Universidad del País Vasco | Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

Trabajo de Fin de Master

Diseño, despliegue e Inteligencia de una herramienta SIEM

AUTOR: UNAI ABRISQUETA SANCHEZ
TUTOR: JUAN JOSÉ UNZILLA GALÁN

CURSO: 2020-2021

FECHA: BILBAO, 21 DE SEPTIEMBRE DE 2021

RESUMEN

El objetivo de este proyecto es implementar herramientas robustas de monitorización de la seguridad de la red. Además, se dotará de inteligencia a estas herramientas mediante módulos complementarios. La razón de ser de este proyecto, nace de la importancia que ha cobrado la ciberseguridad en la sociedad en los últimos tiempos, este crecimiento se debe al crecimiento de los ciberataques en el entorno empresarial. En consecuencia, cada vez es más necesario proteger las infraestructuras corporativas de las empresas y tener un control estricto de la seguridad de la red. Para completar el objetivo, se analizarán diversas alternativas con el fin de encontrar la que más se ajuste al proyecto. Una vez seleccionada la solución, basándose en los requerimientos del proyecto, se diseñará la solución final. Para ello se diseñará e implementará una solución centralizada de monitorización de la seguridad de la red, dotándola de un módulo que aporte inteligencia y aprendizaje automático a la solución. Posteriormente, se evaluará y validará la solución.

Palabras claves: Ciberseguridad, SIEM, log, trafico, red, Machine Learning, inteligencia artificial

LABURPENA

Proiektu honen helburua, sarearen monitorizazioko tresna sendoak inplementatzean datza. Gainera, tresna hauei adimena emango zaie modulu osagarrien bidez. Proiektu honen arrazoia, azken garaietan, eremu industrialean eraso informatikoen hazkundeen ondorioz, zibersegurtasun arloaren garrantziaren hazkundean oinarritzen da. Ondorioz, azpiegitura korporatiboak babestea eta sarearen kontrol zorrotza edukitzea gero eta beharrezkoagoa da. Helburua betetzeko, aukera ezberdinak aztertuko dira, proiektuari hobeto moldatuko dena aukeratzeko helburuarekin. Behin ebazpena aukeratuta, espezifikazioetan oinarrituta, ebazpenaren diseinua egingo da. Horretarako, sarearen monitorizazio zentralizatua diseinatu eta inplementatuko da, halaber, adimena eta ikasketa automatikoa gehituko dion modulu bat gehituz. Ondoren, ebazpena aztertu eta balidatuko da.

Hitz gakoak: Zibersegurtasuna, SIEM, log, trafikoa, sarea, Machine Learning, adimen artifiziala

ABSTRACT

The aim of this project is to implement robust network security monitoring tools. In addition, these tools will be provided with intelligence by means of complementary modules. The raison of this project stems from the importance that cybersecurity has gained in society in recent times due to the growth of cyber-attacks in the business environment. As a result, it is increasingly necessary to protect corporate infrastructures of companies and to have a strict control of network security. To complete the objective, various alternatives will be analyzed in order to find the one that best fits the project. Once the solution has been selected, based on the requirements of the project, the final solution will be designed. To this end, a centralized network security monitoring solution will be designed and implemented, providing it with a module that brings intelligence and automatic learning to the solution. Subsequently, the solution will be evaluated and validated.

Keywords: Cybersecurity, SIEM, log, traffic, network, Machine Learning, artificial intelligence

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ACRÓNIMOS

AAA	**ACCOUNTING, AUTHENTICATION AND AUTHORIZATION**
CLI	**COMMAND LINE INTERFACE**
CRE	**CUSTOM RULE ENGINE**
CVE	**COMMON VULNERABILITIES AND EXPOSURES**
HTTP	**HYPERTEXT TRANSFER PROTOCOL**
HTTPS	**HYPERTEXT TRANSFER PROTOCOL SECURE**
IKE	**INTERNET KEY EXCHANGE**
IoT	**INTERNET OF THINGS**
IP	**INTERNET PROTOCOL**
IT	**INFORMATION TECHNOLOGY**
LEEF	**LOG EVENT EXTENDED FORMAT**
ML	**MACHINE LEARNING**
PAP	**PASO A PRODUCCIÓN**
SCP	**SECURE COPY PROTOCOL**
SIEM	**SECURITY INFORMATION AND EVENT MANAGEMENT**
SOAR	**SECURITY ORCHESTATION, AUTOMATION AND RESPONSE**
URL	**UNIFORM RESOURCE LOCATOR**
VPN	**VIRTUAL PRIVATE NETWORK**

1 INTRODUCCIÓN

Desde que *Internet* vio la luz, su evolución ha sido exponencial y se ha convertido en un elemento fundamental e indispensable en nuestras vidas. Lejos queda esa herramienta diseñada prácticamente con el motivo de uso militar en la década de los *80*. El uso de *Internet* ha ido creciendo tanto en el entorno doméstico como en el entorno empresarial, como la **figura 1** muestra:

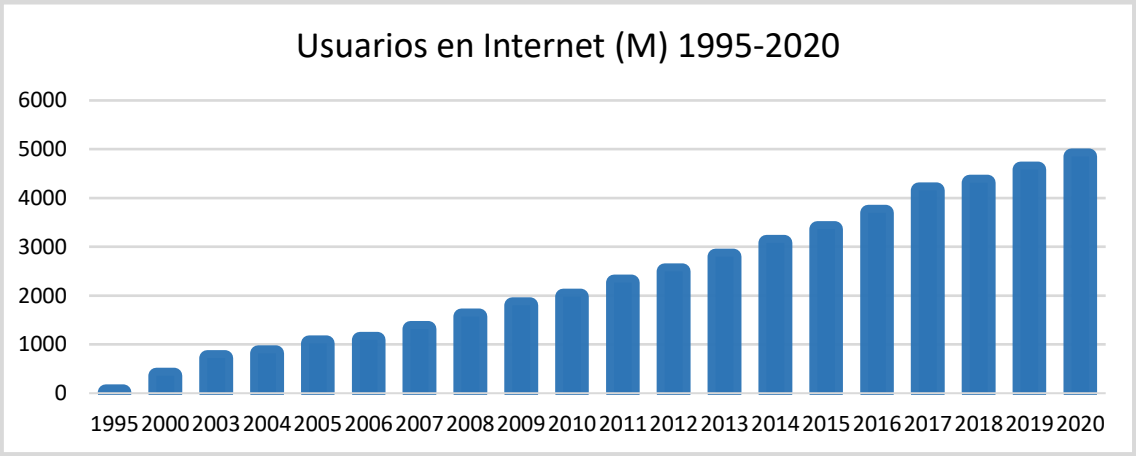


Figura 1: Crecimiento usuarios de internet 1995-2020[1]

La tendencia que sigue el grafico anterior no hará más que crecer con las tecnologías de vanguardia que se están desarrollando para los próximos 5 años; entre otras, IoT (*Internet of Things*), *Smart Cities* y ML (*Machine Learning*). Estas nuevas tecnologías harán que el crecimiento del número de usuarios de *Internet* continúe con una tendencia ampliamente ascendente.

De hecho, ha llegado hasta tal punto que un simple corte de 30 minutos de red puede costar una cantidad ingente de dinero a una empresa, aunque no sea tecnológica. En este ecosistema, afloran varios riesgos que antes se desconocían, los ladrones ya no poseen armas si no un portátil y acceso a *Internet*. En el sector de Industria 4.0 aún se acrecentará más la perdida que supone un corte de servicio, puesto que el número de dispositivos inteligentes será mayor.

Por ello, hoy en día, la seguridad es un bien indispensable para las empresas. El riesgo de delito cibernético cada vez es más alto y las empresas no quieren ver manchado su prestigio ni tener pérdidas económicas con un incidente de este tipo. Para ello, el desarrollo de sistemas informáticos que proporcionen protección, es cada vez más potente.

Continuando con este último punto, actualmente, se dispone de un gran *set* de sistemas informáticos que se dedican meramente a la protección de la red y sus respectivos datos. Esta protección se divide en dos rangos: activa y pasiva. Hay mecanismos que la protegen

activamente y otros, sin embargo, se centran más en la detección de comportamientos sospechosos.

En este proyecto se tratará de buscar la securización de una red corporativa empresarial mediante el despliegue de una herramienta SIEM (*Security Information and Event Management*). Esta herramienta permitirá gestionar los incidentes de seguridad de la red corporativa basándose en los *logs* de los sistemas informáticos.

Además, también cabe hacer referencia al creciente uso de conceptos relacionados con la inteligencia artificial, como redes neuronales o *machine learning*. Estos términos o avances tecnológicos, permitirán crecer sustancialmente a la ciencia y la seguridad no será una excepción. Por lo que, en este proyecto, también se realizará un análisis de *machine learning* enfocado en el ámbito de aplicación de una herramienta SIEM.

2 CONTEXTO

Como bien se ha detallado en la introducción, el mundo IT (*Information Technology*) está en constante crecimiento y esto genera un crecimiento de los activos a proteger por la seguridad informática de una empresa.

Este ecosistema, ofrece un entorno muy amplio para el *hacker*, puesto que cada vez tiene más sistemas que poder explotar. En este punto, puede surgir una duda, ¿Por qué el mayor número de equipos informáticos aumenta el riesgo? Sencillo, no existe el sistema perfecto. Las vulnerabilidades existen y las tienen prácticamente todos los sistemas informáticos.

Además, cuando se descubre una vulnerabilidad nueva en un sistema de uso masivo, esto se expande como la pólvora. Los fabricantes suelen tender a sacar un parche o actualización que pueda solucionarlo, por ello es muy importante mantener siempre actualizado el *software* que usemos. Para conocer las vulnerabilidades pertenecientes a un sistema se usa la CVE (*Common Vulnerabilities and Exposures*), una lista que detalla las vulnerabilidades conocidas.

Pero, ¿el riesgo es solo culpa de los sistemas informáticos? Error, el riesgo es en gran parte por una mala praxis del personal de la empresa. El fallo, normalmente, proviene del factor humano. La seguridad de los sistemas informáticos, sobretodo el control de acceso y la autorización, están construidos sobre una base errónea, probablemente el mayor inconveniente de la seguridad es el no tener en cuenta el factor o esencia del ser humano a la hora de diseñarlos.

Por ello una de las principales bases de los ataques actuales es la ingeniería social. La ingeniería social se basa en obtener información sensible manipulando a usuarios legítimos. Para ello, se tiene en cuenta la poca educación en aspectos de seguridad informática de los usuarios legítimos, su poca predisposición a realizar una praxis correcta de los mecanismos de autorización (contraseñas) y su entorno social o de interés. Combinando estos factores, se puede acceder a un sistema informático sin ningún tipo de *exploit* o programa malicioso, tan solo mediante la persuasión o conocimiento del sujeto.

En base a lo detallado en el párrafo anterior, cabe recalcar que por lo tanto la protección del usuario legítimo y la limitación de sus accesos es esencial. En muchas ocasiones se da por sentado que el usuario legitimo no va a cometer ninguna acción maliciosa, por lo que se pasan por alto varias situaciones como apropiación de credenciales del usuario, protección del sistema frente a un usuario legitimo o caracterización del comportamiento malicioso aplicado al propio usuario legítimo.

2.1 MECANISMOS DE SEGURIDAD

En este ecosistema de seguridad, han surgido mecanismos de seguridad que buscan reforzar estas situaciones o amenazas. Algunos más novedosos o en periodo de prueba y otros más establecidos, entre ellos se encuentran los siguientes:

- NGFW (*Next Generation Firewall*)
- SIEM (*Security Information and Events Management*)
- UEBA (*User and Entity Behaviour Analytics*)
- NAC (*Network Access Control*)
- DLP (*Data Loss Prevention*)
- IPS (*intrusion Prevention System*)
- Cifrado de datos

A continuación, se dividirán distintos subapartados en los que se detallarán principalmente los mecanismos nombrados con anterioridad.

2.1.1 NGFW

Este mecanismo o herramienta hace referencia a los *firewalls* de hoy en día; es decir, los *firewalls* de nueva generación. Estas herramientas o equipos son capaces de realizar más funciones que el mero filtrado de paquetes. Actualmente cumplen las siguientes funciones:

- **Identificación de usuario:** Proporciona servicios AAA (*Authentication, Authorization and Accounting*), por lo cual pueden identificar a usuarios como origen de una comunicación sin tener que determinar su dirección IP. De esta manera se granularizarán los accesos a un usuario, muy útil en entornos con dirección IP (*Internet Protocol*) cambiante, uso de VPNs…
- **Control de aplicaciones:** Esta funcionalidad permite filtrar paquetes en base a la aplicación a la que pertenezcan estos paquetes. Esto proporciona mayor precisión en el filtrado de paquetes.
- **URL Filtering:** Esta característica permite filtrar paquetes en base a URLs (*Uniform Resource Locator*), por lo que como la anterior funcionalidad aporta mayor precisión de filtrado de paquetes.
- **Anti-Virus:** Esta protección permite realizar labores simples de detección de virus a un firewall.
- **Anti-Bot:** Esta operatividad permite comprobar las conexiones a internet en busca de referencias a redes de *Command&Control*, redes que controlan el *malware* remotamente.
- **IPS:** Esta función permite prevenir las intrusiones a nuestro sistema, utilizando un sistema de base de datos de firmas relacionadas con el comportamiento de ataques o *exploits* conocidos.

- **Sandboxing:** Este servicio permite emular la ejecución de un fichero o programa sospechoso en un entorno virtualizado, pudiendo comprobar el efecto adverso que puede generar el mismo.

Por ende, los *firewalls* han dejado de ser un equipo de una única funcionalidad y han adquirido muchas funciones, con el fin de centralizar los mecanismos de seguridad en un único equipo.

2.1.2 SIEM

Este mecanismo nace de las tecnologías SIM (*Security Information Management*) y SEM (*Security Event Management*); es decir, las tecnologías encargadas de la recolección centralizada de *logs* y el análisis de la explotación de la información.

Ofrece el siguiente *set* de funcionalidades:

- Centralización de los eventos de seguridad
- Detección y notificación momentánea de incidentes de seguridad
- Trazabilidad de la actividad
- Ayuda al cumplimiento de la normativa de seguridad vigente
- Concienciación de la organización

Este proyecto se centrará sobre todo en este mecanismo, por lo que posteriormente, se verá en mayor detalle.

2.1.3 UEBA

Este equipamiento estudia el comportamiento habitual de los equipos conectados a una red mediante algoritmia basada en ML, de esta manera comprueba cuál es el funcionamiento normal de una red y detecta que comportamiento anómalo se da en la misma.

Cabe destacar qué es un mecanismo en fase de desarrollo y que, solo es considerado oportuno para una red que tenga una madurez considerable. En el futuro, se prevé que se convertirá en esencial pues pone la vista sobre todo a los problemas dentro de la red que se han dejado de lado.

Está relacionado con el SIEM visto con anterioridad y pretende mejorar su funcionamiento puesto que el SIEM no mejora con el tiempo y no aprende de sus falsos positivos, errores de detección…

2.1.4 NAC

Es una tecnología que permite controlar el acceso de los dispositivos a la red. Proporciona la capacidad para clasificar los dispositivos y darles privilegios concretos o denegarles el acceso. Este control se implementa en nivel de enlace, por lo que proporciona una detección en tiempo real y un control de todos los elementos que acceden a la red.

2.1.5 DLP

Este equipamiento proporciona seguridad ante la pérdida de datos o el envío fraudulento de la misma fuera de la red corporativa. Por lo tanto, es un mecanismo que pretende evitar la fuga de datos. Es un mecanismo muy sensible y solo se recomienda implementar en sistemas con un claro y amplio conocimiento de la infraestructura de red y de seguridad.

2.1.6 IPS

Esta funcionalidad permite la prevención de ataques y ejerce el control de acceso de la red. Últimamente, se incorpora como una funcionalidad más de los NGFW. Los hay de distintos tipos:

- NIPS: Basados en red, buscan tráfico de red sospechoso.
- WIPS: Basados en Wireless, buscan en la red inalámbrica tráfico sospechoso.
- NBA: Basados en el comportamiento de la red, examinan el tráfico inusual como ciertas formas de malware, ataques de denegación de servicios o violaciones de las políticas de seguridad.
- HIPS: Buscan actividades sospechosas en host únicos.

2.1.7 Cifrado de datos

Este método permite proteger la confidencialidad de los datos de una red mediante el uso de algoritmos de cifrado y claves de cifrado. En una época en la que la información se ha convertido tan importante se erige como un método establecido y en uso.

2.2 MEDIDAS DE SEGURIDAD

Las medidas de seguridad son las directivas que se aplican en un entorno para protegerlo. En este caso nos centraremos en el entorno informático, un entorno sensible y vulnerable sin este tipo de directivas.

En muchas ocasiones, se comete el error de pensar que las medidas se dividen en dos subgrupos, las físicas y las tecnológicas. En muchas ocasiones es más importante el enfoque de la psicología, concienciación y formación en materias de seguridad, que tener el cortafuego más potente del mercado.

Por lo cual, en este documento no se obviará la importancia de este tipo de directivas y se remarcaran, dándoles la importancia que poseen. En consecuencia, se dividirán las distintas medidas de seguridad en físicas y no físicas. Las medidas físicas serán aquellas que puedan ser materializadas en una instancia física o virtual que proporciones protección a un sistema. Las medidas no físicas, serán aquellas medidas que no tengan una base que soporte la seguridad.

Entre las medidas de seguridad físicas, se encuentran las siguientes:

- Sistemas de protección física
- Sistemas de protección ante corte eléctricos
- Sistemas de protección de red
- Sistemas de autenticación robustos

Entre las medidas de seguridad no físicas, se encuentran las siguientes:

- Concienciación en materia de ciberseguridad
- Formación informática y en ciberseguridad
- Política de ciberseguridad de la empresa

El primer grupo, son medidas de seguridad que se basan en equipamiento de seguridad y que previenen distintas amenazas que puede sufrir un sistema informático, desde problemas eléctricos a protección frente a *software* malicioso.

El segundo grupo, es más importante quizás que el primero, puesto que trata de que no haya errores en los usuarios legítimos de la red, con los que las anteriores medidas de seguridad suelen ser menos restrictivas. Muchas veces, pese a tener una gran protección, un mero *click* en un correo electrónico puede provocar un gran problema en una red.

Toda esta serie de medidas en el ámbito de la seguridad, se recogen en las certificaciones de ISO 27001 y 27002. De esta manera, el organismo de estandarización ISO busca promover las medidas de seguridad en los entornos de negocio para de esta manera, certificar el buen hacer de una empresa en este ámbito. Esta certificación ha adquirido gran importancia pues se exige de manera contractual en múltiples ocasiones a la hora de colaborar en proyectos empresariales.

Resumiendo, las medidas de seguridad son de vital importancia en el ecosistema empresarial actual ampliamente ligado con la informática.

2.3 AMENAZAS DE SEGURIDAD

Como en el anterior apartado se ha hecho mención de las medidas de seguridad de una manera más teórica, en este apartado se detallarán los ataques cibernéticos actuales. Una amplia comprensión de la problemática actual puede ser de gran ayuda a la hora de desplegar una herramienta de seguridad.

De esta manera, se conocerá ante que debe proteger la herramienta y que tipo de activo debe proteger. En este apartado, en concreto, solo se analizarán los ataques más frecuentes y se detallarán para comprender mejor el *modus operandi* del mismo. Para comprender su funcionamiento, se incluirá la **figura 2** que muestra los pasos que puede tener un ataque:

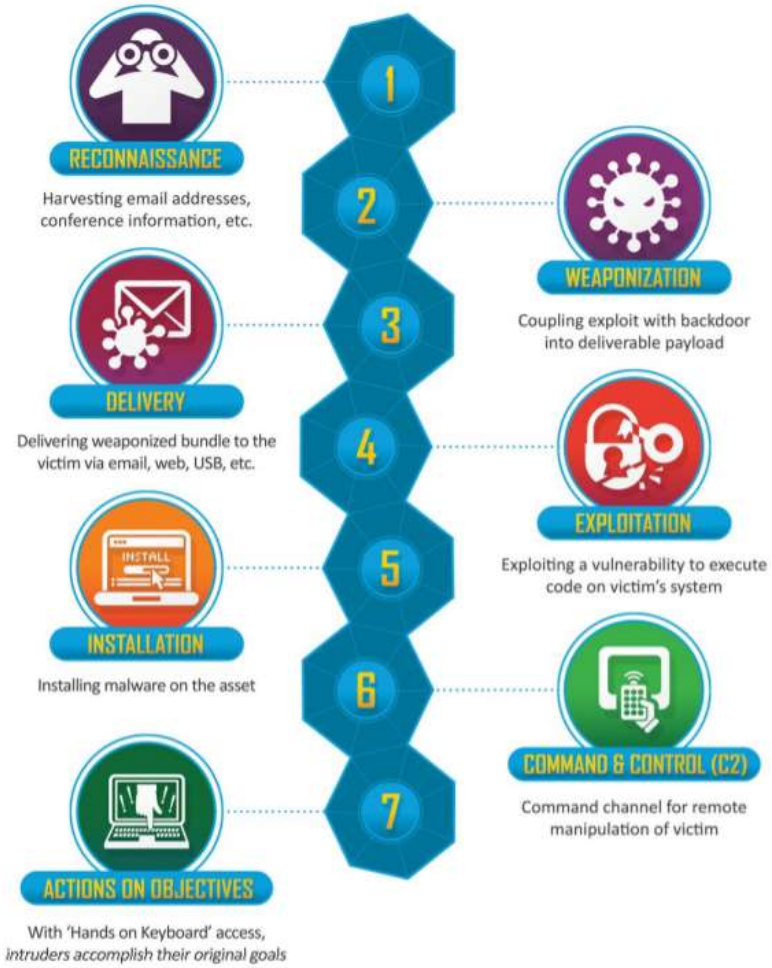


Figura 2: Proceso o pasos de un ataque[2]

Fuente: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

A continuación, se explicará cada paso en más detalle:

- ***Reconnaisance*:** Este paso se basa en lograr información del objetivo de ataque, se consiguen su *email*, nombres y apellidos, información personal… A través de esta información se puede sensibilizar o facilitar una intrusión a su sistema. Este paso está muy relacionado con la ingeniería social mencionada con anterioridad.
- ***Weaponization*:** Este paso trata de conseguir un *payload* o carga maliciosa que pueda interactuar maliciosamente con el equipo infectado.
- ***Delivery*:** Este paso consiste en hacer llegar la carga maliciosa preparada en el paso anterior al atacado, para ello se usa la información lograda en el primer paso.
- ***Exploiting:*** Después de hacer llegar la carga maliciosa al equipo atacado, se tratará de ejecutar código, manipular o explotar su máquina o sistema.
- ***Installation*:** Como parte de la explotación, se puede buscar instalar *malware* o *software* malicioso adicional que pueda generar más vulnerabilidades en el sistema infectado.
- ***Command and Control*:** Gracias al *malware* instalado en el punto anterior cabe la posibilidad de que se genere un canal de comunicación con un servidor remoto que ejecute código malicioso y manipule el sistema infectado.
- ***Actions on objetives*:** En este punto el atacante es dueño del "teclado" del equipo infectado, por lo que lo puede manipular a su antojo y habrá logrado su objetivo, el control de la máquina.

Los pasos anteriores, no se tienen porque dar en todos los ataques, pero sí que son parte de la mayoría de ellos por grupos o en su totalidad. Los ataques más conocidos se aglutinan en la siguiente lista:

- **DDoS** (*Distributed Denial of Service*): Este ataque busca la inutilización momentánea de un sistema informático. Ataca al sistema con tráfico no legitimo para que no pueda albergar peticiones de trafico legítimo y de esta manera entorpecer o acabar con la experiencia de usuario, entre otras. Un ejemplo seria el ataque a una página web, consiguiendo mediante el propio ataque la inutilización del propio portal web, evitando que sus usuarios habituales disfruten del mismo. El impacto puede ser altísimo en páginas web como Amazon y no tan alto en otro tipo de páginas webs más informativas.
- ***Ransomware***: Este tipo de ataque busca principalmente el rédito económico pidiendo un rescate por los datos de una empresa. El ataque cifra todos los datos de una empresa con una clave única que solo el atacante conoce. Este *software* malicioso se suele ejecutar mediante técnicas de ingeniería social, correos maliciosos… Es un ataque muy peligroso puesto que no solo pone en riesgo los datos de tu empresa si no los de tus clientes. Es uno de los ataques más habituales hoy en día y el que más impacto tiene.[5]
- ***Phising***: Este ataque se basa en el envío de correos haciéndose pasar por otra persona para lograr un fin malicioso. Más que un ataque por sí solo, es una manera de entrar a un sistema y lanzar un ataque más peligroso. Busca conseguir contraseñas, la ejecución de un script malicioso… Vulnera un sistema informático robusto basándose en los fallos humanos.

- **MITM** (*Man In The Middle*): Mediante este ataque un atacante busca interceptor tráfico entre dos equipos para descubrir información confidencial. De esta manera, puede lograr contraseñas, numero de cuentas, información personal…

Pese a haber detallado las principales vulnerabilidades, nacen variantes de ellas, vulnerabilidades propias del *software* del sistema operativo o de programas de uso empresarial común, vulnerabilidades de los servidores web…

Todas estas vulnerabilidades se identifican mediante los CVE mencionados con anterioridad y se registran en base de datos de uso público. Para no incurrir en problemas, conviene mantener el *software* que use el ordenador corporativo actualizado.

2.4 SIEM

Una vez explicados los mecanismos, medidas y amenazas, conviene detallar de que se trata la herramienta de seguridad que se va a desplegar y desarrollar en este trabajo. Pese a haber sido introducida en el apartado de mecanismos de una manera breve, cabe detallar de una manera más precisa la herramienta.

Un SIEM es un mecanismo de seguridad pasivo; es decir, no protege sino detecta. Conviene tener un grupo de ingenieros de seguridad encargados de la herramienta para poder revisar las alertas que esta genera, revisarla y acotar su funcionamiento.

El SIEM relaciona los *logs* de una serie de equipos de un sistema o red informático tratando de buscar comportamiento malicioso. En base a estos *logs* y la comparación que pueda hacer con los ataques registrados en su base de datos, crea alertas de distintos niveles, avisando de los problemas que se puedan estar dando en la red.

Como no se encarga de tener que bloquear el tráfico de la red, como los *firewalls*, es capaz de concentrarse en correlacionar el tráfico de una manera muy eficiente. Está pensado para manejar grandes cantidades de tráfico y computarlo. Se trata de que detecte problemas que el *firewall* no es capaz de detectar.

Por ejemplo, el *firewall* no es capaz de correlacionar tráfico de una manera eficiente, el SIEM si, por lo cual, es idóneo para desarrollar patrones de conducta maliciosa. Es capaz de detectar ataques de fuerza bruta, *ransomware*, DDoS, phising y muchos más ataques.

Para recopilar los *logs* se usa el protocolo *syslog* que opera usando el puerto 514 de UDP. Los equipos de seguridad usando el protocolo envían los datos a un equipo o máquina virtual llamada *collector* que se encarga de recopilar los datos y enviárselos al equipo central que se encargara de correlacionar la información. Para ello, es totalmente necesaria la construcción de un canal seguro de comunicación entre el *collector* y el equipo central, para que los datos viajen a través de la red de una manera segura.

El equipo central será capaz de reflejar en su *dashboard* el historial de amenazas del sistema. Además, enviara las alertas por *mail* a los administradores del sistema para que puedan revisar las alertas y solventar las amenazas. También cabe recalcar la importancia de los formatos de los *logs* que se reciben. Una unificación en el propio formato puede generar una mejor relación del SIEM con distintas fuentes de *logs*.

2.5 MACHINE LEARNING APLICADO A CIBERSEGURIDAD

Otro campo que está en reciente efervescencia es el campo de *machine learning*, en este apartado se hará un enfoque a su aplicación en entornos de ciberseguridad. Aun así, para poder entender mejor el campo de *machine learning* será inevitable proporcionar unas pinceladas básicas sobre la materia.

Machine learning hace referencia a una serie de técnicas de desarrollo de *software* que buscan que una herramienta inteligente aprenda de los datos para la aplicación de decisiones futuras. Es decir que se base en su propia experiencia para la aplicación futura de decisiones. Tiene diversos campos de aplicación, ya sea el de la medicina, economía, ciberseguridad…

El proceso que sigue un sistema de ML, suele ser el que muestra la **figura 3**:

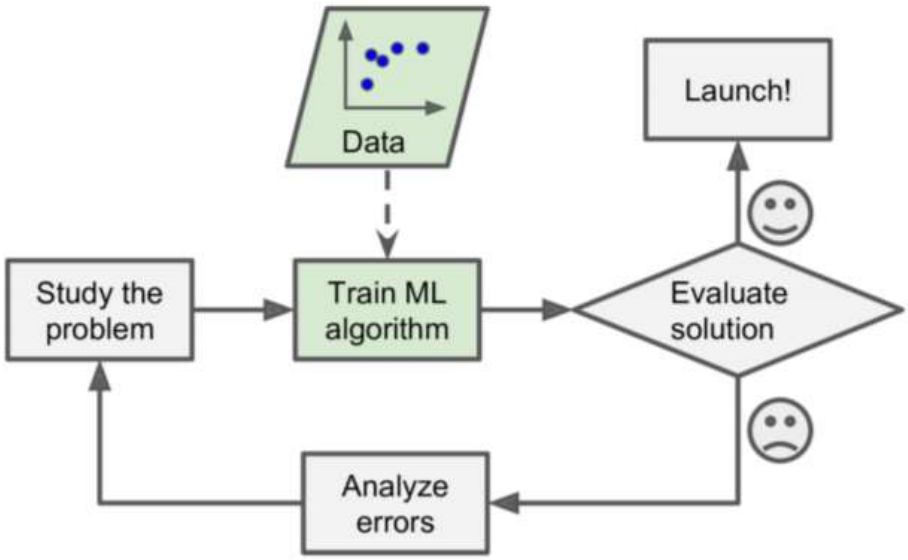


Figura 3: Diagrama de aplicación de Machine Learning

Para poder aprender de la propia experiencia del sistema es necesario otorgar a la algoritmia correspondiente una serie de datos con un formato especifico. Estos datos se agrupan en 3 grupos distintos de datos:

- **Train**: Es el grupo de datos que pretende entrenar al sistema; es decir, una serie de datos que pretende enseñar a la algoritmia los datos que ha habido en el sistema y las decisiones que se han tomado en el pasado sobre esos datos. Por lo que son datos ya procesados con anterioridad, en los que se refleja ya la decisión que se ha tomado respecto a esa línea de datos o a ese registro.
- **Validation**: Es el grupo de datos que validan y ajustan el modelo que se creado en base a los datos de *train*.

- **Test**: Es el grupo de datos que se aplica en la algoritmia después del *dataset* de *validation*; es decir, es el grupo de datos que evalúa la algoritmia que ha sido entrenada con anterioridad.

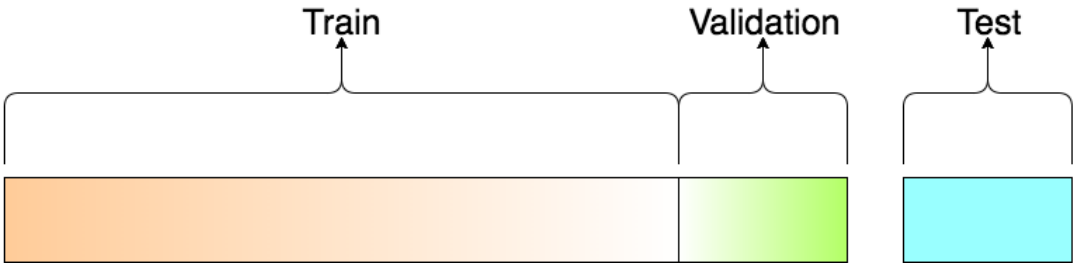


Figura 4: datasets Train, validation y test

Una vez sometido el modelo a estos 3 tipos de datos se podrá decir que tenemos un sistema inteligente capaz de predecir una decisión en base a unas *features* o características de los datos. Las características mencionadas con anterioridad constan de una gran importancia en nuestro modelo y sobre todo la selección de las mismas.

Los datos que se procesen constarán de muchos atributos que habrá que tratar de relacionar con la predicción que queremos hacer y de esta manera disminuir la cantidad de atributos de los datos que se van a procesar y agilizar el proceso.

Otro aspecto ampliamente usado es la clasificación de los propios problemas a los que se les da solución en base a algoritmia o modelos de ML. El siguiente diagrama, mostrado en la **figura 5**, pretende hacer un esbozo de los modelos clásicos de ML:

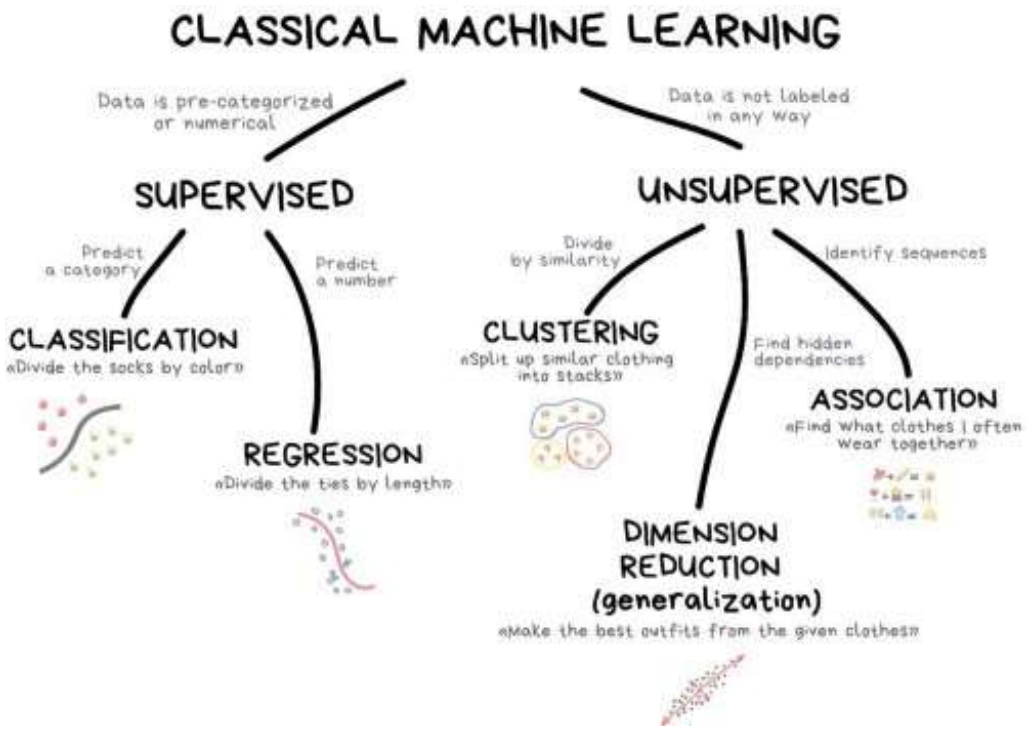


Figura 5: Diagrama de tipos de modelos clásicos de ML

Fuente: https://vas3k.com/blog/machine_learning/?fbclid=IwAR0NjjOJlZt4-KiaBGi11DskcBHAa2d6xaUchkPZdDch7pxS5sbcrZkUBJA

El anterior diagrama comienza por dividir los problemas en base a modelos con datos supervisados o no supervisados. La supervisión se asigna o hace referencia a que los datos estén etiquetados correctamente para su aplicación de la algoritmia de ML.

Por ejemplo, digamos que tenemos unos datos que hacen referencia a los *logs* de un equipamiento de red y queremos clasificar o categorizar cada *log* como trafico malicioso o no malicioso. Si este tráfico esta pre categorizado para poder entrenar el modelo, sería un modelo supervisado y sino sería uno no supervisado.

Dentro de los modelos supervisados, el diagrama muestra dos tipos de modelos: modelos de clasificación y de regresión. Los modelos de clasificación tratan de aportar una clase a cada línea de datos. Los modelos de regresión dan como respuesta un número, que mediante la interpretación humana cobra un significado.

En cuanto a modelos no supervisados, se trata de buscar relaciones entre los datos para poder dar una respuesta correcta. El modelo de *Clustering* busca agrupar los datos por similaridad y así poder procesarlos mejor. El modelo de *Dimension reduction* busca encontrar relaciones ocultas entre los datos para poder procesarlos de una manera más eficiente. El

modelo de asociación intenta identificar secuencias en los datos que pueda ayudar a un mejor procesado futuro de los datos.

Estos han sido los modelos clásicos de ML; sin embargo, el panorama actual ha evolucionado mucho y se ha transformado en una tecnología mucho más sofisticada, para ello se indicará el diagrama mnostrado en la **figura 6**:

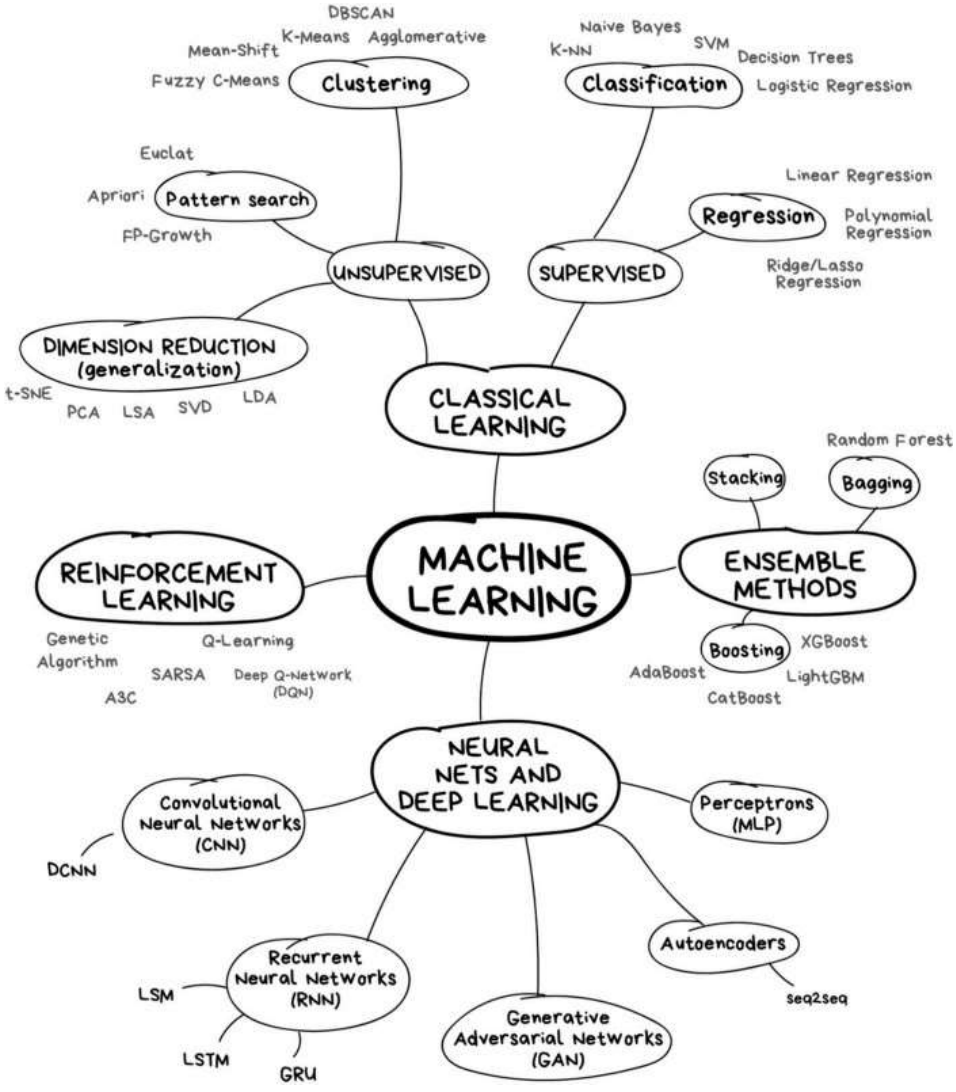


Figura 6: Tipos de modelos actuales de ML

Fuente: https://vas3k.com/blog/machine_learning/?fbclid=IwAR0NjjOJlZt4-KiaBGi11DskcBHAa2d6xaUchkPZdDch7pxS5sbcrZkUBJA

Como se puede apreciar el árbol ha crecido mucho y se han desarrollado distintos métodos más complejos y eficaces para determinados problemas. En este caso se ha

considerado no detallar cada parte de este diagrama porque sería muy extenso; por lo cual, se dejará a modo de mención y en caso de que se quiera hacer mayor hincapié, se hará en los posteriores apartados de este documento.

El resto de información útil para este proyecto de ML se dará en el apartado de análisis alternativas en su apartado relativo.

3 ALCANCE

En este apartado se determinará el alcance del proyecto, que lo que contempla es el diseño y despliegue de una herramienta SIEM, así como el análisis de su funcionamiento y posibles novedades aplicables a la tecnología y su desarrollo.

Este alcance viene concretado por un objetivo principal, a su vez compuesto por varios objetivos secundarios.

El **objetivo principal** del proyecto de este trabajo de fin de master es diseñar e implementar una herramienta SIEM en un entorno corporativo. Para ello será necesario, proporcionar un canal de datos de seguro de la red corporativa del cliente a la red corporativa propia para poder enviar los *logs* al sistema central y que este lo analice.

También convendrá mostrar una interfaz de usuario amigable al propio cliente y darle un servicio con unos mínimos en calidad y atención. Este proyecto convergerá hacia un servicio que se prestara al cliente de soporte y análisis de incidentes, por lo que la relación con el cliente es fundamental.

Pese a entender como éxito el cumplimiento de este objetivo principal, se han definido varios objetivos secundarios necesarios para la consecución de este objetivo principal. La consecución de estos objetivos secundarios será secuencial, puesto que cada objetivo depende de uno o varios de los anteriores. Estos objetivos secundarios se definen en la siguiente lista:

- **Definición de los requerimientos:** El primer objetivo secundario consiste en identificar y definir los requerimientos del sistema. Es decir, la identificación de los requerimientos de seguridad que necesita el sistema.
- **Diseño de la herramienta de seguridad:** Una vez identificados los requerimientos del sistema se pasará al diseño de la herramienta que se pretende desplegar.
- **Implementación de la herramienta:** Una vez diseñada la herramienta, se procederá al despliegue o implementación de la misma, visualizando impacto en cliente.
- **Evaluación del sistema desplegado:** Una vez implementada la herramienta y rellenada con los suficientes datos para madurarla, se pasará al análisis de la misma, refinamiento de su funcionamiento y validación del mismo.
- **Propuesta de mejora:** Una vez realizada la evaluación se procederá a la propuesta de mejora por parte del sistema y como sería más útil la utilización de algoritmos basados en ML. Además, se desarrollará una solución basada en ML para el análisis de tráfico.

4 BENEFICIOS

Este proyecto se justifica partiendo de la base de los beneficios que aporta a varios niveles; es decir, los beneficios que aporta a nivel social, económico y técnico. Mediante estos beneficios que aportaran ventajas a los distintos *stakeholders* de este proyecto en los distintos beneficios mencionados con anterioridad, se justificara la ejecución del proyecto.

Los propios beneficios que pueda traer este proyecto son los que generan un interés en la ejecución del mismo, por ello, se hace un especial hincapié en este apartado, resaltando los distintos beneficios. Por lo que, a continuación, se separaran los distintos beneficios detallándolos con más precisión en subapartados separados.

4.1 BENEFICIOS TECNICOS

Este proyecto será interesante desde el punto de vista técnico, puesto que implementa una tecnología con una madurez suficiente para aportar un grado de seguridad alto a la organización; es decir, la tecnología SIEM.

En esa misma línea, aportará al entorno empresarial del País Vasco, un acercamiento al uso de este tipo de tecnologías y a una mayor concienciación de seguridad. El mero "boca a boca" del cliente del proyecto respecto a otros clientes, puede ayudar a concienciar, a desplegar masivamente la tecnología y a entender mejor la importancia de la misma.

Además, este proyecto poseerá un componente de innovación, relacionado con ML, una tecnología en auge que ganará una mayor visibilidad orientada a seguridad informática. Esta visibilidad ayudara a visualizar la tangibilidad de esta tecnología y no verla como una promesa a nivel de tecnología.

Encima, el propio hecho del uso de ML relacionado con la tecnología SIEM, ayuda a aportar una mejora a una tecnología desarrollada y establecida, haciendo uso de tecnologías emergentes.

4.2 BENEFICIOS ECONÓMICOS

Este proyecto será interesante desde el punto de vista económico, puesto que reportará beneficios económicos directos e indirectos. También habrá varios interesados desde el punto de vista económico; es decir, el cliente y el vendedor.

Desde el punto de vista del cliente, realizará una inversión, que en el futuro le hará ahorrarse dinero en incidentes de seguridad. Los incidentes de seguridad, provocan grandes pérdidas en las empresas industriales y tecnológicas, puesto que pueden poner en jaque el nivel productivo de la empresa, su información…

Todo lo que incremente la protección a nivel de seguridad informática de la empresa, puede generar una mayor estabilidad productiva y por lo cual, un mayor flujo económico.

Desde el punto de vista del vendedor, generara un beneficio económico desde el momento de la venta del producto y la implementación. Además, el propio producto será gestionado por la empresa vendedora lo que reportará beneficios anuales y un mayor trato con el cliente, que podrá reportar más proyectos en el futuro con el mismo.

Resumiendo, se considerará ampliamente beneficioso a nivel económico para las partes con mayor implicación en este proyecto, por lo cual cabe recalcar la gran importancia del proyecto para ambas partes.

4.3 BENEFICIOS SOCIALES

Este proyecto será interesante desde el punto de vista social, puesto que generará una mayor y tranquilidad para los usuarios de la infraestructura tecnológica del cliente. Además, pese a no reparar en ello, sus datos serán protegidos de una manera más solvente.

Mirándolo con otro prisma, los servicios ofrecidos por el cliente también ganaran en calidad y puesto que se dedican al sector de contenidos públicos, sus usuarios ganaran en calidad de visión de los mismos servicios.

Finalmente, cabe recalcar que cuanto más seguros son los servicios tecnológicos más avanza la sociedad en el ala tecnológica y se acrecienta el uso de las nuevas tecnologías.

5 REQUERIMIENTOS

El cliente ha especificado una serie de requerimientos para la ejecución del proyecto que se explicaran a continuación. Primero, se analizarán los requerimientos generales del proyecto y después se segmentarán en la solución de SIEM y la solución de ML.

Los requerimientos de carácter general se muestran a continuación:

- Análisis y reconocimiento de la infraestructura
- Gestión centralizada de ofensas en la infraestructura
- Datación de comunicación entre la empresa del cliente y la empresa ejecutora del proyecto
- Pruebas de validación del funcionamiento y seguimiento del plan de PaP (Paso a Producción)
- Entrega de documentación precisa y completa

Además de estos requerimientos, también se plantean requerimientos para cada segmento de la solución como se ha indicado con anterioridad.

5.1 REQUERIMIENTOS SIEM

La solución de SIEM tendrá unos requerimientos propios que se tendrán que cumplir para satisfacer el alcance del segmento de este proyecto. Los requerimientos específicos de este segmento son los siguientes:

- Investigación, diseño y aprendizaje sobre la implementación de una herramienta SIEM
- Centralización de los *logs* de los equipos de la red del cliente
- Visualización en tiempo real del estado de la red y las ofensas que ha recibido
- *Software* relativo al equipamiento en la última versión recomendada por el fabricante
- Comunicación segura entre el colector de *logs* local y el SIEM central
- Análisis de falsos positivos y refinamiento de las reglas
- Definición del servicio de soporte que se va a desplegar como continuación del proyecto SIEM

5.2 REQUERIMIENTOS ML

La solución de ML tendrá sus propios requerimientos que se tendrán en cuenta para cumplimentar el alcance parcial del proyecto que hace referencia a esta solución. Los requerimientos de esta parte del proyecto son los siguientes:

- Comparación entre distintos modelos de ML, para ver qué modelo converge mejor en cuestiones de exactitud, precisión, matriz de confusión, *recall* y *f1 score*.

- Lograr un modelo con una exactitud mayor del 95 %
- Proporcionar una solución de detección de ataques rápida y eficaz
- Recopilación de diagramas que muestren las métricas de cada modelo

6 ANÁLISIS DE ALTERNATIVAS

Este proyecto presentara diversas variantes para la solución a nivel técnico y económico. Por lo cual, será totalmente necesario un análisis exhaustivo de las variantes o variabilidades que ofrezcan, para poder determinar un diseño basado en una argumentación firme y basada en un trabajo de investigación serio.

Para cada alternativa de solución se presentarán varias variantes y se analizaran los puntos fuertes y débiles de las mismas, posteriormente, se hará un análisis de las principales características de la solución y se ponderaran dándole el nivel de importancia a cada una que merezca.

En este proyecto, se presentarán varias variantes a nivel técnico y económico, que se listan a continuación:

- Análisis de distintas implementaciones SIEM
- Métodos de construcción de canal seguro
- Métodos de aplicación de ML
- Escalabilidad

A continuación, se analizarían estas distintas alternativas por separado eligiendo las distintas soluciones necesarias para la ejecución del proyecto.

6.1 ANÁLISIS DE ALTERNATIVAS TÉCNICAS

En este apartado se analizarán las alternativas que ofrece el proyecto desde el punto de vista técnico con el fin de poder seleccionar la opción que más se ajuste al proyecto. De esta manera, tras un análisis exhaustivo se determinará el núcleo de la solución técnica y se dará un detalle de cómo se ajusta mejor al proyecto que el resto de alternativas.

6.1.1 Análisis de Distintas Implementaciones SIEM

En este apartado, se analizarán las distintas variantes de SIEM de distintos fabricantes que hay para la solución del proyecto en este enfoque. Este análisis de alternativas será amplio puesto que esta solución es parte del grueso del proyecto y la debida justificación de la elección de la implementación brilla por su importancia.

Para la elección de la lista de SIEMs que se van a analizar se ha dirigido el enfoque hacia el típico *cuadrante de Gartner*, esta vez orientado a SIEMs. El *cuadrante de Gartner*[3] tiene la estructura mostrada en la **figura 7**:

Figura 7: Cuadrante mágico de Gartner SIEMs

Fuente: https://www.gartner.com/en/documents/3981040-magic-quadrant-for-security-information-and-event-manage

Como se puede apreciar en la ilustración, está dividida en 4 cuadrantes, el que más interés suscita es el de arriba a la derecha denominado con el anglicismo *leaders*, puesto que estos son los líderes del mercado. Dentro de este cuadrante se elegirán los 4 que estén mar arriba y a la derecha; es decir, IBM (QRadar), Splunk, Exabeam y Securonix.

A continuación, estas alternativas se analizarán exhaustivamente por separado, para luego definir los criterios ponderados y asignárselos a cada fabricante.

6.1.1.1 Alternativas

IBM QRadar[4]

El SIEM de IBM es denominado QRadar, es el principal y el más famoso de los fabricantes de SEIM. Por lo cual, uno de los puntos fuertes es la gran documentación técnica que hay en la red y el IBM *Knowledge Center*, que es el portal de formativo de IBM que cuenta con una gran cantidad de información. Además, IBM cuenta con una gran experiencia en los entornos informáticos y proporciona gran soporte a la hora de plantear las soluciones. También debe de tenerse en cuenta el gran prestigio de un todoterreno transoceánico como IBM.

El SIEM de IBM es muy flexible y se puede adaptar a cambios de configuración en base a las necesidades de cada cliente. Esto aporta una gran adaptabilidad a lo que el cliente quiere y una mayor satisfacción del mismo en las primeras reuniones con fabricante.

A nivel técnico, el SIEM de IBM córrela eventos de manera inteligente y analiza un gran conjunto de actividades, incluidas en la siguiente lista:

- Eventos de seguridad: Se reciben *logs* del firewall y en base a ello se determinan comportamientos maliciosos.
- Eventos de red: Se recibe información del equipamiento de red
- Contexto de actividad de red: Se recibe información de las transacciones de nivel de aplicación
- Actividad Cloud: Se recibe información de entornos SaaS (Softare aas a Service) e IaaS (Infrastructure as a Service) como *salesforce* u *office 365*.
- Contexto de usuarios y equipamiento: Se analiza la gestión de accesos y se escanean vulnerabilidades en este contexto.
- Eventos *endpoint*: Se analiza información de los sistemas finales
- *Logs* de aplicación: *Logs* de aplicaciones corporativas como SAP, bases de datos…
- *Threat Intelligence*: Se analiza información de fuentes como IBM X-Force

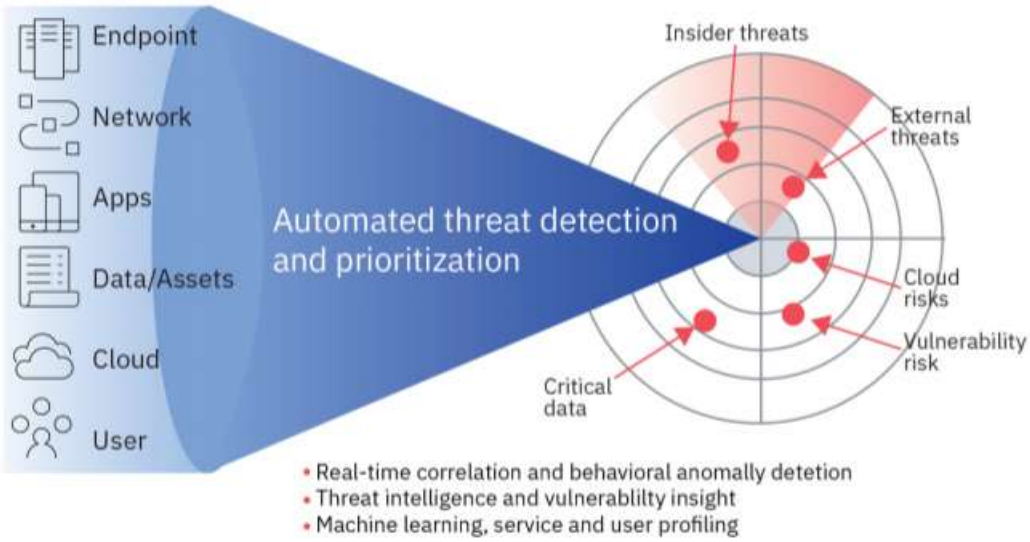


Figura 8: Alcance del SIEM QRadar

Como puede analizarse en la anterior lista, se puede determinar que el SIEM de IBM es bastante completo y realiza un análisis en distintos niveles de la infraestructura. Por lo cual es una solución versátil, adaptable, robusta y con un gran soporte detrás.

Splunk[5]

Esta herramienta pertenece a una empresa homónima, por lo cual se intuye que se dedican principalmente al desarrollo de este SIEM. El propio fabricante lo define como el centro neurálgico de seguridad de la red.

El enfoque de la herramienta se centra en 7 aspectos o funcionalidades:

- Monitorización de seguridad
- Detección de fraude
- Orquestación
- Respuesta a incidentes
- Investigación y forensia de incidentes
- *Endpoints* o equipos de usuario final
- Detección avanzada de amenazas

Figura 9: Alcance del SIEM splunk

Fuente: https://www.splunk.com/en_us/cyber-security.html

Este alcance de esta herramienta determina la falta de flexibilidad de la herramienta puesto que proporciona unas funcionalidades muy concretas y aporta poca caracterización de la herramienta. Además, ofrece menor versatilidad que otros SIEM, pues no dirige el foco hacia tantas áreas de la red como otros.

Exabeam[6]

Esta empresa se dedica a desarrollar SIEM, UEBA y DLPs por lo tanto es una empresa versátil en el sector tecnológico.

Establecen que tienen un SIEM moderno que tiene en cuenta 4 contextos:

- Colección ilimitada de datos de *log*
- Detección e investigación compleja de las amenazas internas
- Automatización y orquestación de la respuesta a amenazas
- Translación a entornos *cloud*

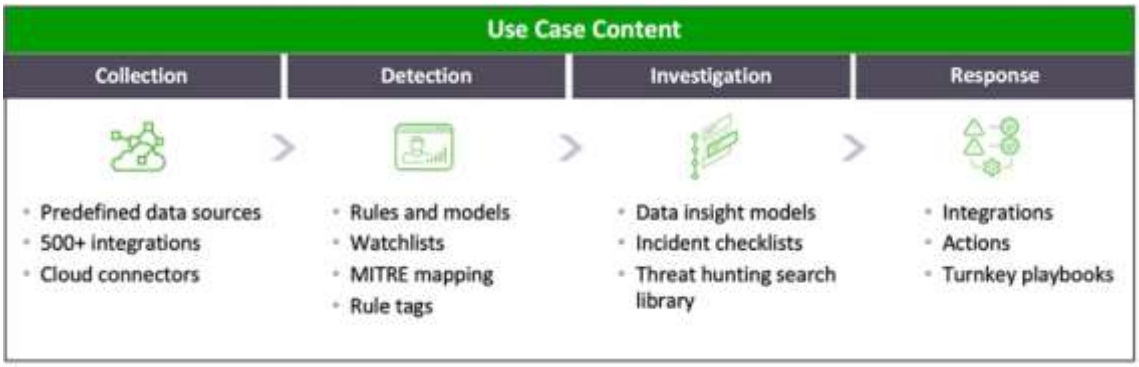


Figura 10: Caso de usos de SIEM Exabeam

Fuente: https://www.exabeam.com/product/fusion-siem/

Este último punto, es de vital importancia a la hora de elegir un SIEM de esta empresa, puesto que su solución está altamente dirigida a clientes con servicios en *cloud*. Por lo cual, es un SIEM poco flexible que se enfoca a un entorno *cloud*, característica que será un punto importante a la hora de determinar la decisión.

También, gran conocedor del entorno *cloud*, ofrece una versión de tipo SaaS, que podría ahorrar gastos de *hosting*.

Securonix[7]

Esta empresa está especializada en plataformas de análisis de seguridad, con productos principales de SIEM, SOAR (*Security Orchestation, Automation and Response*) y UEBA. Por lo cual es una empresa con un enfoque determinado y cuenta con una versatilidad de producto, pero no del todo extensa.

El SIEM de esta empresa, es muy moderno y combina la típica gestión de *logs*, con análisis de comportamiento de usuarios y entidades y soluciones de orquestación, automatización y respuesta en una plataforma *end-to-end security*; es decir, que aporta análisis de seguridad de extremo a extremo. Esta información viene resumida en la **figura 11**:

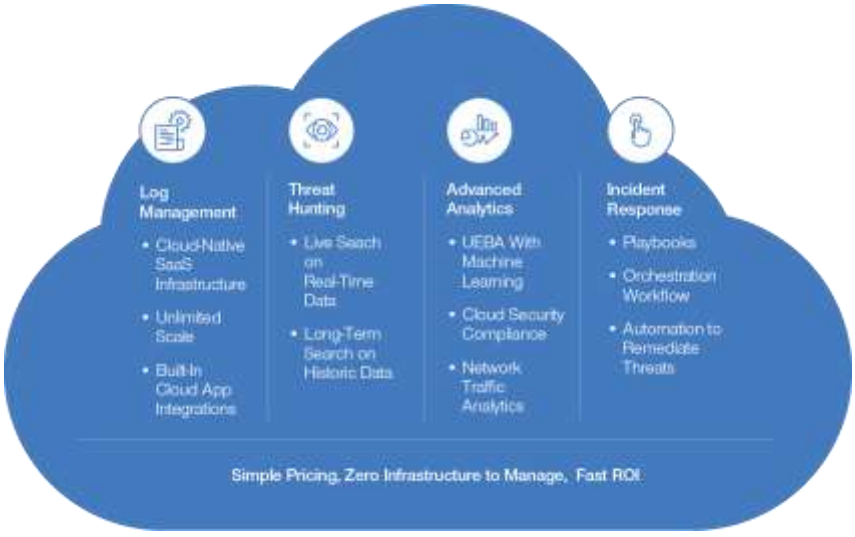


Figura 11: Características SIEM securonix

Fuente: https://www.securonix.com/products/next-generation-siem/

Un gran punto de esta solución es la novedad de su implementación, los SIEM no suelen usar algoritmia de ML para fortalecer su solución, por lo cual puede ser positivo o negativo, en cuanto a la novedad y la poca madurez de la tecnología. En este proyecto se quiere desarrollar una solución propia en este aspecto por lo cual puede ser determinante este punto.

El diagrama de actuación de la herramienta viene expresado por la **figura 12**:

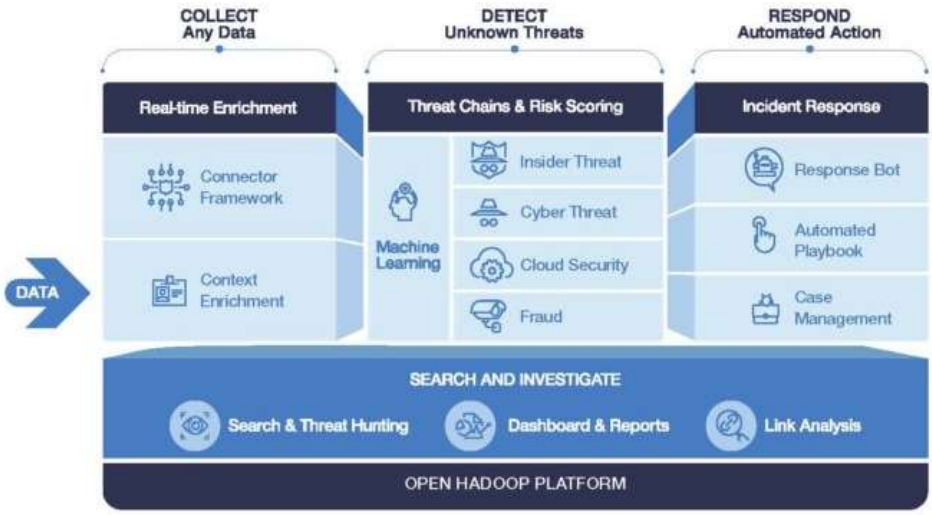


Figura 12: Flujo de actuación de Securonix

Fuente: https://www.securonix.com/products/next-generation-siem/

La actuación se divide en las tres fases que se muestran de colección, detección y respuesta como en otros SIEM, aplicando las novedades respectivas al SIEM.

6.1.1.2 Criterios de elección

Una vez definidas las distintas alternativas SIEM, cabe seleccionar unos criterios de elección adecuados para la posterior elección del fabricante SIEM que más se ajuste al cauce del proyecto. Para ello, se considerarán las características mencionadas con anterioridad en cada uno de los modelos de SIEM.

Los criterios se listan a continuación:

- **Prestigio**: El prestigio de cada uno de los modelos y fabricantes será importante en muchos sentidos: a la hora de necesitar soporte, como gancho de venta, a la hora de solicitar documentación… Por lo cual será un criterio que contenga bastante peso.
- **Experiencia en el sector:** La experiencia que tenga la empresa desarrolladora de la herramienta será muy importante para la elección de la herramienta. Los profesionales experimentados en el sector serán conocedores amplios de infraestructuras y soluciones relacionadas con el proyecto.
- **Documentación:** La documentación acerca del SIEM que contenga el fabricante será muy importante para poder facilitar la instalación y el diseño de la propia herramienta.
- **Funcionalidades:** Las funcionalidades y robustez que ofrezca el SIEM también serán muy importantes.
- **Posibilidad de personalización:** Se medirá la posibilidad del SIEM para ajustarlo a una solución personalizada

- **Precio:** el precio del producto también será importante a la hora de elegir el producto puesto que permitirá mayor competitividad a la hora de adjudicarse el proyecto.

6.1.1.3 Elección de alternativa

A continuación, se expondrá la **tabla 1** con los criterios y la elección de alternativa:

Criterios	IBM Qradar	Splunk	Exabeam	Securonix
Prestigio (25%)	9	7	5	5
Experiencia en el sector(20%)	9	7	5	5
Documentación (5%)	10	7	7	8
Funcionalidades (30%)	7	8	6	9
Posibilidad de personalización(10%)	8	7	8	8
Precio (10%)	6	7	7	8
Total	8,05	7,30	5,90	6,95

Tabla 1: Selección de alternativa SIEM

Conclusión: El SIEM de IBM ha sido el más completo en los criterios de elección. La larga experiencia de IBM en el sector de sistemas informáticos ha sido vital junto a una batería de otras características que lo erigen como el más ajustado a este proyecto.

6.1.2 Construcción de un Canal Seguro

Como bien se ha definido en el apartado de contexto, para la comunicación entre el colector de datos y el SIEM es necesaria la construcción de un canal seguro para la transmisión de los datos a través de la red pública. Para realizar esto, es necesario el uso de la tecnología VPN. Mediante esta tecnología, se pueden conectar dos segmentos de red privada diferenciados a través de la red pública de manera transparente.

Dentro de esta tecnología hay múltiples métodos de implementación, aunque en este caso solo se consideraran alternativas reales IPSEC y SSL VPN. Antes de comenzar con la comparación, hay que destacar cuál es el caso de uso para el que se plantearan las alternativas.

Como bien se ha comentado con anterioridad, lo que se busca es conectar el colector de datos que pertenece a la red corporativa del cliente en cuestión, con la herramienta SIEM que es parte de la red corporativa empresarial de la empresa ejecutora del proyecto. Por lo tanto, estas herramientas informáticas serán parte de redes privadas distintas y necesitan una comunicación confidencial entre ellas, dada la importancia de los datos de la comunicación entre las dos.

Para ello, es necesario el uso de la tecnología VPN para poder conectar estos dos elementos de red de una manera segura y cifrando los datos de la comunicación. Esto se hace a través de túneles seguros que protegen la comunicación.

6.1.2.1 Alternativas

En este apartado se compararán las dos alternativas posibles para la implementación necesaria: IPSEC y SSL. Las características de ambas podrán observarse mediante la **tabla 2**:

IPSEC	SSL
Trabaja en la capa de red	Trabaja sobre la capa de aplicación
Acceso de igual a igual	Acceso remoto
Conexión perdurable entre redes locales: excelente para conectar oficinas	Conexión granular a recursos: óptima para conectar mano de obra remota
Es independiente de la aplicación adoptada	trabaja de acuerdo con los protocolos adoptados por la aplicación
Acceso mediante el software.	Acceso al portal web
–	Limitaciones por privilegio de acceso
Permite cualquier aplicación basada en IP	Permite aplicaciones basadas en web y cliente / servidor

Tabla 2: Comparación entre IPSEC y SSL VPN

Mediante la anterior tabla, se han conocido las características básicas de ambas implementaciones y será posible baremar ambas y poder elegir la alternativa que más convenga en este caso.

6.1.2.2 Criterios de elección

En este apartado se considerarán los distintos criterios que serán vitales a la hora de elegir entre las implementaciones de tecnologías VPN. Se ha optado por seleccionar 3 criterios principales:

- **Transparencia al usuario**: Sera completamente vital que el usuario no tenga que intervenir en la comunicación a través de dicho canal de comunicación.
- **Facilidad de configuración**: Sera importante la facilidad de configuración de túnel entre los dos extremos para así ahorrar costes.
- **Prestaciones:** También será importante que la tecnología proporcione prestaciones altas y capacidades altas para la comunicación.

6.1.2.3 Elección de alternativa

Para el baremo y posterior elección de la alternativa a elegir, se usará una tabla que indicara la importancia que se le ha asignado a cada opción, como en la anterior elección. La elección se hará mediante la **tabla 3**:

Criterios	IPSEC	SSL
Transparencia al usuario (%35)	9	3
Facilidad de configuración (%30)	8	7
Prestaciones (%35)	8	9
Total	8,35	6,3

Tabla 3: Elección tecnología VPN

Conclusión: La transparencia al usuario, la facilidad de configuración y las prestaciones que otorga la tecnología IPSEC han sido vitales para su elección. En este proyecto al no ser necesarias las ventajas web que otorga la tecnología SSL, la decisión se ha inclinado fácilmente hacia IPSEC.

6.1.3 Métodos de Aplicación de ML

Este proyecto constara de un apartado de desarrollo de una solución de ML que trabaje conjuntamente al SIEM para caracterizar eventos de seguridad de una manera inteligente y basándose en la experiencia.

Para ello, es vital el uso de un modelo especifico de *Machine Learning* que se ajuste de la mejor manera posible al problema que presenta este proyecto. Se optarán por los cuatro modelos principales que se suelen usar en problemas de este tipo, estos modelos se listan a continuación:

- LogisticRegression
- RandomForest
- SVM
- NeuralNetworks

6.1.3.1 Alternativas

LogisticRegression

Antes de hablar de modelos de LogisticRegression[8] y LinearRegression[9], cabe tratar de detallar de que se trata la propia regresión. La regresión hace referencia al hecho de intentar predecir el valor de una característica en base a predictores. Los modelos de regresión difieren entre sí en base al número de variables independientes y a la relación entre las variables dependientes e independientes.

En este apartado se detallará el modelo de regresión logística, pero para la mejor comprensión del mismo, se detallará el modelo de regresión lineal con anterioridad.

Los modelos de LinearRegression son una categoría dentro de los modelos de regresión, en este caso el número de variables independientes se limita a uno y la relación entre las variables independientes y las dependientes es lineal.

El próximo grafico puede ayudar a entender mejor como es un modelo de regresión lineal y que es lo que trata de lograr, este grafico se muestra mediante la **figura 13**:

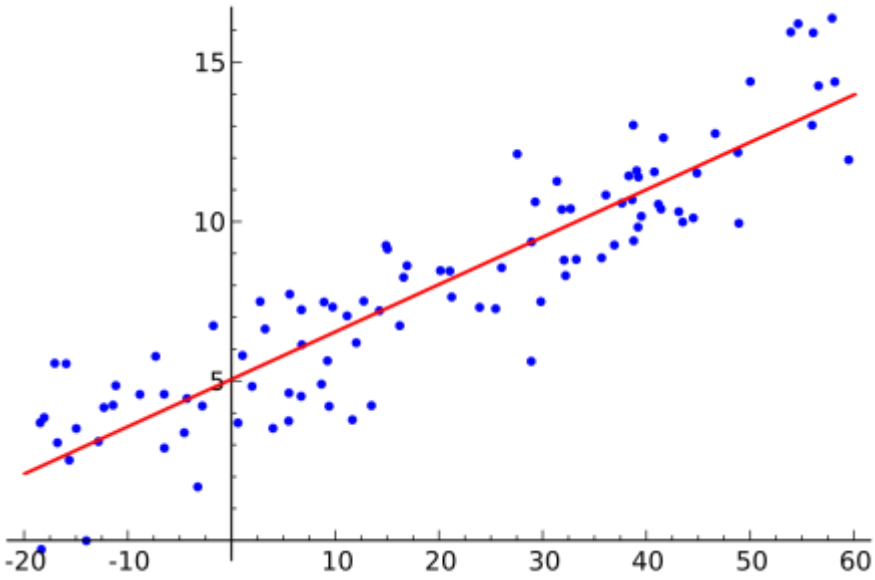


Figura 13: Grafico de regresión lineal

En este caso se dividen las variables independientes, el eje *x* y las variables dependientes, el eje *y*, la línea roja trata de modelizar lo mejor que pueda los puntos azules, tratando de optar a dar una distancia mínima de los puntos a la raya. Esta línea, como se puede observar es una recta por lo que se denomina al modelo regresión lineal.

Como es de prever, este modelo habrá en casos que ajuste muy bien y en otros que ajustará de una manera pésima por su propia característica lineal. La línea roja se suele definir con la siguiente ecuación:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

La anterior ecuación define una recta y sus parámetros cambiantes son la β_0 y β_1. Una vez detallada la regresión lineal, toca describir de la regresión logística.

Este modelo de ML se aplica frecuentemente en problemas de clasificación; es decir, en problemas en los que tienes que predecir de que clase es un dato con una serie de características.

La mayor diferencia con el modelo explicado con anterioridad, se basa en que el modelo de regresión logística clasifica en base a una función sigmoidea, no una recta como el caso anterior, que tiene dos *boundaries* o límites. Mediante la próxima gráfica, mostrada en la **figura 14**, se entiende mejor este aspecto:

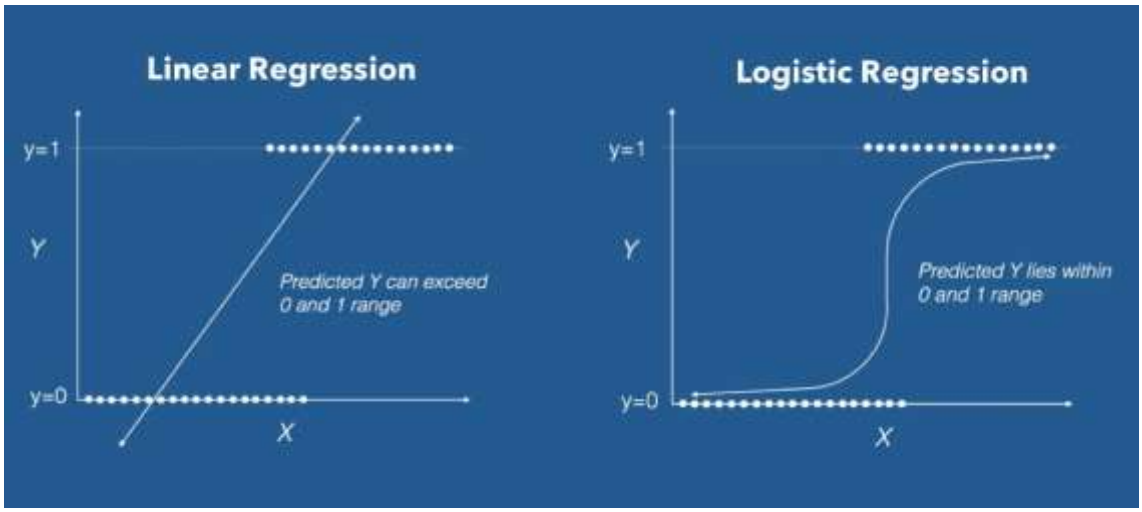


Figura 14: Regresión lineal vs logística

Como puede observarse, en este caso se clasifica con una función distinta que siempre estará entre dos límites. A esta función se le llama la función de logística. Esta función se limita entre el 0 y el 1 y se representa con la siguiente formula:

$$\sigma(Z) = \frac{1}{1 + e^{-Z}} \; \forall \; Z(X) = \; \beta_0 \; + \; \beta_1 X$$

Para la decisión si una entrada pertenece a una clase u otra se define un umbral, qué por encima del mismo se tomará esa entrada como un valor de la clase A y sino de la clase B. Este último aspecto se representa gráficamente mediante el siguiente gráfico mostrado en la **figura 15**:

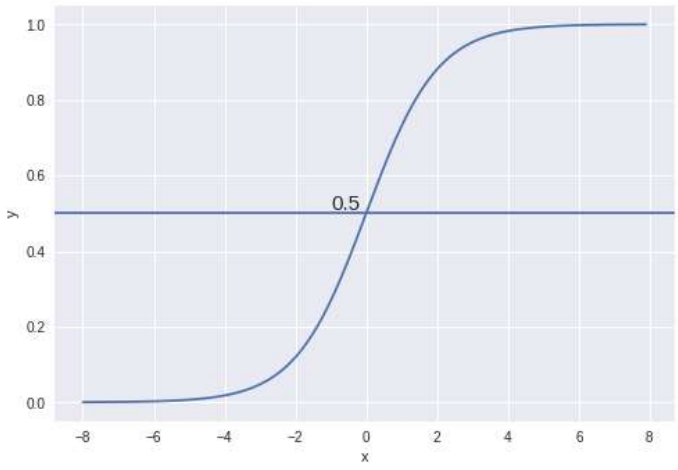


Figura 15: Umbral en la función logística

RandomForest

Los *Random Forest*[10] son una combinación de árboles predictores (arboles de decisión) que predicen independientemente y que operan juntos. Primero cabe reflejar que es un árbol de decisión, un árbol de decisión viene bien detallado en la **figura 16**:

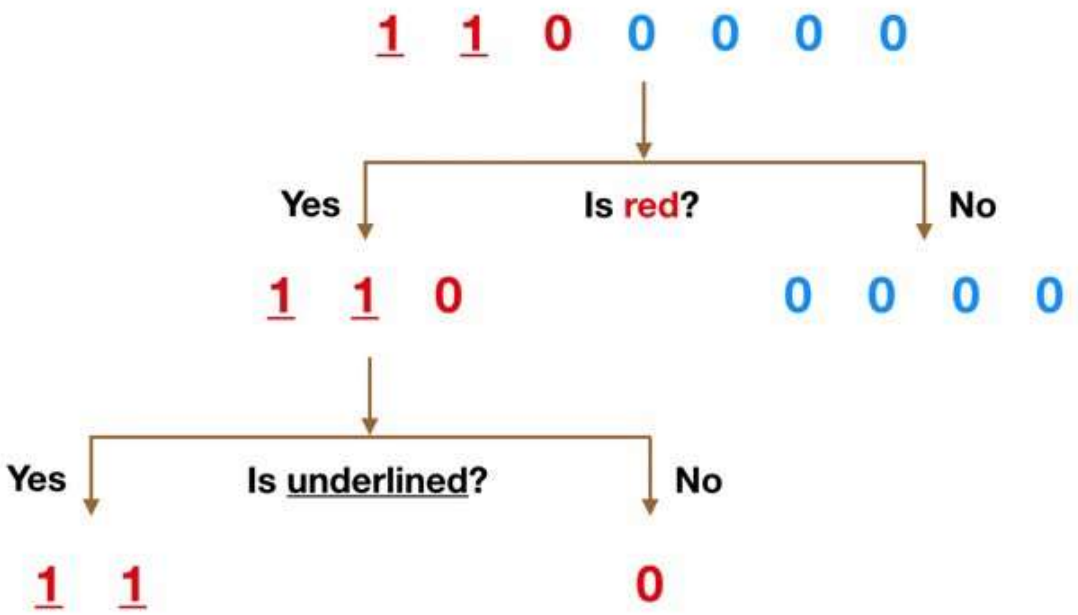


Figura 16: Árbol de decisión en ML

Como se puede apreciar se denomina árbol de decisión a un diagrama en el que se caracteriza la clase de un objeto en base a características dicotómicas. Por ejemplo, en el ejemplo anterior se clasifican una serie de números en base a sus características; es decir, se indica que características tienen los 0s y que características tienen los 1s. En base a este diagrama será sencillo determinar si un número es 0 o 1 en base a sus características.

En el caso de *Random Forest*, se construyen varios árboles de decisión que predicen independientemente una clase, una vez realizadas todas las predicciones la clase con más votos será considerada como la clase predicha. Esto se puede observar en el siguiente diagrama mostrado en la **figura 17**:

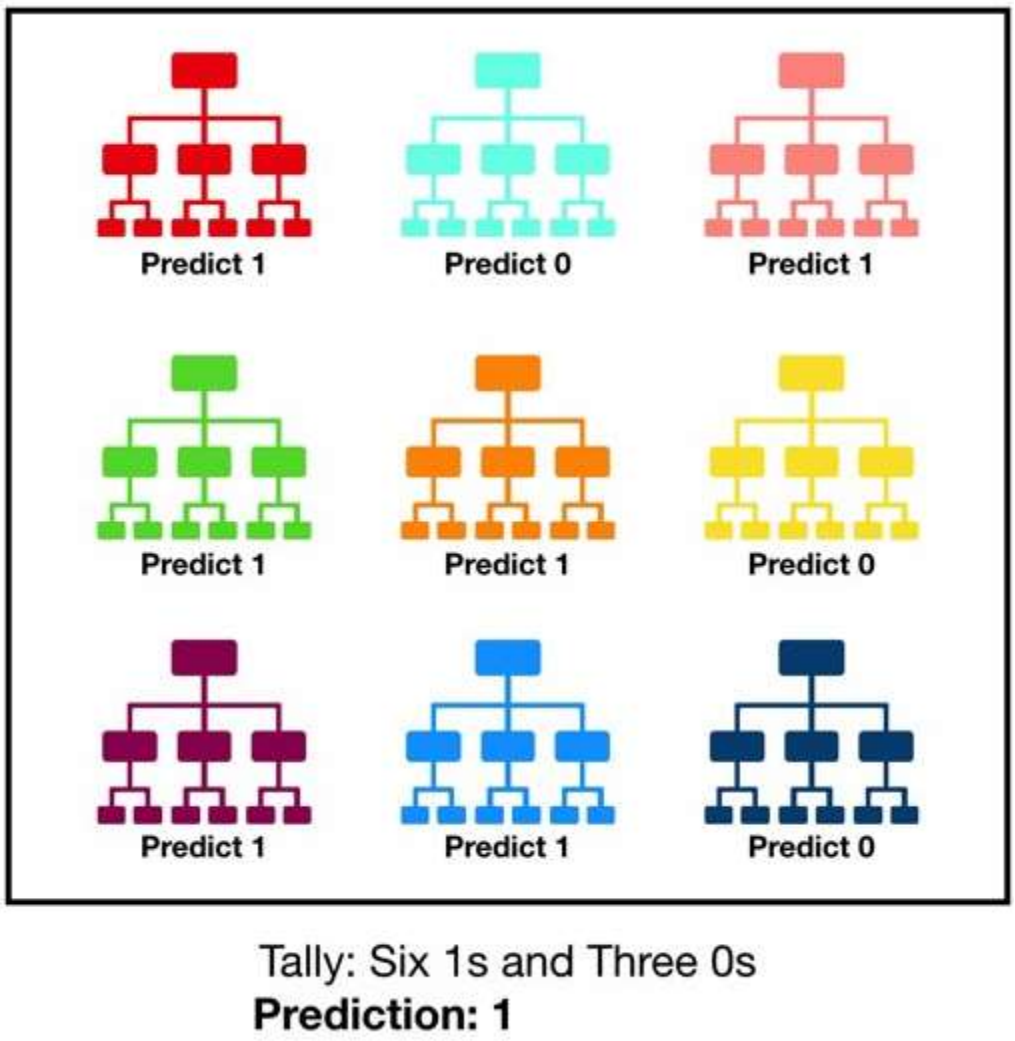


Figura 17: Diagrama de Random Forest

En el diagrama anterior, se puede apreciar como nuestros arboles independientes predicen en 6 ocasiones que el número es un uno y en 3 ocasiones un 0; por lo cual, el resultado será 1. Aunque en este simple ejemplo, parece que es complicado la necesidad de tener que usar un modelo de *Random Forest*, hay en ocasiones que el mero hecho de actuar de manera independiente en más ocasiones es beneficioso.

SVM

El modelo *Support Vector Machine* (SVM)[11] es un algoritmo de ML muy popular usado en problemas de regresión y problemas de clasificación. Este modelo es bastante similar al explicado en la regresión lineal, puesto que también se caracteriza con una función, llamada en esta ocasión hiperplano. Los puntos que están más cercanos al hiperplano son llamados vectores de soporte, que son usados para dibujar la línea perimetral.

Así como otros modelos tratan de minimizar el error entre el valor predicho y el real, este modelo trata de ajustar la mejor línea teniendo en cuenta un ligero umbral. Esto se entiende mejor visualizando el siguiente diagrama mostrado en la **figura 18**:

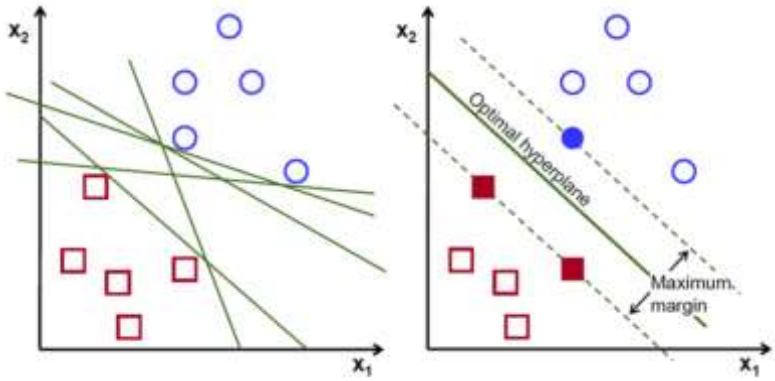


Figura 18: Grafico SVM

En este caso el umbral seria el que va desde el hiperplano a los vectores de soporte y ese umbral es donde un valor que cayera ahí no se podría predecir de una manera correcta. También se puede visualizar que el vector soporte es designado con ese nombre por el mero hecho que ese valor es el que soporta la decisión del modelo ante una muestra.

También se puede aplicar a espacios de más de una dimensión, como muestra la **figura 19**:

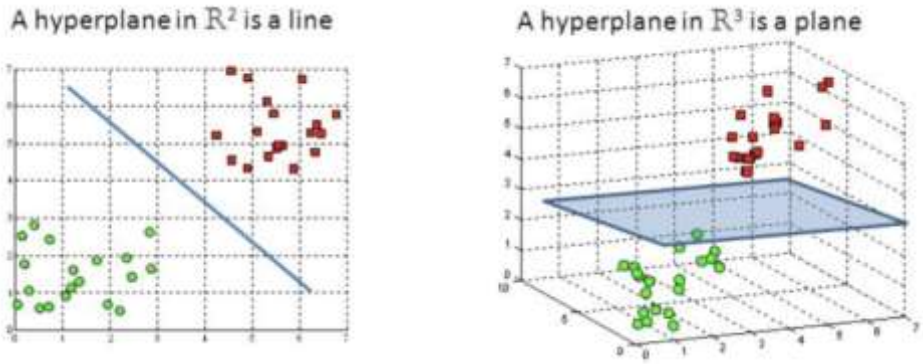


Figura 19: Gráficos SVM bidimensional y tridimensional

Redes Neuronales

Para detallar las redes neuronales[12] primero cabe describir el funcionamiento de una neurona. Una neurona toma un valor de entrada, le aplica una determinada operación matemática y genera una salida, como se puede apreciar en la **figura 20**:

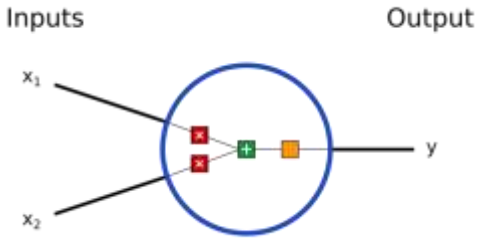


Figura 20: Funcionamiento de una neurona

En esta imagen se puede observar que primero se asigna un determinado peso a cada entrada, de la siguiente manera:

$$x_1 \rightarrow x_1 * w_1$$

$$x_2 \rightarrow x_2 * w_2$$

Posteriormente se suman las entradas ponderadas añadiéndole una constante o desviación:

$$(x_1 * w_1) + (x_2 * w_2) + b$$

Finalmente, se pasan a una función de activación que permitirá lograr un valor de salida, de la siguiente manera:

$$y = f(x_1 * w_1 + x_2 * w_2 + b)$$

Una función que se suele usar habitualmente es la sigmoide, que se puede apreciar a continuación:

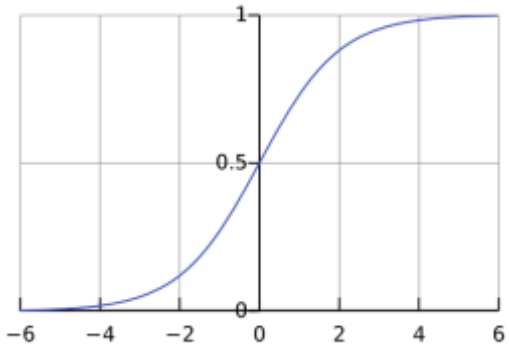


Figura 21: Función sigmoidea

Esta función asigna números sin límites a una salida limitada entre 0 y 1. Los números muy negativos serán 0 y los números muy positivos serán un uno. Una vez explicado este proceso, ya se entiende el funcionamiento de una neurona.

Las redes neuronales son un conjunto de neuronas que reciben varias entradas y entregan una salida concreta. Un diagrama sencillo de las redes, sería el mostrado en la **figura 22**:

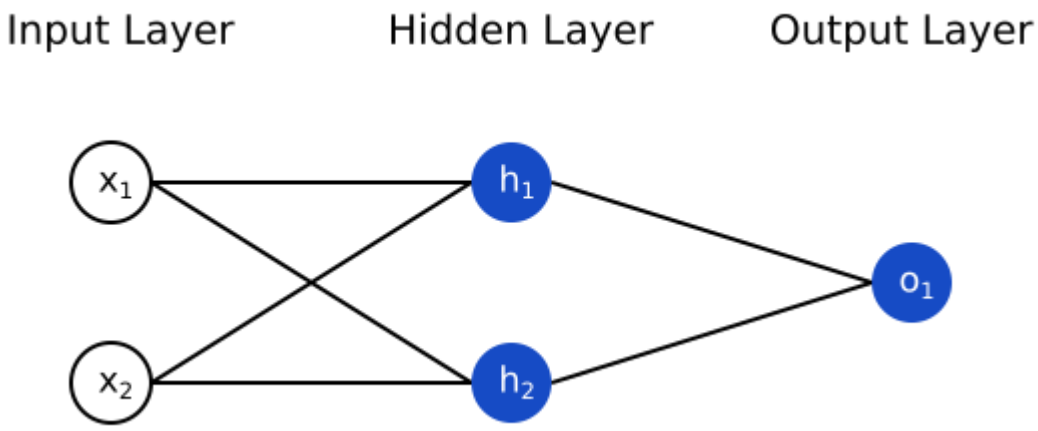


Figura 22: Diagrama de una red neuronal

Esta red tiene dos entradas, una capa oculta de dos neuronas y una capa de salida de una neurona. El aplicativo del funcionamiento es el mismo visto con anterioridad. Pero esto se entenderá mejor con un ejemplo:

Name	Weight (lb)	Height (in)	Gender
Alice	133	65	F
Bob	160	72	M
Charlie	152	70	M
Diana	120	60	F

Tabla 4: Ejemplo Red neuronal I

La red neuronal vista con anterioridad consta de dos entradas por lo que haría falta seleccionar dos características de las muestras, en este caso el peso y la altura. Estas características se pasarán a valores más bajos con el motivo de simplificar las muestras. Además, el campo del sexo se pasará a 0 cuando es masculino y a 1 cuando es femenino. Logrando los siguientes valores:

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1

Tabla 5: ejemplo red neuronal II

Como se ha podido observar las predicciones de nuestro modelo están asociadas a los pesos de cada entrada y a la desviación o *bias* que se le aplique posteriormente. La modificación de estos parámetros modificará la salida que de nuestra red neuronal, por lo cual una buena elección de estos valores mejorará nuestro modelo.

6.1.3.2 Criterios de elección

En este caso, será un poco distinto a otros apartados, puesto que en cuanto a los modelos de *Machine Learning,* conviene analizar todos los datos con distintos modelos y elegir el modelo que nos proporciones mejores resultados.

Entonces, este apartado constara de métricas[13] relacionados con la calidad de un modelo que se tomaran como los propios criterios de elección. Estos criterios se detallarán a continuación:

- *Accuracy*/exactitud: Indica el número de elementos clasificados correctamente en comparación con el número total de artículos.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precisión: Esta métrica representa el número de verdaderos positivos que son realmente positivos en comparación con el número total de valores positivos predichos.

$$precision = \frac{TP}{TP + FP}$$

- Matriz de confusión: Es la matriz que muestra los falsos positivos y negativos y los positivos y negativos reales.

- Sensibilidad: La métrica de sensibilidad muestra la cantidad de verdaderos positivos que el modelo ha clasificado en función del número total de valores positivos.

$$recall = \frac{TP}{TP + FN}$$

- F1 score: Esta métrica es la combinación de las métricas de precisión y sensibilidad y sirve de compromiso entre ellas. La mejor puntuación F1 es igual a 1 y la peor a 0.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Con estas métricas se da por concluido el apartado de definición de criterios de elección.

6.1.3.3 Elección de alternativa

En este apartado se elegirá la alternativa en base a las métricas explicadas en el apartado anterior. Para ello, se presentarán las métricas logradas para cada modelo, así como se referenciará la parte del código en el apartado de metodología y de anexos para poder comprobar el procedimiento habitual para sacar estas métricas.

Como se ha explicado durante el apartado anterior se calcularán 5 tipos de métricas para cada modelo; primero, se presentarán los datos de cada modelo de forma individual y posteriormente, a modo de resumen se incluirán en una tabla que permitirá visualizar de una manera más clara la elección del mejor modelo.

Primero, el modelo de regresión logística logra la siguiente exactitud:

```
La precision del modelo de regresion logistica: %79.93
```

Como se puede observar es un buen nivel de exactitud, aunque no lo suficiente, pero quedan por observar el resto de métricas para poder ver si el sistema es lo suficientemente bueno, para ello mediante los métodos *classification_report* y *confussion_matrix* se sacarán el resto de métricas mostradas en la **figura 23**:

```
              precision    recall  f1-score   support

           0       0.99      0.80      0.89    203300
           1       0.12      0.84      0.21      6701

    accuracy                           0.80    210001
   macro avg       0.56      0.82      0.55    210001
weighted avg       0.97      0.80      0.86    210001

Matriz de confusion:

[[162195  41105]
 [  1047   5654]]
```

Figura 23: Resultados regresión logística balanceada

Como puede observarse el informe de clasificación muestra mucha información, pero la fila que nos atañe es la *weighted avg*, se tendrá en cuenta esta información para la comparación con el resto de modelos, puesto que un criterio es la matriz de confusión. Si no, habría que analizar los resultados positivos y negativos de las métricas. Además, se ha dibujado otra matriz de confusión más visual que pueda ayudar a mejorar la visualización de la misma mediante la **figura 24**:

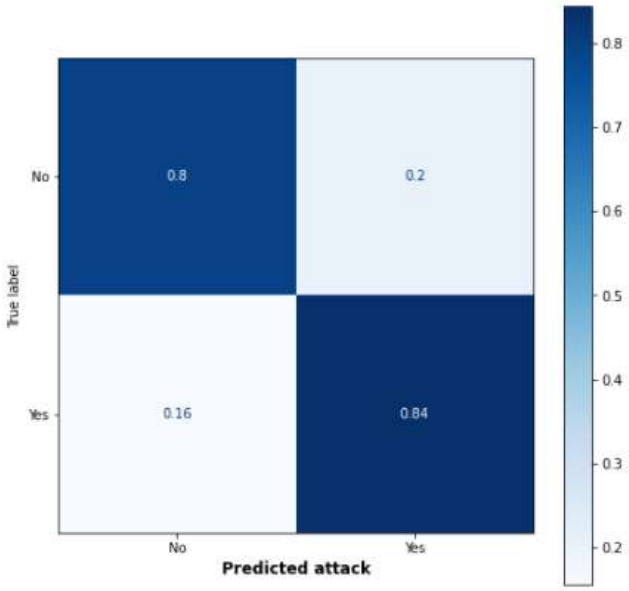


Figura 24: Matriz de confusión de regresión logística balanceada

En este modelo, se ha usado la opción balanceada del mismo para intentar forzar que logre una mejor estimación de los ataques, se ha mejorado este aparatado, pero se ha cedido de sobre manera en el campo de la exactitud. El modelo funciona mejor, pero no termina de converger del todo bien. En el caso qué no usa la opción de balanceo, se logran las siguientes métricas mostradas en la **figura 25**:

```
La precision del modelo de regresion logistica: %96.76

                precision    recall  f1-score   support

             0       0.97      1.00      0.98    203300
             1       0.31      0.01      0.03      6701

      accuracy                           0.97    210001
     macro avg       0.64      0.51      0.50    210001
  weighted avg       0.95      0.97      0.95    210001

Matriz de confusion:

[[203101    199]
 [  6613     88]]
```

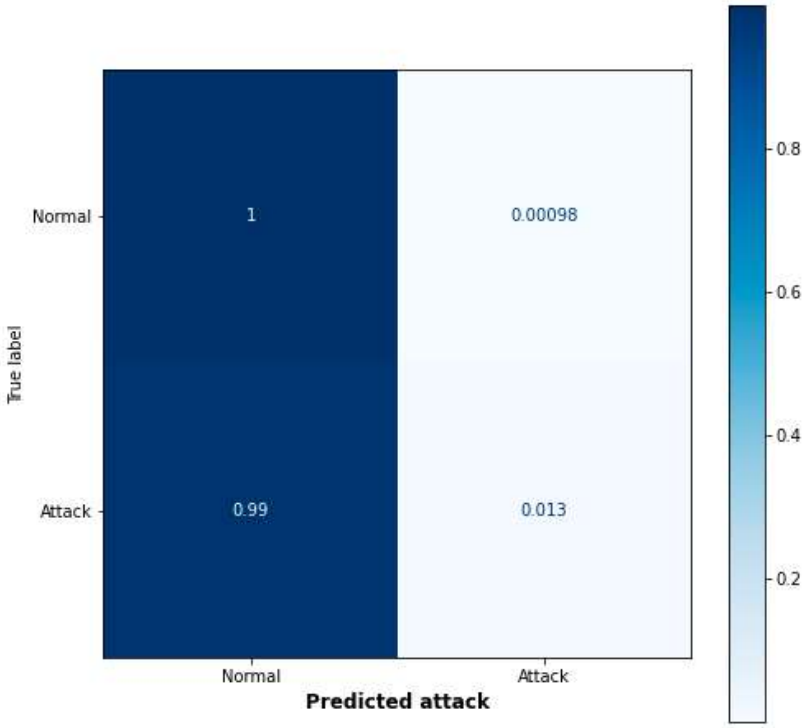


Figura 25: Métricas de regresión logística no balanceada

En este caso, como se puede observar, se logra una mejor exactitud, pero se cede calidad en el resto de métricas. Por lo cual el modelo no termina de ser del todo valido en ninguno de los dos aspectos.

El siguiente modelo a analizar, es el SVC (*Support Vector Classifier*) y las métricas que presenta el modelo son las mostradas en la **figura 26**:

```
La precision del modelo de Support vector machine: %96.68
              precision    recall  f1-score   support

           0       0.97      1.00      0.98    203300
           1       0.28      0.03      0.05      6701

    accuracy                           0.97    210001
   macro avg       0.63      0.51      0.52    210001
weighted avg       0.95      0.97      0.95    210001

Matriz de confusion:

[[202856    444]
 [  6526    175]]

Out[17]: Text(0.5, 0, 'Predicted attack')
```

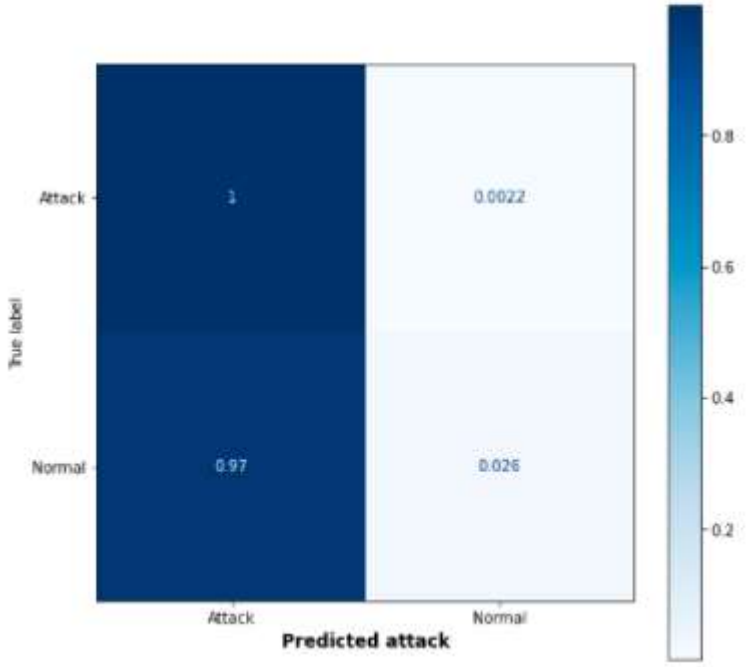


Figura 26: Métricas del modelo SVC

Este modelo, pese a mostrar una exactitud alta, no es capaz de detectar bien los ataques, como muestra la matriz de confusión, puesto que la distribución es casi de 1 en el caso de los falsos negativos. Pese a usar el método balanceado como en la regresión logística, no se mejoran los resultados de ninguna de las maneras. Además, este modelo no es aconsejado para *datasets* con un gran número de datos, por su gran necesidad de capacidad de computación.

El tercer modelo es el modelo RandomForestClassifier y sus métricas son las mostradas en la **figura 27**:

```
La precision del modelo de Random Forest: %99.80
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    203300
           1       0.96      0.98      0.97      6701

    accuracy                           1.00    210001
   macro avg       0.98      0.99      0.98    210001
weighted avg       1.00      1.00      1.00    210001

Matriz de confusion:

[[203053    247]
 [   163   6538]]
Out[52]: Text(0.5, 0, 'Predicted attack')
```

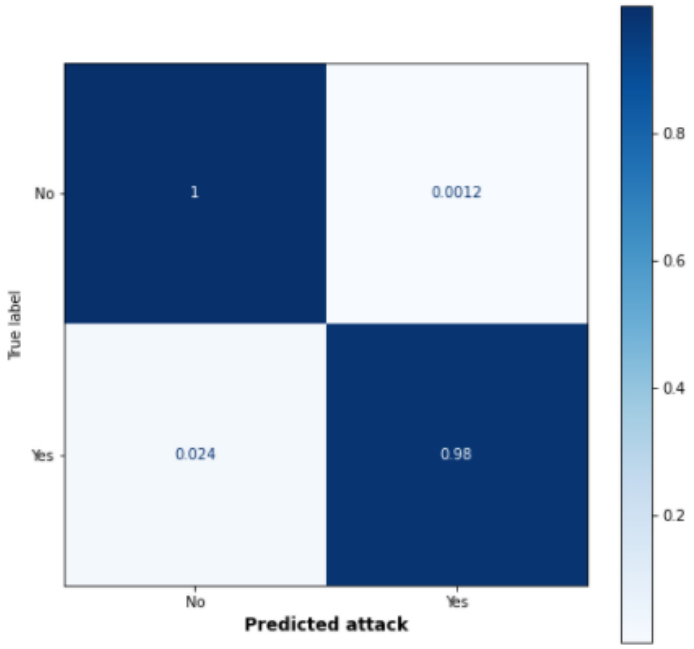


Figura 27: Métricas del modelo RandomForestClassifier

Como se puede observar, las métricas son excelentes en todos los sentidos por lo que se deriva que es un modelo muy ajustado a la solución. Se detectan bien tanto los ataques como el trafico normal. Este modelo sería capaz de identificar un ataque con el 98 % de probabilidad, unos valores muy buenos. Además, la exactitud del modelo es muy buena rozando el 100 %, el objetivo de convergencia un modelo de inteligencia artificial.

El último modelo es el de redes neuronales y sus métricas son las mostradas en la **figura 28**:

```
La precision del modelo de Redes neuronales: %96.76
              precision    recall  f1-score   support

           0       0.97      1.00      0.98    203300
           1       0.31      0.01      0.03      6701

    accuracy                           0.97    210001
   macro avg       0.64      0.51      0.50    210001
weighted avg       0.95      0.97      0.95    210001

Matriz de confusion:

[[203101    199]
 [  6613     88]]
```

Out[41]: Text(0.5, 0, 'Predicted attack')

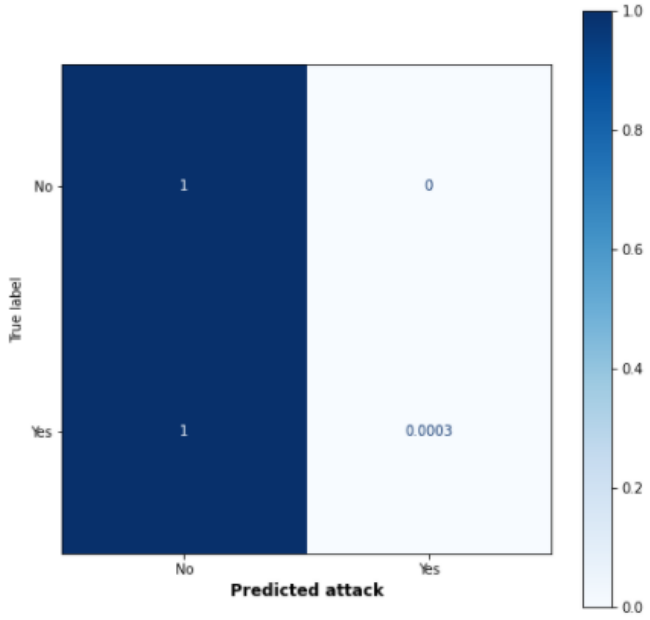


Figura 28: Métricas del modelo de redes neuronales

En este caso, el modelo queda muy desbalanceado. Mediante elmetodo de validación cruzada; es decir, del uso de los distintos atributos del modelo para la obtención del mejor modelo, se han logrado unos mejores resultados, como se aprecia en la **figura 29**:

```
La precision del modelo de Redes neuronales: %96.06
              precision    recall  f1-score   support

           0       0.99      0.97      0.98    203300
           1       0.44      0.82      0.57      6701

    accuracy                           0.96    210001
   macro avg       0.72      0.89      0.78    210001
weighted avg       0.98      0.96      0.97    210001

Matriz de confusion:

[[196191   7109]
 [  1174   5527]]
```

Out[17]: Text(0.5, 0, 'Predicted attack')

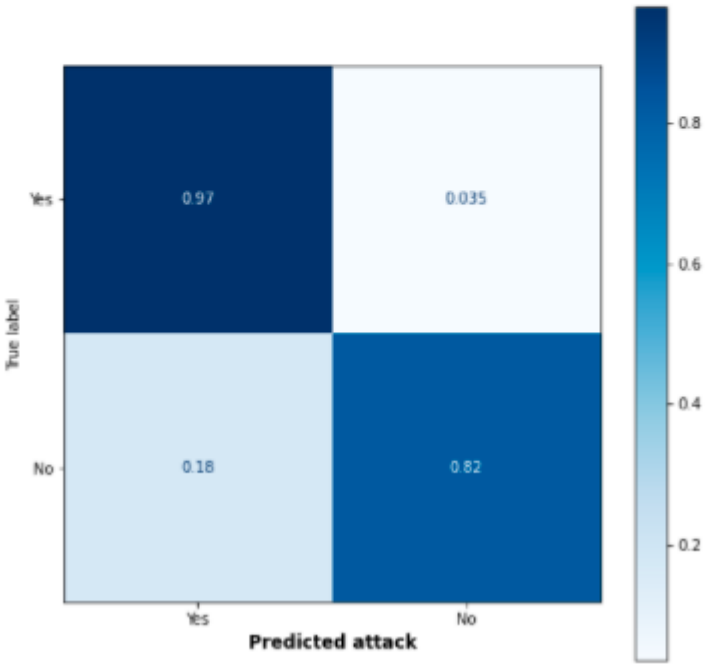


Figura 29: Resultados de redes neuronales con validación cruzada

Como puede observarse, los resultados han mejorado considerablemente, mejorando sustancialmente el modelo de redes neuronales.

En el caso de no haber logrado ningún modelo que se adecue a la solución o el problema que se presenta, se debería de aplicar técnicas de *downsampling* o *oversampling*. Estas técnicas se basan en usar datos creados manualmente que sean de la clase minoritaria o restar *logs* de la clase mayoritaria, para entrenar el modelo de una manera más balanceada

Una vez calculadas y mostradas todas las métricas, a modo de resumen se incluirán los resultados de cada modelo en la **tabla 6**:

Criterios	LogisticRegresion	SVC	RandomForest	RedNeuronal
Accuracy (%15)	0,80	0,97	0,99	0,96
Precisión(%15)	0,97	0,95	1	0,98
Matriz de confusión (%40)	0,85	0,4	0,98	0,85
Recall (%15)	0,80	0,97	1	0,96
F1 score (%15)	0,86	0,95	1	0,97
Total	0,855	0,736	0,99	0,92

Tabla 6: Evaluación de modelos de ML

Conclusión: Como se puede observar en la siguiente tabla la opción elegida es la RandomForest, esta elección se basa en las métricas que ha obtenido en cada uno de los criterios de selección. En el caso de ML, su propio funcionamiento favorece la obtención de métricas y la comparación de modelos.

Además, cabe destacar que los otros modelos hubieran sido peor valorados si se hubieran tenido en cuenta, no los parámetros medios, sino sus partes positivas y negativas ponderadas. En este caso no se ha efectuado de esta manera, porque ya se ha tenido en cuenta la matriz de confusión que es la fuente de los cálculos del resto de métricas.

6.2 Análisis de Alternativas de Negocio

En este apartado se tratará de analizar las alternativas que puede ofrecer el negocio o el proyecto desde el punto de vista del desarrollo de negocio. Para ello, se analizará uno de los puntos fuertes del propio análisis del desarrollo de negocio; es decir, la escalabilidad.

6.2.1 Análisis de la Escalabilidad del proyecto

En este apartado se analizará la escalabilidad del proyecto; es decir, como converge el proyecto ante el crecimiento de la infraestructura del cliente, del caudal o del número de usuarios. Es muy importante realizar un buen enfoque en este tipo de circunstancias, puesto que también forma parte de una buena planificación con vistas al futuro.

Como bien se ha destacado en la introducción del proyecto, la conectividad a través de internet no para de crecer, por lo que no se puede obviar este aspecto en el alcance de este proyecto.

Para un correcto análisis de la escalabilidad, lo primero es reflejar como se cobrará a la empresa ejecutora del proyecto el uso de la herramienta de QRadar, puesto que esta es la herramienta de uso que ha sido seleccionada en las alternativas técnicas. En este aspecto se muestra la **tabla 7** con las tarificaciones de QRadar:

Producto	Precio (€ * event / s)
IBM QRadar Console	$3{,}06 \times 10^{-6}$
IBM Event Collector	$0{,}127 \times 10^{-6}$

Tabla 7: Tarificaciones IBM

Estos precios no son referenciados a ninguna página oficial, puesto que cada cliente de IBM recibirá su oferta personalizada y no hay ninguna tabla de tarificaciones al uso.

Como se puede observar en la tabla anterior la tarificación de IBM se escala en base a los eventos/logs por segundo que reciben los componentes de la solución de IBM. Esto es un modelo OPEX (*Operational Expenditures*); es decir, la tarificación fluctúa en base al uso del propio producto.

Para hacer un análisis de escalabilidad, primero hay que analizar los *logs* que se están generando actualmente en la infraestructura del cliente para ello se analizara lo que está recibiendo el *firewall* actual, qué actualmente es el equipo que recibe los *logs* de la infraestructura, con los datos se ha creado la **tabla 8**:

Horario	Trafico (*log*/s)
Horas no laborales	60
Horas pico	170
Horario laboral	120
Media diaria	100

Tabla 8: Cantidad de tráfico actual en la red

Como se puede observar, el tráfico se ha dividido en tres fases, el horario laboral medio, las horas punta y el horario no laboral. Teniendo en cuenta que la tarificación del proyecto será lo que se consuma en un semestre, y posteriormente, una vez implantada la solución se cobrará como servicio. Este servicio se tarificará anualmente, por lo que esto, puede antojar una visibilidad de cómo puede evolucionar la tarificación. Aun así, cabe destacar que esta información solo es la relativa a servidores; puesto que aun los *logs* de los *host* y algunos servidores no se están colectando por el *firewall*.

Analizando el tráfico de la red y viendo que el tráfico de la red esta segmentado de la siguiente manera como muestra la **figura 30**:

Figura 30. Distribución del tráfico

- El % 5 del tráfico es perteneciente a los servidores y equipamiento monitorizados por el *firewall* de la infraestructura de la red
- El % 95 del tráfico pertenece a mas servidores y los *hosts* de la red

También para caracterizar cada tipo de usuario cabe analizar qué tipos de equipos y su distribución en la red, para ello se presenta el diagrama de la **figura 31**:

Figura 31. Distribución del tráfico por equipos

- El % 55 del tráfico es el perteneciente a *hosts*; es decir, ordenadores de mesa, portátiles o móviles.
- El % 5 del tráfico pertenece a sensores que posee la infraestructura del cliente
- El % 15 del trafico pertenece a Servidores que componen la red
- El % 15 del trafico pertenece a los distintos equipamientos de red que componen la red
- El % 10 del trafico pertenece al equipamiento de seguridad de la red

Como puede observarse la distribución de los equipos que componen la red gráficamente, está bastante controlada y ahora mismo se está monitorizando el %30 del tráfico; puesto que el equipamiento de red y seguridad y un porcentaje pequeño de los servidores están siendo monitorizados en su actualidad.

Como se puede observar en las tablas que reflejan el número de eventos que se están registrando actualmente, 100 EPS (*Events Per Seconds*), puesto que este sería el %30 del tráfico total cabria calcular el total, que supondría 333,33 EPS.

Una vez analizado el número de EPS que genera la red del cliente, habría que observar lo que soportara el colector de eventos y la consola central de IBM, esta información se muestra en la **tabla 9**:

Producto	EPS máximos
Consola central	40.000[11]
Colector de eventos	2.500[12]

Tabla 9: EPS máximos de los productos de IBM

Estos valores serán importantes a la hora de estimar como el crecimiento de la red puede afectar a la adquisición de nueva infraestructura para seguir soportando la red del cliente.

Una vez detallados todos los datos, se puede confirmar que el sistema que se ha implementado, momentáneamente soporta de manera airada toda la red del cliente. Aun así, en este apartado es necesario el análisis de la evolución anual de la red del propio cliente con el paso del tiempo.

Como se ha indicado en la introducción el mundo de Internet está en constante crecimiento y con la nueva revolución industrial crecerá aún más el número de dispositivos conectados a la red. Por lo cual, hay que tener en cuenta los índices de crecimiento que prevén para los próximos años y aplicarlo a esta solución para proveer una solución escalable.

Según el informe anual de internet de la empresa Cisco[14], el número de usuarios crecerá aproximadamente un 6 % para 2023, como muestra el gráfico mostrado en la **figura 32**:

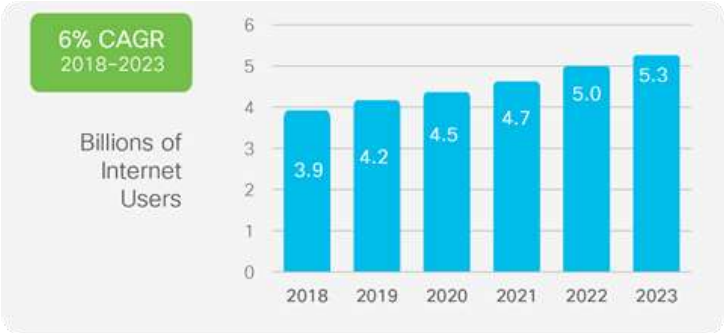


Figura 32: Crecimiento de usuarios (2018-2023)

En cuanto a los dispositivos conectados a la red de manera general, se estima que en 5 años crezca un 10 %, como muestra la gráfica de la **figura 33**:

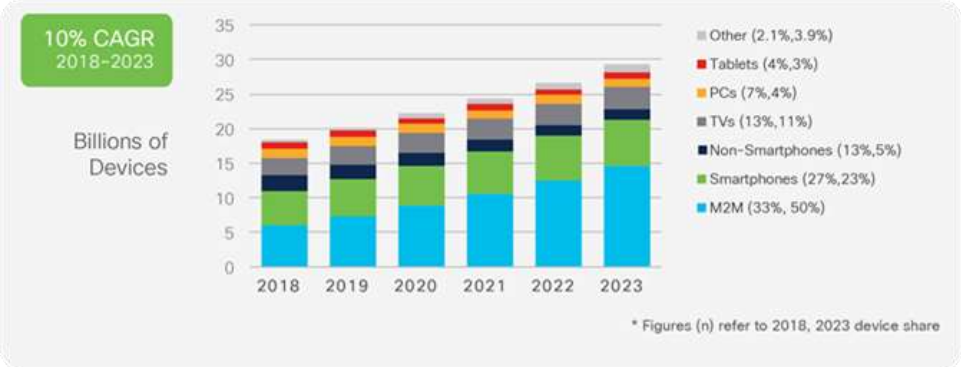


Figura 33: Crecimiento de dispositivos conectados (2018-2023)

En cuanto a las tecnologías más emergentes o boyantes se encuentra IoT, que además tiene una relación muy estrecha con el crecimiento de eventos en las redes, según el mismo estudio el crecimiento será de un 19 % entre los años 2018-2023, como muestra en la gráfica de la **figura 34**:

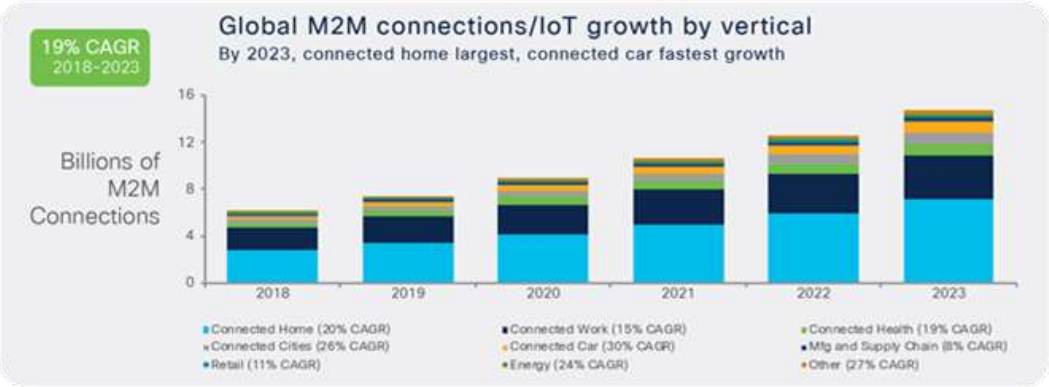


Figura 34: Crecimiento de conexiones IoT (2018-2023)

Como estima el estudio anterior el crecimiento que se vive actualmente en la interconexión de dispositivos es muy creciente. Aun así, cabe destacar que un plan de escalabilidad se suele hacer teniendo en cuenta mínimo 5 años desde la ejecución del proyecto. Estos estudios solo nos aportan datos de los próximos 2-3 años.

Se ha optado por incluir datos de este estudio puesto que los datos están soportados y revisados con los datos anuales de los años ya completados. Para poder tener esta visibilidad se incluirá la gráfica mostrada en la **figura 35** que aporta datos de 2019 a 2030:

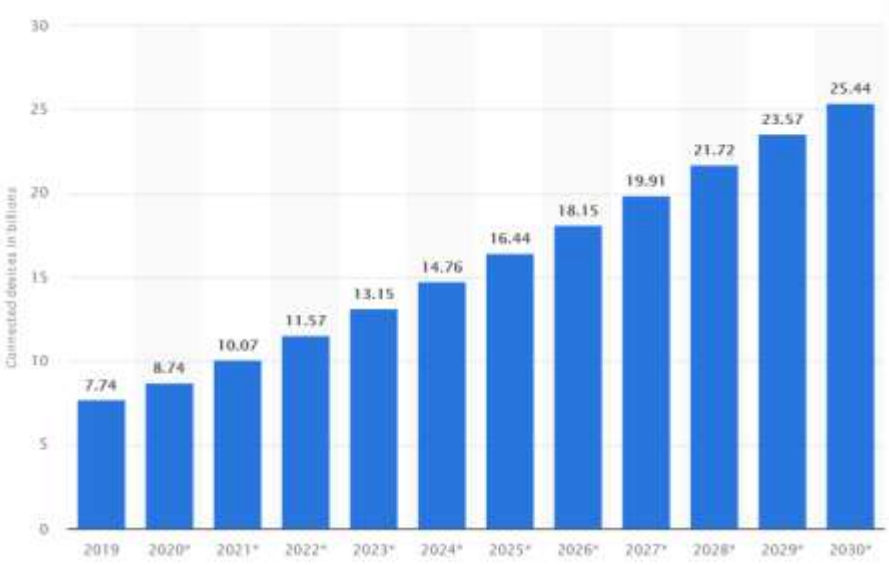


Figura 35: Grafico de crecimiento de dispositivos conectados Statista (2019-2030)[15]

Como se puede observar en el grafico anterior, los datos que aporta, coinciden con el prestigioso informe de Cisco. En este caso, sí que aporta datos válidos para el estudio de escalabilidad del proyecto que se centrara entre los años 2020 y 2026.

Analizando estos años, se estima un crecimiento de los dispositivos conectados del 108 %. Además, este crecimiento también supone una mayor cantidad de tráfico por la característica de interconexión de los nuevos dispositivos. Se estima que ese crecimiento del 108 % aplicado a la infraestructura de la empresa, supondría un crecimiento del 540 % en los eventos que creara la infraestructura del cliente. Para entender mejor estos números se muestra la **tabla 10**:

Estados	EPS
Estado actual 2020	333,33
Estado futuro 2026	1800

Tabla 10. Evolución de EPS (2020-2026)

Como se puede observar, el sistema que se ha implantado escala perfectamente con la estimación que se ha hecho de cara a los próximos 5-6 años.

Teniendo en cuenta que las anteriores gráficas y tablas son estimaciones, aunque bien infundadas, estimaciones, conviene tener preparado y acordado un plan de contingencia con el cliente y que en ese caso de sufrir un crecimiento fuera del control se tomen medidas. En este caso se implantaría otro colector de eventos, puesto que es el dispositivo que más puede peligrar para poder soportar el tráfico de la red.

Como trimestralmente se hará un estudio del crecimiento del tráfico de la red del cliente, esto se podrá prever y solucionar sin mayores problemas. Para apreciar cómo puede crecer la tarificación en esos años, puesto que se usa el modelo OPEX, se muestra el gráfico mediante la **figura 36**:

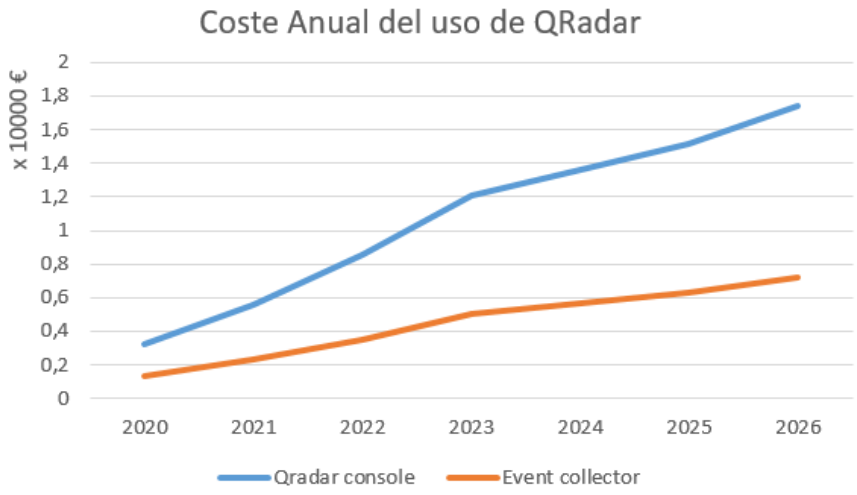


Figura 36: Evolución de coste anual de la solución

Como se puede observar en el grafico el coste del servicio subirá anualmente, por lo que se ha tenido en cuenta a la hora de redactar el contrato con el cliente y fijar la duración del coste del servicio.

Concluyendo, se ha realizado un estudio de la escalabilidad del proyecto y el servicio a futuro que supondrá este mismo, subrayando la robustez de este proyecto en los próximos 5-6 años, reafirmando la falta de un redimensionamiento de la solución. Por lo que quedan concluido el estudio de la escalabilidad del mismo.

7 ANÁLISIS DE RIESGOS

En este apartado se revisarán los posibles riesgos que pueden influenciar tanto en la planificación como en la ejecución del propio proyecto. La identificación de estos riesgos proporciona una seguridad para la gestión de los mismos.

Este apartado se dividirá en 3 subapartados, el primero, se basará en una descripción de cada riesgo que puede influir al proyecto. Posteriormente, mediante una matriz de riesgos en las que se relacionara la probabilidad y el impacto de los mismos, se evaluara cada uno de los riesgos identificados. Finalmente, se detallarán las medidas de contingencia para cada uno de estos riesgos.

7.1 DESCRIPCIÓN DE RIESGOS

Para el análisis, el primer punto es la identificación de los riesgos que pueden influir el proyecto, en este caso se han identificado 3 riesgos principales:

- **A: La enfermedad COVID-19.** Este riesgo atañe a la enfermedad que ha paralizado la economía a nivel mundial. Son múltiples las consecuencias negativas que puede conllevar este riesgo. Por ejemplo, puede que el gobierno confine a la población e incapacite la capacidad de reunirse físicamente o trabajar en la oficina. También puede incurrir en la baja de uno de los integrantes de este proyecto. Cualquiera de estas consecuencias puede acarrear retrasos en el proyecto, así como una gestión ineficiente del mismo.

  Probabilidad: Media

  Impacto: Medio

- **B: Retrasos en la planificación.** Este riesgo está relacionado con los retrasos que pueden conllevar cualquiera de los inconvenientes que puede sufrir el proyecto. Estos retrasos pueden desencadenar en el descontrol del proyecto, así como un sobrecoste del mismo, disminuyendo las ganancias del proyecto. Entre los posibles motivos del retraso se incluyen una mala planificación del proyecto, bajas repentinas de los trabajadores, falta de reuniones de seguimiento, mala praxis del jefe del proyecto… Como se ha indicado antes, todos estos problemas pueden costar retrasos y dinero a la empresa ejecutora del proyecto.

  Probabilidad: alta

  Impacto: medio

- **C: Problemas de interoperabilidad.** En este proyecto, por el carácter propio del proyecto, convivirán equipamientos de capacidad de proveedores o fabricantes. Esta

interoperabilidad en ocasiones da problemas sobre todo cuando se va a tratar los *logs* de estos equipamientos. La falta de interoperabilidad entre equipamientos de fabricantes distintos puede incurrir en retrasos y sobrecostes en el propio proyecto.

Probabilidad: baja

Impacto: grave

Una vez definidos los distintos riesgos del proyecto, se debe evaluar cada uno de ellos de una manera gráfica mediante la matriz de probabilidad-impacto.

7.2 Evaluación de Riesgos

En este apartado se evaluarán los riesgos definidos en el apartado anterior mediante la matriz de probabilidad impacto. En este matriz se segmentarán tanto la probabilidad como el impacto en 5 clases distintas. Además, se usará un código de colores para ver en qué zona de riesgo queda cada riesgo identificado.

La matriz de probabilidad-impacto se muestra en la **tabla 11**:

		Probabilidad				
		Muy baja	Baja	Media	Alta	Muy alta
Impacto	Muy leve					
	Leve					
	Medio			A	B	
	Grave		C			
	Muy grave					

Tabla 11: Matriz de probabilidad-impacto

7.3 Plan de Contingencia

En este apartado se presentarán las distintas medidas que formarán el plan de contingencia para los riesgos definidos en los apartados anteriores.

- **A: La enfermedad COVID-19.** Para no tener problemas con esta enfermedad se comprobará el posible acceso remoto a los recursos necesarios para la configuración, implantación y diseño de la solución. Además, en caso de una baja necesaria de uno de los trabajadores del proyecto se definirán posibles sustitutos para cada uno de los participantes del proyecto. Además, en el trabajo se tomarán las medidas de prevención necesarias para evitar el contagio por la enfermedad del virus.
- **B: Retrasos en la planificación.** Para no ser influenciados por este riesgo se harán reuniones de seguimiento semanales del proyecto en las que mediante el *software* ágil

Trello se evaluara la ejecución del proyecto. También, se definirá un diagrama GANTT con hitos y tareas definidas en el tiempo para el mayor control del mismo En caso de sufrir retrasos inevitables se tomarán 3 directivas. La primera, dedicar más recursos al proyecto sin aumentar mucho el sobrecoste del mismo. El segundo, aumentar las horas diarias que se le dediquen al proyecto promoviéndolo por encima de otros proyectos de la empresa. El tercero, en caso de no poder aplicar los otros dos se hablará con el cliente para poder renegociar los plazos del proyecto.

- **C: Problemas de interoperabilidad.** Este riesgo tiene un componente de aleatoriedad o falta de control por la parte ejecutora del proyecto. Para poder solventar los problemas derivados por este riesgo, se negociará con IBM contractualmente que no se van a dar problemas de este tipo y en caso de que sucedan, se pedirá una compensación económica por parte de IBM. Con esta compensación se podrá hacer frente a los sobrecostes que genere el retraso del proyecto.

8 DISEÑO DE LA SOLUCIÓN

En esta apartado se detallará la solución del despliegue de la herramienta centralizada de IBM, QRadar. Para ello primero se analizará la infraestructura que compone el cliente, con el fin de conocer mejor la situación desde la que se parte en este proyecto.

Posteriormente, se analizará la arquitectura base de QRadar. Aunque ya se ha tratado anteriormente lo que es un SIEM, su arquitectura de despliegue no se ha tratado, puesto que varía mucho de la elección de la propia herramienta especifica SIEM. En este caso, como en el apartado de análisis de alternativas ha salido seleccionado el SIEM de IBM, se analizará esta arquitectura.

Después, se explicará la arquitectura propuesta para la red del cliente. En este apartado se detallará la propuesta realizada para el proyecto, analizando las nuevas implantaciones.

Finalmente, se detallará el diseño del módulo de ML, definiendo la arquitectura de bajo y alto nivel del mismo.

8.1 ARQUITECTURA ACTUAL DEL CLIENTE

En este apartado se analizará **infraestructura del cliente** y **la infraestructura propia**; es decir, la de la empresa ejecutara del proyecto. Esta diferenciación será esencial para el resto del proyecto, puesto que es la nomenclatura que se usará en el documento.

Para entender mejor la interconexión actual del cliente, conviene analizar el esquema de red mostrado en la **figura 37**:

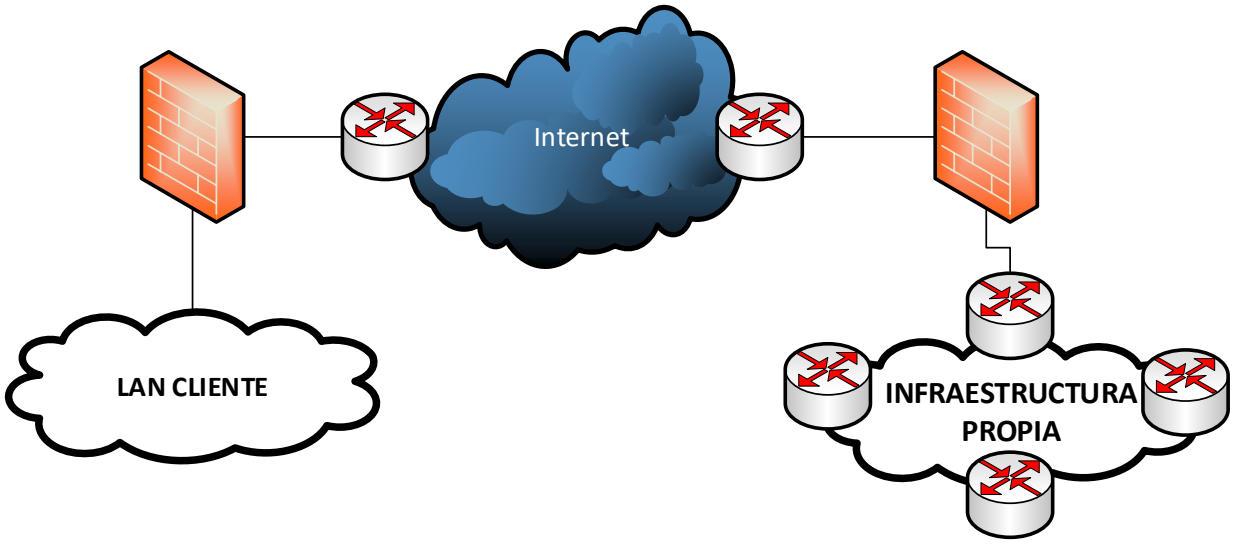


Figura 37: Red de interconexión actual

Como se puede observar, ahora mismo las dos infraestructuras, se sitúan protegidas por un *firewall*, que protege la red interna de cada una de ellas. Ahora mismo, entre el segmento de red de seguridad de la infraestructura propia y la infraestructura del cliente no hay ningún tipo de comunicación directa y segura. Solo se puede acceder a las redes accesibles desde internet; por lo cual, no existe ningún canal seguro ni comunicación con la red interna.

Para ello, posteriormente se explicará la manera de proporcionar un canal seguro entre las dos redes.

También cabe destacar, que la red del cliente no posee ningún colector de eventos que le ayude a monitorizar el estado de la infraestructura, ni tan siquiera internamente. La infraestructura posee una monitorización superficial (estado de los equipos y enlaces) a través del *software* de monitorización NAGIOS.

Los únicos eventos que se analizan son los que recibe el *firewall*. En este caso el *firewall* es del fabricante Checkpoint y posee la funcionalidad Smartevent, que permite analizar los eventos y generar alertas en el *firewall*. Aun así, esto no tiene toda la visibilidad de la red y tampoco monitoriza el estado interno de las redes.

Tampoco se posee ninguna solución de ML para el análisis de los *logs* del tráfico. Toda esta información, permite conocer que es lo que se va a implementar en la red del cliente para conocer mejor la solución. También, es una manera de justificar la ejecución del proyecto, puesto que en este apartado la infraestructura tiene una necesidad de protección.

Analizando la red del cliente en concreto, sí que posee infraestructura dela que se hara uso para el despliegue de la solución. Para visualizar mejor este apartado dela solución, se presenta el esquema mostrado en la **figura 38**:

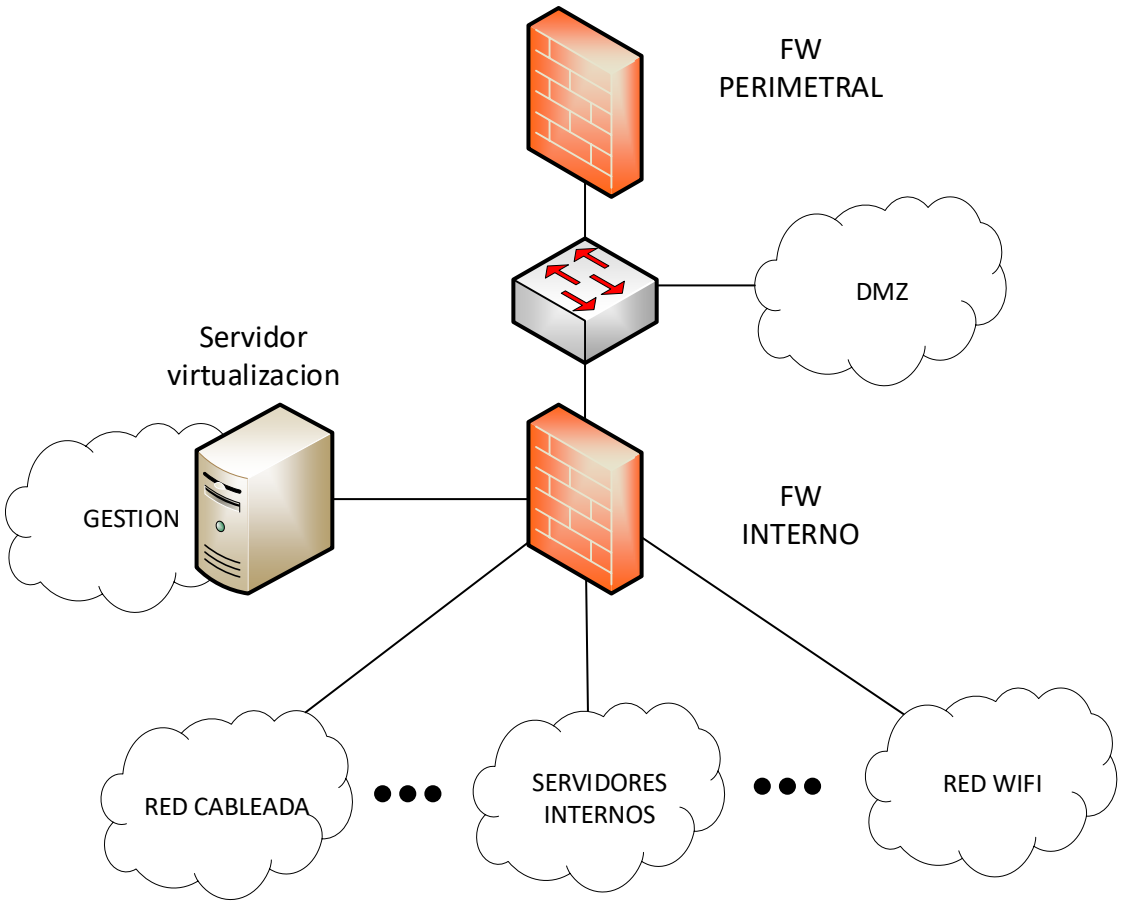


Figura 38: Arquitectura actual del cliente

Analizando el esquema anterior, se puede observar la arquitectura interna del cliente, mostrando la segmentación de redes que posee el cliente (no se han añadido todas por falta de importancia) y la protección de dos niveles que posee. En este caso, como se explicará en los siguientes apartados, se hará uso del servidor de virtualización para desplegar el servidor de eventos en la infraestructura del cliente.

También cabe remarcar que la red del servidor es una red de gestión que tendrá permisos para acceder al resto de la red, lo que ayudará a la hora del despliegue. Para el acceso a los *logs* por parte de la colectora de eventos hará falta comunicación con el resto de redes, por eso se ha añadido esta segmentación en el esquema.

Una vez analizada la infraestructura actual del cliente, es el turno de analizar la parte de la infraestructura propia. Para ello se analizará la infraestructura propia mediante su esquema de red:

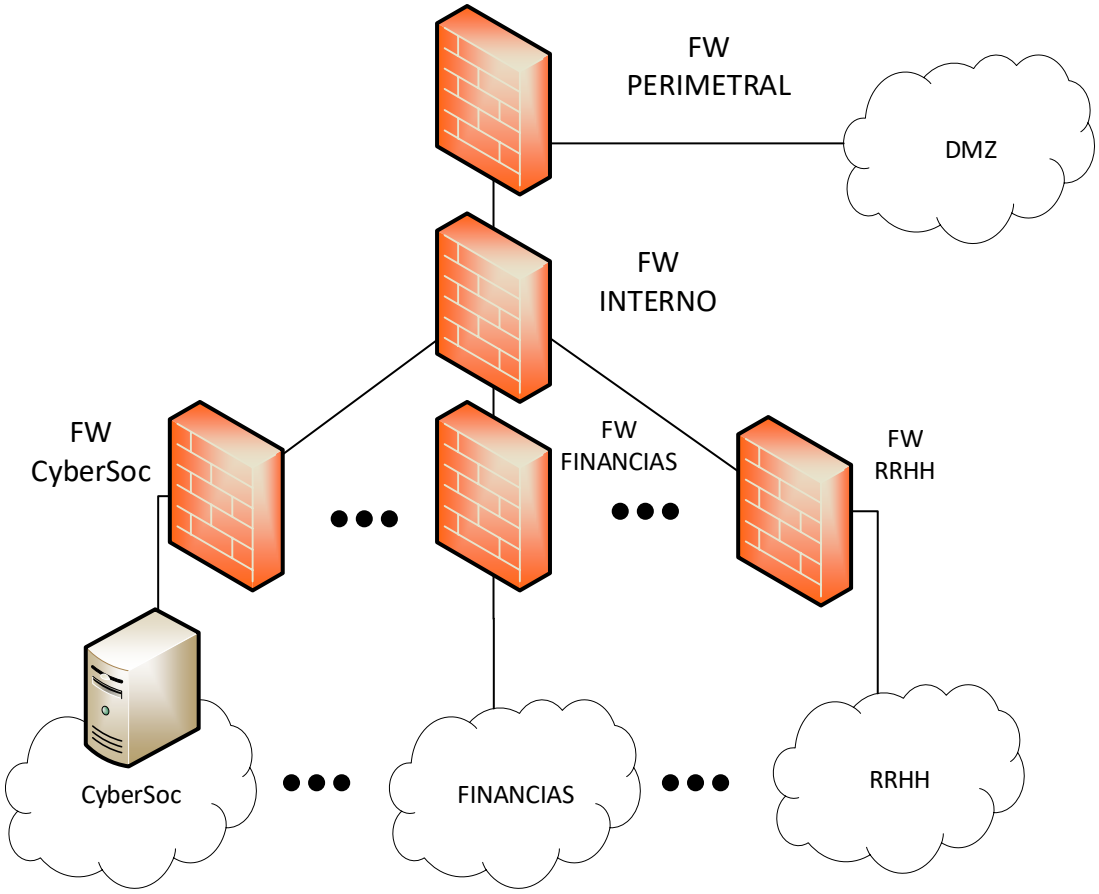


Figura 39: Infraestructura actual propia

Como se puede observar, esta infraestructura es mucho más potente a nivel de seguridad, al ser una empresa que se dedica a ofrecer servicios profesionales de seguridad, redes y comunicaciones. La seguridad se divide en tres niveles, un *firewall* perimetral que gestiona las entradas y salidas del tráfico hacia internet, el *firewall* interno que gestiona la comunicación entre los distintos departamentos de la empresa y, por último, el *firewall* que protege cada sector de la empresa.

Además, la infraestructura posee muchos elementos extras de seguridad que no se han incluido en este esquema por estar fuera del alcance del proyecto. La red que atañe a este proyecto es la del cybersoc, la red que alojara al SIEM y desde la que se monitorizara la red del cliente. Este Cybersoc será un SOC (*Support Operation Center*) dedicado a ciberseguridad. Un SOC es un equipo dedicado al análisis de alertas en una infraestructura a través de un sistema de *ticketing* que relaciona las incidencias con un portal *web* gestionable.

Como en el anterior caso, posee un servidor de virtualización para la integración del SIEM en el mismo que se aprovechara para este proyecto. Con este breve detallado de la infraestructura del cliente y la propia se da por finalizado este apartado.

8.2 ARQUITECTURA BASE DE QRADAR

Primero, cabe describir una arquitectura de IBM QRadar[16] más generalista y luego analizar la topología que se va a construir en el caso de este proyecto en concreto. IBM plantea la siguiente arquitectura de su herramienta en su documentación técnica:

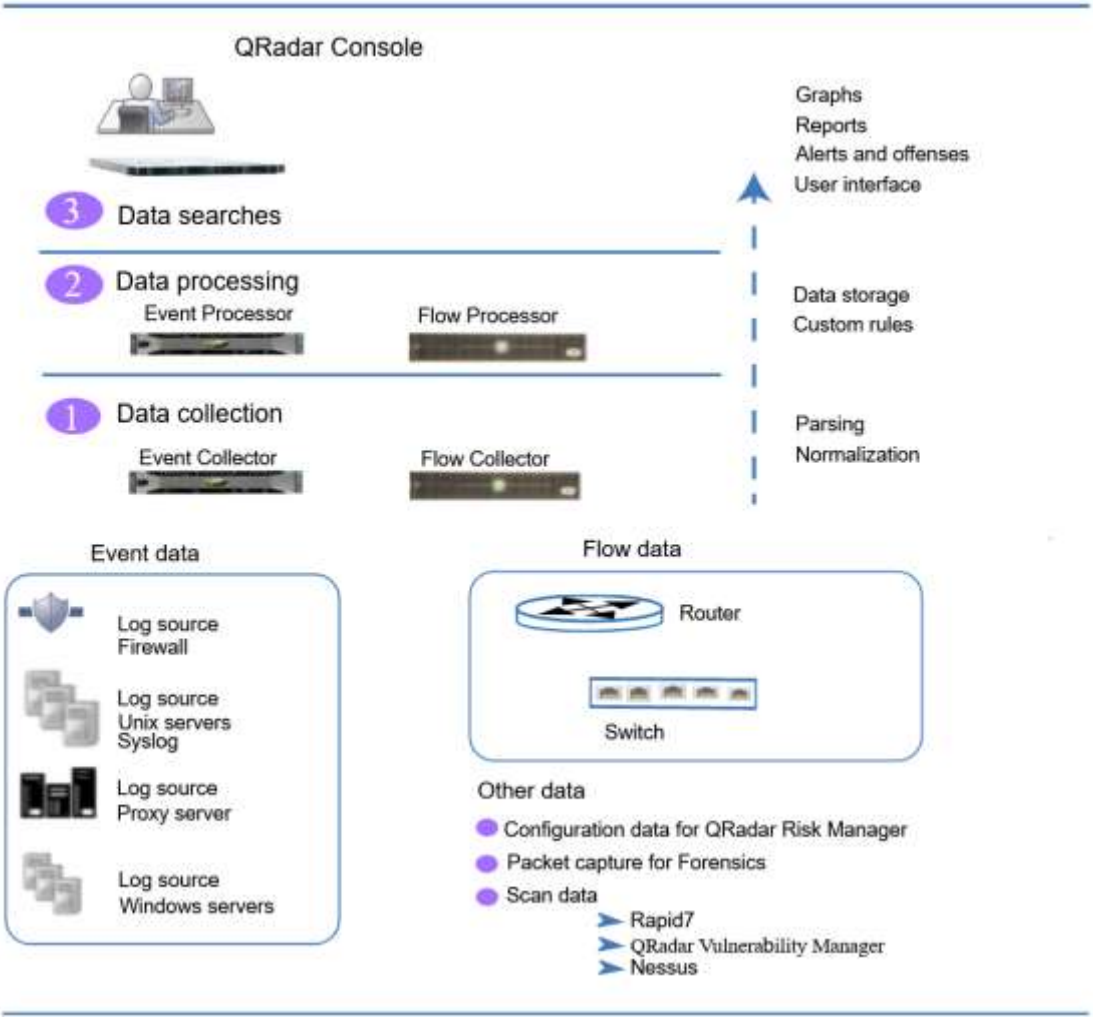


Figura 40: Arquitectura IBM QRadar

Analizando el esquema de la imagen, se observa que hace una segmentación en dos niveles.

El primer nivel es la diferenciación entre eventos y flujos; es decir, si va a colectar solo logs o también va a colectar tráfico de los equipamientos de nivel tres haciendo uso de puertos de *monitor* o *mirroring* que dupliquen el tráfico y se lo envíen a la gestora central de QRadar. En esta solución, no se generará una colección del tráfico, por lo que solo se analizaran los eventos, para evitar un flujo gigante de datos.

El otro nivel que diferencia es el de análisis de los datos que recibe. Lo separa en tres fases: La colección de los datos, su procesado y la búsqueda de amenazas de los datos. Dada la figura, parece que esta búsqueda de tráfico es manual, pero lógicamente es un proceso automatizado y totalmente transparente para el cliente.

Ahora, se precisa explicar estos pasos con más detenimiento, la primera capa, la de colección de los datos, se trata de una colección de datos pertenecientes a la red del propio cliente. En este punto hay dos posibilidades, una solución del tipo all-in-one; es decir, montar el propio SIEM en la red del cliente o; en su defecto, usar una solución centralizada en el que se despliegan colectores de tráfico o de *logs* como el *QRadar Event Collector*.

Estos datos se reciben en este caso en el *collector*, que los parsea y normaliza para enviárselo a la capa de procesado. La función principal del SIEM de QRadar se centra en esta capa, puesto que el preparado de los datos es una de las partes esenciales.

Para entender mejor esta capa, cabe detallar a qué se refiere este texto por eventos. Los eventos son; por ejemplo, *logins* de usuarios, *emails*, conexiones VPN, bloqueos del *firewall* y muchos más eventos a nivel de red o aplicación.

La siguiente capa, sería la capa de procesado. En este punto, se realiza el análisis del trafico colectado en la capa anterior. Para ello se definen una serie de reglas que harán *match* o no con los eventos coleccionados. Al motor que aplica estas reglas se le llama CRE (Custom Rules Engine). Este motor genera alertas y avisos de ofensas y además guarda los eventos en su base de datos, al menos temporalmente (se guardan durante tiempos largos 1-3 meses).

La última capa, se basa en un atractivo *backend* llamado QRadar *dashboard*, en el que se podrán realizar informes, gráficos, análisis y búsqueda de tráfico.

En estas capas se han mencionado varios componentes de la arquitectura como la QRadar *console*, el *event collector* y el *event processor*, la **figura 41** muestra los distintos componentes de la arquitectura y sus funciones:

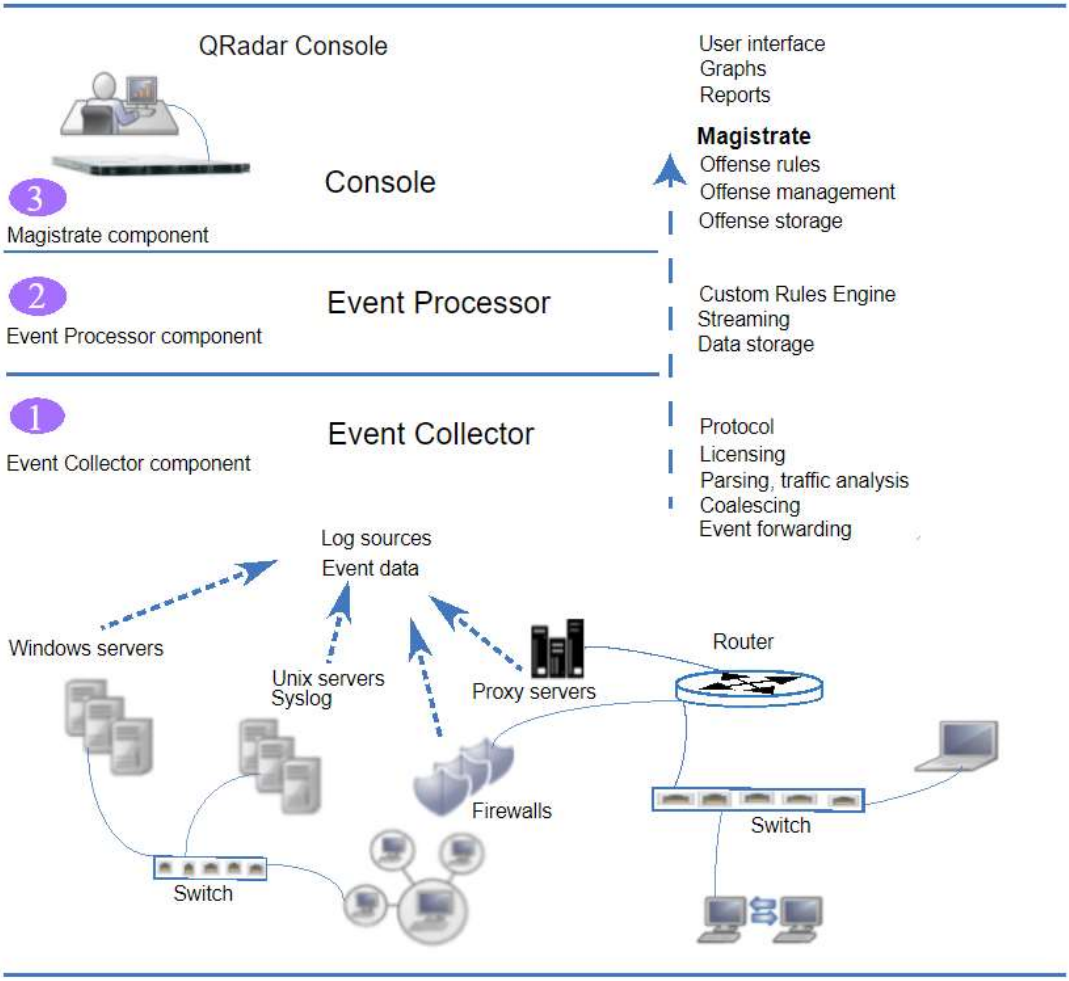


Figura 41. Componentes de la arquitectura SIEM

Los componentes mostrados en la imagen anterior, son los que implementan las capas descritas anteriormente. Es decir, la QRadar *console* implementa búsqueda de datos, generando reglas de ofensa, gestión de ofensas y el almacenamiento de la mismas.

El *event processor*, implementa la capa de procesamiento de eventos y aplica las CRE, almacena los datos que recibe o los almacena en un nodo de datos, para ello usa Ariel, una base datos basada en series de tiempos. También, envía datos a tiempo real a la QRadar *console*.

El *event collector*, como antes se ha dicho, es el encargado de la capa más importante que es la caracterización y envío de datos. Usa el protocolo *syslog*, que se analizara con más detalle en la metodología, para la recepción de los eventos, hace *licenisng checking*, puesto que QRadar permite analizar un numero de eventos por días en base a la licencia que se haya comprado, coge los eventos y los parsea a un formato más legible para QRadar, normalmente LEEF (Log Event Extended Format). También analiza el tráfico y lo normaliza, pudiendo así fusionar tráfico

que esta correlacionado; además, envía los eventos a otros sistemas de *syslog* y otros sistemas SIEMs.

También hay nodos opcionales que ayudan como el *data node* mencionado con anterioridad, que ayuda al *event processor* a almacenar los eventos para mantener más ligero al mismo. También está el QRadar *App Host*, que trata de quitarle capacidad de procesado a la QRadar *console*, mostrando un entorno dedicado para aplicaciones específicas que requieren de muchas capacidades como *machine learning* o *user behaviour analytics*.

Con estas descripciones la arquitectura base queda completamente desgranada, ahora conviene poner el foco en la topología de despliegue en la red del cliente; es decir, ir a un caso más práctico.

8.3 ARQUITECTURA QRADAR EN CLIENTE

En este apartado se dará un análisis detallado de la arquitectura exacta que se va a desplegar en el cliente. Para ello se analizará que será necesario desplegar o implementar, referenciando a la parte de metodología donde se detallará cómo realizar o implementar este despliegue más concretamente.

En primera instancia se pretende mostrar la arquitectura o topología de red del despliegue para poder entender mejor la solución y el diseño de la misma. Esta topología se muestra en la **figura 42**:

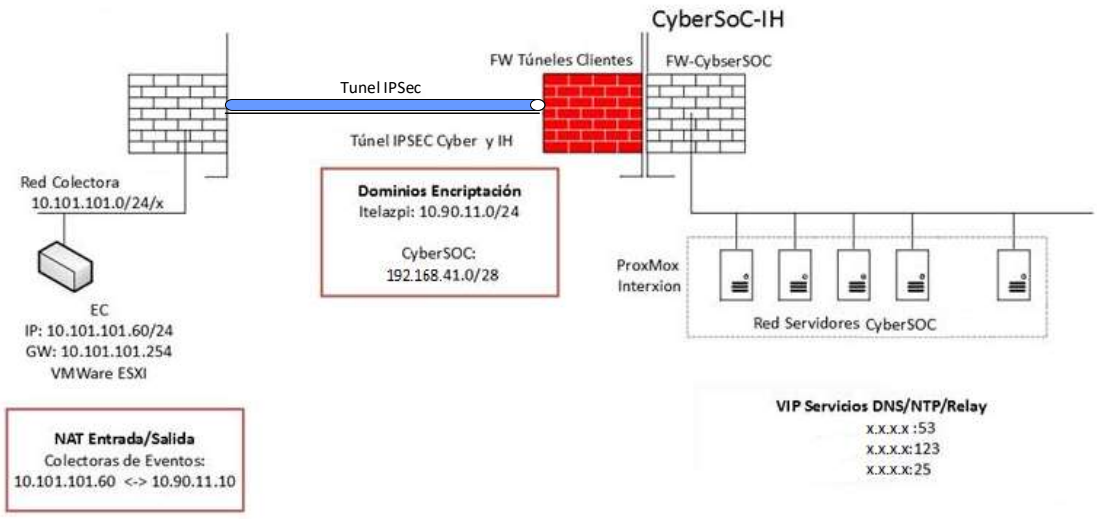


Figura 42: Topología del despliegue SIEM en cliente

Como se puede observar, la topología del cliente será muy sencilla, aprovechando su infraestructura, se configurará el *firewall* perimetral del cliente y de la empresa para construir un túnel VPN entre ellos, como se detallará en la topología.

Una vez construido el túnel de IPSec, se tendrá un canal seguro entre el cliente y la empresa para poder enviar los eventos nombrados en el apartado anterior. Para ello, habrá que desplegar un IBM *event collector*, el identificado como EC en el esquema. Para evitar lanzar rutas a través de sistemas OSPF o BGP a través del túnel, se hará un NAT a un dominio de encriptación específico para poder acceder a ello.

La máquina virtual va sobre VMWare ESXi y se explicara en el apartado de metodología como se ha hecho el *deployment*. Por lo que el flujo de tráfico sería el detallado en la **figura 43**:

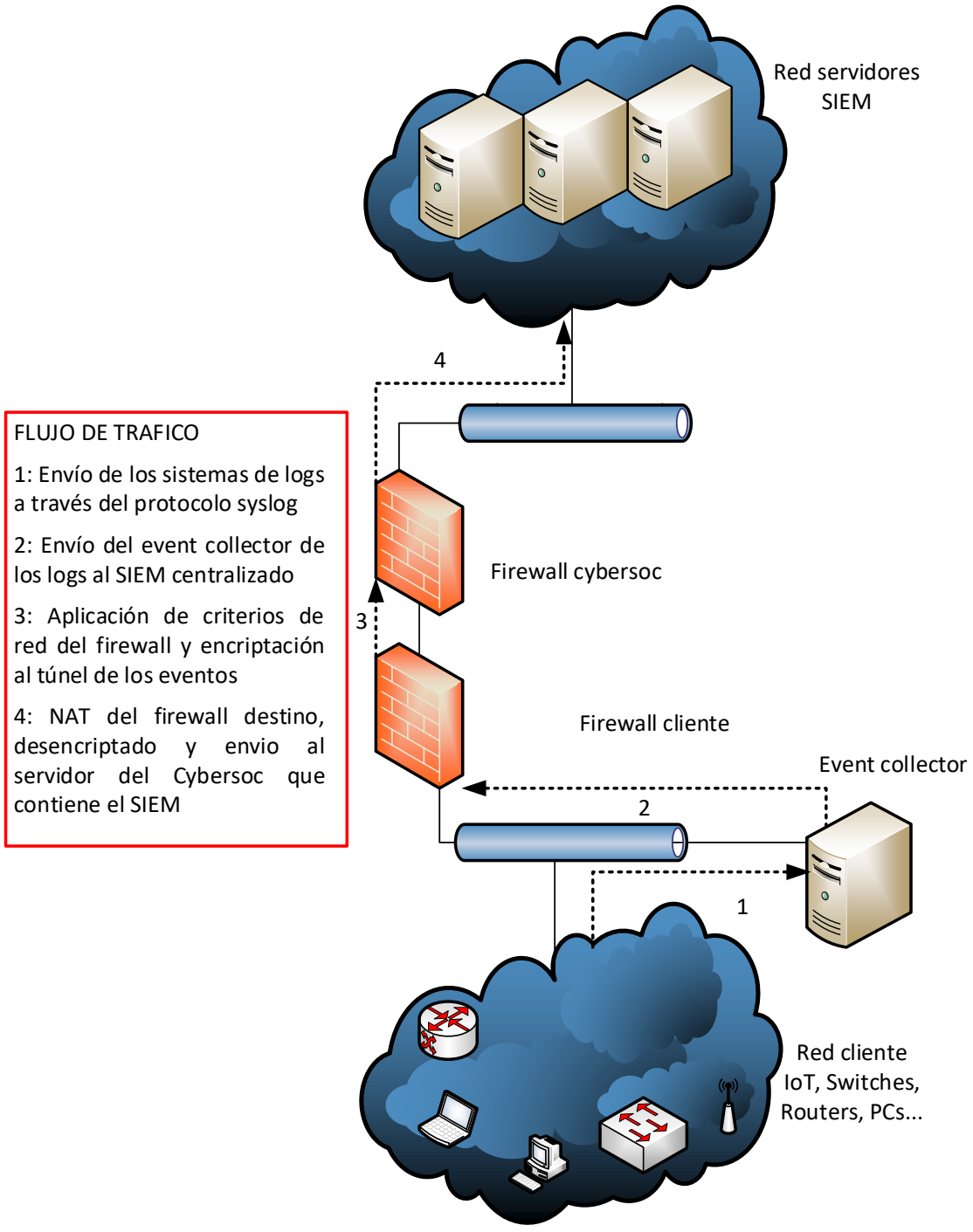


Figura 43: Flujo de tráfico entre equipos del cliente y el SIEM

La propia figura anterior describe perfectamente el flujo con la tabla textual que contiene en el margen izquierdo. Este despliegue, permite a su vez tener una arquitectura que podrá tener una gran escalabilidad al tener los colectores de eventos alojados en las redes de los

clientes y los procesadores de eventos y la consola alojados en la red corporativa de la parte ejecutora del proyecto.

En este esquema, se ha obviado el rol de los colectores de eventos del host como por ejemplo *sysmon* o *wincollect*. Estas máquinas guardan los *logs* de las maquinas *host* de la infraestructura cliente y se lo envían al servidor de *event collector* de QRadar.

Para esta recopilación de eventos, es necesario instalar *software* en el cliente o en el *host* relativo al *client-side* de los *softwares* mencionados con anterioridad. Todo este despliegue se detallará con mayor detalle en el apartado de metodología.

Aun así, también se considera necesario evaluar y representar el diseño del despliegue de estas máquinas y el transcurso que siguen los *logs* a través de la red hasta llegar al nodo *event collector*. Esto se puede observar a través del diagrama mostrado en la **figura 44**:

Figura 44: Diagrama de colección de logs

En este caso, se usará la funcionalidad de Windows *sysmon* para poder enviar los eventos de cada *host* a un Windows Server que tenga instalado el *software* propietario de Windows Wincollect. Este programa recabara todos los *logs* de los *hosts* y se los enviara a QRadar *event collector* haciendo uso del protocolo *syslog*. Este diagrama más funcional es el que explica de forma conjunta el paquete de *software* necesario de cada elemento de la red. Aun así, cabe detallar a nivel comunicativo como se realizaría este envío de *logs*. En el diagrama de la **figura 45** se puede observar la comunicación de una manera más precisa:

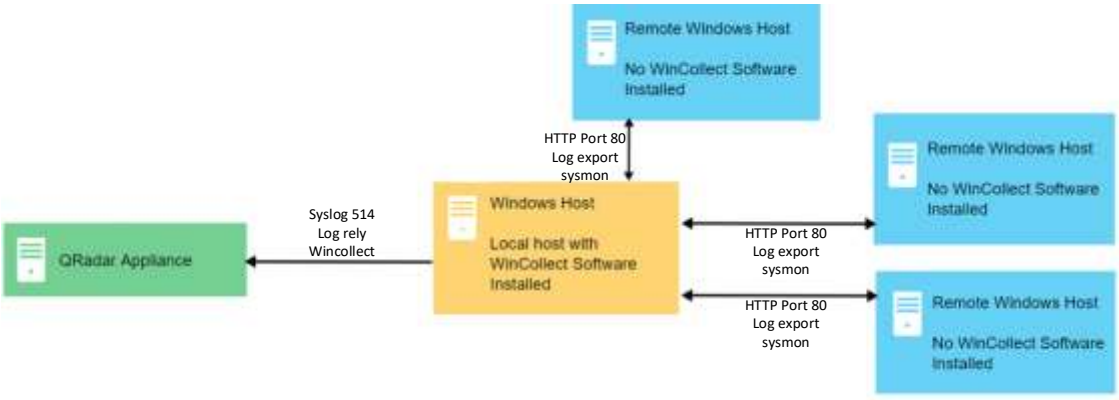


Figura 45: Reenvió de logs en windows

Como se puede observar la comunicación entre los *hosts* y el servidor Wincollect se hace a través del puerto 80, haciendo uso del protocolo HTTP. Después el servidor reenvía sus datos al *event collector* de IBM.

Por último, en este apartado se ha obviado cómo se hace el reenvío de *logs* en sistemas que no poseen sistema operativo Windows; es decir, en los sistemas Linux (MacOS no se encuentra en la red del cliente). Los sistemas Linux tienen la capacidad de reenviar por si mismos sus eventos haciendo uso de la dependencia *rsyslog*. La comunicación quedaría más simple que la anterior, como muestra el diagrama mostrado en la **figura 46**:

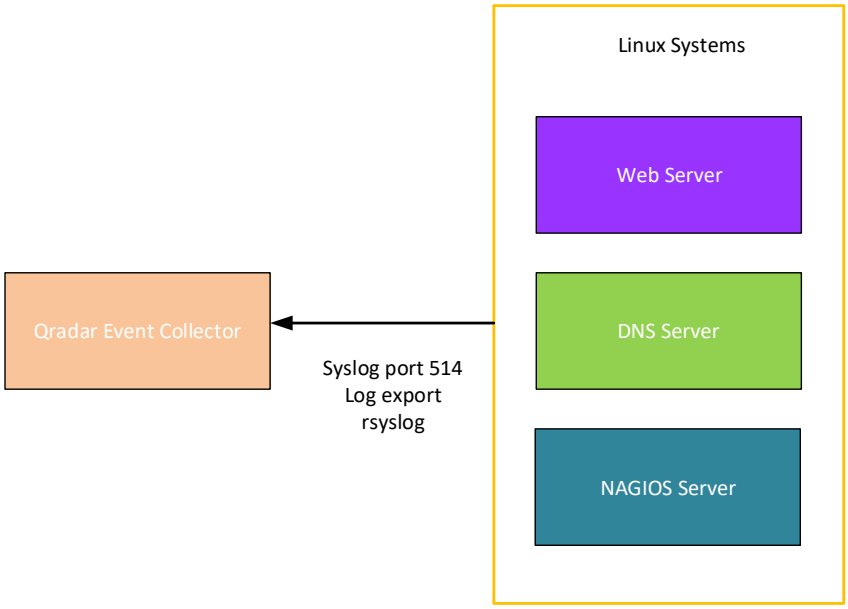


Figura 46. Envío de logs sistemas Linux

Una vez clarificado el diseño del entorno IBM y pautado su despliegue se puede dar por finalizada el apartado del diseño de QRadar.

8.4 DISEÑO DEL MÓDULO DE MACHINE LEARNING

En este apartado se detallará el diseño de alto nivel del módulo de *Machine Learning*, para ello se detallará un procedimiento para el desarrollo del mismo, marcando las pautas a seguir para el correcto funcionamiento del módulo.

Pese a haber trazado un *background* sobre *Machine Learning* en los apartados de Contexto y diseño de alternativas, cabe recalcar la vital importancia de cada fase de un desarrollo de ML. Para ello se presentará el diagrama de la **figura 47** que aportará una visión más grafica de las fases del subproyecto de ML:

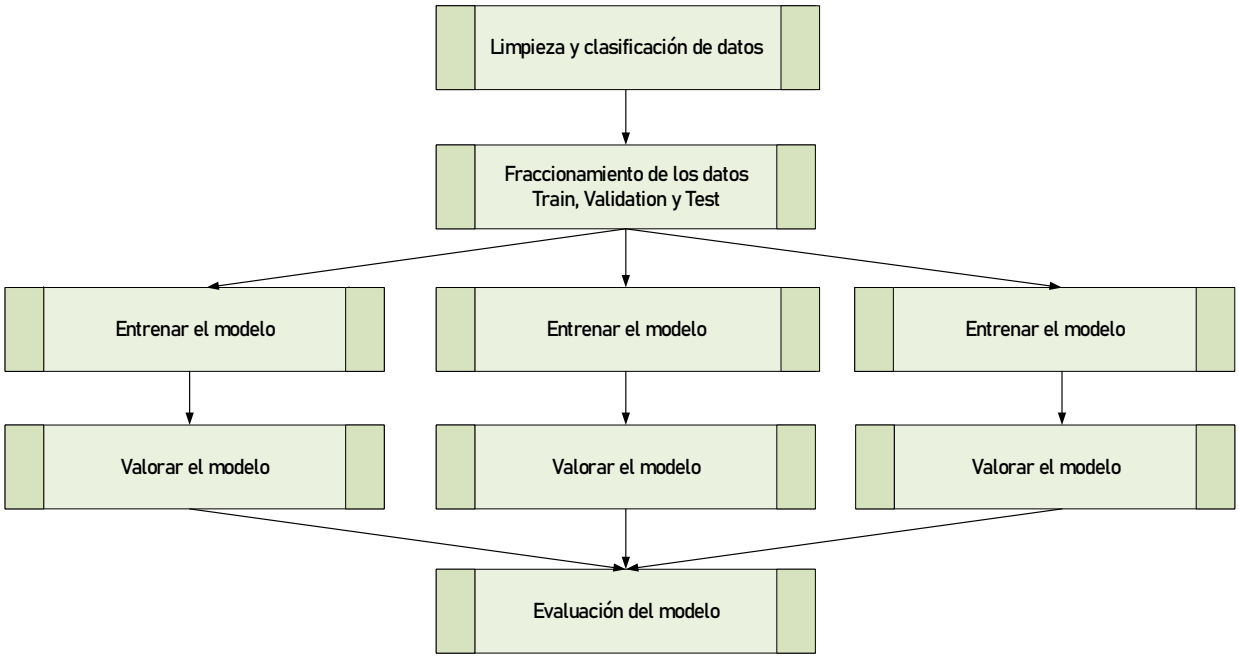


Figura 47: Diagrama del proceso de la solución Machine learning

El diagrama mostrado, resume los pasos que se darán a la hora de desarrollar el módulo de ML, ahora se detallará cada paso como se llevará a cabo para el caso de uso concreto de esta solución.

1. **Limpieza y clasificación de los datos:** Se conseguirán los datos de los *logs* de los registros del *event processor*, se limpiarán quitando los falsos positivos e información no importante. Los datos se clasificarán en los distintos tipos de vulnerabilidades que identifica QRadar.
2. **Fraccionamiento de los datos:** Se fraccionarán los datos en 3 grupos, ya explicados en el contexto, de *train, test y validation*.
3. **Entrenamiento del modelo:** Una vez fraccionados los datos, se usarán los datos de *train* para entrenar cada modelo que se haya considerado, en este caso, los mencionados en el apartado de alternativas de diseño.

4. **Valoración del modelo:** En este apartado se desarrollará código para sacar gráficas y estadísticas que diagnostiquen el modelo y su calidad, dichas estadísticas son las que se han tenido en cuenta para la elección del modelo en las alternativas de diseño.
5. **Evaluación del modelo:** En este apartado, se compararán las estadísticas de cada modelo visto en el apartado anterior para poder lograr una mayor comprensión de la elección del modelo final.

Una vez aclarada la estructura del proceso de ML, cabe recalcar que el modelo elegido finalmente, se entrenara con más datos aun de los que ya lo entrenaron en el apartado anterior. Como esta parte de "re-entrenamiento" es muy repetitiva y no aporta gran contenido a este trabajo se obviará en la memoria.

Con el detallado del proceso de desarrollo de la herramienta o módulo de ML, también se considera oportuno mostrar la arquitectura funcional de este módulo. Es decir, cómo se va a pautar el proceso anterior a nivel de uso de *software*, recursos y máquinas.

Para ello, se aportará un diagrama que logre representar gráficamente el funcionamiento, este se muestra a continuación mediante la **figura 48**:

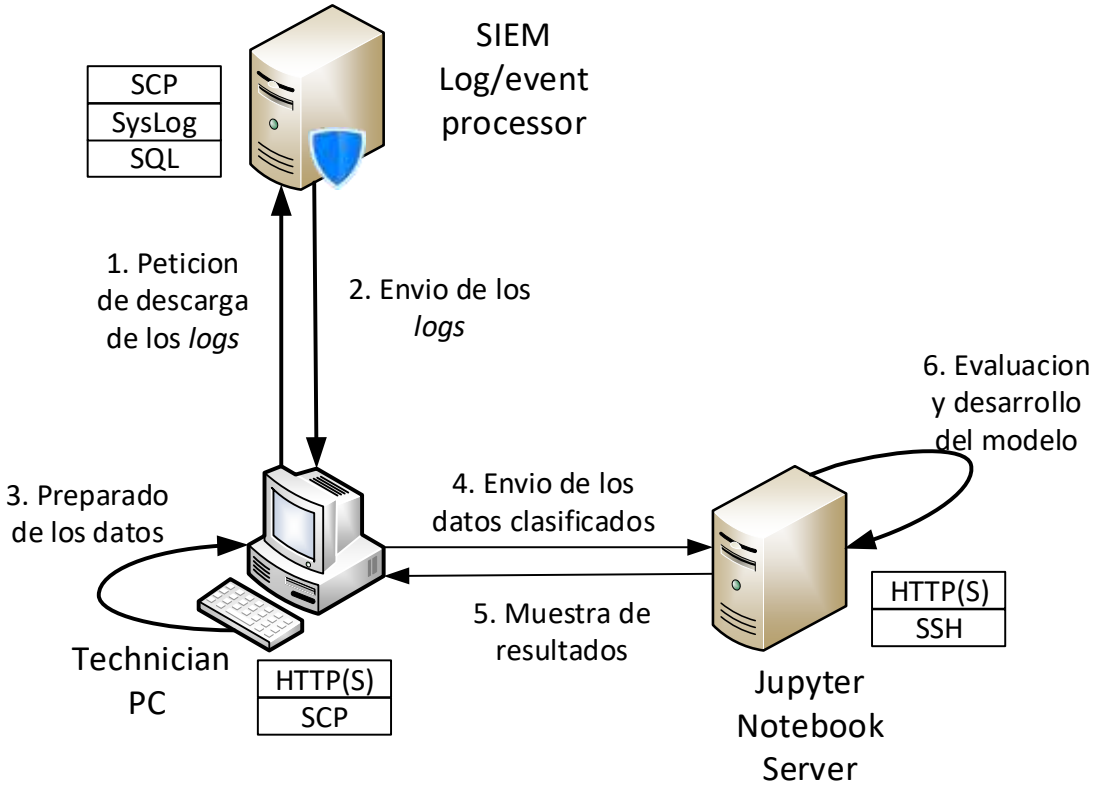


Figura 48: Arquitectura funcional del desarrollo del módulo ML

El anterior esquema de red, pretende detallar cómo será la arquitectura funcional del desarrollo del módulo de ML. Hasta este punto, no se había hablado de cómo se iba a implementar la solución a nivel de *hardware* y *software*, por lo que partiendo de este diagrama se debe detallar la estructura del desarrollo en este aspecto.

Primero, se hará uso de un ordenador corporativo para la conexión con los dos servidores; es decir, el servidor SIEM que contiene el procesador de eventos y el *data node* que guarda los eventos que ha procesado. Para la conexión con este servidor se usará el clásico protocolo SCP (Secure Copy Protocol) para la transferencia de archivos de manera segura, para la conexión con el servidor se usará el cliente de Windows WinSCP. Este protocolo actúa sobre una conexión ssh, por lo que hace uso del puerto 22. Los detalles sobre el uso de la herramienta se detallarán en la metodología.

El ordenador corporativo también hará uso de un navegador web por el que se conectará al servidor que tenga el *software* relativo al desarrollo del módulo de *Machine Learning*. En este caso el servidor es un servidor con *Jupyter Notebook* que escucha al tráfico HTTPS (Hypertext Transfer Protocol Secure) a través del puerto 8888.

Este servidor, contendrá todo el *software* que incluye Anaconda, *software* para el desarrollo de en Python; en concreto, se hará uso de Jupyter Notebook. Una implementación que proporciona un formato web al uso del lenguaje de programación Python. El uso de esta herramienta no ha estado sujeto a debate, puesto que es la clásica herramienta para proyectos de ciberseguridad relacionados con ML.

Además, externalizar este *software* a un servidor y no tenerlo en la maquina *host*, ayudará a un mejor procesado de los datos, fomentando una mayor velocidad de procesamiento.

9 METODOLOGÍA

En este apartado se añadirán conceptos y procedimientos relacionados con la implementación de la solución. Esta implementación se fragmentará en dos subapartados principales, como ya ha sucedido en el resto de apartados. Por un lado, se analizará la implantación del SIEM de IBM, QRadar y por el otro, se describirá el desarrollo de la herramienta o del módulo de ML.

En el apartado de la implantación del SIEM, se detallarán los procedimientos realizados durante el despliegue de la solución relativos a la colección de *logs*, el despliegue de la colectora de eventos en la infraestructura del cliente, el afinamiento de las políticas y la gestión de los informes. También se hará un breve análisis del *dashboard* que verá el cliente, donde podrá comprobar el estado de su infraestructura y las ofensas que ha recibido.

En el módulo de ML, se detallarán los pasos que se han codificado con el fin de llegar a una comprensión más sencilla de la solución.

9.1 IMPLEMENTACIÓN IBM QRADAR

Como se ha mencionado con anterioridad, en este apartado se detallará la implantación de la solución SIEM en la infraestructura del cliente. Para ello primero se detallará el despliegue de la colectora de eventos en la infraestructura del cliente.

9.1.1 Despliegue de la maquina IBM *event collector*

Lo primero en un despliegue de una solución SIEM es el despliegue de la máquina que colectara los *logs* de las máquinas de la infraestructura del cliente. Para ello, lo primero es analizar la infraestructura que posee el cliente para poder aplicar el tipo de máquina que más se ajuste a su entorno.

En este caso, el cliente posee un servidor Huawei con la plataforma de virtualización VMware ESXI en la versión 6.5.0.33, esta versión solo posee la solución del cliente ligero o cliente *web*. Una vez conocida la plataforma sobre la que se va a implementar, cabe saber los requerimientos mínimos de la máquina que se va a desplegar que se incluyen a continuación:

- Memoria RAM: 16 GB
- CPU: 16 núcleos
- Disco: 256 GB
- Red: una interfaz

Como se puede observar es una máquina que necesita bastantes recursos por la cantidad de datos que recibirá. También es analizable que se ha optado por dotar la máquina virtual de una sola interfaz de red. Esta opción combina la gestión y el flujo de los datos por la misma

interfaz de red, la elección ha sido en base a la operativa habitual del cliente con su servidor de virtualización.

Una vez aclarados los requerimientos de la máquina, en el caso del *event collector* de QRadar la implementación se hace en base a una imagen de tipo ISO, por lo que primero habrá que crear la máquina virtual y luego, insertar la imagen ISO desde un *datastore* del servidor virtual. El *datastore* es un almacén de datos del servidor de virtualización que se suele usar para guardar las máquinas virtuales y las imágenes ISO.

Primero se hará la creación de la máquina, para ello se elegirá el nodo en cuestión sobre el que se creará la máquina y se seleccionará la opción de nueva máquina virtual. Una vez seleccionada esta opción se mostrará un menú como el de la **figura 49**:

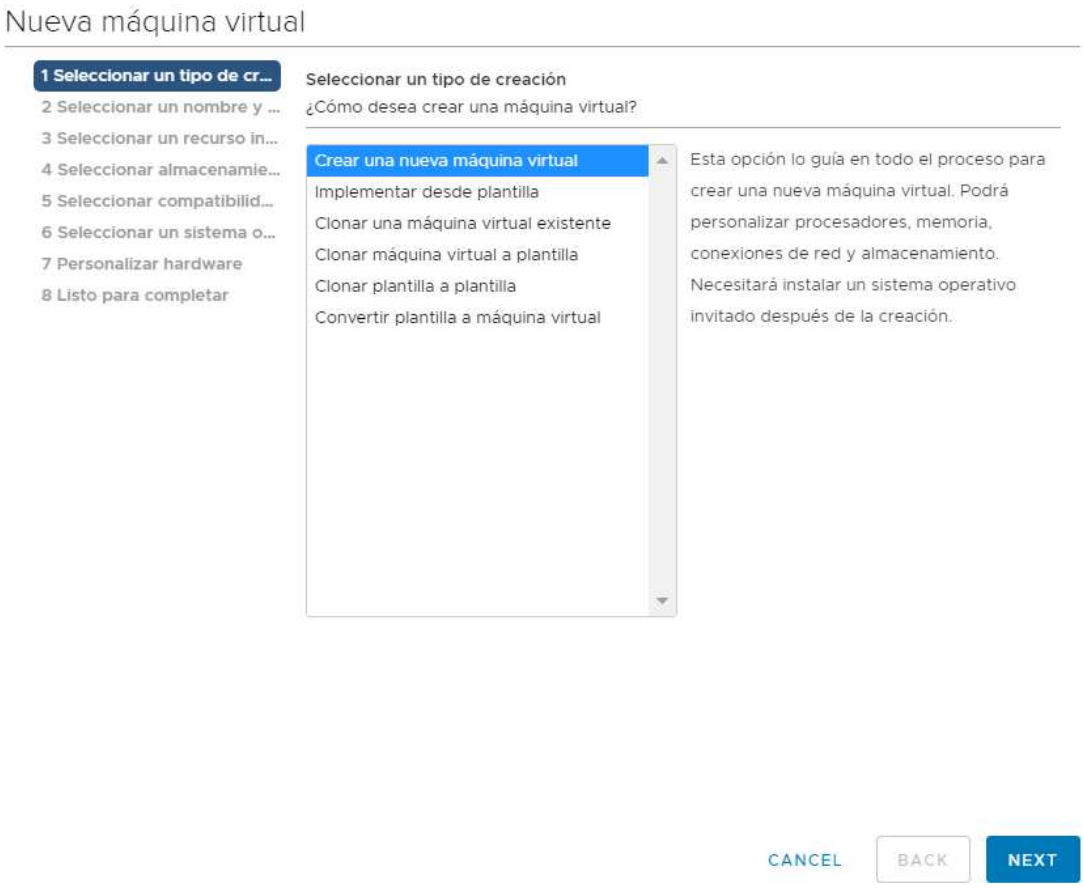


Figura 49: Menú creación nueva máquina virtual I

Se elegirá la primera opción, puesto que como se ha comentado antes se creará la maquina vacía y luego se insertará la imagen ISO. El dialogo mostrado por la **figura 50** será el correspondiente a asignar el nombre de la máquina virtual y su ubicación:

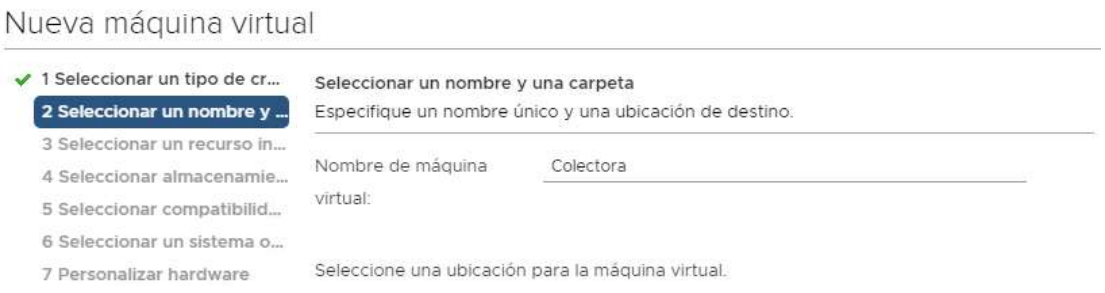


Figura 50: Menú creación máquina virtual II

Una vez asignado el nombre y la ubicación (no se ha mostrado en la imagen por motivo de confidencialidad) posteriormente se asignará uno de los servidores en el que se instalará la máquina virtual, como muestra la **figura 51**:

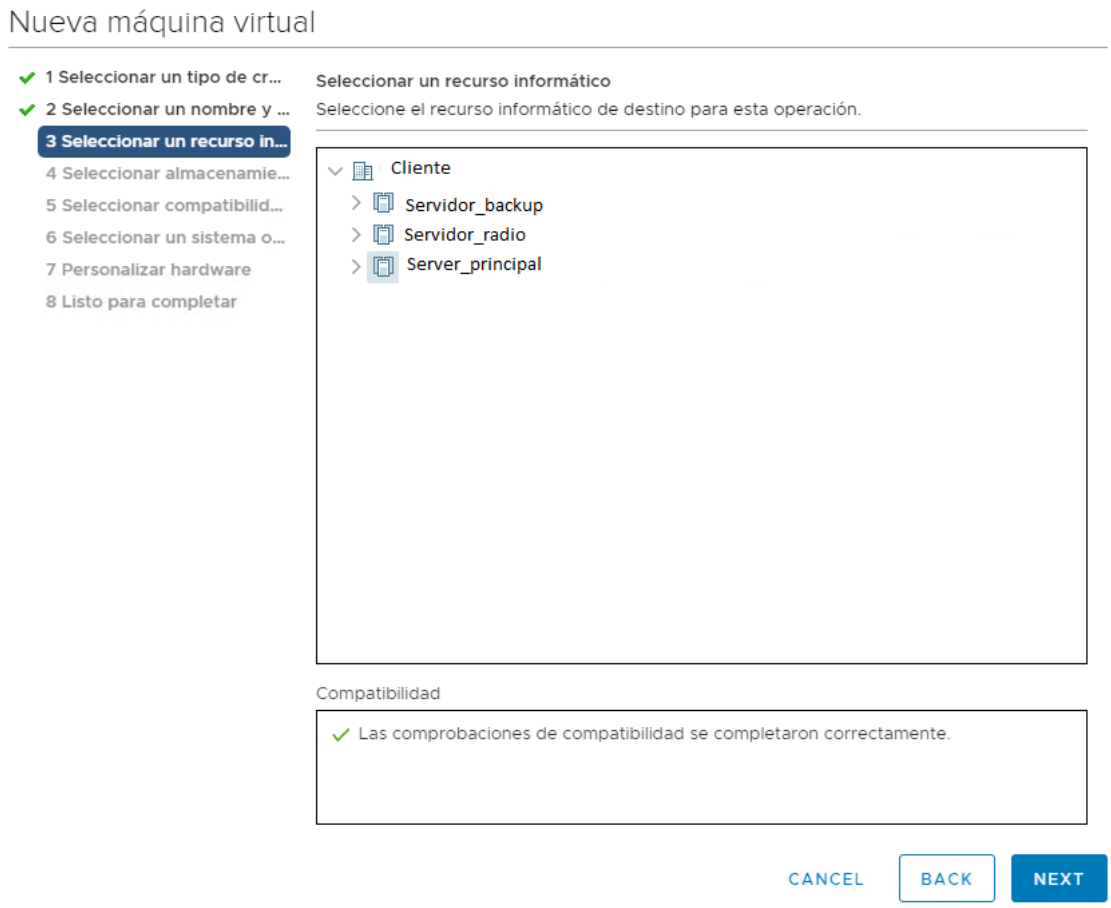


Figura 51: Menú creación máquina virtual III

Posteriormente, se asignará el disco donde se almacenarán los datos de la máquina virtual de la manera que muestra la **figura 52**:

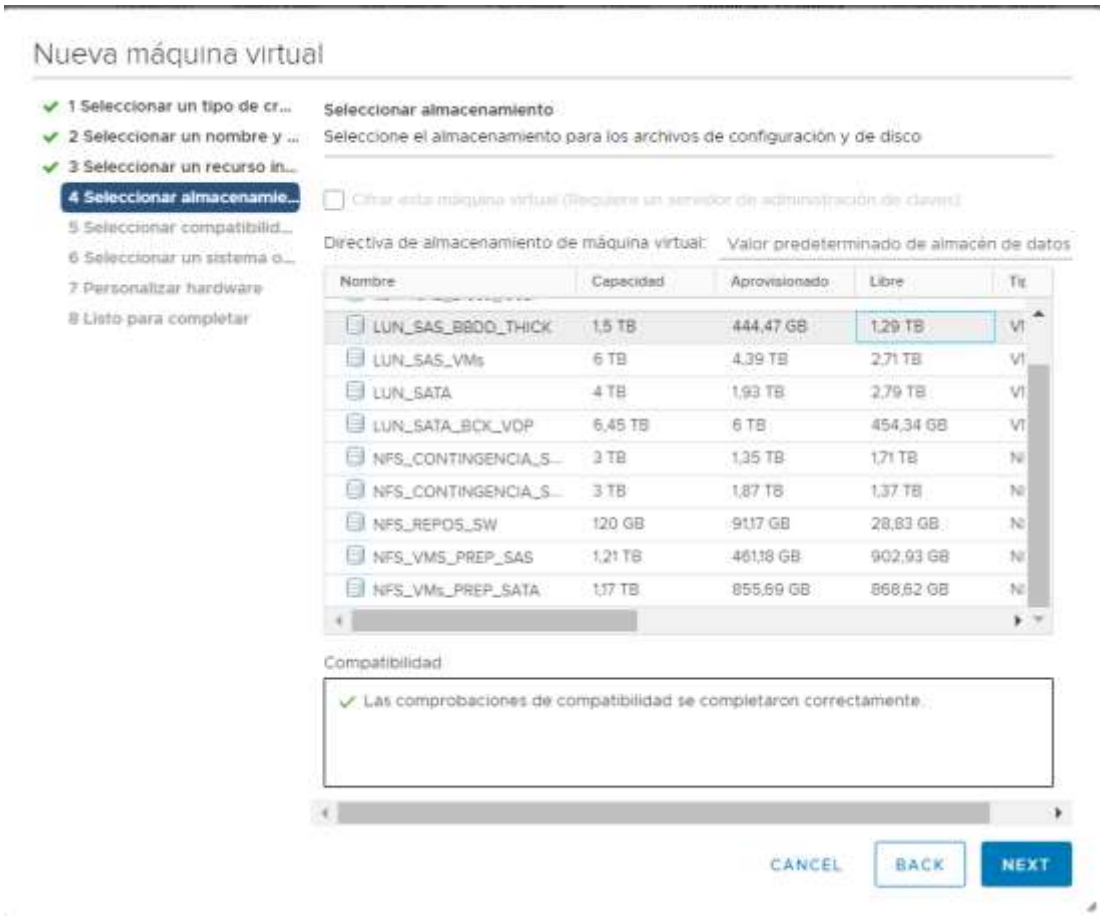


Figura 52: Menú creación de la máquina virtual IV

Como se ha mencionado antes habrá que elegir un disco con suficiente espacio para alojar la máquina. Lo siguiente será elegir el sistema operativo de la máquina que se va a implementar es este caso es una RHEL 7 (Red Hat Enterprise Linux), puesto que se elegirá la opción mostrada en la **figura 53**:

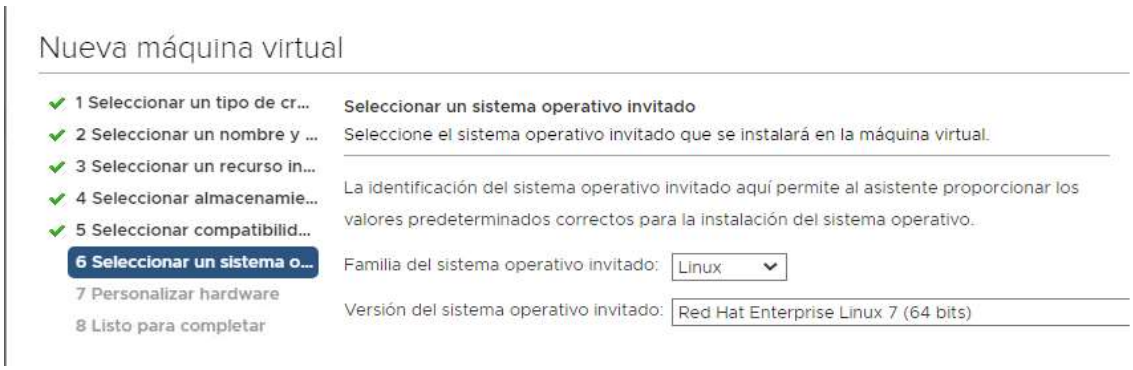


Figura 53: Menú creación de máquina virtual V

Lo siguiente será personalizar el *hardware* de la máquina virtual, de la manera que muestra la **figura 54**:

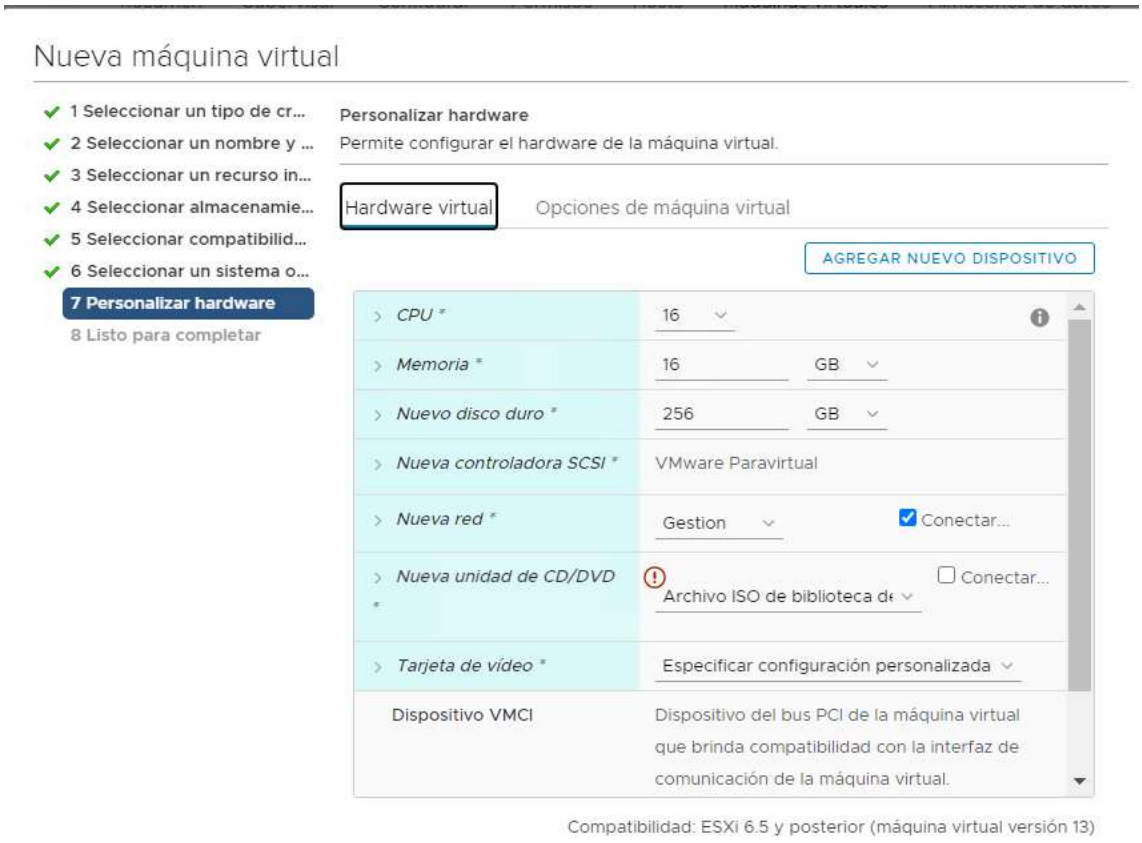


Figura 54: Menú creación de máquina virtual VI

Como se puede observar se han añadido los requerimientos que pedía la máquina para su correcto funcionamiento, la imagen ISO no se ha conectado aún, puesto que se hará al iniciar la máquina. Después de este paso se creará la máquina y estará lista para insertarle la imagen. Cabe mencionar que después de arrancar la imagen el menú de instalación comprueba si la maquina tiene los requerimientos necesarios y en caso afirmativo, se auto instala en un proceso que dura unas 2 horas aproximadamente.

Durante este proceso, se dan una cantidad de instalación de dependencias necesarias para el correcto funcionamiento del sistema, por eso es un proceso bastante largo en el tiempo.

Después tocaría instalar las dependencias necesarias usando el CLI (*Command Line interface*), estas dependencias son las relativas al *software* propietario de IBM. Durante este proceso, también se asignarán características al sistema como su *hostname*, usuario y contraseña y dirección IPv4.

Este proceso de configuración inicial se define en los apartados de los anexos que se incluyen al final del documento.

9.1.2 Integración de la colectora en QRadar console

Al finalizar este *setup*, se integrará esta colectora o sonda en la consola central de la infraestructura propia. Para ello se accederá a la consola central y se entrará como administrador de la manera que muestra la **figura 55**:

Figura 55: QRadar console panel de administrador

Seleccionando la opción *admin* se entrará al panel de configuración que da opciones de administrar la consola. Una vez dentro del menú de administrador, se mostrará el menú de la **figura 56**:

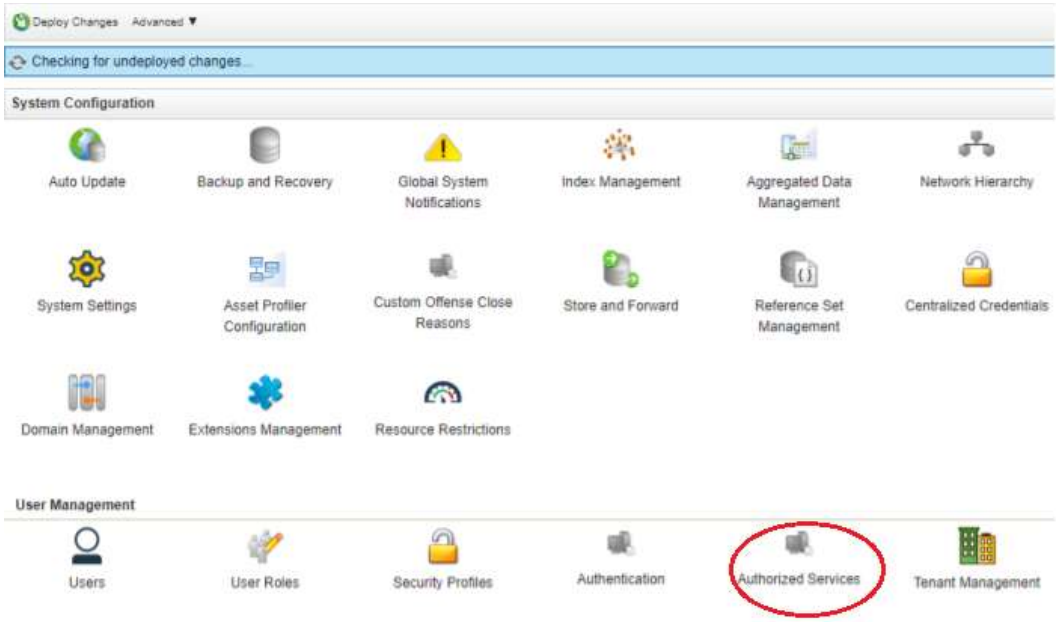


Figura 56. Menú de administración

Bajo *Authorized Services* se definirá la fuente de *logs* que se ha desplegado en la infraestructura del cliente. A esta fuente de *logs* en el argot de ingeniería de sistemas informáticos se le denomina sonda. Esta sonda se agregará de la manera que muestra la **figura 57**:

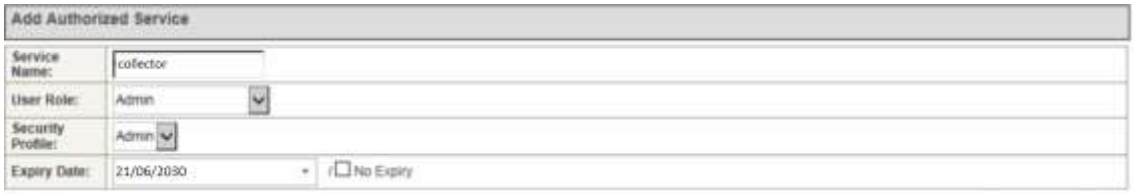


Figura 57: Agregar el servicio de la sonda a la consola central

Lo siguiente será, bajo el menú de *data sources*, definir la sonda y su acceso, para ello primero se accede al menú de *log sources* de la manera que muestra la **figura 58**:

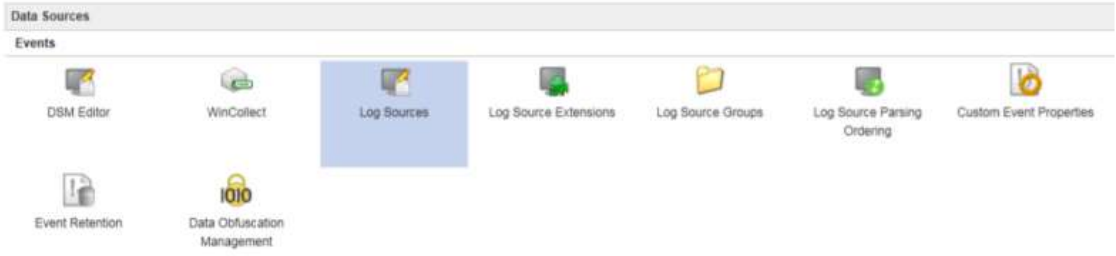


Figura 58: Menú data sources

Bajo este menú, se añadirá una fuente de *logs* con las características que muestra la **figura 59**:

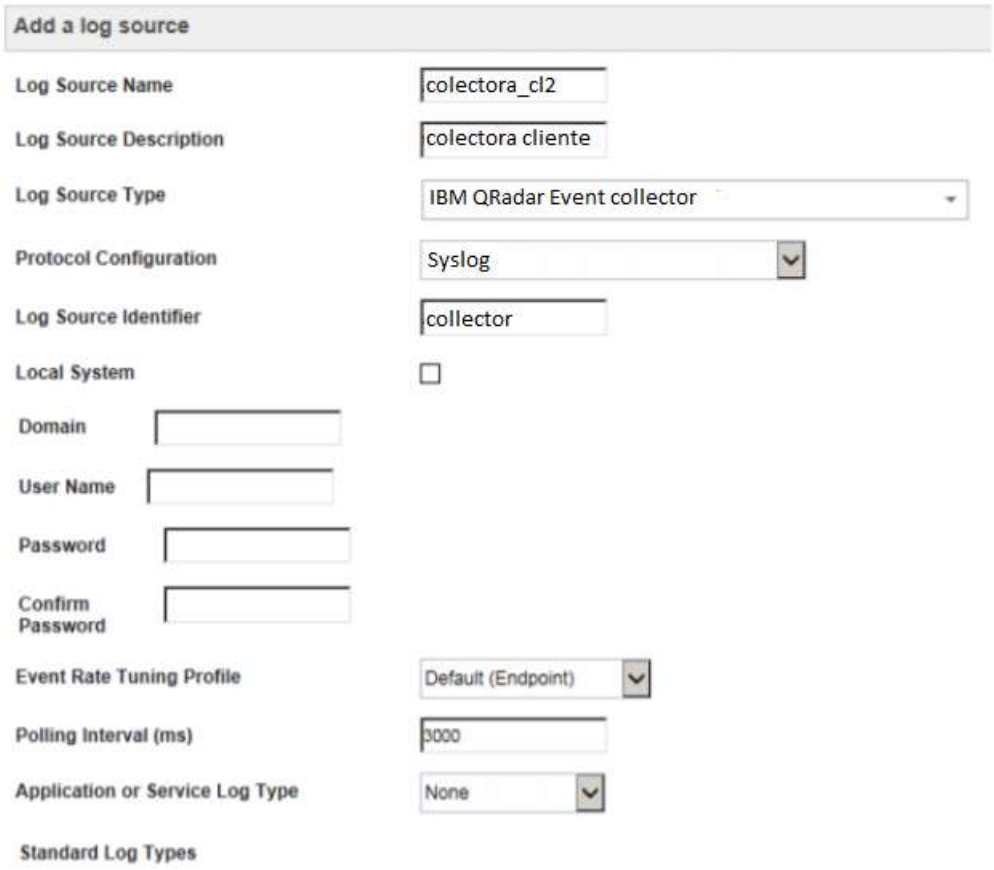


Figura 59. Añadido de la sonda QRadar

Los campos más importantes serían los siguientes:

- **Log Source Type**: Aquí se debe de elegir el tipo de fuente que es, en esto caso *IBM QRadar Event collector*.
- **Protocol configuration**: Aquí se define el protocolo que se usara para la comunicación, en este caso *syslog*.
- **User name and passwords**: En estos campos habrá que definir el usuario y contraseña de la máquina.
- **Log source identifier:** En este campo hay que definir el *Authorized service* creado con anterioridad.

Un campo que es curioso que no aparezca a la hora de definir la fuente es la dirección IPv4 de la sonda, esta no se define porque la sonda será la que envíe los *logs* a la consola central,

esta consola central mirara entre sus *authorized services* y autorizara los *logs* de la sonda en caso de que esté autorizado.

Esta es una opción más segura de personalización de la herramienta porque también cabe la posibilidad de aceptar cualquier fuente de *logs* y después rechazarla, en caso necesario, manualmente. Una vez finalizado este último punto, la sonda ya estaría integrada en la consola central.

9.1.3 Implantación de *software syslog* en *host*

Este apartado detallara el despliegue del *software* para recibir los *logs* de los *host* tanto de Linux como de Windows. Esta implantación se distinguirá en dos tipos:

- Sistemas Windows
- Sistemas Linux

El despliegue y configuración será distinto en ambos casos por lo que se distinguirá en distintos apartados las configuraciones de ambos.

El despliegue del *software* en Linux será muy sencillo y no necesitará ningún tipo de elemento extra.

En el caso de Windows sí que se diferencian 3 pasos:

- Despliegue del *software sysmon* en los *host* y algunos servidores
- Despliegue de *software* Wincollect en un Windows *server*
- Configuración del *event collector* para recibir los *logs*

Este procedimiento se proporcionará en el sector de anexos bajo la nomenclatura despliegue de colectores de eventos *host*.

9.1.4 Construcción de canal seguro colectora cliente-consola central

En este apartado se detallarán los pasos que hay que dar en el *client-side* para configurar el canal seguro contra la infraestructura propia. Primero, hay que subrayar el hecho de la importancia de los elementos de cada punto del túnel seguro.

En este caso, se constará de un *firewall* del fabricante Checkpoint en el *client-side* y un *firewall* Fortinet en la infraestructura propia. El foco de este apartado se centrará principalmente en la parte del cliente.

Checkpoint distingue la configuración de túneles para cuando son entre equipos de mismo fabricante o distinto, a este último caso le llama *3rd party*. Por lo que se empezara a detallar el procedimiento seguido para la implantación de este tipo de túnel.

Primero, habrá que crear un objeto del tipo *interoperable device* con la dirección IP del otro extremo del túnel, para ello habrá que hacerlo bajo el menú de la **figura 60**:

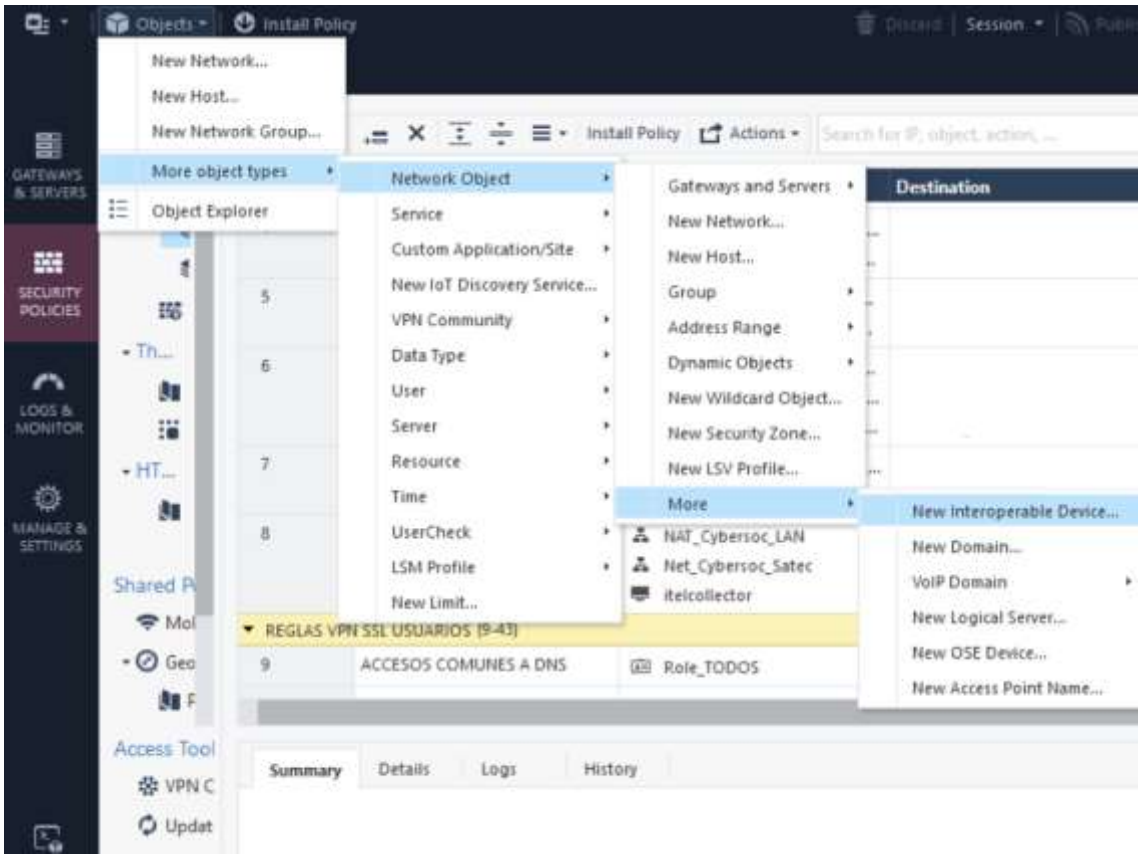


Figura 60: Menú interoperable device

Clickando en la opción que nos interesa, nos llevara al menú correspondiente, que se muestra en la **figura 61**:

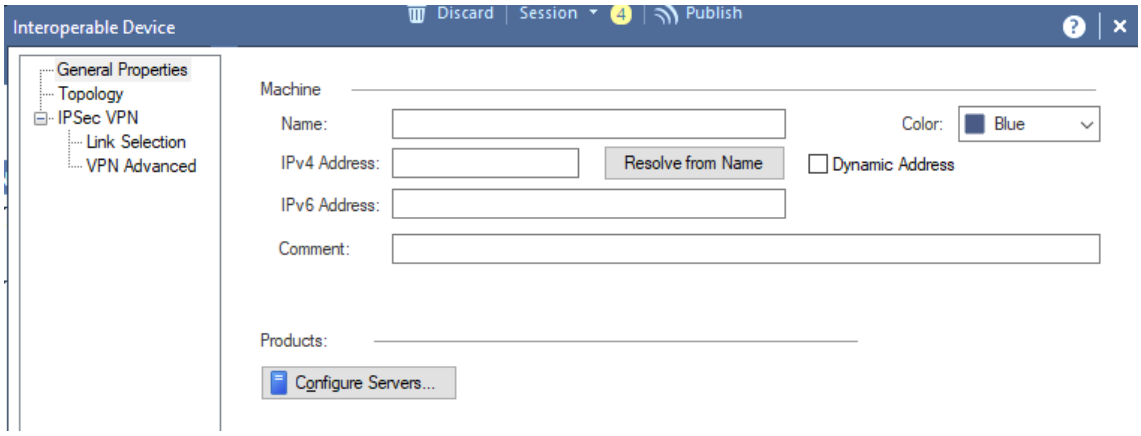


Figura 61: Personalización del interoperable device

Se rellenarían los campos de *IPv4 address y name* correctamente definiendo el otro extremo de la comunicación IPSEC. Una vez creado el otro extremo, se creará una comunidad VPN (*Virtual Private Network*), en la que se configurará el túnel IPSEC VPN. Esto se hará en el apartado del *firewall mostrado en la* ***figura 62***:

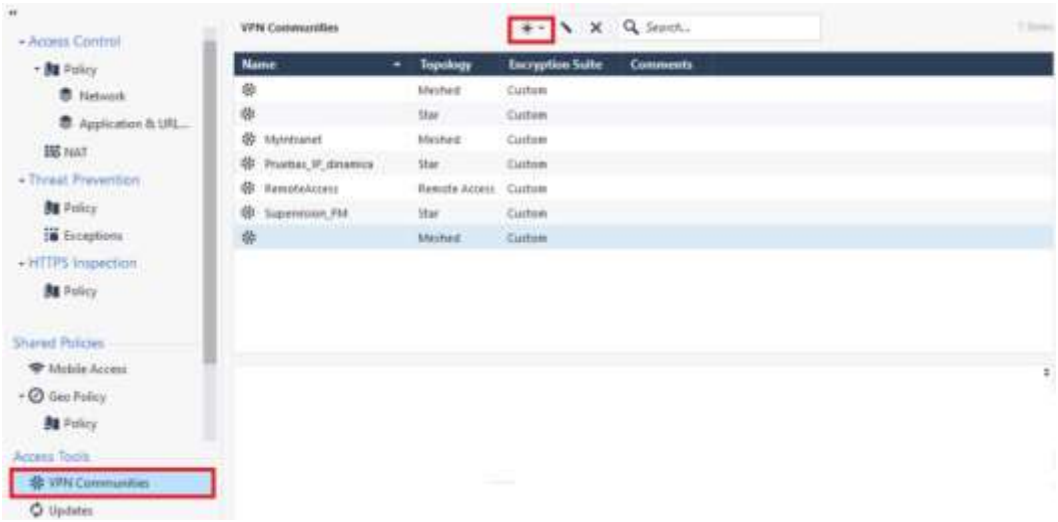


Figura 62: Menú VPN community

Se clicará en la estrella para crear una nueva y se seleccionara el tipo Meshed que es el que aplica al caso de *3rd party* VPN. A continuación, se mostrarán los parámetros de configuración que habrá que ajustar para el túnel VPN:

- Primero habrá que definir los *gateways* o extremos de la comunicación y las redes que se anunciaran tras esos *Gateway*, bajo la nomenclatura VPN *domain*:

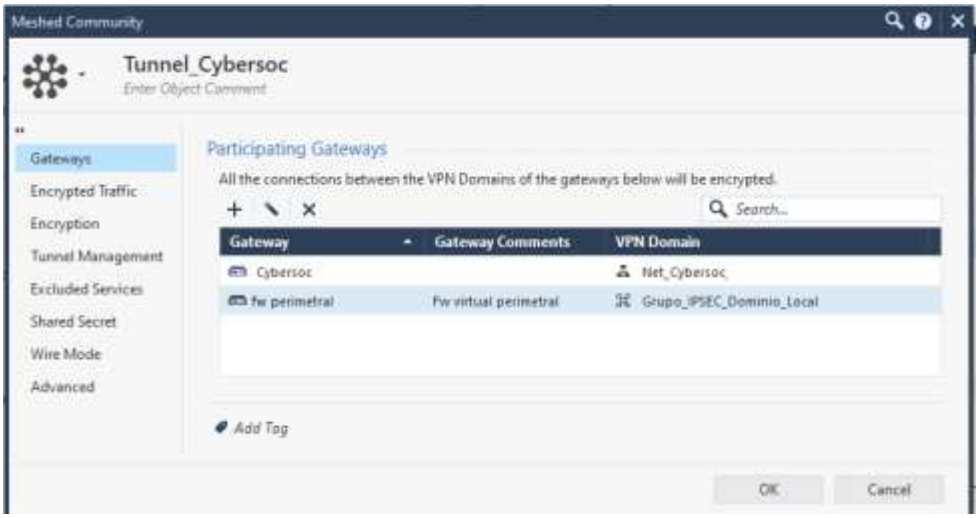


Figura 63: Definición de gateways VPN community

- Lo siguiente será establecer la configuración de encriptación o definición del *cypher suite*, esta configuración tiene que ser igual en los dos extremos, y tiene la estructura mostrada en la **figura 64**:

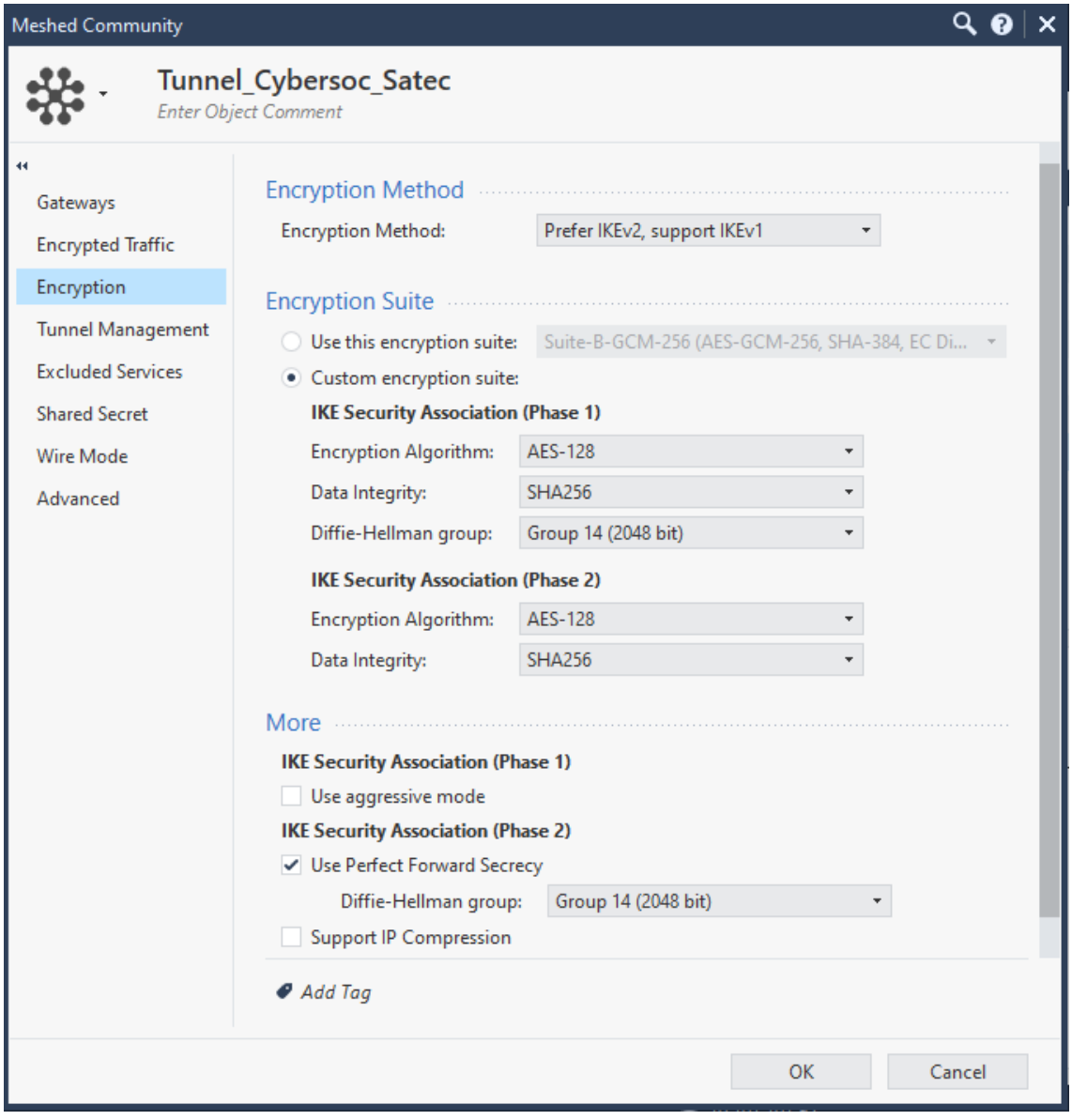


Figura 64: Definición cypher suite

- Lo siguiente es definir la contraseña bajo el menú *shared secret* que también tendrá que ser igual en ambos extremos.
- Después, habrá que definir los tiempos de renegociación de las dos fases de IKE (*Internet Key Exchange*), en este caso Checkpoint recomienda marcar el tiempo de la fase 2 cinco

horas mayores que el del extremo *3rd party*. Se han definido estos tiempos, para no saturar el túnel:

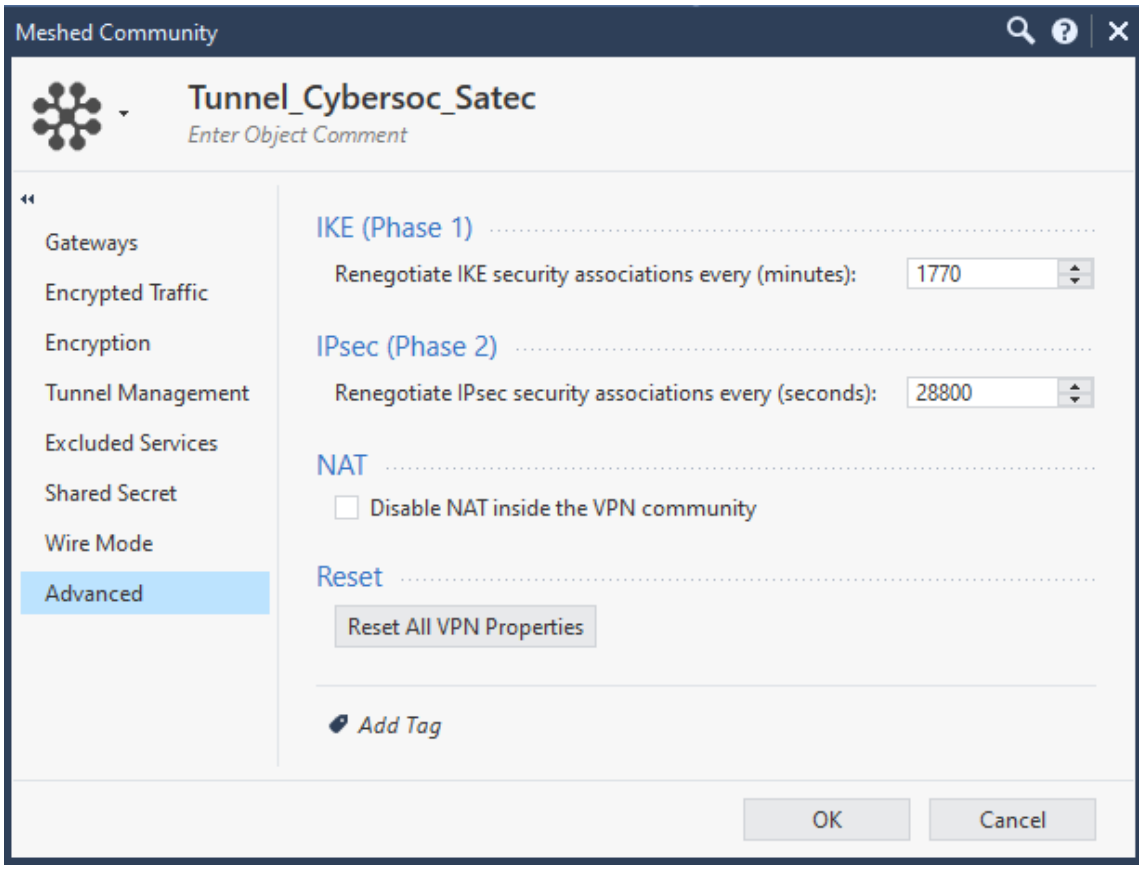


Figura 65: Definición tiempos de renegociación IKE

Una vez configurada esta última funcionalidad la comunidad VPN estará formada. Pero, para habilitar la comunicación falta un último paso, habilitar el tráfico a través del *firewall*, puesto que un túnel VPN no levanta hasta que no ve tráfico por el túnel.

Para ello, se ha definido la regla mostrada en la **figura 66** en el *firewall*:

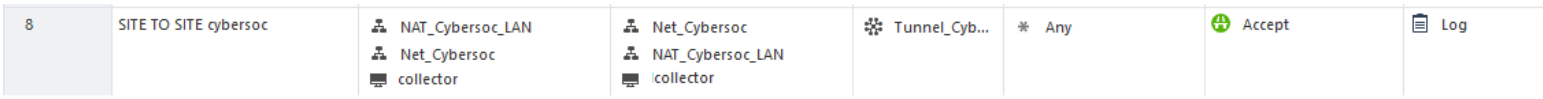


Figura 66: regla del firewall para tráfico VPN

En esta regla se permite el tráfico entre los dos *sites*; es decir, entre la red del *cybersoc* y la red del cliente y la colectora de eventos. En el campo VPN habrá que incluir la comunidad que se ha creado antes. Una vez generada la regla, el túnel habrá levantado si el otro extremo está configurado.

9.2 CONTINUIDAD DE SERVICIO SIEM

Por la propia característica de esta solución, se presentará en este apartado como continuará después del proyecto la gestión del propio producto SIEM. En este caso, como bien se ha definido en el alcance del proyecto el proyecto no queda solo en el diseño e implantación de la herramienta.

Este proyecto se convertirá en un servicio que se prestará anualmente al propio cliente y por lo tanto cabe detallar como se va a gestionar este servicio pese a ser parte de otro proyecto como tal; puesto que tiene una gran relación con este proyecto.

La explicación de este servicio se detallará mediante 3 principales puntos:

- Explicación del *dashboard* del SIEM al que tendrá acceso el cliente para ver las amenazas
- Informes trimestrales con los datos más relevantes relativos a las ofensas que ha sufrido la red
- Funcionamiento del SoC que se va a establecer y su modo de establecimiento

9.2.1 *Dashboard* IBM QRADAR

Como se ha mencionado anteriormente, el cliente tendrá acceso al *dashboard* del sistema que le presentará una serie de datos y graficas relativas al estado de la red. Para cerciorar la correcta protección de este portal puesto que sus datos son muy sensibles, este se alojará en la red propia y el cliente accederá mediante una conexión remota al mismo.

El portal constará de diversos apartados en los que, con una interfaz gráfica amigable, podrá observar el estado de la red y las ofensas que se han registrado. Para comprender mejor la estructura de un portal de este tipo, se incluirán unas representaciones graficas del mismo.

Primero, se incluye la estructura que tiene la página inicial del propio portal:

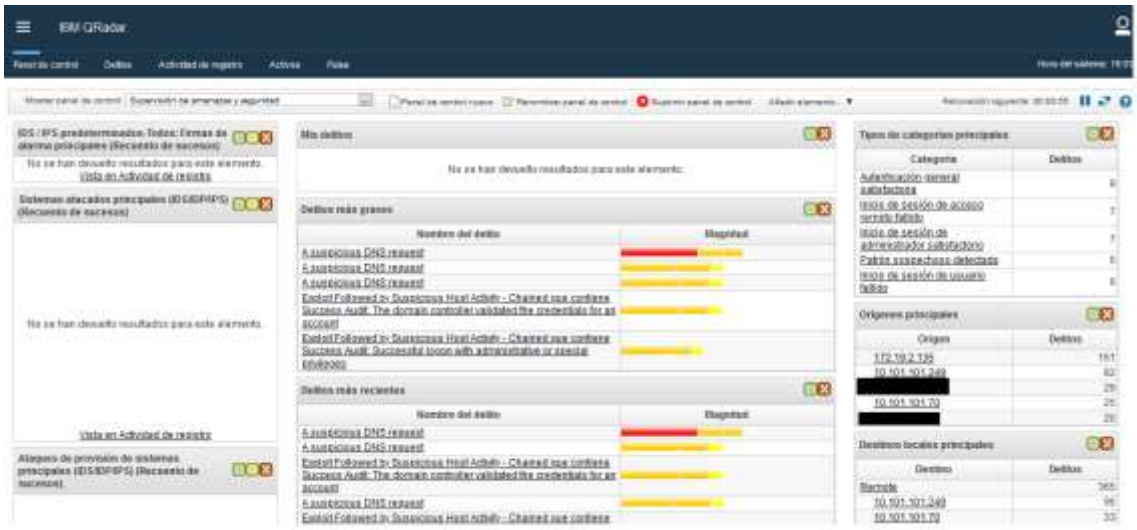


Figura 67: Panel de control del dashboard QRadar

En el panel de control se muestra información relativa y a modo de resumen de las ofensas que se han recibido. Se muestran los delitos que se han recibido, de que tipo, su impacto, desde que orígenes, a que destinos… Esta primera funcionalidad sirve para poder evaluar si ha habido algún nuevo problema en la red.

La siguiente funcionalidad que ofrece el portal, es el apartado de delitos, en el que de manera más extensa pueden observarse los delitos que ha sufrido la infraestructura con una información muy detallada sobre ello. El registro de delitos tiene el formato que se muestra en la **figura 68**:

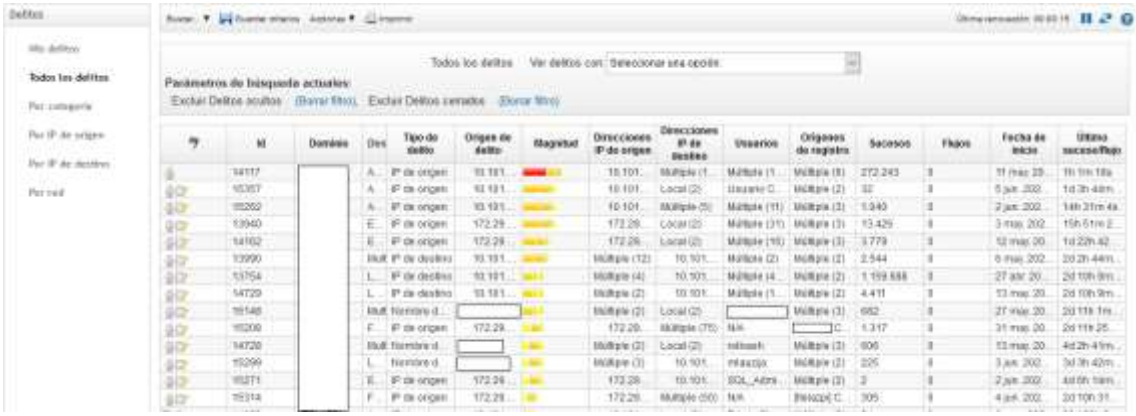


Figura 68: Registro de delitos del dashboard de QRadar

En este registro se puede observar mucha información relativa a cada problema de seguridad en la red; además, se puede ordenar o a agrupar por categoría de ataque, por IPs de origen y destino y por la red que ha sufrido la ofensa.

La siguiente funcionalidad es la actividad de registro, qué es un registro en tiempo real de los *logs* que llegan a la consola central de QRadar, el formato del mismo se puede observar a través de la **figura 69**:

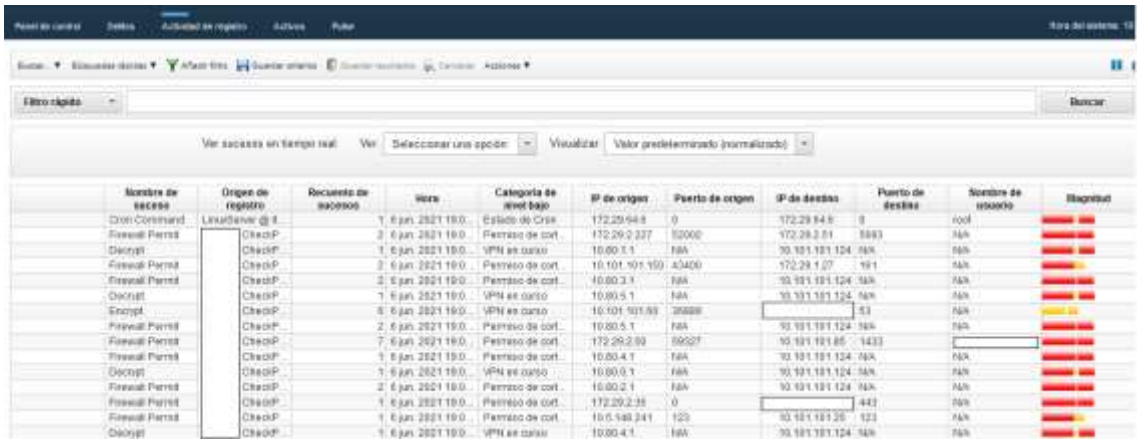


Figura 69: Actividad de registro del dashboard de QRadar

Este registro se puede utilizar para comprobar que se están recibiendo los *logs* de manera correcta, los *logs* que se reciben por segundo…

La siguiente funcionalidad es la de los activos que componen la infraestructura del cliente, por la criticidad de hacer pública esta funcionalidad, solo se mostraran las columnas que posee la tabla que muestra esta funcionalidad:

Figura 70: Formato de la funcionalidad de activos del dashboard de QRadar

Por último, la funcionalidad *pulse* muestra gráficos relativos al estado de la red al cliente, probablemente es a la funcionalidad que más uso del cliente, puesto que es la más amigable para el usuario. A modo de ejemplo se muestra este segmento de la funcionalidad:

Figura 71: Funcionalidad grafica de la herramienta

Como se puede observar mediante una interfaz gráfica y amigable se muestran datos relativos a la red, estos datos y gráficos serán útiles para el informe trimestral que se presente al cliente.

9.2.2 Informes trimestrales sobre el estado de la red

Como continuación de este proyecto, se plantearán reuniones trimestrales con el cliente, en las que se presentara el estado de la red, los trabajos realizados durante esos meses, las ofensas más importantes…

El cliente podrá conocer de una manera más cercana la solución, aportando ideas nuevas y logrando así, una solución más personalizada. Tanto en ciberseguridad y más específicamente en tareas de auditoria, los informes son muy importantes, por lo tanto, en estas reuniones se presentará un informe detallando todos los aspectos mencionados anteriormente.

El informe tendrá los siguientes apartados:

- **Introducción**: Se explicará lo que se tratará en el informe
- **Análisis** de ofensas: Se realizará un análisis en varios apartados de las ofensas que ha sufrido la red
    - **Ofensas abiertas por severidad:** se presentarán las ofensas abiertas actualmente ordenadas por severidad en un grafico
    - **Ofensas cerradas por severidad:** Se presentarán las ofensas que se hayan cerrado o solucionado ordenadas por severidad
    - **Resolución de ofensas:** Se presentará que solución se ha dado a cada ofensa
- **Análisis del trafico:** Se analizará el tráfico de la red en varios aspectos
    - **Conexiones *Malware*:** Se mostrarán gráficamente las conexiones de *malware* que se han detectado

- **Conexiones *botnets*:** Se mostrarán gráficamente las conexiones hacia *botnets* de *command and control* que se han encontrado
- **Servicios de anonimización:** Conexiones que ocultan la IP real del *host* a través de VPNs o Proxys.
- **Conexiones *command and control*:** Se mostrarán las conexiones de tipo *command and control* que se han identificado
- **Trafico de IPs de mala reputación:** Se presentará el tráfico que se ha identificado de IPs de reputación baja o mala

9.2.3 Despliegue del SoC

El despliegue del SoC en la infraestructura del cliente, permitirá gestionar las ofensas que se produzcan en tiempo real, dando un soporte de 24x7. De esta manera se aprovechará el rendimiento de la propia herramienta al máximo.

Como es habitual en los SoC se segmentará el mismo en 3 niveles:

- **N1:** Se encargarán de la actividad de monitorización continua de las alarmas de los clientes, siguiendo el proceso de clasificación y triage, y aplicando procedimientos de actuación definidos ante los diferentes casos de uso. Se encargarán de descartar falsos positivos e identificar acciones de remediación estándar como p.ej. parches de seguridad y proceden al escalado de las alarmas que requieren un análisis más profundo.
- **N2:** Compuesto por expertos de ciberseguridad con capacidad para analizar incidentes, interpretar eventos y alertas de vulnerabilidad, definir acciones de mitigación y generar los informes. Se encargarán de evaluar el riesgo real de las vulnerabilidades, priorizarlas e identificar acciones de remediación personalizada.
- **N3:** Está formado por ciberanalistas con conocimientos muy avanzados en los diferentes campos que componen la ciberseguridad, que aportan ya escenarios de investigación avanzada o análisis forense ante un incidente. También realizan actividades de caza de amenazas en el ciclo de revisión, sin necesidad de que exista un incidente de seguridad y proporcionar soluciones a problemas y ajustes en el SIEM antes de que se produzca el mismo. También realizan actividades de hacking ético o auditorías de seguridad.

Estos 3 niveles serán gestionados por la figura de *Security Manager*. Un Service Manager especializado para los servicios avanzados de seguridad, se encargará del seguimiento y control de actividades y el reporte de las mismas al cliente, coordinando internamente los distintos componentes del servicio definidos en el modelo anterior, que provienen de la aportación de capacidades de distintos departamentos de la empresa que presta el servicio.

Además de definir los niveles que tendrá el SoC, también cabe definir el formato de las alertas que llegaran al cliente, para poder comprobar que formato siguen este tipo de mensajes o alertas de ofensas. El formato de las mismas se puede observar a través de la **tabla 12**:

¿Qué notificar?	Descripción
Asunto	Frase que describe de forma general el incidente. [xxxx-Cyber] - ##ID ICD## Análisis Ofensa Nº xxxxx Ej: [xxxx-Cyber] - ##2021100231## Necesario Análisis - Nº Ofensa 11092
Descripción	Describir con detalle lo sucedido.
Afectado	Indicar si se trata de un usuario o varios los afectados
Fecha y Hora del Incidente	Indicar con la mayor precisión posible cuándo ha ocurrido el ciberincidente.
Fecha y hora de detección del incidente	Indicar con la mayor precisión posible cuándo se ha detectado el ciberincidente.
Taxonomía del incidente	Posible clasificación del ciberincidente en función de la taxonomía descrita. Se especificará: clasificación y tipo de incidente.
Recursos afectados	Indicar la información técnica sobre el número y tipo de activos afectados por el ciberincidente, incluyendo direcciones IP, sistemas operativos, aplicaciones, versiones…
Origen del incidente	Indicar la causa del incidente si se conoce. Apertura de un fichero sospechoso, conexión de un dispositivo USB, acceso a una página web maliciosa, etc.
Impacto	Impacto estimado en la entidad, en función del nivel de afectación del ciberincidente.
Adjuntos	Incluir documentos adjuntos que puedan aportar información que ayude a conocer la causa del problema o a su resolución (capturas de pantalla, ficheros de registro de información, correos electrónicos, etc.)

Tabla 12: Formato alerta CyberSoC

Como se puede observar mediante estas alertas se plantea generar información para el cliente y para el técnico que tenga que solventar o revisar la ofensa.

9.3 DESARROLLO DE SOLUCIÓN MACHINE LEARNING

En este apartado se detallará el desarrollo de la solución de ML mencionada en los apartados anteriores. Primero se analizará cómo se usa el *software* Jupyter Notebook. Primero se ejecutará el programa, en este caso desde un entorno Windows Server. Al ejecutar el programa, se levantará un servidor web que escuchará por defecto en el puerto 8888.

Para conectarse a este servidor, se abrirá el navegador y se introducirá la dirección IP del servidor con el puerto mencionado anteriormente, formando un formato como el que se muestra a continuación:

< IP_servidor_Jupyter>:8888

Para observar cual es la estructura del portal web se muestra la **figura 72**:

Figura 72: Página de inicio Jupyter

Una vez en esta página, crearemos un *notebook* nuevo usando el botón *New*, una vez creado nos redirigirá a una página que nos permitirá empezar a desarrollar el programa.

En un inicio se definirá el *dataset* que se va a usar; en este caso, se usara como fuente los *logs* que recibe el QRadar en un periodo de tiempo. Estos *logs* están marcados con dos etiquetas o clases. La primera, define si se trata de un ataque o no y es una variable de tipo *binary*; es decir, posee los estados 0 y 1. La segunda clase, determina qué tipo de ataque se trata; es decir, si es un ataque de reconocimiento, *Exploits*, *DoS*…

En el apartado de anexos se dará más información acerca del resto de características que se tienen en cuenta para el desarrollo del programa.

Lo primero que hará el programa será cargar este *dataset* mostrando que se carga correctamente, para realizar esto se usará la librería de Python *pandas*. Esta librería proporciona prácticamente casi todas las operaciones que puedan idearse con un bloque de datos.

Para cargar el *dataset* y ver sus primeras columnas se ingresan los siguientes comandos:

```
df = pd.read_csv('QRadar_logs.csv', sep = ';', decimal = '.')

df.head(5)
```

Logrando el resultado que muestra la **figura 73**:

	srcip	sport	dstip	dsport	proto	state	dur	sbytes	dbytes	sttl	...	ct_ftp_cmd
0	5916600	1390	1491711266	53	udp	CON	0.001055	132.0	164.0	31.0	...	0.0
1	5916600	33661	1491711269	1024	udp	CON	0.036133	528.0	304.0	31.0	...	0.0
2	5916606	1464	1491711267	53	udp	CON	0.001119	146.0	178.0	31.0	...	0.0
3	5916605	3593	1491711265	53	udp	CON	0.001209	132.0	164.0	31.0	...	0.0
4	5916603	49664	1491711260	53	udp	CON	0.001169	146.0	178.0	31.0	...	0.0

5 rows × 49 columns

Figura 73: Tabla de datos del dataset de ML

También cabe aclarar que el *dataset* posee aún más columnas o *features*, pero la interfaz gráfica no muestra más información.

Así mismo, se considera interesante mostrar los valores que toman las clases o *labels* del modelo, esto se comprueba con las siguientes líneas de código:

```
pd.value_counts(df.values[:,-1])
pd.value_counts(df.values[:,-2])
```

```
0.0    677786
1.0     22215
dtype: int64
```

Figura 74: Cantidad de entradas de ataques y no ataques del dataset

```
Generic           7522
Exploits          5409
 Fuzzers          5051
Reconnaissance    1759
DoS               1167
Backdoors          534
Analysis           526
Shellcode          223
Worms               24
dtype: int64
```

Figura 75: Cantidad de cada tipo de ataque en el dataset

Posteriormente, se modificaran las variables categóricas; es decir, las variables no numéricas mediante la función *LabelEncoder()*. Esto se realizará mediante el siguiente bloque de funciones:

```
X=df.values[:,:-2]

enc=LabelEncoder()
```

```
X[:,4]=enc.fit_transform(X[:,4])

enc=LabelEncoder()
X[:,5]=enc.fit_transform(X[:,5])

enc=LabelEncoder()
X[:,13]=enc.fit_transform(X[:,13])

enc=LabelEncoder()
X[:,39]=enc.fit_transform(X[:,39])
```

Pero con esto, no finaliza el preprocesamiento de los datos puesto que estos datos hay que limpiarlos y afinarlos; por ejemplo, convertir las entradas en el mismo tipo de dato, quitar los valores vacíos y lo mismo para la *label* o salida, esto se hace mediante el siguiente código:

```
X=X[:,4:]
X=X.astype(float)

imp=SimpleImputer(missing_values=np.nan,strategy='mean')
imp.fit(X)# habría que usar solo el training
X=imp.transform(X)
np.mean(X,axis=1)

y1=df.values[:,-1]
y1=y1.astype('int')
y2=df.values[:,-2]
```

Una vez limpiado el *dataset* habrá que separar el *dataset* en test y train. Para ello se usará la siguiente función:

```
Xtr, Xte, ytr, yte= train_test_split(X,y,test_size=0.3,random_state=0)
```

Mediante el anterior método, se separa el *dataset* con una distribución 70/30 en *train* y *test*. Te separa tanto la entrada del sistema como la salida. Posteriormente se cambiará la escala de los datos para no tener mucha diferencia entre las distintas *features*, esto se hace de la siguiente manera:

```
sc=StandardScaler()

sc.fit(Xtr)

Xtr=sc.transform(Xtr)
```

```
Xte=sc.transform(Xte)
```

Una vez preparados los datos, tocaría definir los distintos modelos que se han visto en el análisis de alternativas, para entrenar estos modelos se usaran los datos que se han preparado con anterioridad; además, se sacan las estadísticas necesarias para la evaluación de los modelos y en las siguientes trazas de código se muestra el proceso:

Modelo de regresión logística:

```
regL = LogisticRegression(class_weight='balanced')

regL.fit(Xtr,ytr)

ypred=regL.predict(Xte)

acc=100*accuracy_score(yte1,ypred)

print('La precision del modelo de regresion logistica: %{Acc:.2f}'.format(Acc=acc))

print(classification_report(yte1,ypred))

print('Matriz de confusion:\n')

print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(regL, Xte1, yte1,\

                      normalize='true',cmap=plt.cm.Blues, display_labels={'Normal','Attack'})

ax=plt.gca()

fig=plt.gcf()

fig.set_size_inches(8,8)

ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')
```

En este caso de usa un modelo de regresión logística balanceado, que permite obtener mejores resultados a la hora de analizar datos conuna clase dominante.

SVC:

```
n_estimators=50

svm=OneVsRestClassifier(BaggingClassifier(LinearSVC(class_weight='balanced'),
max_samples=1.0 / n_estimators, n_estimators=n_estimators),n_jobs=-1)

svm.fit(Xtr,ytr)

ypred=svm.predict(Xte)
```

```
acc=100*accuracy_score(yte1,ypred)

print('La precision del modelo de Support vector machine: %{Acc:.2f}'.format(Acc=acc))


print(classification_report(yte1,ypred))

print('Matriz de confusion:\n')

print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(svm, Xte1, yte1,\

                      normalize='true',cmap=plt.cm.Blues, display_labels={'Normal','Attack'})

ax=plt.gca()

fig=plt.gcf()

fig.set_size_inches(8,8)

ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')
```

En este caso el modelo SVC, hay que trabajarlo más que el resto puesto que con *datasets* con tantos datos como el de este proyecto, el modelo tarda muchísimo tiempo en converger, para ello se hace uso de los métodos OneVsRestClassifier() y BaggingClassifier(), puesto que ayudan a realizar trabajos paralelamente, a segmentar el análisis de características…

Random Forest:

```
rfc=RandomForestClassifier(n_estimators=100,max_features=4,
max_samples=0.75,random_state=0,oob_score=True)

rfc.fit(Xtr,ytr)

ypred=rfc.predict(Xte)

print('La precisión del modelo de Random Forest: %{Acc:.2f}'.format(Acc=acc))

print(classification_report(yte1,ypred))

print('Matriz de confusión:\n')

print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(svm, Xte1, yte1,\

                      normalize='true',cmap=plt.cm.Blues, display_labels={'Normal','Attack'})

ax=plt.gca()
```

```
fig=plt.gcf()
fig.set_size_inches(8,8)
ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')
```

En este caso, se hace uso de los atributos clásicos de *Random Forest*, se genera el número máximo de estimadores, el número máximo de características de cada estimador, las muestras que usara cada uno…

Redes Neuronales:

```
# mlp clasico
mlpc2=MLPClassifier(hidden_layer_sizes=(50,50,50),activation='relu',max_iter=200,\
        early_stopping=True,validation_fraction=0.15,random_state=0)
mlpc2.fit(Xtr,ytr)
ypred=mlpc2.predict(Xte)
# GridSpace mlp
mlp = MLPClassifier(max_iter=100)
parameter_space = {
   'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
   'activation': ['tanh', 'relu'],
   'solver': ['sgd', 'adam'],
   'alpha': [0.0001, 0.05],
   'learning_rate': ['constant','adaptive'],
}
from sklearn.model_selection import GridSearchCV

mlp = GridSearchCV(mlp, parameter_space, n_jobs=-1, cv=3)
mlp.fit(Xtr1,ytr1)
ypred=mlp.predict(Xte1)
```

```
print('La precisión del modelo de Redes neuronales: %{Acc:.2f}'.format(Acc=acc))

print(classification_report(yte1,ypred))

print('Matriz de confusión:\n')

print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(svm, Xte1, yte1,\

                      normalize='true',cmap=plt.cm.Blues, display_labels={'Normal','Attack'})

ax=plt.gca()

fig=plt.gcf()

fig.set_size_inches(8,8)

ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')
```

En el caso de las redes neuronales, al observar resultados muy desbalanceados se ha tratado de hacer uso de métodos de validación cruzada mediante el método GridSearchCV, para conocer el mejor modelo de redes neuronales para este proyecto.

Como se puede observar, la definición, entrenamiento y comprobación de cada modelo se hace de la misma manera. Este proceso ya se explicó en el diseño de la solución por lo que la reiteración en la explicación no aportaría a este proyecto. Después de entrenar y comprobar el proyecto se han buscado las métricas que puedan hacer que se decante por el mejor modelo. Esta información se incluirá en el apartado de análisis de alternativas.

Además, para más detalle del código total que se ha usado para este proyecto se podrá acudir al apartado de anexos.

10 PLANIFICACIÓN DEL PROYECTO

En la planificación de este proyecto ha sido muy importante el uso de métodos de planificación, de esta manera se minimizan varios de los riesgos vistos en el análisis de los mismos. En este caso la planificación se separa en dos apartados: la relativa a KanBan y la relativa a la planificación clásica paquetes de trabajo y diagrama Gantt.

Primero se mencionará como se ha usado la metodología KanBan y la implementación de la misma y después, se tratará con detalle la planificación clásica.

10.1 PLANIFICACIÓN KANBAN

Como se ha mencionado en la introducción, para la correcta realización de un proyecto, es necesario o muy útil el uso de tecnologías de gestión de proyectos. En este caso, se ha usado la tecnología KanBan, cabe mencionar que la plataforma más conocida que usa esa metodología es Trello, por lo cual es la que se ha usado.

Lo primero, cabe recalcar que Trello es una plataforma gratuita, característica que le ha hecho ser elegible para un proyecto docente. Una vez aclarado el anterior aspecto, es necesario registrarse en la plataforma para poder hacer uso de la misma. La plataforma consta de una versión un web y una versión de escritorio. En este caso, se ha optado por la versión web.

Después, se ha creado un tablero llamado “Proyecto Cybersoc QRadar” para la realización del proyecto. El creador, lo ha compartido con su compañero para que ambos puedan gestionar la planificación del proyecto. De manera automática, Trello te crea tres listas o conjunto de tarjetas: TO DO, DOING y DONE. De esta manera, se puede tener controlado el proyecto, granulado en las distintas tareas que deben ejecutarse. Opcionalmente, se podrían añadir más listas en proyectos más complejos, pero en este caso no se ha estimado necesario. Por lo cual, Trello tendría la apariencia mostrada en la **figura 76**:

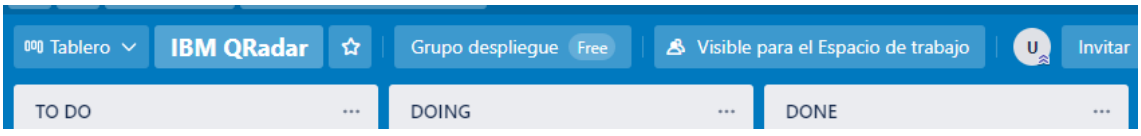


Figura 76: Listas trello

Posteriormente se ha pasado a añadir tarjetas, es decir, a definir las tareas que iban a realizarse, logrando el resultado mostrado en la **figura 77**:

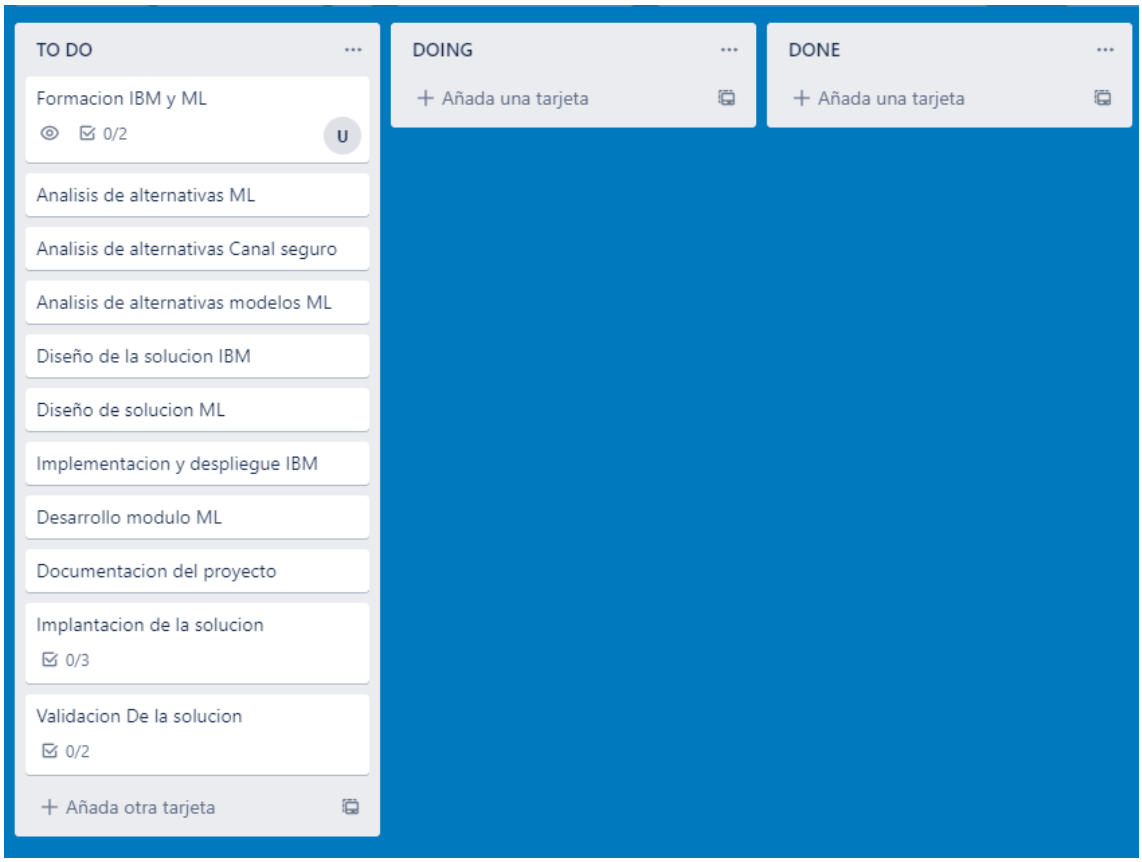


Figura 77: Tarjetas trello el proyecto

Como se puede apreciar, hay tareas que ya han sido comentadas y asignadas, esto ayuda a una gestión más eficiente de asignación de recursos. Cuando las tareas se comiencen a realizar pasarían a la lista de DOING. Una vez finalizada la tarea, pasaría al apartado DONE. Cabe recalcar que cada actualización de cada tarea se verá reflejada en su tarjeta correspondiente.

Para ver como se vería la tarjeta y sus funcionalidades, se analizará una en concreto:

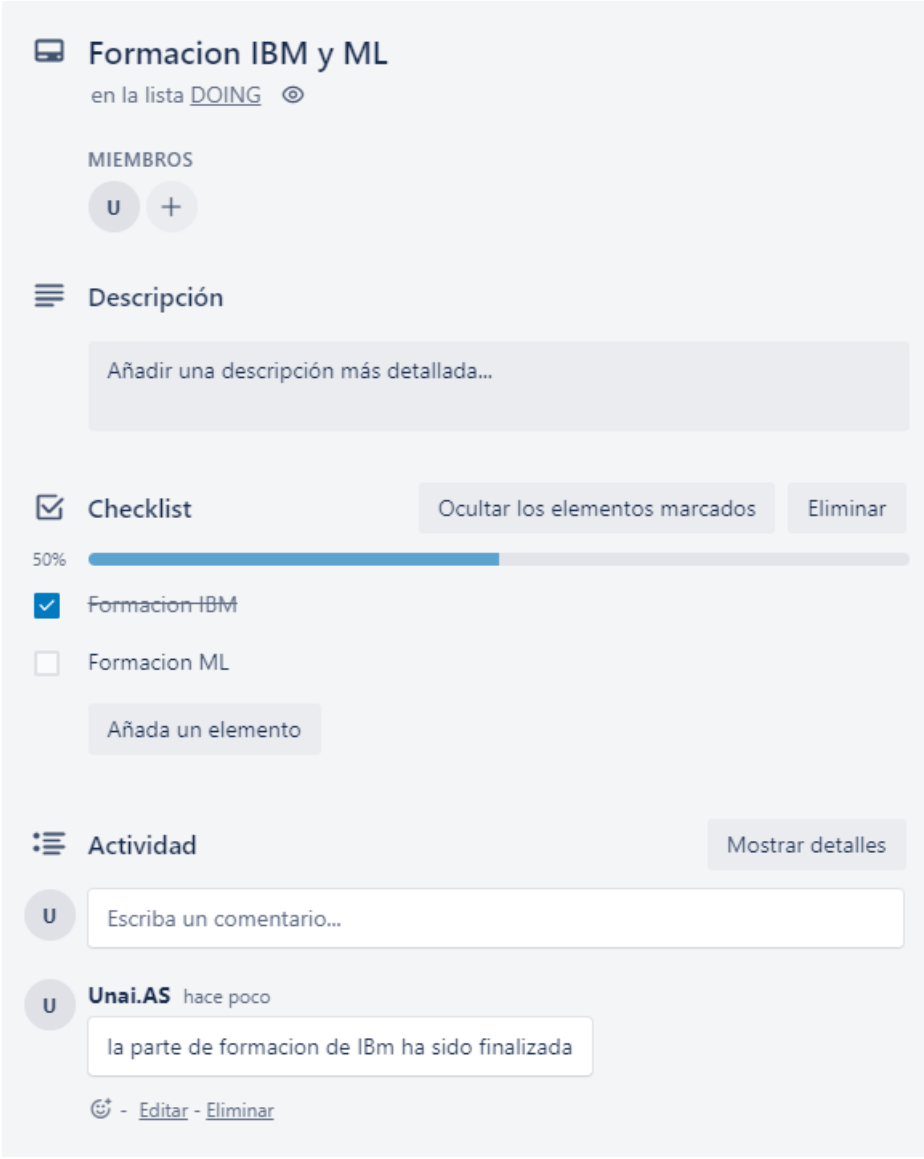


Figura 78: Estructura tarjeta trello

Como se puede observar, se han añadido comentarios actualizando el estado de la tarjeta, y también se ha asignado el recurso que la ejecutara. Hay varios campos que se pueden rellenar, como la fecha de vencimiento, *checklist* de subtareas…

Una vez que finalice el proyecto, el tablero cogerá la forma que muestra la **figura 79**:

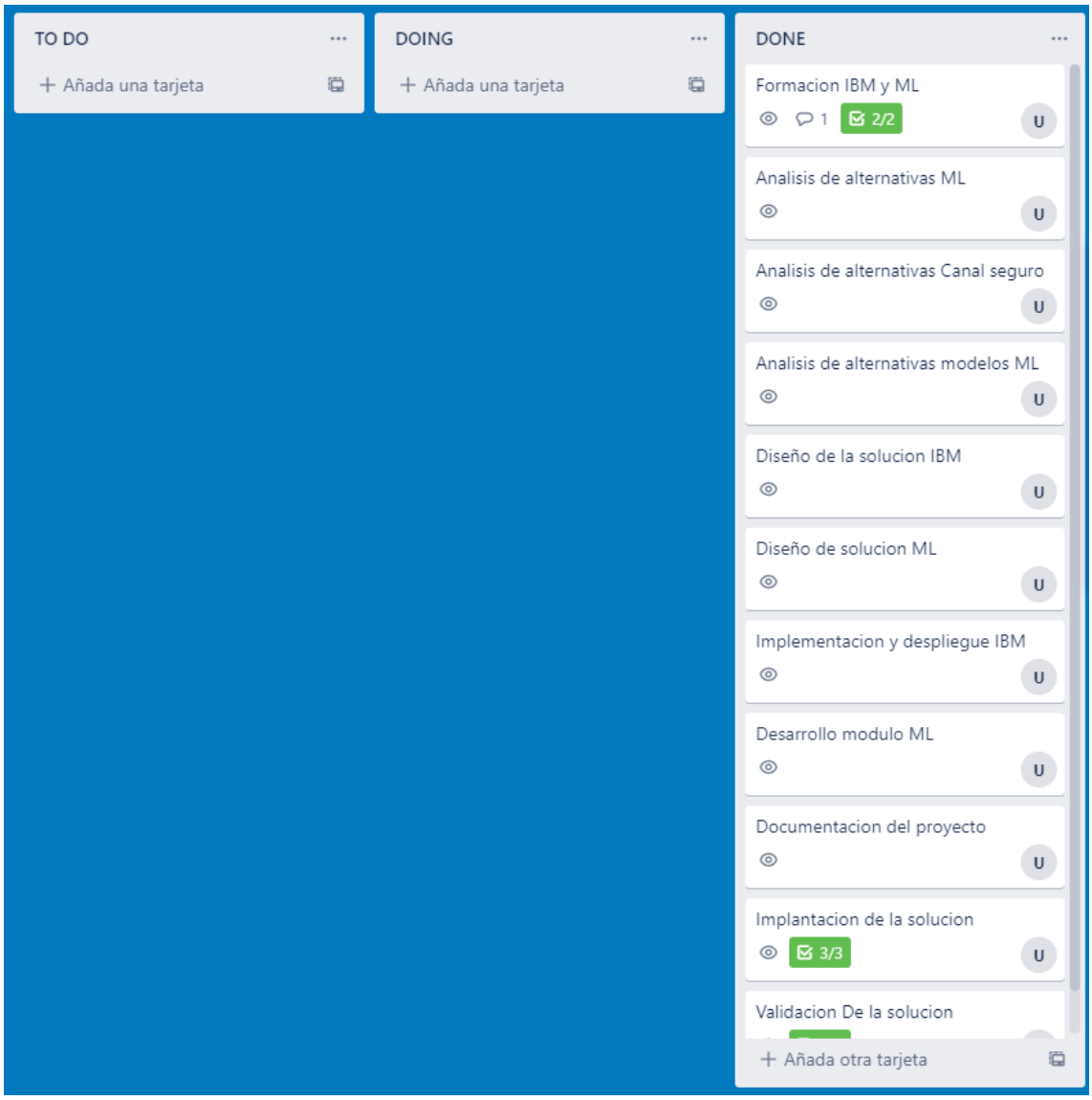


Figura 79: Tablero trello finalizado

De la misma manera este proyecto en singular se añadiría al tablero de la empresa realizadora y se revisaría semanalmente, como esta parte es más del funcionamiento global y no tanto del proyecto en singular, no se incluirá en este documento.

Con el análisis de estas funcionalidades, se da por finalizada la explicación de las tecnologías KanBan.

10.2 GRUPO DE TRABAJO

Para poner en marcha este proyecto, es necesario el uso de recursos humanos y su correcta planificación, utilización y gestión. Para ello será importantísimo tener muy claras las funciones de cada uno de cara a realizar un proyecto de forma controlada.

A continuación, se define el grupo de trabajo que formara este proyecto:

- **Ingeniero junior:** Este trabajador será parte de los trabajos más simples del proyecto y formará parte de la configuración.
- **Ingeniero senior:** Este trabajador será el que tomará importancia en el diseño de la solución, dará soporte al ingeniero junior y tendrá contacto con el cliente.
- **Jefe de proyecto:** Este trabajador será el encargado de la parte de la gestión del proyecto, reuniones internas y gestión de recursos.

En el siguiente apartado se definirán los paquetes de trabajo que componen el proyecto.

10.3 PAQUETES DE TRABAJO

El proyecto se agrupará en distintos paquetes de trabajo, que a su vez estarán constituidos de tareas específicas que permitirán segmentar el proyecto y facilitar el control del mismo. Cabe recalcar la importancia de este apartado y método, puesto que tener el control de la planificación del proyecto es esencial. A continuación, se detallan los paquetes:

LP1: Fase de formación y conocimiento del proyecto

A1.1 Formación en Machine Learning

- Duración: 10 días
- Descripción: El ingeniero junior se formará en la temática de ML, para ello, usará la documentación disponible en internet y tendrá soporte de una persona más experta como el ingeniero senior, aunque no se le considerara para esta tarea
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador y material de oficina.

A1.2 Formación IBM QRadar

- Duración: 10 días
- Descripción: En este apartado el ingeniero junior tomará parte de la formación de IBM QRadar, lo hará a través de IBM *Knowledge center*
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador y material de oficina.

LP2: Análisis de alternativas de diseño

A2.1 Análisis de fabricantes

- Duración:5 días
- Descripción: Se hará un análisis de los distintos fabricantes de SIEM y se valoraran o ponderaran en busca de la mejor solución
- Recursos humanos: Ingeniero senior e Ingeniero junior
- Recursos técnicos: Ordenador

A2.2 Análisis alternativas modelos Machine Learning

- Duración: 5 días
- Descripción: Dentro de ML, se analizarán los distintos modelos en busca del modelo que mejor métricas presente
- Recursos humanos. Ingeniero junior
- Recursos técnicos: Ordenador

A2.3 Análisis tecnologías *tunneling*

- Duración: 1 día
- Descripción: Se analizarán las distintas topologías de *tunneling* que existen, mediante su ponderación y valoración, se elegirá la tecnología de uso
- Recursos humanos: Ingeniero senior e Ingeniero junior
- Recursos Tecnicos: Ordenador.

LP3: Diseño de la solución; SIEM y ML

A3.1 Diseño del despliegue de red

- Duración: 5 días
- Descripción: Se diseñará el despliego de red de las herramientas del proyecto
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador y programa Visio

A3.2 Diseño del canal seguro

- Duración: 20 días
- Descripción: Se diseñará la conexión de canal seguro entre el SIEM y los colectores de *logs*
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador

A3.3 Diseño de módulo de ML

- Duración: 5 días
- Descripción: Se hará el diseño de alto nivel para el desarrollo del módulo de ML
- Recursos técnicos: Ordenador

LP4. Implementación y Desarrollo

A4.1 Configuración del *core* IBM

- Duración: 20 días
- Descripción: Se configurarán los parámetros relativos al funcionamiento del SIEM de IBM
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador, Servidor SIEM y licencias de equipamientos.

A4.2 Despliegue de herramientas *log collectors*

- Duración. 15 días
- Descripción: Despliegue, configuración e instalación de los colectores de eventos
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador, servidores y servidor VMWare.

A4.3 Configuración del canal seguro

- Duración. 15 días
- Descripción: Construcción del canal seguro para la comunicación entre el SIEM y los colectores
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador, Checkpoint firewall y Fortinet Firewall

A4.4 Diseño de reglas CRE

- Duración:10 días
- Descripción: Diseño y afinamiento de las reglas que capturan amenazas en el SIEM
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador y SIEM Qradar.

A4.5 Desarrollo modulo ML

- Duración: 1 día
- Descripción: Codificación del módulo de ML
- Recursos humanos: Ingeniero junior

- Recursos técnicos: Ordenador y servidor Jupyter notebook

LP5: Pruebas, validación y puesta en producción

A5.1 Pruebas de conectividad de equipamiento de seguridad

- Duración: 1 día
- Descripción: pruebas de conectividad en el despliegue.
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador

A5.2 Pruebas comunicación cliente y servidor syslog

- Duración: 1 día
- Descripción: Pruebas de comunicación entre los servidores y los clientes de *syslog*.
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador

A5.3 Pruebas de reglas CRE

- Duración: 1 día
- Descripción: Pruebas y validación de las reglas CRE, mediante el uso de comportamiento malicioso interno.
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador

A5.4 Afinamiento de falso positivos

- Duración: 1 día
- Descripción: Personalización de alertas y detecciones para borrar los falsos positivos.
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador

A5.5 Prueba del módulo ML

- Duración: 1 día
- Descripción: Prueba del módulo de ML y su nivel de precisión
- Recursos humanos: Ingeniero junior
- Recursos técnicos: Ordenador

LP6: Gestión del Proyecto

A6.1 Seguimiento del proyecto

- Duración: Duración del proyecto
- Descripción: Durante el proyecto la gestión relativa: reuniones periódicas, comunicación con el cliente…
- Recursos humanos: Jefe del proyecto, Ingeniero junior e ingeniero senior.
- Recursos técnicos: Trello y 3 ordenadores

A6.2 Redacción y documentación del proyecto

- Duración: 20 días
- Descripción: Al acabar el proyecto redacción del diseño, metodología y aspectos más importantes
- Recursos humanos: Ingeniero junior, Ingeniero senior y jefe de proyecto

10.4 Entregables e Hitos del Proyecto

Para poder llevar un seguimiento correcto del proyecto, se definirán varios entregables y fechas límite de los mismos. De este modo, se llevará un seguimiento continuo del proyecto y no se obviará el objetivo final. El proyecto comenzó el 6 de enero de 2021 y deberá acabar el 7 de mayo de 2021, además dentro del proyecto se definirán más restricciones que se engloban en la **tabla 13**:

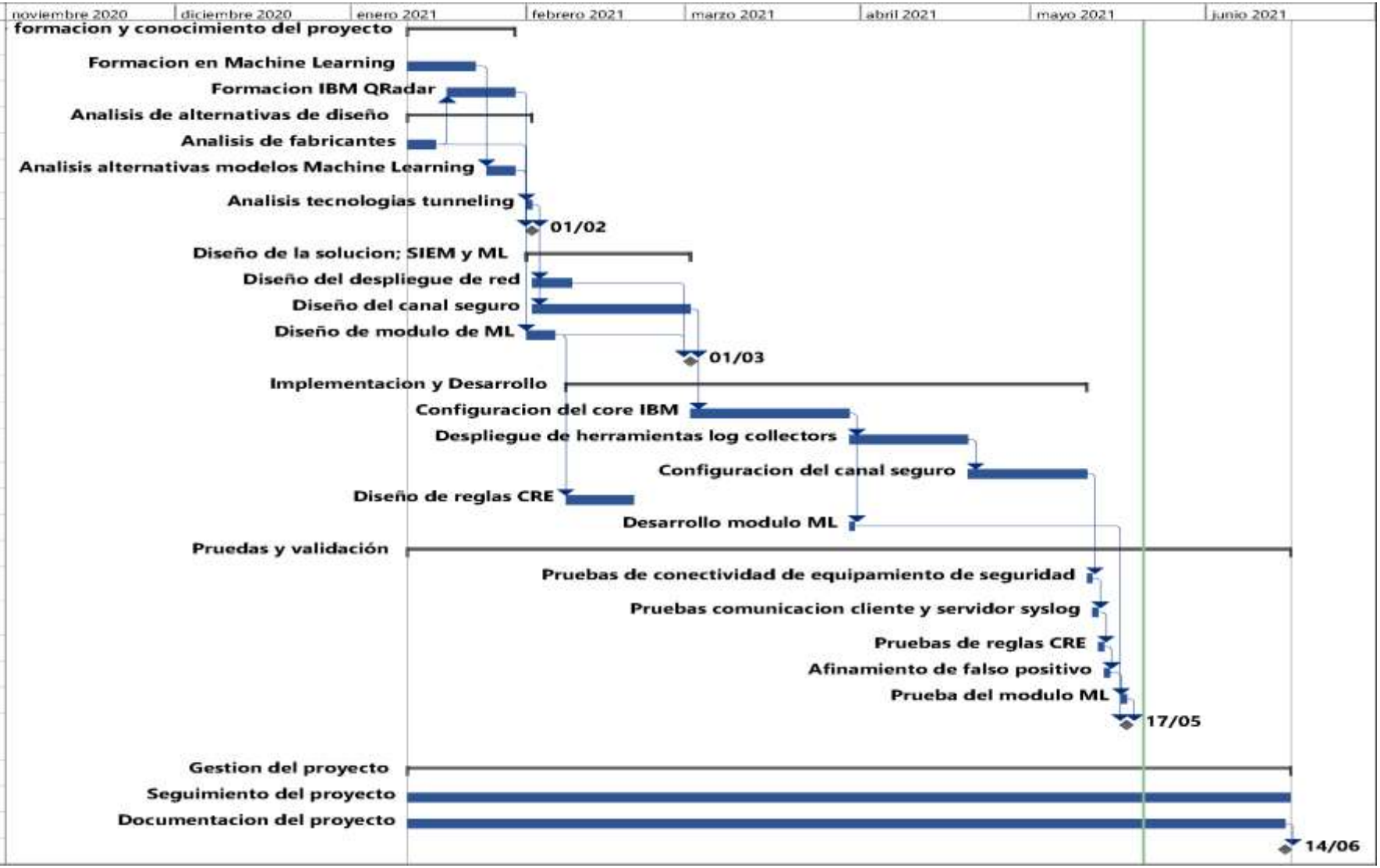
Hito	Entregable	Data
Finalización de análisis de alternativas	Análisis de alternativas	01/02/2021
Fin del diseño	documentación del diseño	01/03/2021
Finalización de configuración y validación	Informe PaP (Paso a Producción)	17/05/2021
Fin del proyecto	Memoria del proyecto	14/06/2021

Tabla 13: Hitos y entregable

Estos hitos y entregables ayudaran a seguir de una manera más detallada el proyecto puesto que los puntos de control darán información acerca del estado del proyecto.

10.5 DIAGRAMA GANTT

Id	Nombre de tarea
1	**Fase de formacion y conocimiento del proyecto**
2	Formacion en Machine Learning
3	Formacion IBM QRadar
4	**Analisis de alternativas de diseño**
5	Analisis de fabricantes
6	Analisis alternativas modelos Machine Learning
7	Analisis tecnologias tunneling
8	Entrega de analisis de alternativas
9	Diseño de la solucion; SIEM y ML
10	Diseño del despliegue de red
11	Diseño del canal seguro
12	Diseño de modulo de ML
13	Fin del diseño
14	**Implementacion y Desarrollo**
15	Configuracion del core IBM
16	Despliegue de herramientas log collectors
17	Configuracion del canal seguro
18	Diseño de reglas CRE
19	Desarrollo modulo ML
20	**Pruedas y validación**
21	Pruebas de conectividad de equipamiento de seguridad
22	Pruebas comunicacion cliente y servidor syslog
23	Pruebas de reglas CRE
24	Afinamiento de falso positivo
25	Prueba del modulo ML
26	Finalizacion de implementacion y validacion
27	**Gestion del proyecto**
28	Seguimiento del proyecto
29	Documentacion del proyecto
30	Finalizacion del proyecto



11 ASUNCIÓN DE GASTOS

El presupuesto de este proyecto está compuesto por la suma total de los costes asociados al mismo, ya sean humanos como materiales. Cada uno de ellos se ve representado en distintas partidas.

En este apartado se calculan, por un lado, los costes humanos y por otro el precio total de cada elemento material con su correspondiente periodo de amortización.

11.1 HORAS INTERNAS

En esta partida se incluyen las horas dedicadas por todos aquellos trabajadores del proyecto para llevar a cabo las tareas asociadas al mismo. Estas horas conllevan un coste, que se puede calcular conocido el coste horario y el número de horas invertidas. De esta forma es posible obtener el coste total de las horas internas.

Este proyecto consta de tres trabajadores, un ingeniero junior que será el responsable del diseño y la configuración de la solución que se ha implementado. También se contará con un ingeniero senior que dará soporte al ingeniero junior a la hora de plantear la solución y ayudará al trato con el cliente. Por último, se contará con un jefe de proyecto que se ocupará de la gestión del mismo.

El desglose de las horas de estos tres trabajadores se presenta en la **tabla 14**:

Horas internas			
Trabajador	Coste de horas(€/h)	Horas(h)	Coste(€)
Ingeniero junior	20 €/h	448 h	8.960,00 €
Ingeniero senior	35 €/h	112 h	3.920,00 €
Jefe del proyecto	45 €/h	56 h	2.520,00 €
Total			15.400,00 €

Tabla 14: Horas internas del proyecto

El montante total de las horas internas es 15.400 €, aun así, cabe destacar que este proyecto se convertirá en un servicio al cliente que compondrá de un flujo de horas continuo, por lo que se presenta como un proyecto estratégico.

11.2 GASTOS

La partida de gastos incluye todo aquello que se haya usado para llevar a cabo el proyecto y que, debido a su uso, no pueda volver a ser empleado en proyectos posteriores. El desglose de los gastos del proyecto se presenta en la **tabla 15**:

Gastos			
Gasto	Cantidad	Coste total (/unidad)	Coste(€)
Material fungible	-	50,00 €	50,00 €
Qradar console	1	80.000,00 €	80.000,00 €
Qradar Event collector	1	25.000,00 €	25.000,00 €
Internet	-	80,00 €	80,00 €
Electricidad	-	50,00 €	50,00 €
Licencias Qradar	1	6.000,00 €	6.000,00 €
Total			111.050,00 €

Tabla 15: Gastos del proyecto

Los gastos ascienden a 111.050 €, por lo que se determinan como el componente con más peso económico. En el sector tecnológico el equipamiento es caro y tiene un coste elevado, los detalles de la tarificación del producto y su utilización se presentan en el plan de escalabilidad.

11.3 AMORTIZACIONES

En esta partida se recoge el coste de aquellos materiales que ya se encuentran disponibles previamente al inicio del proyecto, como es el caso de los servidores VMware. Por tanto, las amortizaciones son el coste de los activos fijos que se utilizan para el proyecto. La parte amortizable del material a imputar en el proyecto es proporcional al tiempo que este se destina al proyecto dentro de su vida útil.

Se hará uso de tres ordenadores portátiles de precio 1000 €. Para un equipo de esas características se considera una vida útil de unos 5 años.

Para la documentación y seguimiento del proyecto se utiliza el software propietario Microsoft Office, con una duración igual a la del portátil y un coste de 300€.

Además, los servidores VMware, cuyo coste asciende a 25.000€ y de los que se espera que disponga de una vida útil de 10 años se utilizarán para este proyecto y otro de forma simultánea.

El desglose de las amortizaciones puede verse en la **tabla 16**:

Amortizaciones					
Material	Cantidad	Coste total (/unidad)	Vida útil(h)	Tiempo de uso(h)	Coste final(€)
Ordenadores	3	1.000,00 €	43800 h	2664 h	182,47 €
Servidor VMWare	2	25.000,00 €	87600 h	800 h	456,62 €
Licencia Office 365	3	300,00 €	43800 h	60 h	1,23 €
Total					640,32 €

Tabla 16: Amortizaciones del proyecto

Como se puede apreciar, las amortizaciones componen la menor parte del coste del proyecto.

11.4 COSTE TOTAL

Finalmente, para obtener el coste total, se realiza la suma de los costes parciales de los apartados anteriores, además de un porcentaje de los anteriores, para considerar los gastos indirectos que no se pueden imputar al proyecto.

Coste total	
Concepto	Coste(€)
Horas internas	15.400,00 €
Amortización	640,32 €
Gastos	111.050,00 €
Total	127.090,32 €

Tabla 17. Coste total del proyecto

Como se puede observar en la tabla anterior, el coste total del proyecto es de 127.090,32 €. Este coste refleja que el proyecto de implantación de un SIEM es un proyecto costosa y para empresas con una gran madurez en seguridad.

12 CONCLUSIONES

En este proyecto se ha logrado la securización y vigilancia de una infraestructura de red privada, proporcionando una mayor madurez en el sector de la ciberseguridad. Mediante la implantación de un sistema SIEM y el CyberSoC que vigilara las incidencias que el mismo cree, se le ha proporcionado una capa más de seguridad a la infraestructura cliente.

Adicionalmente, se ha diseñado una solución de *Machine Learning* que ayudara a identificar el trafico malicioso de una manera más veloz y dinámica. Esta solución no es una mera herramienta o solución de ciberseguridad, también aporta un valor diferencial y un paso de gigante hacia la modernización y automatización del cliente.

En términos globales, el proyecto ha aportado seguridad, prestigio y automatización al cliente, siendo altamente fructífero para ambas partes del proyecto. Además, las características del proyecto fomentan una gran relación con el cliente.

La relación futura con el cliente se verá altamente influenciada gracias a la ejecución de este proyecto. Además, también será de ayuda para futuros proyectos de índole similar con otros clientes, aportando experiencia, confianza y valor.

Fuera de la relación empresarial del proyecto, también es un proyecto con un alto valor tecnológico y científico. Este proyecto ayuda a sentar precedentes en el mundo de la ciberseguridad, puesto que innova usando conceptos modernos como ML. Además, comienza a normalizar la implantación de sistemas como SIEM en el entorno industrial del País Vasco.

En el entorno social, al ser el cliente un proveedor de servicios públicos, la sociedad ganara en tranquilidad, continuidad de estos servicios públicos y transparencia en la gestión de la seguridad.

Concluyendo, el proyecto ha aportado valores en el ámbito económico, tecnológico y social, considerando estos 3 de los puntos fuertes de la ejecución de un proyecto. Además, ha sentado una base y referencia para futuros proyectos en el ámbito de la ciberseguridad, tanto a nivel local de la empresa como a nivel global de la industria por la publicidad de este documento y su información.

13 ANEXOS

13.1 CONFIGURACIÓN INICIAL DEL *EVENT COLLECTOR*

En este apartado se incluye el proceso de configuración inicial de la colectora de eventos, justo después del despliegue de la misma, dicho proceso se detalla a continuación:

- Primero se elige el tipo de QRadar que se va a desplegar, en este caso virtual o instalación *software*:

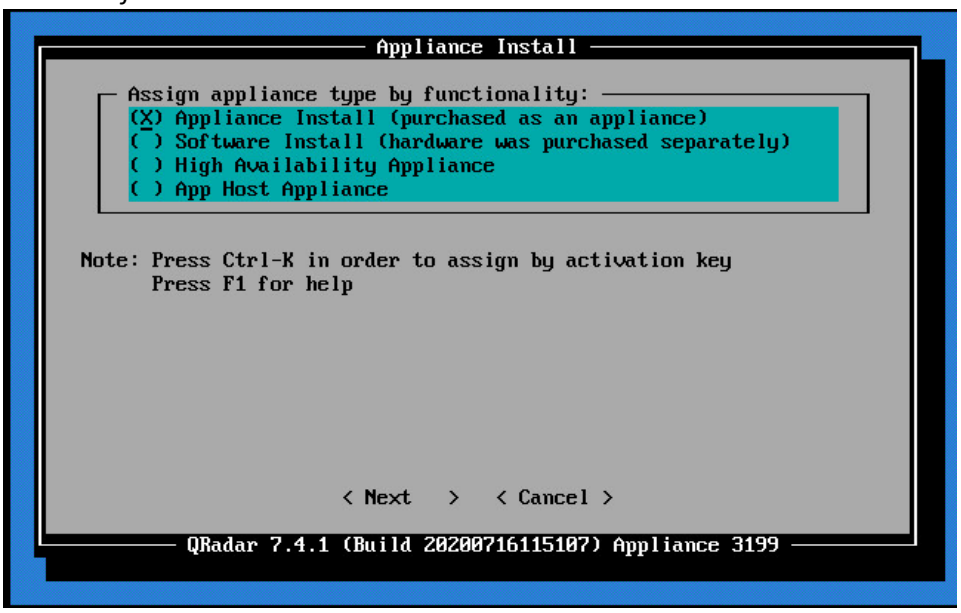


Figura 80: Tipo de instalación QRadar

- Posteriormente, se elige que funcionalidad de QRadar se va a desplegar, en este caso el *event collector*:

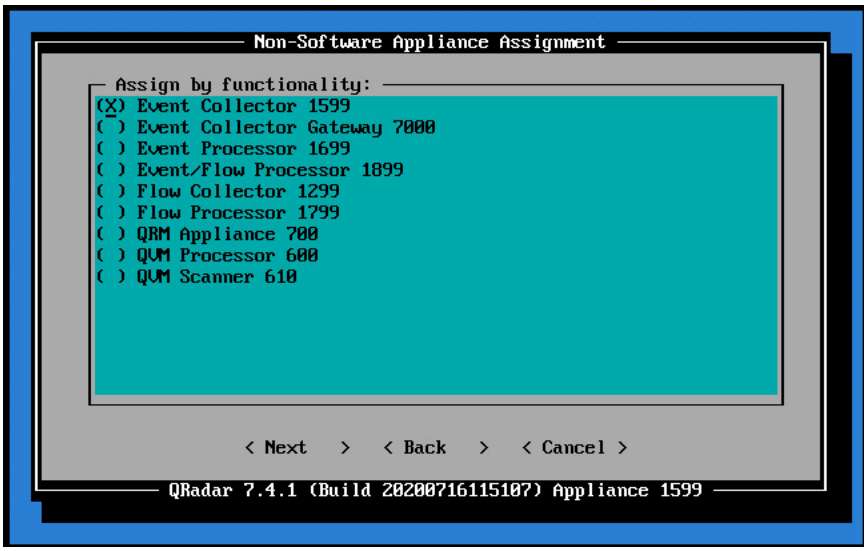


Figura 81: Elección de event collector

- Después, abraque elegir si se va a montar un despliegue de *high availability* o no:

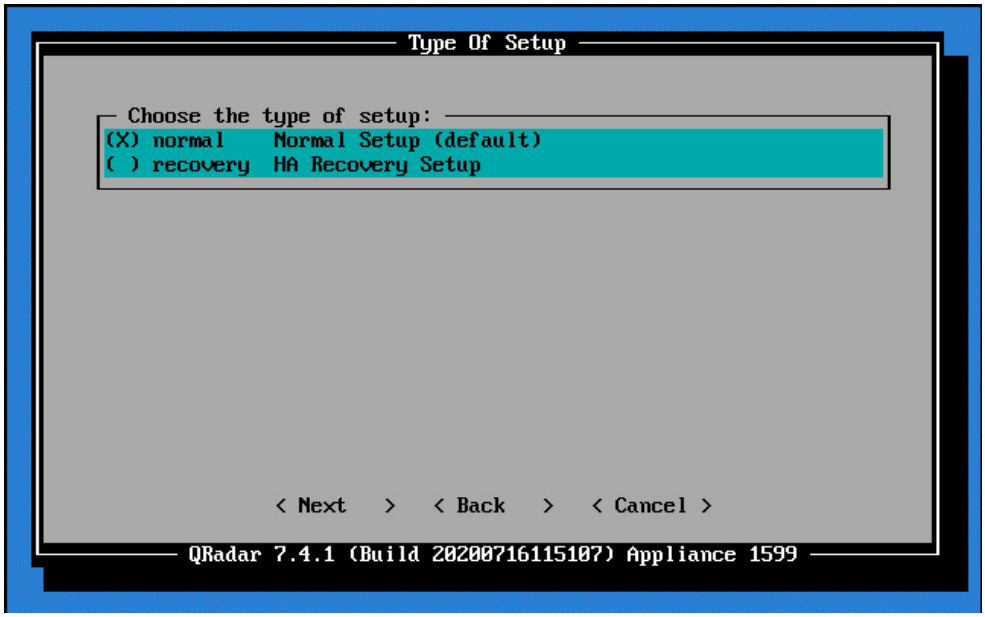


Figura 82: Tipo de despliegue (standalone o HA)

- Configuración a nivel de IP (IPv4 o IPv6):

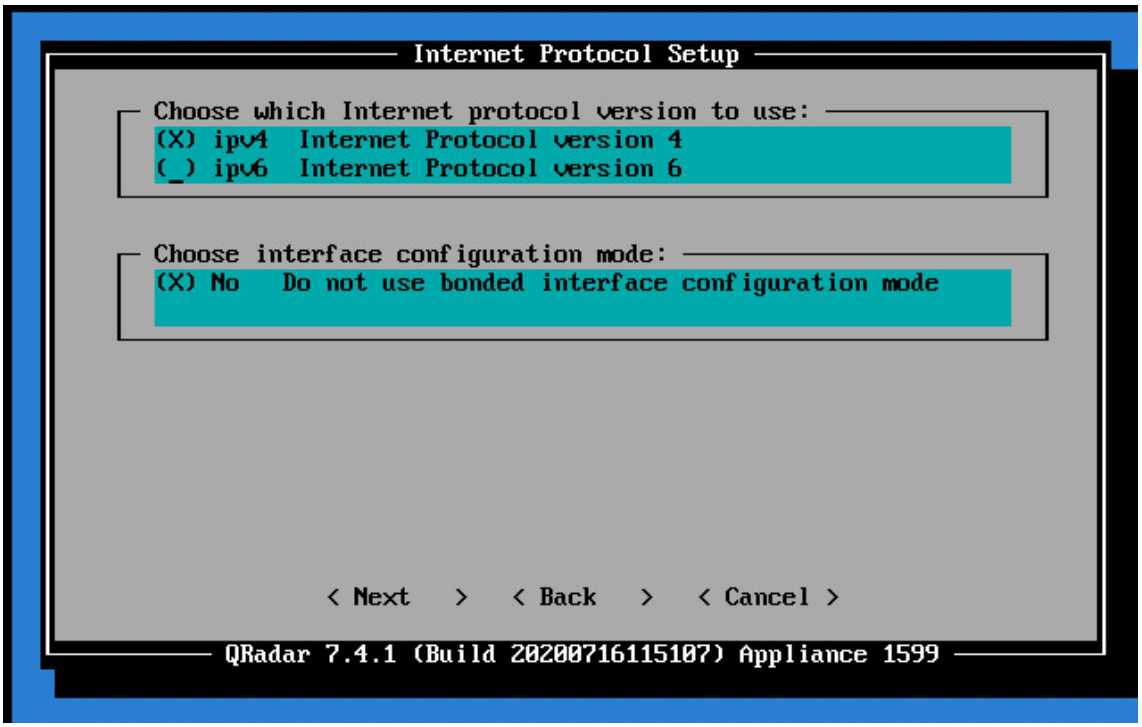


Figura 83: Elección de versión IP de event collector

- Lo siguiente será configurar la dirección IP y demás configuraciones de red de la maquina:

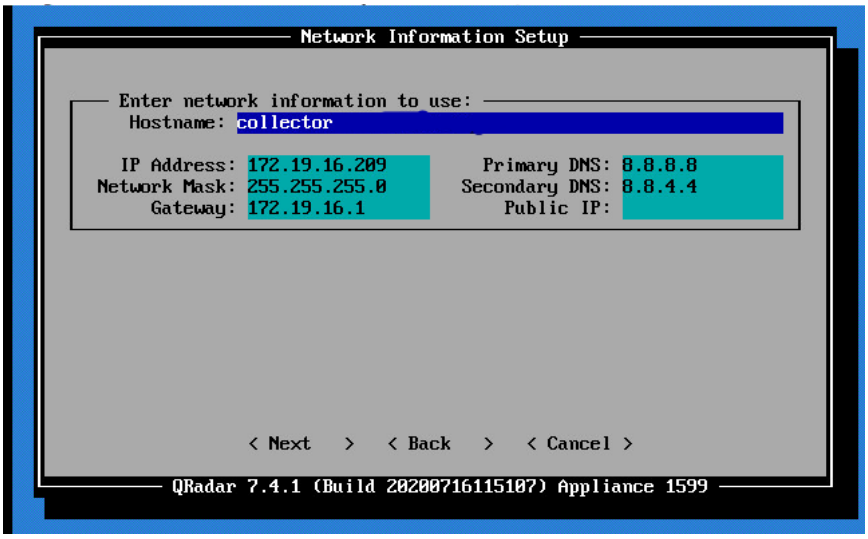


Figura 84: Configuración de red de la colectora

- configuración de la interfaz de gestión:

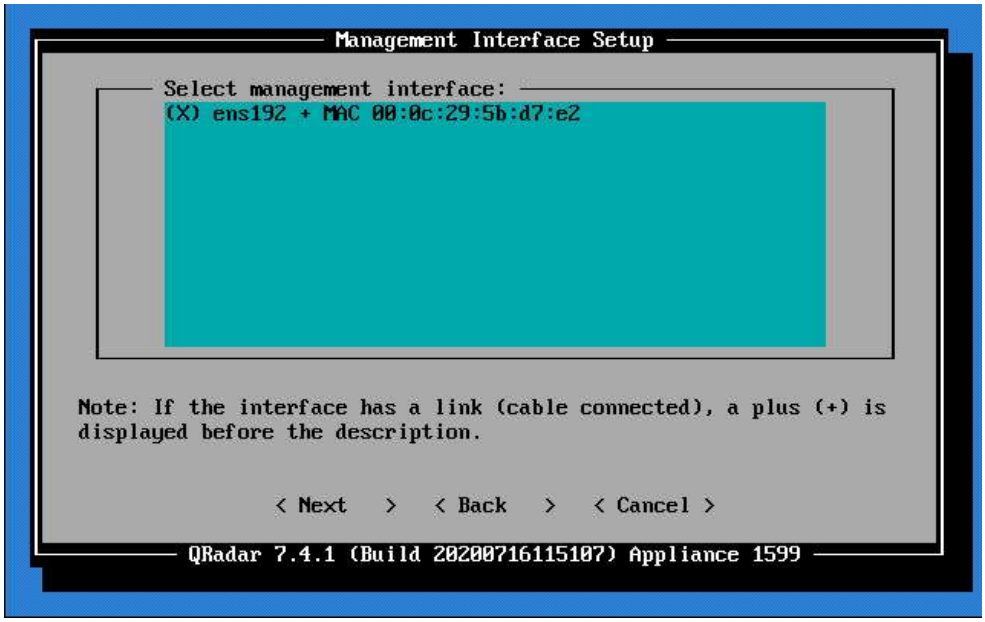


Figura 85: Elección de la interfaz de gestión

Una vez finalizado este proceso la maquina quedara completamente configurada.

13.2 DESPLIEGUE DE COLECTORES DE EVENTOS DE HOST

En este apartado se describirá el proceso que se ha seguido para desplegar los servidores Wincollect y los clientes *sysmon* en la arquitectura del cliente, también se añadirá como se ha realizado la configuración en los sistemas Linux.

13.2.1 Despliegue en sistemas Windows

Para comenzar, se desplegará Wincollect en un Windows Server y posteriormente, se instalará su consola de gestión. Es muy recomendable que se desplieguen en un servidor dedicado.

Se ejecuta el instalador *wincollect-7.2.9-103.x64* y se clica en siguiente:

Figura 86: Instalacion Wincollect I

Se acepta el acuerdo de licencia:

Figura 87: Instalación Wincollect II

Se introduce el usuario de Windows con el que se va a lanzar el instalador.

Organization: EMPRESA

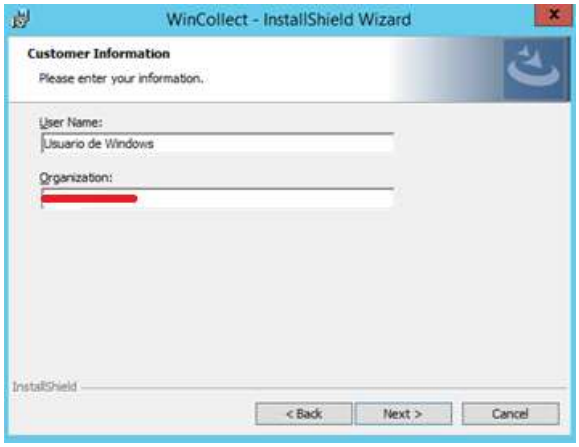


Figura 88: Instalación Wincollect III

Se selecciona el directorio donde se va a instalar *Wincollect* o se deja el de por defecto:

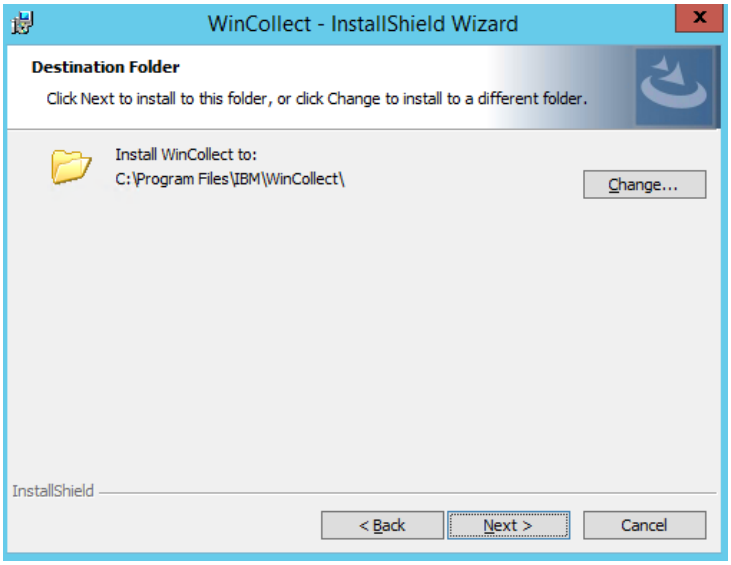


Figura 89: Instalación Wincollect IV

Setup Type: Seleccionar *Stand Alone*

Figura 90: Instalación Wincollect V

Clicar en "*Create a Log Source*"

Log Source Name: wincollect-sysmon

Log Source Identifier: win-sys

En *Event Logs,* seleccionar los que se muestran en la captura:

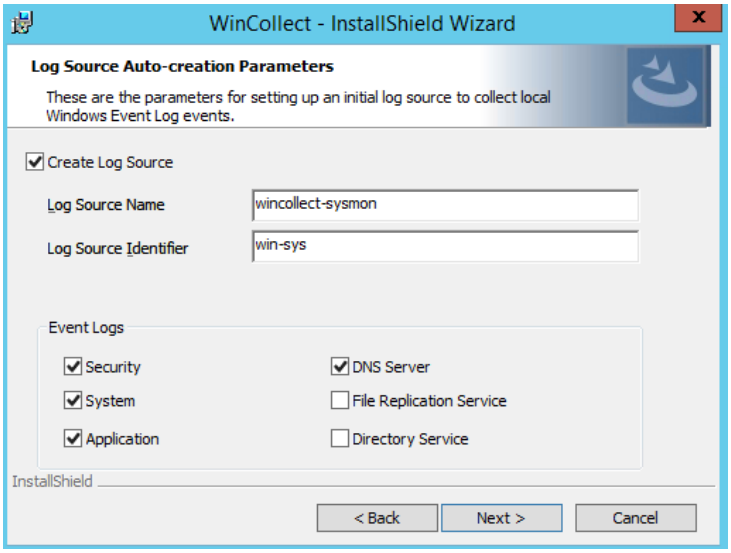


Figura 91: Instalación Wincollect VI

Destination Name: QRadar

Hostname / IP: 10.101.101.160

Figura 92: Instalación Wincollect VII

En *Event Rate Tuning Profile* seleccionar *Typical Server –500/750* tal y como se muestra a continuación:

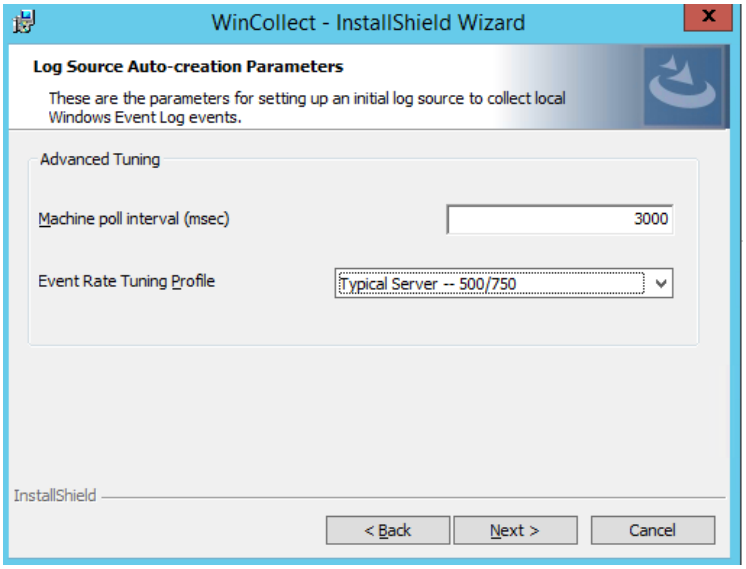


Figura 93: Instalación Wincollect VIII

Clicar en *Next*:

Figura 94: Instalación Wincollect IX

Clicar en *Next*:

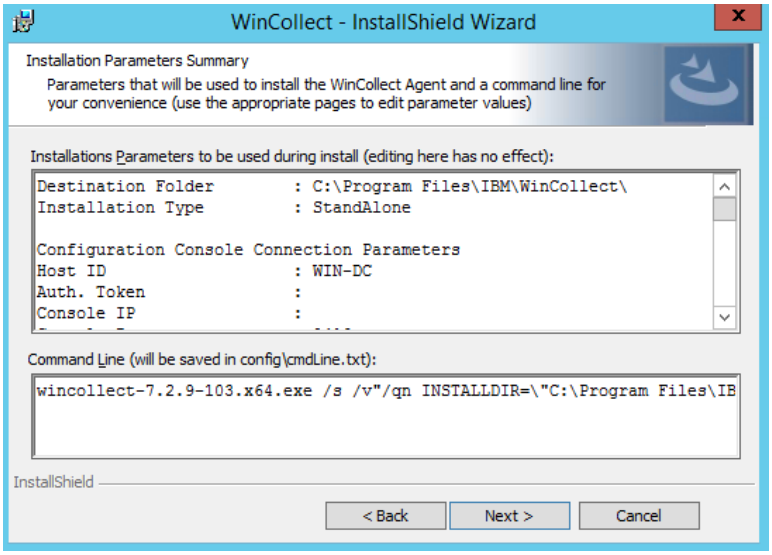


Figura 95: Instalación Wincollect X

Clicar en *Install:*

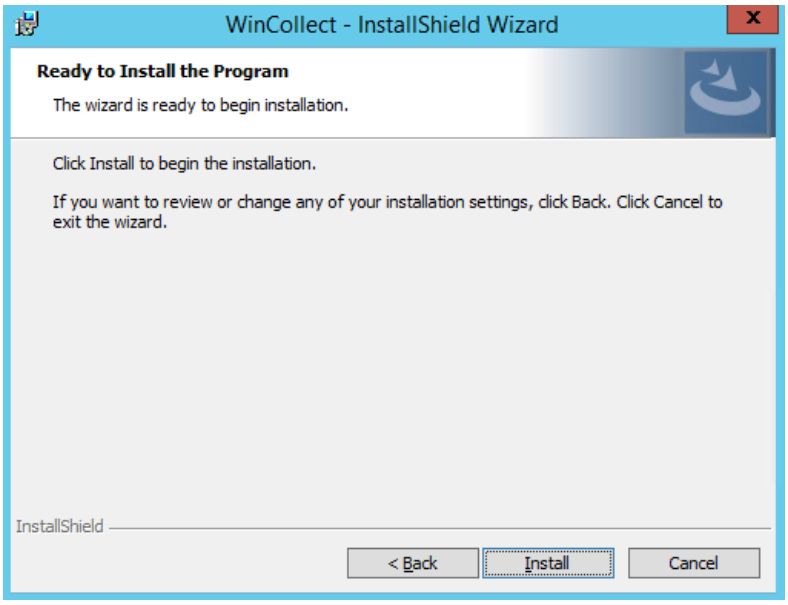


Figura 96: Instalación Wincollect XI

Una vez realizado el procedimiento anterior, se habrá dejado instalado el servidor Wincollect. Después de instalar el servidor tocaría instalar la consola de gestión de Wincollect para poder gestionar de una manera más sencilla la colección de eventos.

Para ello se ejecuta *wincollect-standalone-patch-installer-7.2.9-103* para proceder con la instalación de su consola de gestión.

Clicar en Sí:

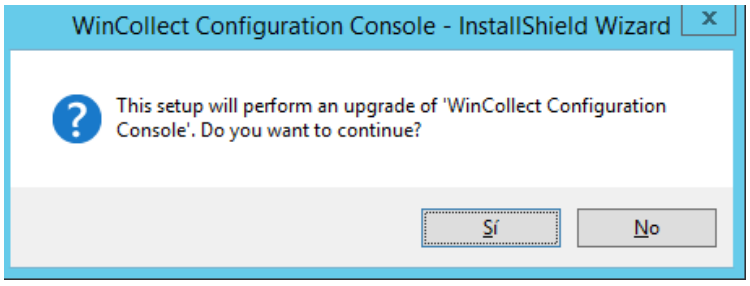


Figura 97: Instalación consola gestion Wincollect I

Clicar en *Next:*

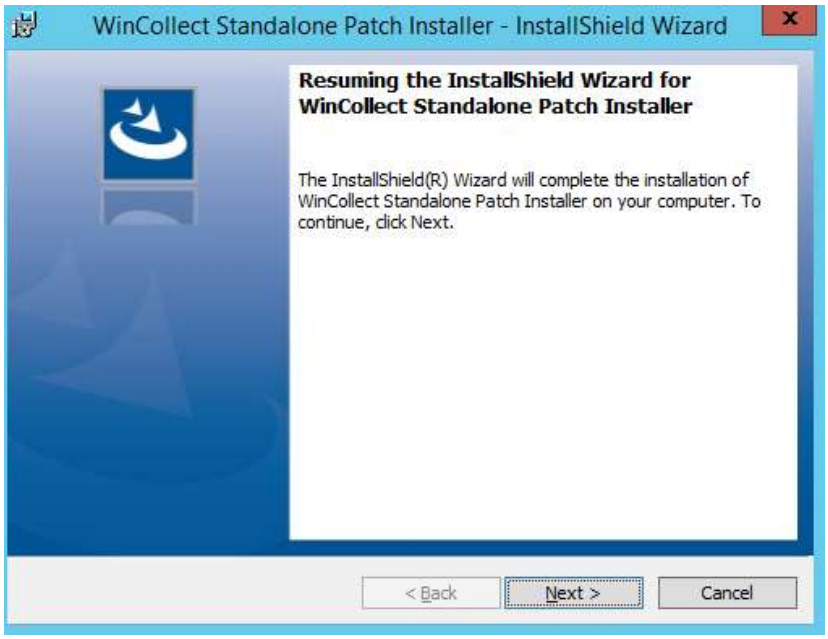


Figura 98: Instalación consola gestión Wincollect II

Se acepta el acuerdo de licencia:

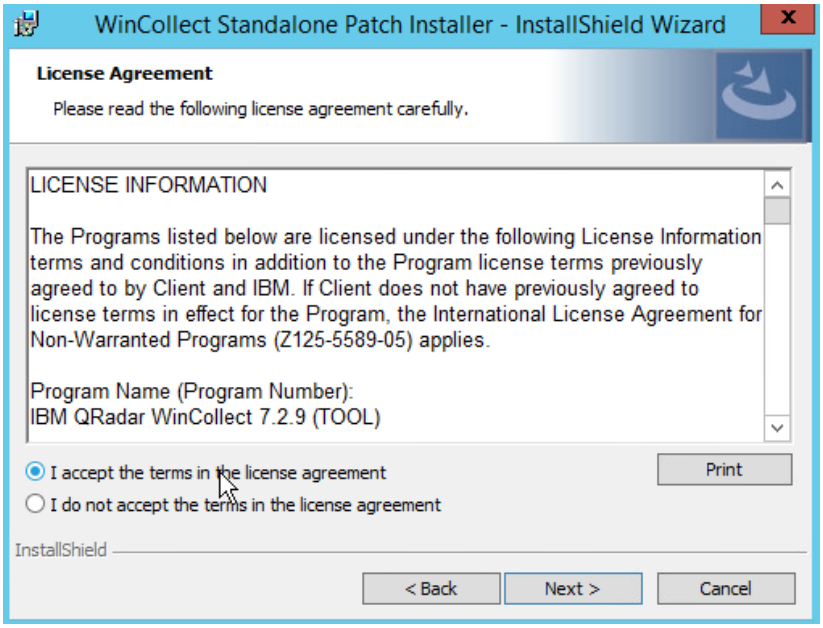


Figura 99: Instalación consola gestión Wincollect III

Se introduce el usuario de Windows con el que se va a lanzar el instalador.

Organization: Empresa

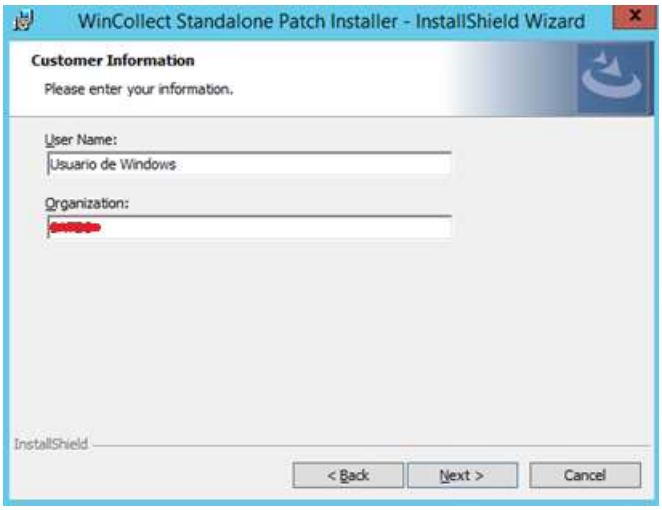


Figura 100: Instalación consola gestión Wincollect IV

Clicar en Next:

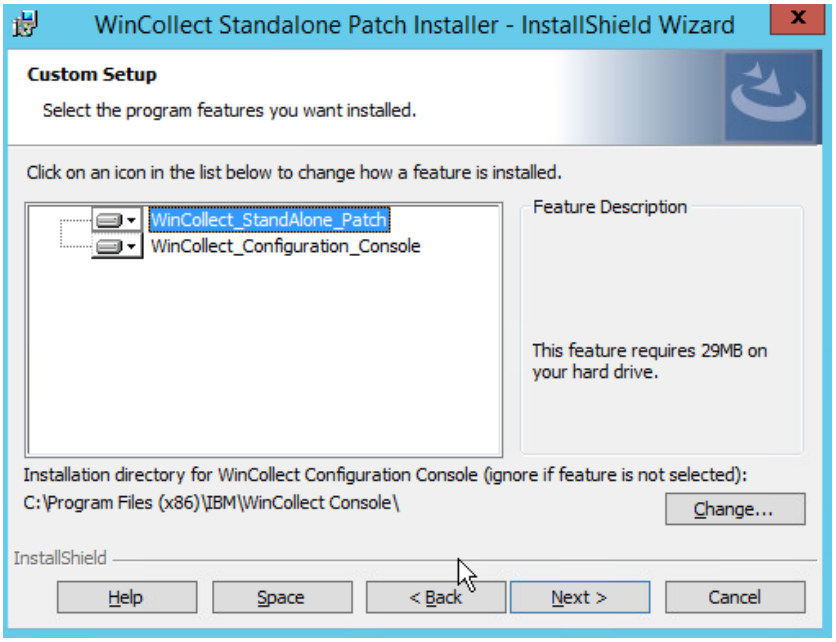


Figura 101: Instalación consola gestión Wincollect V

Clicar en *Install*:

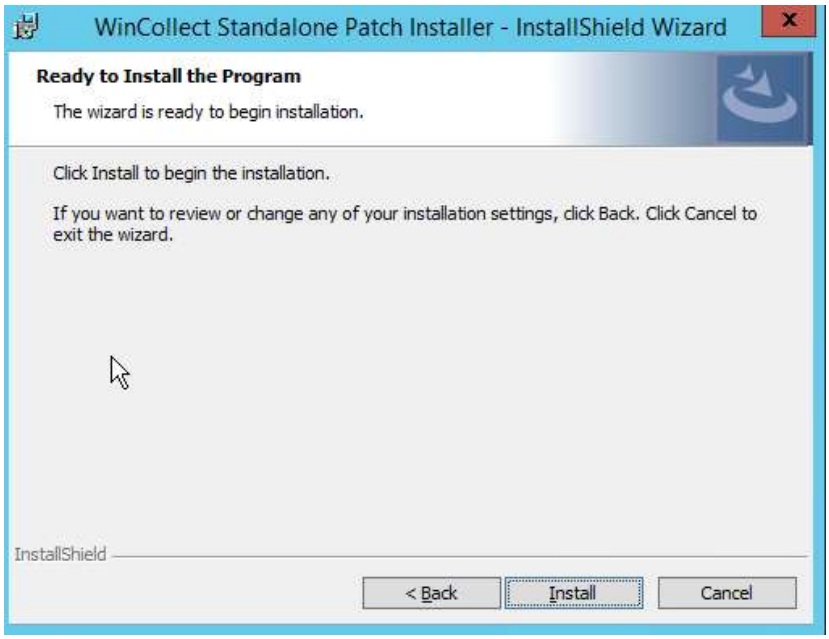


Figura 102: Instalación consola gestión Wincollect VI

Con esto se finalizará la instalación de la consola de gestión y habrá que instalar el *software* sysmon en los clientes Windows.

Una vez instalado *Wincollect*, se procede a instalar y configurar *Sysmon* en los PCs o servidores que se quiera monitorizar y en el *Windows Server* que actúa como colector.

En primer lugar, tendremos que ejecutar como administrador el Símbolo del Sistema y ubicarnos en el directorio donde se encentra la carpeta *Sysmon* proporcionada.

A continuación, ejecutar el siguiente comando:

```
sysmon.exe -accepteula -i sysmonconfig-export.xml
```

En primer lugar, se comprueba que el servicio *winrm* está corriendo tanto en el Windows Server que recopila los eventos como en los servidores y PCs que se quiera monitorizar. Para ello, se ejecutará a través de PowerShell el siguiente comando:

```
winrm quickconfig
```

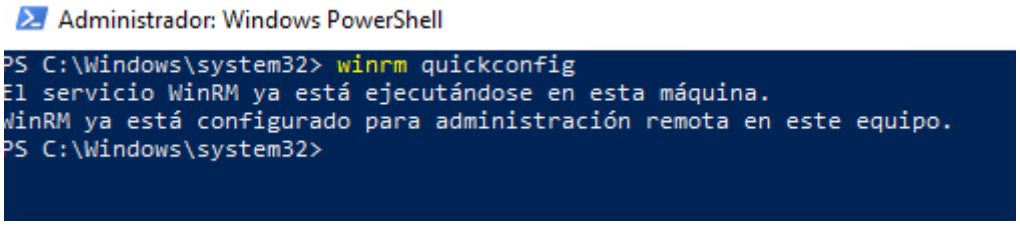


Figura 103: comprobación estado sysmon

Una vez instalado *Sysmon* y verificado que el servicio *winrm* está ejecutándose, se procede a reenviar sus eventos hacia el Windows Server donde se encuentra instalado *Wincollect*.

Desde los sistemas Windows que se quieran monitorizar, se deberá indicar el equipo que tiene permisos para leer los eventos. Para ello, se tendrá acceder al panel de *Administración del Equipo de Windows*, clicar en *Usuarios y grupos locales* →*Grupos*, y acceder a los *Lectores del registro de eventos* tal y como se muestra en la siguiente captura:

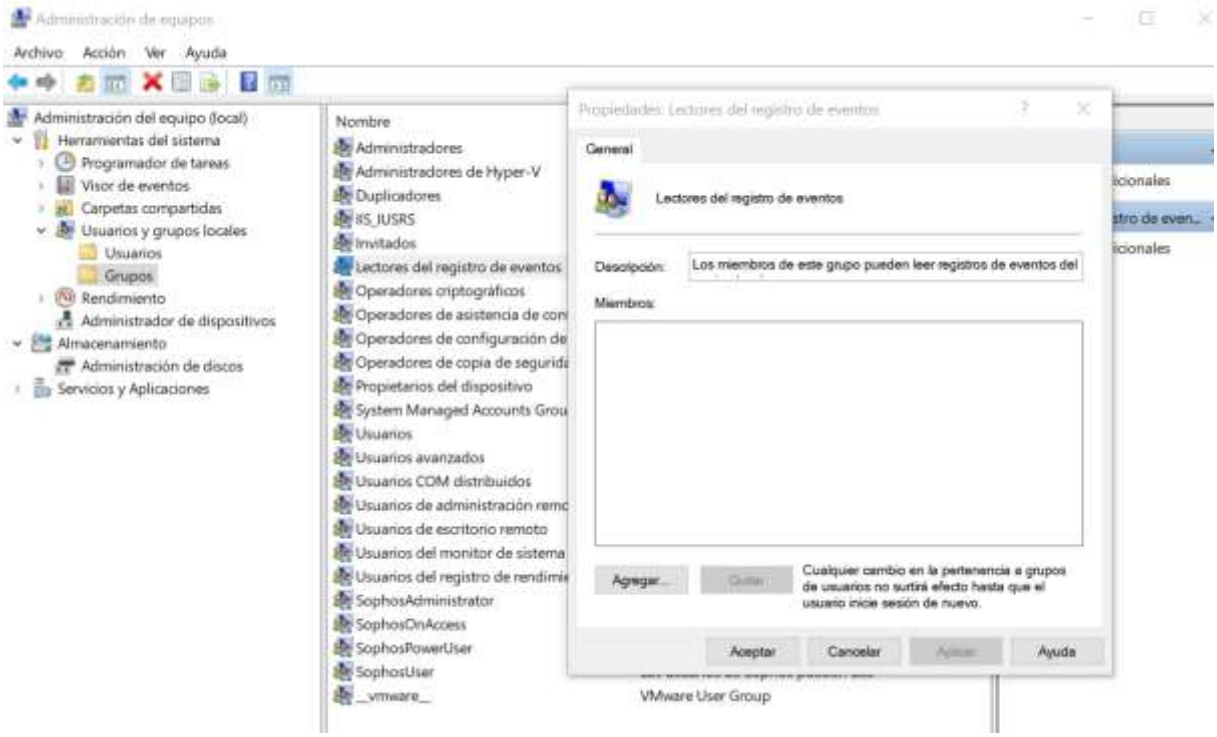


Figura 104: Asignación permisos de lectura de eventos I

Una vez se ha abierto el panel de *Lectores del Registro de eventos*, se tendrá que agregar el Windows Server en el que se encuentra desplegado *Wincollect.* Para ello, clicar en *Agregar*→*Tipos de objeto* y seleccionar *Equipos*:

Figura 105: Asignación permisos de lectura de eventos II

A continuación, se introducirá el *hostname* del Windows Server y se clicará en *Comprobar nombres* para verificar que es visible desde el equipo desde donde se quieren enviar los eventos.

En la siguiente captura, se muestra un ejemplo con uno de los PCs de Empresa en dominio:

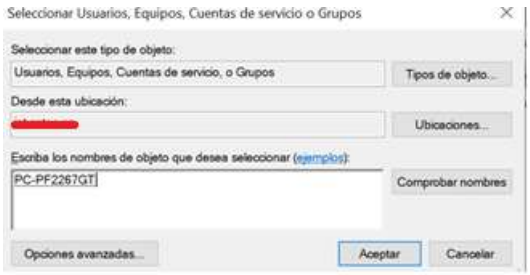


Figura 106: Asignación permisos de lectura de eventos III

Una vez agregado, clicar en aceptar:

Figura 107: Asignación permisos de lectura de eventos IV

Tras realizar este cambio, *Wincollect* tendrá acceso para leer los eventos del equipo.

Una vez se ha dado permiso al Windows Server para leer los eventos, se tendrá que crear una suscripción en el Windows Server para poder recibirlos.

Para crear una suscripción, se deberá abrir el *Visor de Eventos*, clicar con el botón derecho del ratón en *Suscripciones* y seleccionar *Crear una suscripción* tal y como se muestra a continuación:

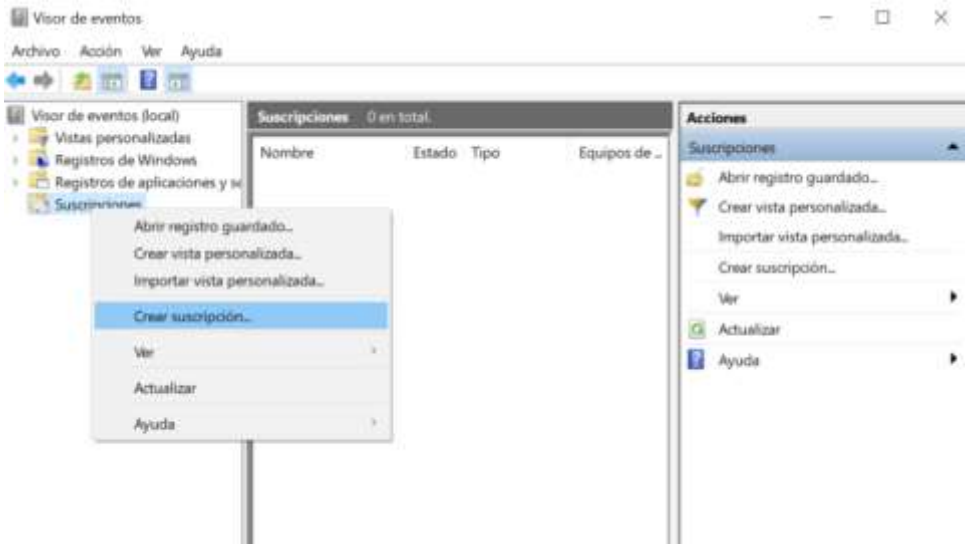


Figura 108: Subscripcion del servidor syslog

Una vez en el panel *de Propiedades suscripción: Eventos de Sysmon,* se dará un nombre a la suscripción y, en *Tipos de Suscripción y equipos de origen*, se marcará la opción *Iniciada por el equipo de origen* tal y como se muestra en la siguiente captura:

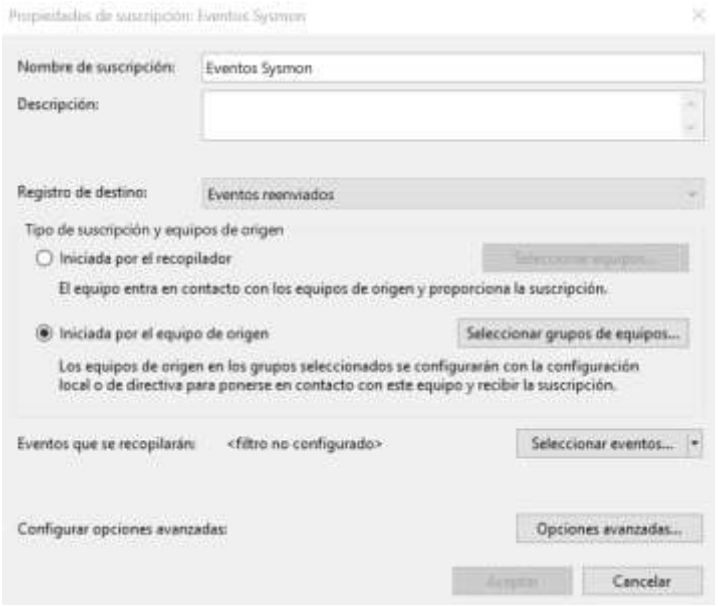


Figura 109: Configuración suscripción I

Posteriormente clicar en *Seleccionar grupos de equipos → Agregar equipos de dominio* e introducir el *hostname* del equipo del que se quieren recibir los logs.

Después de haber verificado que hay conectividad con el equipo mediante el botón de *Comprobar nombres* tal y como se indicaba en los anteriores parrafos, se tendrán que configurar los eventos que se van a recopilar.

En el mismo panel *Propiedades suscripción: Eventos de Sysmon,* clicar en *Seleccionar eventos:*

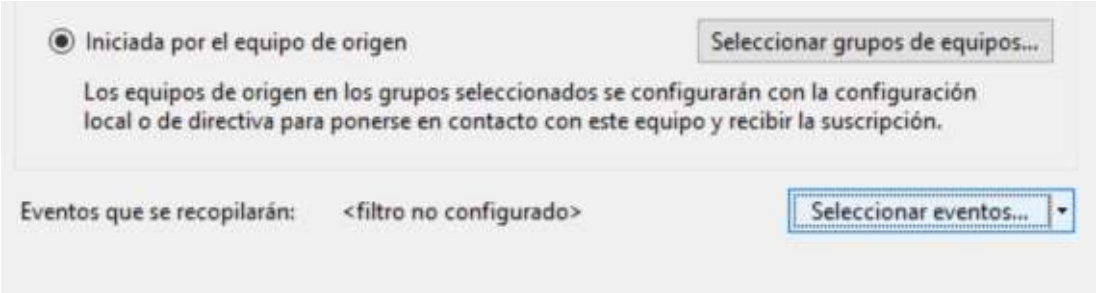


Figura 110: Configuración suscripción II

En *Nivel del Evento*, marcar todas las casillas. Seleccionar *Por registro* y desplegar *Registros de Aplicaciones y servicios →Microsoft →Windows* y seleccionar *Sysmon* tal y como se muestra a continuación

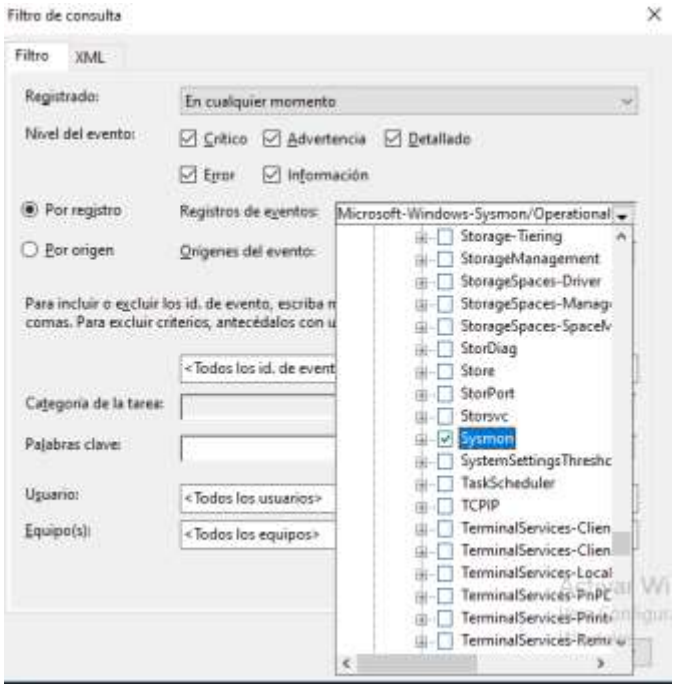


Figura 111: Configuración suscripción III

Aceptar:

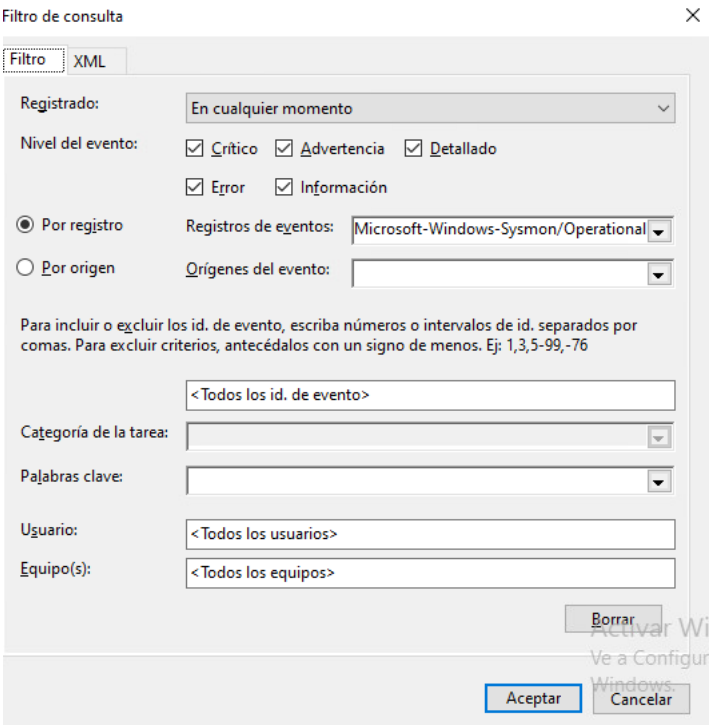


Figura 112: Configuración suscripción IV

Clicar en *Opciones avanzadas y en Optimización de entrega del evento*, seleccionar *Minimizar latencia*

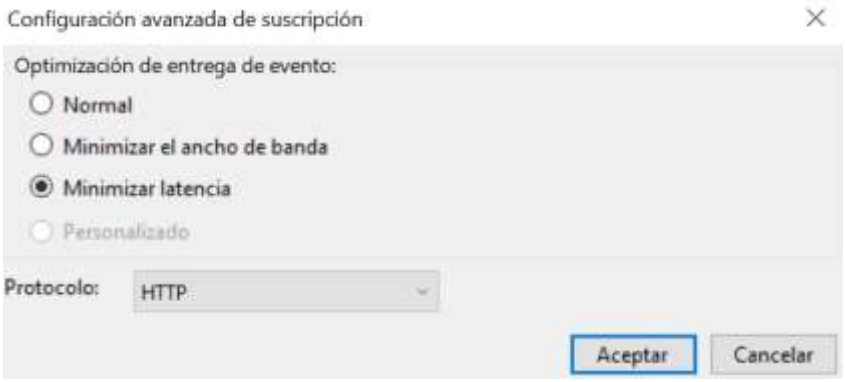


Figura 113: Configuración suscripción V

Finalmente, clicar en *Aceptar* para salir de la ventana de *Optimización de entrega de evento* y nuevamente en *Aceptar* para crear la subscripción.

Se podrán verificar que se reciben los logs de *Sysmon* en el mismo *Visor de Eventos* accediendo a los *Registros de Windows → Eventos reenviados.*

Cabe destacar, que para que se envíen correctamente los eventos desde los equipos Windows al Windows Server se tendrá que permitir en los FWs a través del puerto 80 (HTTP)

Una vez se ha desplegado *Sysmon* y *Wincolect*, se procede a configurar la consola de *Wincollect.* En primer lugar, nos ubicamos en *Devices →wincollect-sysmon* tal y como se muestra a continuación:

Figura 114:Configuración consola Wincollect I

Posteriormente, en *XPath Query*, introducimos la siguiente consulta:

```
<QueryList>
 <Query Id="0" Path="Microsoft-Windows-Sysmon/Operational">
  <Select Path="Microsoft-Windows-Sysmon/Operational">*</Select>
 </Query>
</QueryList>
```

Figura 115: Configuración consola Wincollect II

Cada vez que mandemos los eventos de un nuevo equipo al Windows Server con *Wincollect* es necesario crear un nuevo dispositivo (*Add New Device*) tal y como se muestra en la siguiente imagen:

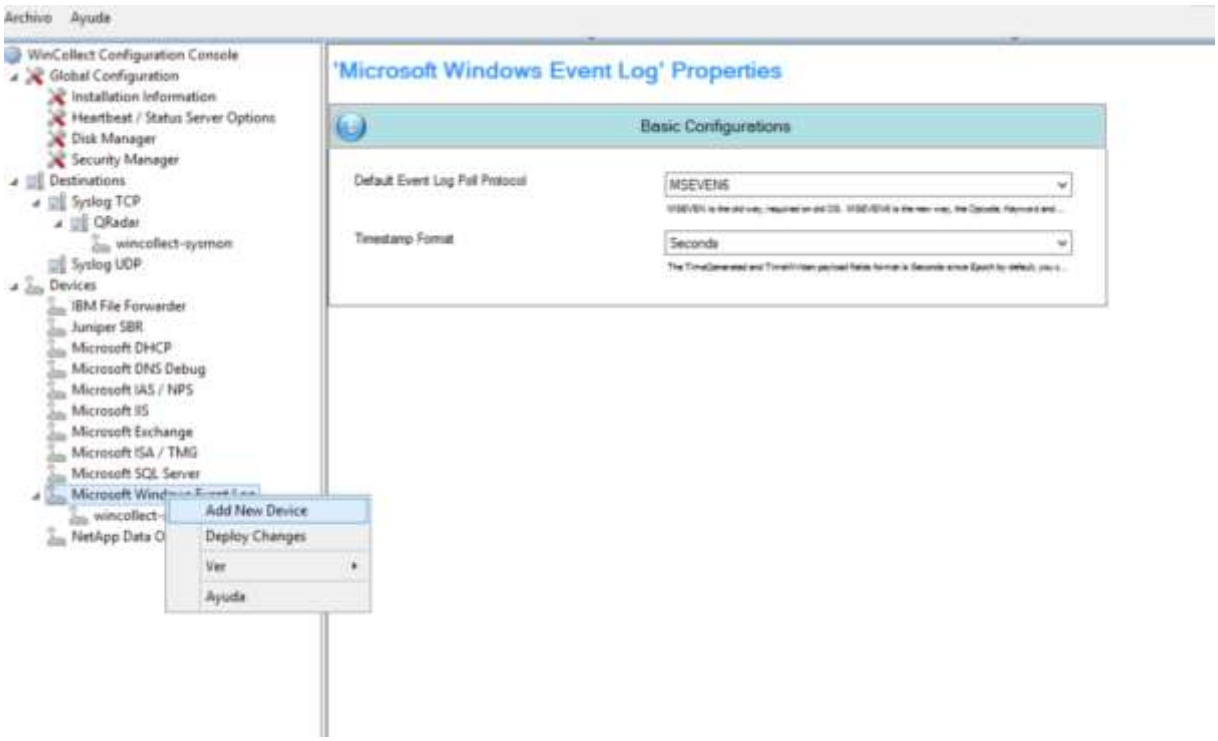


Figura 116: Configuración consola Wincollect III

En *Device Address*, se tendrá que introducir la IP con la que se presentan los eventos del nuevo equipo añadido:

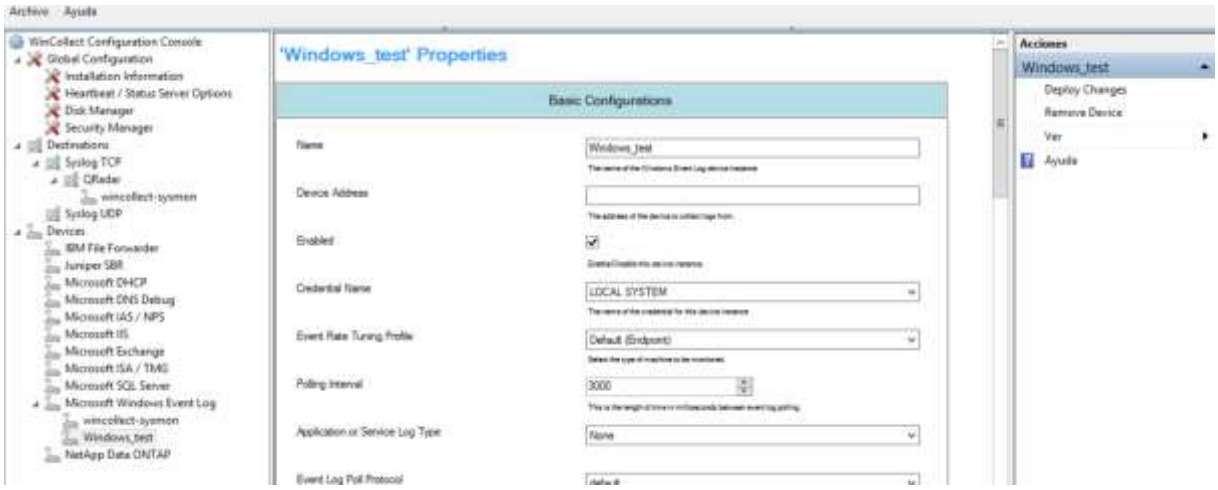


Figura 117: Configuración consola Wincollect IV

Y por último, tal y como se indico anteriormente, se deberá incluir la consulta en *XPath Query*:

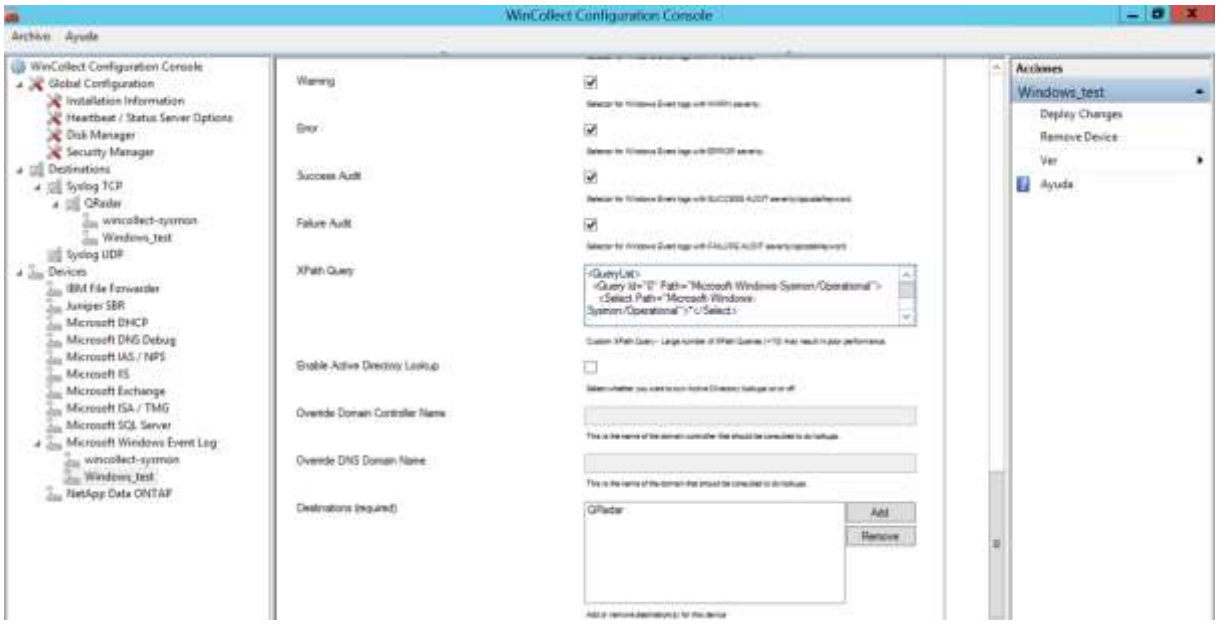


Figura 118: Configuración consola Wincollect V

Finalmente, en el panel de arriba la derecha, se tendrá que clicar sobre *Deploy Changes* para aplicar los cambios aplicados.

Con este procedimiento ya estaria configurado el reenvio de evento de maquinas Windows.

13.2.2 Despliegue en sistemas Linux[17]

Como ya se menciono en el diseño, el reenvio en las maquinas Linux se realizara directamente contra la colectora de QRadar sin usar ningun servidor auxiliar. Para ello habra que analizar si la dependencia *rsyslog* esta instalada, con el siguiente comando:

```
rpm -qa | grep -i rsyslog rsyslog-5.10.1-0.11.1
```

En caso de no estar instalado, habrá que instalar la dependencia con el instalador correspondiente al sistema, con *zypper* por ejemplo se haría de la siguiente manera:

```
zypper install rsyslog
```

Como viene siendo habitual en instalaciones en entornos Linux, habrá que retocar el fichero de configuración alojado en /etc/sysconfig/syslog. Este archivo contiene la configuración del servicio *syslog*, habría que añadir la siguiente línea:

```
SYSLOG_DAEMON="rsyslogd" RSYSLOGD_COMPAT_VERSION=”4?
```

Con esta línea se indica que el demonio *rsyslogd* será usado para el protocolo *syslog*. También habrá que modificar el archivo de configuración del servicio *rsyslog*, alojado en /etc/rsyslog.d/remote.conf, añadiendo la siguiente línea:

```
@@:IP Address of QRadar event Collector:514 #send all log events to QRadar via tcp
```

Con esta configuración ya se enviarían los *logs* al QRadar.

13.3 CARACTERÍSTICAS DEL *DATASET* DE ML

En este apartado se incluirá el formato del *dataset* de ML:

No.	Name	Type	Description
1	srcip	nominal	Source IP address
2	sport	integer	Source port number
3	dstip	nominal	Destination IP address
4	dsport	integer	Destination port number
5	proto	nominal	Transaction protocol
6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	sbytes	Integer	Source to destination transaction bytes
9	dbytes	Integer	Destination to source transaction bytes
10	sttl	Integer	Source to destination time to live value
11	dttl	Integer	Destination to source time to live value
12	sloss	Integer	Source packets retransmitted or dropped
13	dloss	Integer	Destination packets retransmitted or dropped
14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
19	swin	integer	Source TCP window advertisement value
20	dwin	integer	Destination TCP window advertisement value
21	stcpb	integer	Source TCP base sequence number
22	dtcpb	integer	Destination TCP base sequence number
23	smeansz	integer	Mean of the ?ow packet size transmitted by the src
24	dmeansz	integer	Mean of the ?ow packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time

31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	integer	No of flows that has a command in ftp session.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

Tabla 18: Características dataset ML

13.4 CÓDIGO DE LA SOLUCIÓN DE ML

En este apartado se incluye el código completo del programa desarrollado para la solución de ML, para que, en caso de querer analizar el código, se pueda observar. A continuación, se muestra el código:

```
# Se cargan las librerias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Modelos
from sklearn.svm import SVC, LinearSVC
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.neural_network import MLPClassifier
from imblearn.ensemble import BalancedRandomForestClassifier
from sklearn.linear_model import MultiTaskLassoCV

# Preprocesado, por ejemplo feature stanardization
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PolynomialFeatures

# Separación de los datos train/test
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import RandomizedSearchCV
from sklearn.impute import SimpleImputer
# Distribuciones
from scipy.stats import uniform

# Evaluación de los modelos
from sklearn.metrics import r2_score
from sklearn.metrics import explained_variance_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import _regression
#metricas de los modelos
from sklearn.metrics import accuracy_score, recall_score, precision_score, confusion_matrix,
roc_curve, roc_auc_score
from sklearn.metrics import precision_recall_curve, average_precision_score, f1_score
```

```
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, roc_auc_score

from sklearn.metrics import plot_confusion_matrix, plot_roc_curve, plot_precision_recall_curve

# cargamos los datos

df = pd.read_csv('UNSW-NB15_1_nh.csv',header=None, thousands='.',sep=';',low_memory=False)
df.head(5)


# se listan las clases
pd.value_counts(df.values[:,-2])
pd.value_counts(df.values[:,-1])

# preprocesamiento de los datos

X=df.values[:,:-2]

# codificación de los strings a numeros
enc=LabelEncoder()
X[:,4]=enc.fit_transform(X[:,4])

enc=LabelEncoder()
X[:,5]=enc.fit_transform(X[:,5])

enc=LabelEncoder()
X[:,13]=enc.fit_transform(X[:,13])

enc=LabelEncoder()
X[:,39]=enc.fit_transform(X[:,39])

X=X[:,4:]
X=X.astype(float)

np.mean(X,axis=1)
imp=SimpleImputer(missing_values=np.nan,strategy='mean')
imp.fit(X)# habria que usar solo el training
X=imp.transform(X)
np.mean(X,axis=1)

y1=df.values[:,-1]
y1=y1.astype('int')
y2=df.values[:,-2]
Xtr1, Xte1, ytr1, yte1= train_test_split(X,y1,test_size=0.3,random_state=0)
```

```
Xtr2, Xte2, ytr2, yte2= train_test_split(X,y2,test_size=0.3,random_state=0)

# regresión logística

regL=LogisticRegression(class_weight='balanced')
regL=regL.fit(Xtr1,ytr1)
ypred=regL.predict(Xte1)
acc=100*accuracy_score(yte1,ypred)
print('La precision del modelo de regresion logistica: %{Acc:.2f}'.format(Acc=acc))
print(classification_report(yte1,ypred))
print('Matriz de confusion:\n')
print(confusion_matrix(yte1,ypred))

# Visualizacion de la matriz de confusion
plot_confusion_matrix(regL, Xte1, yte1,\
                      normalize='true',cmap=plt.cm.Blues,display_labels={'Yes','No'})
ax=plt.gca()
fig=plt.gcf()
fig.set_size_inches(8,8)
ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')

# SVM
# SVM clásico balanceado
n_estimators=50
svm=OneVsRestClassifier(BaggingClassifier(LinearSVC(class_weight='balanced'),
max_samples=1.0 / n_estimators, n_estimators=n_estimators),n_jobs=-1)

# SVM usando randomized search
kf=KFold(n_splits=3)
search_space=dict(C=uniform(loc=1,scale=100))
svc=LinearSVC(max_iter=20000)
svcL=RandomizedSearchCV(svc,search_space,random_state=0,scoring='accuracy',cv=kf,n_it
er=20)
ypred=svcL.fit(Xtr1,ytr1)
svm=svm.fit(Xtr1,ytr1)
ypred=svm.predict(Xte1)
acc=100*accuracy_score(yte1,ypred)
print('La precision del modelo de Support vector machine: %{Acc:.2f}'.format(Acc=acc))

print(classification_report(yte1,ypred))
print('Matriz de confusion:\n')
print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(svm, Xte1, yte1,\
                      normalize='true',cmap=plt.cm.Blues, display_labels={'Normal','Attack'})
ax=plt.gca()
fig=plt.gcf()
```

```
fig.set_size_inches(8,8)
ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')

# Random forest

rfc=RandomForestClassifier()
rfc=rfc.fit(Xtr1,ytr1)
ypred=rfc.predict(Xte1)

acc=100*accuracy_score(yte1,ypred)
print('La precision del modelo de Random Forest: %{Acc:.2f}'.format(Acc=acc))

print(classification_report(yte1,ypred))
print('Matriz de confusion:\n')
print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(rfc, Xte1, yte1,\
                      normalize='true',cmap=plt.cm.Blues, display_labels={'Yes','No'})
ax=plt.gca()
fig=plt.gcf()
fig.set_size_inches(8,8)
ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')

# red neuronal
# mlp clasico
sc=StandardScaler()
sc.fit(Xtr1)
Xtr1=sc.transform(Xtr1)
mlp=OneVsRestClassifier(MLPClassifier(solver='lbfgs',  alpha=1e-5,hidden_layer_sizes=(5,  2),
random_state=1))
mlp=mlp.fit(Xtr1,ytr1)
ypred=mlp.predict(Xte1)

# GridSpace mlp
mlp = MLPClassifier(max_iter=100)
parameter_space = {
   'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
   'activation': ['tanh', 'relu'],
   'solver': ['sgd', 'adam'],
   'alpha': [0.0001, 0.05],
   'learning_rate': ['constant','adaptive'],
}
from sklearn.model_selection import GridSearchCV

mlp = GridSearchCV(mlp, parameter_space, n_jobs=-1, cv=3)
mlp.fit(Xtr1,ytr1)
ypred=mlp.predict(Xte1)
```

```
acc=100*accuracy_score(yte1,ypred)
print('La precision del modelo de Redes neuronales: %{Acc:.2f}'.format(Acc=acc))

print(classification_report(yte1,ypred))
print('Matriz de confusion:\n')
print(confusion_matrix(yte1,ypred))

plot_confusion_matrix(mlp, Xte1, yte1,\
                      normalize='true',cmap=plt.cm.Blues, display_labels={'Yes','No'})
ax=plt.gca()
fig=plt.gcf()
fig.set_size_inches(8,8)
ax.set_xlabel('Predicted attack', fontsize=12, fontweight='bold')
```

14 REFERENCIAS

[1] "Internet Growth Statistics 1995 to 2021 - the Global Village Online." https://www.internetworldstats.com/emarketing.htm (accessed Feb. 01, 2021).

[2] "The Cyber Kill Chain explained – along with some 2020 examples – osintme.com." https://www.osintme.com/index.php/2020/05/31/the-cyber-kill-chain-explained-along-with-some-2020-examples/ (accessed Feb. 01, 2021).

[3] "2020 SIEM Gartner Magic Quadrant." https://logrhythm.com/uk-gartner-magic-quadrant-siem-report-2020/ (accessed Feb. 25, 2021).

[4] "IBM QRadar SIEM - Detalles - España | IBM." https://www.ibm.com/es-es/products/qradar-siem/details (accessed Feb. 01, 2021).

[5] "Security, SIEM and Fraud | Cyber Security Solutions | Splunk." https://www.splunk.com/en_us/cyber-security.html (accessed Mar. 25, 2021).

[6] "Fusion SIEM | Exabeam." https://www.exabeam.com/product/fusion-siem/ (accessed Mar. 25, 2021).

[7] "Next-Gen SIEM Solution | Security Information and Event Management | Securonix." https://www.securonix.com/products/next-generation-siem/ (accessed Mar. 25, 2021).

[8] "Logistic Regression — Detailed Overview | by Saishruthi Swaminathan | Towards Data Science." https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc (accessed Mar. 11, 2021).

[9] "Introduction to Machine Learning Algorithms: Linear Regression | by Rohith Gandhi | Towards Data Science." https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a (accessed Mar. 11, 2021).

[10] "Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu | Towards Data Science." https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (accessed Mar. 11, 2021).

[11] "Machine Learning Basics: Support Vector Regression | by Gurucharan M K | Towards Data Science." https://towardsdatascience.com/machine-learning-basics-support-vector-regression-660306ac5226 (accessed Mar. 11, 2021).

[12] "Machine Learning for Beginners: An Introduction to Neural Networks | by Victor Zhou | Towards Data Science." https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9 (accessed Mar. 11, 2021).

[13] "Selección de métricas para los modelos de aprendizaje automático | Fayrix." https://fayrix.com/machine-learning-metrics_es (accessed Mar. 11, 2021).

[14] "Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco." https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed Apr. 30, 2021).

[15] A. Holst, "IoT connected devices worldwide 2019-2030," *Statista*, 2021.

https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed Apr. 30, 2021).

[16] "QRadar architecture overview - IBM Documentation." https://www.ibm.com/docs/en/qsip/7.4?topic=deployment-qradar-architecture-overview (accessed Mar. 30, 2021).

[17] "Configure Linux OS to send audit logs to QRadar® - Forums - IBM Support." https://www.ibm.com/mysupport/s/question/0D50z00006PEFCD/configure-linux-os-to-send-audit-logs-to-qradar?language=en_US (accessed Apr. 11, 2021).