28871</value>
            </key>
            <key name="100_109">
                <value type="value">13781</value>
            </key>
            <key name="100_178">
                <value type="value">119</value>
            </key>
            <key name="100_179">
                <value type="value">219</value>
            </key>
            <key name="100_190">
                <value type="value">1009</value>
            </key>
            <key name="100_191">
                <value type="value">-2</value>
            </key>
            <key name="100_192">
                <value type="value">701</value>
            </key>
            <key name="100_193">
                <value type="value">441</value>
            </key>

            <!-- 101- -->
            <key name="101_63">
                <value type="value">103</value>
            </key>
            <key name="101_205">
                <value type="value">31</value>
            </key>
            <key name="101_100">
                <value type="value">5</value>
            </key>
            <key name="101_101">
                <value type="value">49</value>
            </key>
            <key name="101_102">
                <value type="value">119</value>
            </key>
            <key name="101_105">
                <value type="value">500</value>
            </key>
            <key name="101_106">
                <value type="value">1</value>
            </key>

            <!-- 107- -->
            <key name="107_3">
                <value type="value">16.2</value>
            </key>
            <key name="107_77">
```

```
            <value type="value">15.9</value>
        </key>
        <key name="107_78">
            <value type="value">512.1</value>
        </key>
        <key name="107_79">
            <value type="value">433.4</value>
        </key>
        <key name="107_90">
            <value type="value">344.4</value>
        </key>
        <key name="107_130">
            <value type="value">-0.44013</value>
        </key>
        <key name="107_131">
            <value type="value">43.0</value>
        </key>
        <key name="107_132">
            <value type="value">41.2</value>
        </key>
        <key name="107_141">
            <value type="value">12.1</value>
        </key>
        <key name="107_200">
            <value type="value">91</value>
        </key>
        <key name="107_201">
            <value type="value">98.8</value>
        </key>
        <key name="107_202">
            <value type="value">110</value>
        </key>
        <key name="107_203">
            <value type="value">85.1</value>
        </key>
        <key name="107_204">
            <value type="value">85.2</value>
        </key>
        <key name="107_205">
            <value type="value">410</value>
        </key>
        <key name="107_206">
            <value type="value">592</value>
        </key>
        <key name="107_207">
            <value type="value">1.5</value>
        </key>
        <key name="107_208">
            <value type="value">44.7</value>
        </key>
        <key name="107_209">
            <value type="value">11.9</value>
        </key>
        <key name="107_210">
            <value type="value">221.45</value>
        </key>
        <key name="107_211">
            <value type="value">13.4</value>
        </key>
```

```
            <key name="107_212">
                <value type="value">0.000402</value>
            </key>

            <!-- 109- -->
            <key name="109_3">
                <value type="value">16.2</value>
            </key>
            <key name="109_7">
                <value type="value">15.9</value>
            </key>
            <key name="109_8">
                <value type="value">880</value>
            </key>
            <key name="109_10">
                <value type="value">344.4</value>
            </key>
            <key name="109_40">
                <value type="value">41.2</value>
            </key>
            <key name="109_41">
                <value type="value">12.1</value>
            </key>

            <key name="empty">
                <value type="value">""</value>
            </key>
        </key_value_mappings>
    </databus>
</core>
```

S7-300 template 14

```
<core>
    <template>
        <!-- General information about the template -->
        <entity name="unit">371</entity>
        <entity name="vendor">IPMI</entity>
        <entity name="description">Creates a simple IPMI device</entity>
        <entity name="protocols">IPMI</entity>
        <entity name="creator">Lukas Rist</entity>
    </template>
    <databus>
        <!-- Core value that can be retrieved from the databus by key -->
        <key_value_mappings>
          <key name="SystemName">
              <value type="value">"DoE"</value>
          </key>
        </key_value_mappings>
    </databus>
</core>
```

IPMI template 15

```
<core>
    <template>
        <!-- General information about the template -->
        <entity name="unit">382</entity>
        <entity name="vendor">Kamstrup</entity>
        <entity name="description">Register clone of an existing Kamstrup 382
smart meter</entity>
```

```
        <entity name="protocols">Kamstrup</entity>
        <entity name="creator">Johnny Vestergaard</entity>
    </template>
    <databus>
        <!-- Core value that can be retrieved from the databus by key -->
        <key_value_mappings>
            <key name="power_simulator">
                <value
type="function">conpot.protocols.kamstrup.usage_simulator.UsageSimulator</val
ue>
            </key>
            <key name="register_1024">
                <value type="value">0</value>
            </key>
            <key name="register_1">
                <value type="value">0</value>
            </key>
            <key name="register_2">
                <value type="value">0</value>
            </key>
            <key name="register_13">
                <value type="value">71832712</value>
            </key>
            <key name="register_14">
                <value type="value">0</value>
            </key>
            <key name="register_1054">
                <value type="value">228</value>
            </key>
            <key name="register_1055">
                <value type="value">229</value>
            </key>
            <key name="register_1056">
                <value type="value">224</value>
            </key>
            <key name="register_1076">
                <value type="value">511</value>
            </key>
            <key name="register_1077">
                <value type="value">422</value>
            </key>
            <key name="register_1078">
                <value type="value">144</value>
            </key>
            <key name="register_1080">
                <value type="value">1000</value>
            </key>
            <key name="register_1081">
                <value type="value">5499</value>
            </key>
            <key name="register_1082">
                <value type="value">895</value>
            </key>
            <key name="register_3">
                <value type="value">0</value>
            </key>
            <key name="register_4">
                <value type="value">0</value>
            </key>
```

```
            <key name="register_5">
                <value type="value">0</value>
            </key>
            <key name="register_6">
                <value type="value">0</value>
            </key>
            <key name="register_1025">
                <value type="value">0</value>
            </key>
            <key name="register_1033">
                <value type="value">0</value>
            </key>
            <key name="register_1034">
                <value type="value">0</value>
            </key>
            <key name="register_1035">
                <value type="value">0</value>
            </key>
            <key name="register_1036">
                <value type="value">0</value>
            </key>
            <key name="register_15">
                <value type="value">0</value>
            </key>
            <key name="register_16">
                <value type="value">0</value>
            </key>
            <key name="register_17">
                <value type="value">0</value>
            </key>
            <key name="register_18">
                <value type="value">0</value>
            </key>
            <key name="register_1027">
                <value type="value">0</value>
            </key>
            <key name="register_20">
                <value type="value">0</value>
            </key>
            <key name="register_21">
                <value type="value">0</value>
            </key>
            <key name="register_22">
                <value type="value">0</value>
            </key>
            <key name="register_23">
                <value type="value">0</value>
            </key>
            <key name="register_24">
                <value type="value">0</value>
            </key>
            <key name="register_25">
                <value type="value">0</value>
            </key>
            <key name="register_26">
                <value type="value">0</value>
            </key>
            <key name="register_27">
                <value type="value">0</value>
```

```
            </key>
            <key name="register_28">
                <value type="value">0</value>
            </key>
            <key name="register_29">
                <value type="value">0</value>
            </key>
            <key name="register_30">
                <value type="value">0</value>
            </key>
            <key name="register_31">
                <value type="value">0</value>
            </key>
            <key name="register_32">
                <value type="value">0</value>
            </key>
            <key name="register_33">
                <value type="value">0</value>
            </key>
            <key name="register_34">
                <value type="value">0</value>
            </key>
            <key name="register_35">
                <value type="value">0</value>
            </key>
            <key name="register_36">
                <value type="value">0</value>
            </key>
            <key name="register_37">
                <value type="value">0</value>
            </key>
            <key name="register_38">
                <value type="value">0</value>
            </key>
            <key name="register_39">
                <value type="value">0</value>
            </key>
            <key name="register_40">
                <value type="value">0</value>
            </key>
            <key name="register_41">
                <value type="value">0</value>
            </key>
            <key name="register_42">
                <value type="value">0</value>
            </key>
            <key name="register_43">
                <value type="value">0</value>
            </key>
            <key name="register_44">
                <value type="value">0</value>
            </key>
            <key name="register_45">
                <value type="value">0</value>
            </key>
            <key name="register_46">
                <value type="value">0</value>
            </key>
            <key name="register_1071">
```

```
                <value type="value">0</value>
            </key>
            <key name="register_1072">
                <value type="value">0</value>
            </key>
            <key name="register_1073">
                <value type="value">0</value>
            </key>
            <key name="register_50">
                <value type="value">0</value>
            </key>
            <key name="register_51">
                <value type="value">1258679</value>
            </key>
            <key name="register_52">
                <value type="value">0</value>
            </key>
            <key name="register_53">
                <value type="value">0</value>
            </key>
            <key name="register_54">
                <value type="value">21000002</value>
            </key>
            <key name="register_55">
                <value type="value">22201011</value>
            </key>
            <key name="register_56">
                <value type="value">1000</value>
            </key>
            <key name="register_57">
                <value type="value">0</value>
            </key>
            <key name="register_58">
                <value type="value">0</value>
            </key>
            <key name="register_1083">
                <value type="value">34353</value>
            </key>
            <key name="register_1084">
                <value type="value">256</value>
            </key>
            <key name="register_1086">
                <value type="value">101110</value>
            </key>
            <key name="register_1205">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1092">
                <value type="value">1</value>
            </key>
            <key name="register_1037">
                <value type="value">0</value>
            </key>
            <key name="register_1038">
                <value type="value">0</value>
            </key>
            <key name="register_1112">
                <value type="value">30</value>
```

```
            </key>
            <key name="register_1113">
                <value type="value">30</value>
            </key>
            <key name="register_1114">
                <value type="value">30</value>
            </key>
            <key name="register_1039">
                <value type="value">99000</value>
            </key>
            <key name="register_1121">
                <value type="value">0</value>
            </key>
            <key name="register_1026">
                <value type="value">0</value>
            </key>
            <key name="register_1126">
                <value type="value">382003175115322177893719318328 6</value>
            </key>
            <key name="register_19">
                <value type="value">0</value>
            </key>
            <key name="register_1047">
                <value type="value">9441543881752250126</value>
            </key>
            <key name="register_1049">
                <value type="value">0</value>
            </key>
            <key name="register_1050">
                <value type="value">0</value>
            </key>
            <key name="register_1028">
                <value type="value">0</value>
            </key>
            <key name="register_1051">
                <value type="value">0</value>
            </key>
            <key name="register_1189">
                <value type="value">0</value>
            </key>
            <key name="register_1202">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1203">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1204">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1206">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1207">
                <value
type="value">340282366920938463463374607431768211455</value>
```

```
            </key>
            <key name="register_1208">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1209">
                <value
type="value">340282366920938463463374607431768211455</value>
            </key>
            <key name="register_1029">
                <value type="value">100</value>
            </key>
            <key name="register_1058">
                <value type="value">2276916355582011806331 39</value>
            </key>
            <key name="register_1115">
                <value type="value">60</value>
            </key>
            <key name="register_1059">
                <value type="value">0</value>
            </key>
            <key name="register_1060">
                <value type="value">0</value>
            </key>
            <key name="register_1030">
                <value type="value">46828625</value>
            </key>
            <key name="register_1061">
                <value type="value">0</value>
            </key>
            <key name="register_1062">
                <value type="value">0</value>
            </key>
            <key name="register_1063">
                <value type="value">0</value>
            </key>
            <key name="register_1064">
                <value type="value">0</value>
            </key>
            <key name="register_1065">
                <value type="value">0</value>
            </key>
            <key name="register_1031">
                <value type="value">0</value>
            </key>
            <key name="register_1066">
                <value type="value">0</value>
            </key>
            <key name="register_1067">
                <value type="value">0</value>
            </key>
            <key name="register_1068">
                <value type="value">0</value>
            </key>
            <key name="register_1069">
                <value type="value">0</value>
            </key>
            <key name="register_1070">
                <value type="value">0</value>
```

```
            </key>
            <key name="register_1074">
                <value type="value">0</value>
            </key>
            <key name="register_1075">
                <value type="value">0</value>
            </key>
            <key name="register_1079">
                <value type="value">315</value>
            </key>
            <key name="register_1181">
                <value type="value">433534329705531658</value>
            </key>
            <key name="register_1001">
                <value type="value">15085488</value>
            </key>
            <key name="register_1002">
                <value type="value">203513</value>
            </key>
            <key name="register_1003">
                <value type="value">140727</value>
            </key>
            <key name="register_1004">
                <value type="value">283</value>
            </key>
            <key name="register_1005">
                <value type="value">53011401</value>
            </key>
            <key name="register_1010">
                <value type="value">15085488</value>
            </key>
            <key name="register_1021">
                <value type="value">0</value>
            </key>
            <key name="register_1023">
                <value type="value">0</value>
            </key>
            <key name="reboot_signal">
                <value type="value">0</value>
            </key>
            <key name="software_version">
                <value type="value">'5.5 (E5)'</value>
            </key>
            <key name="access_control_status">
                <value type="value">'DISABLED'</value>
            </key>
            <key name="access_control_1">
                <value type="value">'0.0.0.0'</value>
            </key>
            <key name="access_control_2">
                <value type="value">'0.0.0.0'</value>
            </key>
            <key name="access_control_3">
                <value type="value">'0.0.0.0'</value>
            </key>
            <key name="access_control_4">
                <value type="value">'0.0.0.0'</value>
            </key>
            <key name="access_control_5">
```

```
            <value type="value">'0.0.0.0'</value>
        </key>
        <key name="device_name">
            <value type="value">''</value>
        </key>
        <key name="nameserver_1">
            <value type="value">'0.0.0.0'</value>
        </key>
        <key name="nameserver_2">
            <value type="value">'0.0.0.0'</value>
        </key>
        <key name="nameserver_3">
            <value type="value">'0.0.0.0'</value>
        </key>
        <key name="mac_address">
            <value type="value">'00:13:EA:00:00:00'</value>
        </key>
        <key name="use_dhcp">
            <value type="value">'YES'</value>
        </key>
        <key name="ip_addr">
            <value type="value">'192.168.1.210'</value>
        </key>
        <key name="ip_gateway">
            <value type="value">'192.168.1.1'</value>
        </key>
        <key name="ip_subnet">
            <value type="value">'255.255.255.0'</value>
        </key>
        <key name="ip_addr_dhcp">
            <value type="value">'192.168.0.1'</value>
        </key>
        <key name="ip_gateway_dhcp">
            <value type="value">'192.168.0.254'</value>
        </key>
        <key name="ip_subnet_dhcp">
            <value type="value">'255.255.255.0'</value>
        </key>
        <key name="kap_a_server_hostname">
            <value type="value">'pwr_ctrl_mgmt01.int.local'</value>
        </key>
        <key name="kap_a_server_ip">
            <value type="value">'10.232.15.242'</value>
        </key>
        <key name="kap_a_server_port">
            <value type="value">'50'</value>
        </key>
        <key name="kap_b_server_ip">
            <value type="value">'0.0.0.0'</value>
        </key>
        <key name="kap_b_server_port">
            <value type="value">'50'</value>
        </key>
        <key name="channel_a_meternumber">
            <value type="value">'A1 06 A1 02 B7 34 12 00 00 03'</value>
        </key>
        <key name="channel_b_meternumber">
            <value type="value">'A1 06 A1 02 B7 34 12 00 00 03'</value>
        </key>
```

```
            <key name="channel_a_port">
                <value type="value">'1025'</value>
            </key>
            <key name="channel_b_port">
                <value type="value">'1027'</value>
            </key>
            <key name="kap_ack_server">
                <value type="value">'NO'</value>
            </key>
            <key name="kap_local_port">
                <value type="value">'800'</value>
            </key>
            <key name="alarm_server_status">
                <value type="value">'DISABLED'</value>
            </key>
            <key name="alarm_server_ip">
                <value type="value">''</value>
            </key>
            <key name="alarm_server_port">
                <value type="value">'4000'</value>
            </key>
            <key name="kap_server_lookup">
                <value type="value">'0 - none'</value>
            </key>
            <key name="software_watchdog">
                <value type="value">'3600'</value>
            </key>
            <key name="kap_missing_warning">
                <value type="value">'60'</value>
            </key>
            <key name="keep_alive_timer">
                <value type="value">'10'</value>
            </key>
            <key name="serial_settings_a">
                <value type="value">'Auto'</value>
            </key>
            <key name="serial_settings_b">
                <value type="value">'115200,8,E,1'</value>
            </key>
            <key name="channel_a_connect_socket">
                <value type="value">'0 - None'</value>
            </key>
            <key name="channel_b_connect_socket">
                <value type="value">'0 - None'</value>
            </key>
        </key_value_mappings>
    </databus>
</core>
```

Kamstrup 384 template 16

9.1.2 imágenes Docker ELK

```
FROM alpine

# add config files and templates
ADD config/ /root/config/
```

```
# install apps
RUN sed -i 's/dl-cdn/dl-2/g' /etc/apk/repositories && \
    apk -U --no-cache add openjdk8-jre aria2 curl git \
              libzmq nss bash libc6-compat

# Install and config
RUN git clone --depth=1 https://github.com/dtag-dev-sec/listbot /etc/listbot
&& \
    cd /root/config/ && \
    mkdir -p /usr/share/logstash/ && \
    aria2c -s 16 -x 16
https://artifacts.elastic.co/downloads/logstash/logstash-6.7.0.tar.gz && \
    tar xvfz logstash-6.7.0.tar.gz --strip-components=1 -C
/usr/share/logstash/ && \
    /usr/share/logstash/bin/logstash-plugin install logstash-filter-
translate && \
    /usr/share/logstash/bin/logstash-plugin install logstash-output-syslog
&& \
    aria2c -s 16 -x 16 -o GeoLite2-ASN.tar.gz
http://geolite.maxmind.com/download/geoip/database/GeoLite2-ASN.tar.gz && \
    tar xvfz GeoLite2-ASN.tar.gz --strip-components=1 -C
/usr/share/logstash/vendor/bundle/jruby/2.3.0/gems/logstash-filter-geoip-
5.0.3-java/vendor && \
    cd /root/config/ && \
    cp update.sh /usr/bin/ && \
    chmod u+x /usr/bin/update.sh && \
    mkdir -p /etc/logstash/conf.d && \
    cp logstash.conf /etc/logstash/conf.d/ && \
    cp elasticsearch-template-es6x.json
/usr/share/logstash/vendor/bundle/jruby/2.3.0/gems/logstash-output-
elasticsearch-9.3.2-java/lib/logstash/outputs/elasticsearch/ && \

# Get user, groups, configs
    addgroup -g 5000 logstash && \
    adduser -S -H -s /bin/bash -u 5000 -D -g 5000 logstash && \
    chown -R logstash:logstash /usr/share/logstash && \
    chown -R logstash:logstash /etc/listbot && \
    chmod 755 /usr/bin/update.sh && \

# Remove temporal files
    rm -rf /root/* && \
    rm -rf /tmp/* && \
    rm -rf /var/cache/apk/*

# check status
HEALTHCHECK --retries=20 CMD curl -s -XGET 'http://127.0.0.1:9600'

# Start logstash
#USER logstash:logstash
CMD update.sh && exec /usr/share/logstash/bin/logstash -f
/etc/logstash/conf.d/logstash.conf --config.reload.automatic --java-
execution
```

Logstash Dockerfile 17

```
# Input section
input {

# Conpot
```

```
  file {
    path => ["/data/conpot/log/*.json"]
    codec => json
    type => "ConPot"
  }
}

# Filter Section
filter {

# Conpot
  if [type] == "ConPot" {
    date {
      match => [ "timestamp", "ISO8601" ]
    }
    mutate {
      rename => {
        "dst_port" => "dest_port"
        "dst_ip" => "dest_ip"
      }
    }
  }

# Drop if parse fails
if "_grokparsefailure" in [tags] { drop {} }

# Add geo coordinates / ASN info / IP rep.
  if [src_ip]  {
    geoip {
      cache_size => 10000
      source => "src_ip"
      database =>
"/usr/share/logstash/vendor/bundle/jruby/2.3.0/gems/logstash-filter-geoip-
5.0.3-java/vendor/GeoLite2-City.mmdb"
    }
    geoip {
      cache_size => 10000
      source => "src_ip"
      database =>
"/usr/share/logstash/vendor/bundle/jruby/2.3.0/gems/logstash-filter-geoip-
5.0.3-java/vendor/GeoLite2-ASN.mmdb"
    }
    translate {
      refresh_interval => 86400
      field => "src_ip"
      destination => "ip_rep"
      dictionary_path => "/etc/listbot/iprep.yaml"
    }
  }

# In some rare conditions dest_port, src_port, status are indexed as string,
forcing integer for now
  if [dest_port] {
    mutate {
        convert => { "dest_port" => "integer" }
    }
  }
  if [src_port] {
    mutate {
```

```
        convert => { "src_port" => "integer" }
    }
  }
  if [status] {
    mutate {
        convert => { "status" => "integer" }
    }
  }

}

# Output section
output {
  elasticsearch {
    hosts => ["elasticsearch:9200"]
  }
}
```

Logstash config 18

```
FROM alpine

# Add config files and templates
ADD config/ /root/config/

# Install apps
RUN sed -i 's/dl-cdn/dl-2/g' /etc/apk/repositories && \
    apk -U --no-cache add openjdk8-jre bash aria2 curl nss

# Get and install packages
RUN cd /root/config/ && \
    mkdir -p /usr/share/elasticsearch/ && \
    aria2c -s 16 -x 16
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-
6.7.0.tar.gz && \
    tar xvfz elasticsearch-6.7.0.tar.gz --strip-components=1 -C
/usr/share/elasticsearch/ && \
    cd /root/config/ && \
    mkdir -p /usr/share/elasticsearch/config && \
    cp elasticsearch.yml /usr/share/elasticsearch/config/ && \

# Get user, groups, configs
    addgroup -g 5000 elasticsearch && \
    adduser -S -H -s /bin/ash -u 5000 -D -g 5000 elasticsearch && \
    chown -R elasticsearch:elasticsearch /usr/share/elasticsearch/ && \
    rm -rf /usr/share/elasticsearch/modules/x-pack-ml && \

# Remove temporal files
    rm -rf /root/* && \
    rm -rf /tmp/* && \
    rm -rf /var/cache/apk/*

# check status
HEALTHCHECK --retries=20 CMD curl -s -XGET
'http://127.0.0.1:9200/_cat/health'

# Start ELK
USER elasticsearch:elasticsearch
CMD ["/usr/share/elasticsearch/bin/elasticsearch"]
```

Elasticsearch Dockerfile 19

```
cluster.name: ssorocluster
node.name: "ssorocluster-node-01"
xpack.ml.enabled: false
path:
    logs: /data/elk/log
    data: /data/elk/data
http.host: 0.0.0.0
http.cors.enabled: true
http.cors.allow-origin: "*"
```

Elasticsearch cluser config 20

```
FROM node:10.15.2-alpine

# Add config files and templates
ADD config/ /root/config/

# Setup env and apt
RUN sed -i 's/dl-cdn/dl-2/g' /etc/apk/repositories && \
    apk -U --no-cache add curl aria2

# Get and install packages
RUN cd /root/config/ && \
    mkdir -p /usr/share/kibana/ && \
    aria2c -s 16 -x 16 https://artifacts.elastic.co/downloads/kibana/kibana-
6.7.0-linux-x86_64.tar.gz && \
    tar xvfz kibana-6.7.0-linux-x86_64.tar.gz --strip-components=1 -C
/usr/share/kibana/ && \

# Setup user, groups and configs
    cd /root/config/ && \
    sed -i 's/#server.basePath: ""/server.basePath: "\/kibana"/'
/usr/share/kibana/config/kibana.yml && \
    sed -i 's/#kibana.defaultAppId: "home"/kibana.defaultAppId:
"dashboards"/' /usr/share/kibana/config/kibana.yml && \
    sed -i 's/#server.host: "localhost"/server.host: "0.0.0.0"/'
/usr/share/kibana/config/kibana.yml && \
    sed -i 's/#elasticsearch.hosts:
\["http:\/\/localhost:9200"\]/elasticsearch.hosts:
\["http:\/\/elasticsearch:9200"\]/' /usr/share/kibana/config/kibana.yml && \
    sed -i 's/#server.rewriteBasePath: false/server.rewriteBasePath: false/'
/usr/share/kibana/config/kibana.yml && \
    sed -i "s/#005571/#e20074/g"
/usr/share/kibana/src/legacy/core_plugins/kibana/public/index.css && \
    sed -i "s/#007ba4/#9e0051/g"
/usr/share/kibana/src/legacy/core_plugins/kibana/public/index.css && \
    sed -i "s/#00465d/#4f0028/g"
/usr/share/kibana/src/legacy/core_plugins/kibana/public/index.css && \
    echo "xpack.infra.enabled: false" >> /usr/share/kibana/config/kibana.yml
&& \
    echo "xpack.logstash.enabled: false" >>
/usr/share/kibana/config/kibana.yml && \
    echo "xpack.canvas.enabled: false" >>
/usr/share/kibana/config/kibana.yml && \
    echo "xpack.spaces.enabled: false" >>
/usr/share/kibana/config/kibana.yml && \
    echo "xpack.apm.enabled: false" >> /usr/share/kibana/config/kibana.yml
&& \
    rm /usr/share/kibana/node/bin/node && \
    ln -s /usr/bin/node /usr/share/kibana/node/bin/node && \
```

```
    rm -rf /usr/share/kibana/optimize/bundles/* && \
    /usr/share/kibana/bin/kibana --optimize && \
    addgroup -g 5000 kibana && \
    adduser -S -H -s /bin/ash -u 5000 -D -g 5000 kibana && \
    chown -R kibana:kibana /usr/share/kibana/ && \

# Remove temporal files
    rm -rf /root/* && \
    rm -rf /tmp/* && \
    rm -rf /var/cache/apk/*

# Check status
HEALTHCHECK --retries=20 CMD curl -s -XGET 'http://127.0.0.1:5601'

# Start kibana
STOPSIGNAL SIGKILL
USER kibana:kibana
CMD ["/usr/share/kibana/bin/kibana"]
```

Kibana Dockerfile 21

```
services:

# Logstash service
  logstash:
    build: logstash/.
    container_name: logstash
    image: "ssoro/logstash"
    restart: always
    depends_on:
      elasticsearch:
        condition: service_healthy
    env_file:
     - /opt/etc/compose/elk_environment
    volumes:
     - /data:/data
     - /root/elk/logstash/logstash.conf:/etc/logstash/conf.d/logstash.conf

# Elasticsearch service
  elasticsearch:
    build: elasticsearch/.
    container_name: elasticsearch
    image: "ssoro/elasticsearch"
    restart: always
    ports:
     - "127.0.0.1:9201:9200"
    environment:
     - bootstrap.memory_lock=true
     - ES_JAVA_OPTS=-Xms1024m -Xmx1024m
     - ES_TMPDIR=/tmp
    cap_add:
     - IPC_LOCK
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    mem_limit: 4g
```

```
    volumes:
     - /data:/data

# Kibana service
  kibana:
    build: kibana/.
    container_name: kibana
    image: "ssoro/kibana"
    restart: always
    stop_signal: SIGKILL
    depends_on:
      elasticsearch:
        condition: service_healthy
    ports:
     - "127.0.0.1:5602:5601"
```

ELK Docker-compose 22