



## ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster en Ciberseguridad

### **DETECCIÓN Y ALERTA TEMPRANA EN GEM**

Autor  
Gonzalo Herreros Diezhandino

Dirigido por  
Juan Carlos Cortinas  
Gregorio López

Madrid  
Julio 2021

**Gonzalo Herreros Diezhandino**, declara bajo su responsabilidad, que el Proyecto con título **Detección y Alerta Temprana en GEM** presentado en la ETS de Ingeniería (ICAI) de la Universidad Pontificia Comillas en el curso académico 2020/21 es de su autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: .....

Fecha: 08 / 07 / 2021

Autoriza la entrega:

EL DIRECTOR DEL PROYECTO

**Juan Carlos Cortinas**

Fdo.: .....

Fecha: 08 / 07 / 2021

V. B. DEL COORDINADOR DE PROYECTOS

**Gregorio López**

Fdo.: .....

Fecha: 09 / 07 / 2021

## **AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO**

### ***1º. Declaración de la autoría y acreditación de la misma.***

El autor D.Gonzalo Herreros Diezhandino **DECLARA** ser el titular de los derechos de propiedad intelectual de la obra: DETECCIÓN Y ALERTA TEMPRANA EN GEM, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

### ***2º. Objeto y fines de la cesión.***

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, los derechos de digitalización, de archivo, de reproducción, de distribución y de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

### ***3º. Condiciones de la cesión y acceso***

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- (a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- (b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- (c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- (d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- (e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- (f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

### ***4º. Derechos del autor.***

El autor, en tanto que titular de una obra tiene derecho a:

- (a) Que la Universidad identifique claramente su nombre como autor de la misma

- (b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- (c) Solicitar la retirada de la obra del repositorio por causa justificada.
- (d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

#### ***5º. Deberes del autor.***

- (a) El autor se compromete a:
- (b) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- (c) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- (d) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- (e) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

#### ***6º. Fines y funcionamiento del Repositorio Institucional.***

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 8 de Julio de 2021

ACEPTA

Fdo.: .....

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:





## ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster en Ciberseguridad

### **DETECCIÓN Y ALERTA TEMPRANA EN GEM**

Autor  
Gonzalo Herreros Diezhandino

Dirigido por  
Juan Carlos Cortinas  
Gregorio López

Madrid  
Julio 2021



# Resumen

El objetivo principal del trabajo es la creación de una plataforma que incorpore las principales herramientas de monitorización de una organización, que junto con la conexión a sondas externas que proporcionen una visión de los activos de esta misma desde el exterior se convierta en la herramienta central para gestionar la seguridad de la entidad.

Este proyecto surge como consecuencia de la continuación del trabajo inicial realizado por el tutor Juan Carlos Cortinas, dónde realizó una herramienta que se encargaba de visualizar los activos externos de una organización y mezclar esa información con las posibles vulnerabilidades públicas existentes de dichos activos.

En cuanto a la plataforma se la ha dotado de nueva funcionalidad como es la visualización en tiempo real de las alertas generadas por las diferentes herramientas de monitorización, con la posibilidad de modificar los dashboard de cada una de las herramientas y configurarlos a gusto del consumidor. Además, cabe la posibilidad de poder añadir más herramientas en un futuro debido a la facilidad de escalado de la que dispone la plataforma.

Finalmente, la plataforma denominada CETA cobra un valor aún mayor con la incorporación de la correlación de los datos proporcionados por las diferentes herramientas. De este modo, cabe la posibilidad de observar las alertas generadas de forma conjunta en vez de manera individual. Y no sólo eso, la implementación de un algoritmo de Machine Learning como es el clusterizado, permite la clasificación de todas las alertas recibidas en tiempo real.

En definitiva, un sistema totalmente completo y revolucionario de cara a ser una referencia en la monitorización de organizaciones.



# Abstract

The main objective of the work is the creation of a platform that incorporates the main monitoring tools of an organization, which together with the connection to external probes that provide a view of its assets from the outside becomes the central tool for managing the security of the company.

This project arises as a consequence of the continuation of the initial work carried out by Juan Carlos Cortinas, where he created a tool that was in charge of visualizing the external assets of an organization and mixing that information with the possible existing public vulnerabilities.

The platform has been equipped with new functionality, such as the real-time display of the alerts generated by the different monitoring tools, with the possibility of modifying the dashboards of each of the tools and configuring them to suit the consumer. In addition, it is possible to add more tools in the future due to the ease of scaling that the platform has.

Finally, the platform called CETA takes on an even greater value with the incorporation of the correlation of the data provided by the different tools. In this way, it is possible to observe the alerts generated jointly instead of individually. And not only that, the implementation of a Machine Learning algorithm such as clustering, allows the classification of all alerts received in real time.

In short, a totally complete and revolutionary system in order to be a reference in the monitoring of organizations.



*A mi familia y amigos.*

*Ver más allá de la niebla...  
y no esperar a que se disipe.*

ANÓNIMO



# Agradecimientos

Tras varios meses hoy es el día, escribo este apartado de agradecimientos para finalizar mi trabajo de fin de máster. Ha sido un período de aprendizaje intenso tanto a nivel científico como personal. La realización de este trabajo ha tenido un gran impacto en mí y me gustaría agradecer a todas aquellas personas que han aportado durante este proyecto.

Primero de todo me gustaría agradecer el trabajo y la ayuda recibida por parte de Juan Carlos Cortinas. Pese a no haber podido coincidir presencialmente ambos en la oficina y haber podido trabajar mano a mano, siempre ha estado en todo momento atento a los problemas o consultas que yo tenía. Es por esto, que es a él al primero que agradezco su ayuda. Agradecer también a Gregorio y Javier, directores del máster en Ciberseguridad en la Universidad Pontificia Comillas por haber hecho posible y realidad que este proyecto saliese adelante, además del seguimiento realizado corrigiendo los asuntos que iban surgiendo. También destacar a todos los profesores del máster por su dedicación y ayuda en cualquier consulta que les he ido realizando en la realización del proyecto, no puedo estar más agradecido.

Por último y no menos importante, gracias a mis amigos y familia por haberme ayudado y guiado en este proceso. Gracias por el feedback recibido durante la realización de la plataforma para posibles mejoras o incorporaciones para la optimizar visualización de los datos.

¡Muchas gracias a todo!

Gonzalo Herreros Diezhandino  
Valladolid 07/07/2021



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Origen del proyecto . . . . .	2
1.2. El proyecto . . . . .	2
1.3. Motivación . . . . .	3
1.4. Metodología de trabajo . . . . .	4
1.5. Documento y consideraciones previas . . . . .	5
<b>2. Estado del Arte</b>	<b>7</b>
2.1. Introducción . . . . .	7
2.2. Monitorización externa . . . . .	8
2.3. Monitorización interna . . . . .	10
<b>3. Entorno Tecnológico</b>	<b>13</b>
3.1. Herramientas . . . . .	13
3.1.1. Shodan . . . . .	13
3.1.2. CVE API . . . . .	14
3.1.3. Suricata . . . . .	15
3.1.4. Snort . . . . .	16
3.1.5. Ossec . . . . .	17
3.1.6. Wazuh . . . . .	17
3.1.7. AlienVault OSSIM . . . . .	18
3.2. Entorno de desarrollo . . . . .	18
3.2.1. Red Hat (CENTOS 8) . . . . .	19
3.2.2. Security Onion . . . . .	19
3.2.3. Ubuntu . . . . .	20
3.2.4. Windows 10 . . . . .	21
3.3. Otras herramientas . . . . .	21
3.3.1. ELK Stack . . . . .	21
3.3.2. Pyhton . . . . .	22
3.3.3. MongoDB . . . . .	22
3.3.4. Overleaf . . . . .	22
3.3.5. BitBucket . . . . .	23
3.3.6. Github . . . . .	23
3.3.7. SublimeText . . . . .	23
3.3.8. VirtualBox . . . . .	24
3.3.9. Syslog-ng . . . . .	24

3.3.10. Pulled Pork . . . . .	24
<b>4. Plan de Desarrollo de Software</b>	<b>27</b>
4.1. Introducción . . . . .	27
4.1.1. Propósito . . . . .	27
4.1.2. Usuarios . . . . .	27
4.1.3. Alcance . . . . .	27
4.1.4. Resumen . . . . .	28
4.2. Visión general . . . . .	28
4.2.1. Objetivos . . . . .	28
4.2.2. Restricciones y suposiciones . . . . .	28
4.2.3. Características . . . . .	29
4.2.4. Metodología de desarrollo . . . . .	29
4.2.5. Entregables . . . . .	30
4.3. Organización . . . . .	30
4.3.1. Roles . . . . .	30
4.3.2. Reuniones del proyecto . . . . .	31
4.4. Planificación y estimación . . . . .	31
4.4.1. Estimación temporal . . . . .	31
4.4.2. Estimación de costes . . . . .	32
4.4.3. Desviación del proyecto . . . . .	32
<b>5. Seguimiento del Proyecto</b>	<b>33</b>
5.1. Introducción . . . . .	33
5.2. Sprints . . . . .	34
5.2.1. Sprint 1: Reuniones - Inicio proyecto . . . . .	34
5.2.2. Sprint 2: Puesta a Punto CETA . . . . .	36
5.2.3. Sprint 3: Suricata . . . . .	37
5.2.4. Sprint 4: Snort . . . . .	39
5.2.5. Sprint 5: Ossec . . . . .	40
5.2.6. Sprint 6: Wazuh . . . . .	42
5.2.7. Sprint 7: AlienVault OSSIM . . . . .	43
5.2.8. Sprint 8: Correlación Datos . . . . .	45
5.2.9. Sprint 9: Clustering . . . . .	46
5.2.10. Sprint 10: Documentación y Revisión final . . . . .	47
<b>6. Plan de Gestión de Riesgos</b>	<b>49</b>
6.1. Introducción . . . . .	49
6.1.1. Identificación del riesgo . . . . .	50
6.1.2. Análisis cualitativo de riesgos . . . . .	50
6.1.3. Plan de respuesta al riesgo y contingencia . . . . .	50
6.2. Gestión de riesgos . . . . .	51
6.3. Control de riesgos . . . . .	54

<b>7. Análisis</b>	<b>57</b>
7.1. Análisis de requisitos . . . . .	57
7.1.1. Introducción . . . . .	57
7.1.2. Requisitos funcionales . . . . .	57
7.1.3. Requisitos no funcionales . . . . .	59
7.1.4. Requisitos de información . . . . .	60
7.2. Diagrama de actores . . . . .	60
7.3. Diagrama de casos de uso . . . . .	61
7.4. Casos de Uso . . . . .	62
7.4.1. Ver Datos Shodan . . . . .	62
7.4.2. Ver Datos CVEs Organización . . . . .	63
7.4.3. Ver Datos Suricata . . . . .	64
7.4.4. Ver Datos Snort . . . . .	64
7.4.5. Ver Datos Ossec . . . . .	65
7.4.6. Ver Datos Wazuh . . . . .	66
7.4.7. Ver Datos AlienVault . . . . .	66
7.4.8. Ver Correlación Datos . . . . .	67
7.4.9. Ver Clusterizado Datos . . . . .	68
<b>8. Diseño e Implementación</b>	<b>69</b>
8.1. Entorno de desarrollo . . . . .	69
8.2. CETA . . . . .	70
8.2.1. Puesta a punto . . . . .	70
8.2.2. Suricata . . . . .	73
8.2.3. Snort . . . . .	76
8.2.4. Ossec . . . . .	80
8.2.5. Wazuh . . . . .	85
8.2.6. AlienVault . . . . .	89
8.3. Correlación de datos . . . . .	89
8.3.1. Creación Dataset . . . . .	89
8.3.2. Correlación . . . . .	91
8.4. Clusterizado . . . . .	92
8.4.1. Algoritmo . . . . .	92
<b>9. Pruebas</b>	<b>95</b>
9.1. Suricata . . . . .	95
9.1.1. Regla . . . . .	95
9.1.2. Ataque . . . . .	95
9.1.3. Visualización . . . . .	96
9.2. Snort . . . . .	96
9.2.1. Regla . . . . .	96
9.2.2. Ataque . . . . .	96
9.2.3. Visualización . . . . .	97
9.3. Ossec . . . . .	97
9.3.1. Regla . . . . .	97
9.3.2. Ataque . . . . .	98
9.3.3. Visualización . . . . .	98

<b>10. Conclusiones y Líneas Futuras</b>	<b>99</b>
10.1. Conclusiones . . . . .	99
10.1.1. Conclusiones CETA . . . . .	100
10.2. Líneas futuras . . . . .	101
10.2.1. Adición de nuevas herramientas de monitorización . . . . .	101
10.2.2. Adición de nuevas sondas externas . . . . .	101
10.2.3. Mejora algoritmo de clasificación . . . . .	102
10.2.4. Implementación de algoritmo para detección de anomalías . . . . .	102
10.2.5. Implementación algoritmo de predicción . . . . .	102
10.2.6. Mejora de rendimiento . . . . .	102
<b>Appendix</b>	<b>103</b>
<b>A. Manual de Usuario</b>	<b>103</b>
A.1. Introducción . . . . .	103
A.2. Inicio . . . . .	103
A.3. Shodan . . . . .	106
A.4. CVE Search . . . . .	107
A.5. Suricata . . . . .	110
A.6. Snort . . . . .	110
A.7. Ossec . . . . .	111
A.8. Wazuh . . . . .	111
A.9. Correlation . . . . .	112
<b>B. Soporte Digital</b>	<b>113</b>
<b>Bibliografía</b>	<b>115</b>

# Índice de figuras

1.1. Tipología de ataques . . . . .	3
1.2. Sectores atacados . . . . .	4
3.1. Shodan . . . . .	14
3.2. CVE-API . . . . .	14
3.3. Suricata . . . . .	16
3.4. Snort . . . . .	16
3.5. Ossec . . . . .	17
3.6. Wazuh . . . . .	18
3.7. AlienVault . . . . .	18
3.8. Centos . . . . .	19
3.9. Security Onion . . . . .	20
3.10. Ubuntu . . . . .	20
3.11. Windows 10 . . . . .	21
3.12. ELK . . . . .	22
3.13. Python . . . . .	22
3.14. MongoDB . . . . .	22
3.15. Overleaf . . . . .	23
3.16. Bitbucket . . . . .	23
3.17. Github . . . . .	23
3.18. SublimeText . . . . .	24
3.19. VirtualBox . . . . .	24
3.20. Syslog-ng . . . . .	24
3.21. Pulled Pork . . . . .	25
5.1. Flujo de Trabajo Scrum . . . . .	34
5.2. Sprint 1: Reuniones - Inicio proyecto . . . . .	35
5.3. Sprint 2: Puesta a Punto CETA . . . . .	37
5.4. Sprint 3: Suricata . . . . .	39
5.5. Sprint 4: Snort . . . . .	40
5.6. Sprint 5: Ossec . . . . .	42
5.7. Sprint 6: Wazuh . . . . .	43
5.8. Sprint 7: AlienVault OSSIM . . . . .	44
5.9. Sprint 8: Correlación Datos . . . . .	46
5.10. Sprint 9: Clustering . . . . .	47
5.11. Sprint 10: Documentación y Revisión final . . . . .	48

7.1. Casos de uso: Responsable de seguridad . . . . .	61
8.1. Esquema de red . . . . .	69
8.2. CETA: Cyber Early Threat Alarm . . . . .	70
8.3. CETA: Flujo de datos . . . . .	72
8.4. Suricata: Flujo de datos . . . . .	76
8.5. Suricata: Visualización . . . . .	76
8.6. Snort: Flujo de datos . . . . .	80
8.7. Snort: Visualización . . . . .	80
8.8. Ossec: Flujo de datos . . . . .	84
8.9. Ossec: Visualización . . . . .	85
8.10. Wazuh: Flujo de datos . . . . .	88
8.11. Wazuh: Visualización . . . . .	88
8.12. Correlación: Visualización . . . . .	92
9.1. Prueba: Suricata . . . . .	96
9.2. Prueba: Snort . . . . .	97
9.3. Prueba: Ossec . . . . .	98
A.1. Manual de usuario: Carpetas . . . . .	103
A.2. Manual de usuario: initRequirements . . . . .	104
A.3. Manual de usuario: AlertSystemMain . . . . .	104
A.4. Manual de usuario: Inicio Servicios . . . . .	104
A.5. Manual de usuario: Inicio CVE Search . . . . .	105
A.6. Manual de usuario: CETA . . . . .	105
A.7. Manual de usuario: Buscar Shodan . . . . .	106
A.8. Manual de usuario: CETA Shodan . . . . .	107
A.9. Manual de usuario: CETA Shodan 1 . . . . .	107
A.10. Manual de usuario: Buscar CVE Search . . . . .	108
A.11. Manual de usuario: CETA CVE Search . . . . .	109
A.12. Manual de usuario: CETA CVE Search 1 . . . . .	109
A.13. Manual de usuario: Suricata . . . . .	110
A.14. Manual de usuario: Snort . . . . .	110
A.15. Manual de usuario: Ossec . . . . .	111
A.16. Manual de usuario: Wazuh . . . . .	111
A.17. Manual de usuario: Correlation . . . . .	112
A.18. Manual de usuario: Correlation Dashboard . . . . .	112

# Índice de cuadros

4.1. Roles en el proyecto . . . . .	30
4.2. Eventos scrum . . . . .	31
4.3. Sprints del proyecto . . . . .	32
6.1. Riesgo 01 . . . . .	51
6.2. Riesgo 02 . . . . .	51
6.3. Riesgo 03 . . . . .	51
6.4. Riesgo 04 . . . . .	52
6.5. Riesgo 05 . . . . .	52
6.6. Riesgo 06 . . . . .	52
6.7. Riesgo 07 . . . . .	52
6.8. Riesgo 08 . . . . .	53
6.9. Riesgo 09 . . . . .	53
6.10. Riesgo 10 . . . . .	53
6.11. Riesgo 11 . . . . .	53
6.12. Riesgo 12 . . . . .	54
6.13. Riesgo 13 . . . . .	54
7.1. Requisitos Funcionales 1 . . . . .	58
7.2. Requisitos Funcionales 2 . . . . .	59
7.3. Requisitos No Funcionales . . . . .	60
7.4. Requisitos de Información . . . . .	60
7.5. Actor: Responsable de seguridad de la organización . . . . .	61
7.6. CU-01:Ver Datos Shodan . . . . .	62
7.7. CU-01:Curso Normal . . . . .	62
7.8. CU-02:Ver Datos CVEs Organización . . . . .	63
7.9. CU-02:Curso Normal . . . . .	63
7.10. CU-03:Ver Datos Suricata . . . . .	64
7.11. CU-03:Curso Normal . . . . .	64
7.12. CU-04:Ver Datos Snort . . . . .	64
7.13. CU-04:Curso Normal . . . . .	65
7.14. CU-05:Ver Datos Ossec . . . . .	65
7.15. CU-05:Curso Normal . . . . .	65
7.16. CU-06:Ver Datos Wazuh . . . . .	66
7.17. CU-06:Curso Normal . . . . .	66
7.18. CU-07:Ver Datos AlienVault . . . . .	66

7.19. CU-07:Curso Normal . . . . .	67
7.20. CU-08:Ver Correlación Datos . . . . .	67
7.21. CU-08:Curso Normal . . . . .	67
7.22. CU-09:Ver Clusterizado Datos . . . . .	68
7.23. CU-09:Curso Normal . . . . .	68
8.1. Correlation Dataset . . . . .	90

# Acrónimos

<i>ICAI</i>	Insitituto Católico de Artes e Industrias
<i>CETA</i>	Cyber Early Threat Alarm
<i>TFM</i>	Trabajo Fin de Máster
<i>IDS</i>	Intrusion Detection System
<i>HIDS</i>	Host-based Intrusion Detection System
<i>NIDS</i>	Network Intrusion Detection System
<i>EDR</i>	Endpoint Detection and Response
<i>SIEM</i>	Security Information and Event Management
<i>CVE</i>	A unique, alphanumeric identifier assigned by the CVE Program. Each identifier references a specific vulnerability.



# Capítulo 1

## Introducción

La sociedad actual tiende a ser cada vez más tecnológica, a tal punto que hoy en día, casi todas las personas tienen un su haber un dispositivo electrónico, ya sea móvil, portátil, smartwatch etc. Esto mismo está ocurriendo con el mundo empresarial, son muchas ya las organizaciones cuyo soporte se encuentra centralizado en torno a la tecnología y son escasas aquellas que aún no lo han hecho, igual porque no lo necesitan o porque se encuentran en un momento inicial de migración hacia la nueva tecnología.[1]

Todo esto, hace que los antiguos ladrones hayan tenido que adaptarse también a las nuevas tecnologías y que hayan tenido que implementar nuevas técnicas de evasión y ocultación para poder acometer su acción. De hecho, la aparición de la pandemia mundial provocada por el COVID-19 ha intensificado y aumentado el número de ataques realizados por estos ciberdelincuentes.[2]

He aquí dónde nace la importancia de tener protegido en todo momento aquella información confidencial o de alto valor. En el caso del entorno empresarial esta necesidad se acentúa más, ya que la información almacenada puede contener un alto valor y su robo pondría en un estado de gravedad a dicha organización.

Existen múltiples mecanismos de protección frente a amenazas, entre las que pueden destacar, por ejemplo, implementación de seguridad física, establecimiento de comunicaciones seguras, implementación de seguridad lógica etc. Pero sin duda una de las importantes consiste en la monitorización de los eventos producidos en una empresa para la detección de posibles acciones inusuales.[3]

La monitorización de una organización implica recopilar y analizar información para detectar comportamientos sospechosos o cambios no autorizados. De este modo, se pueden llegar a resolver y detectar posibles acciones maliciosas o incluso realizar un análisis de la organización y de su funcionamiento.

## 1.1. Origen del proyecto

El proyecto fue iniciado por el tutor de este mismo, Juan Carlos Cortinas, en la elaboración de una herramienta que se encargase de la monitorización externa de una organización, a través de la utilización de sondas con repositorios abiertos como son Shodan[4] y CVE[5].

A la hora de monitorizar una organización existen 2 visiones:[6]

- Monitorización externa. Consiste en observar la organización desde fuera y ver qué posibles activos se encuentran visibles al exterior y pueden ser vulnerables.
- Monitorización interna. Consiste en observar la organización desde dentro, el flujo de datos entre los dispositivos de la organización y las acciones realizadas en cada uno de ellos y por cada uno.

Es por esto, que se propuso la continuación del proyecto para unificar en una única herramienta la monitorización externa e interna de una organización.

## 1.2. El proyecto

Este proyecto solicitado tiene como objetivo principal desarrollar una herramienta que se encuentre alojada en una máquina independiente y que se encargue de recoger la información proporcionada por todos los sistemas de monitorización disponibles en una organización y las sondas externas que evalúen los activos que estas disponen de cara al exterior. De este modo, con todos estos datos, se podrá tener un sistema central a través del cuál podamos gestionar toda la seguridad de nuestra organización y que por tanto, nos permita detectar anomalías o situaciones de funcionamiento incorrecto de la organización.

Para ello, habrá que realizar una plataforma de visualización dentro de la máquina sobre la que se encuentra CETA (a partir de ahora se llamará así a la herramienta central) y realizar todas las conexiones pertinentes para que la máquina central reciba la información necesaria de las herramientas de monitorización disponibles. La plataforma de visualización, será la encargada de correlacionar los datos recibidos para poder así tener una herramienta aún más sofisticada y que permita detectar con prontitud un posible futuro ataque.

Con todo esto, cualquier organización será capaz de conocer el estado de sus sistemas e infraestructuras en tiempo real. Procediendo así a estar en todo momento alerta de cualquier vulnerabilidad nueva encontrada y de las posibles fugas de información, convirtiendo a esta herramienta en sistema preventivo de alta fiabilidad.

En definitiva, el objetivo principal es el de disponer de un sistema de alerta temprana que nos permita actuar con prontitud a partir de diversas sondas (SIEM, sondas de vulnerabilidades, CMDB, etc.).

A partir de estos objetivos principales, se derivan unos objetivos secundarios que se comentan a continuación:

- Afianzar la planificación de un proyecto software

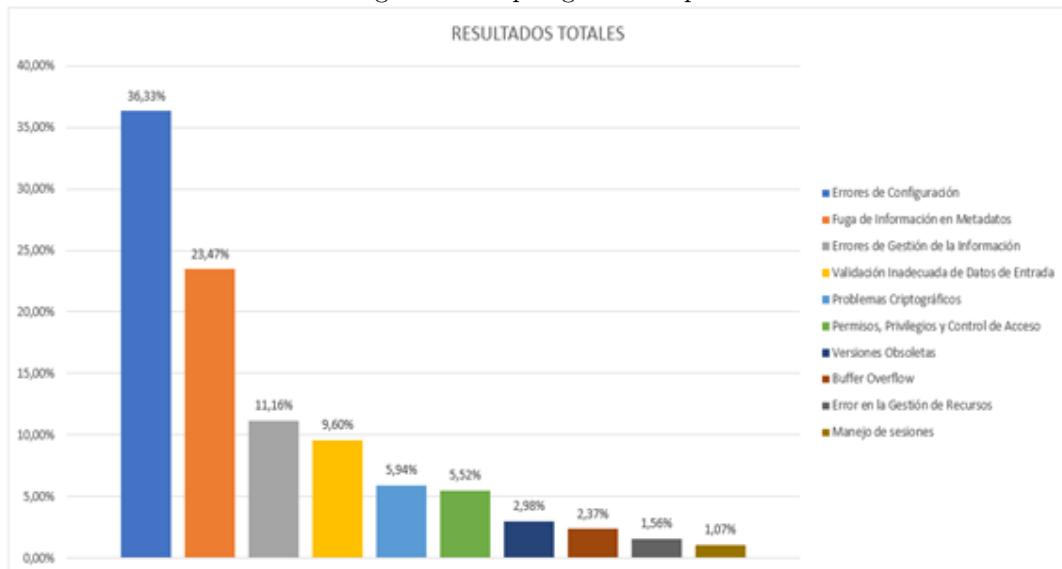
- Afianzar el manejo de herramientas para la monitorización de sistemas
- Elaboración de una memoria de un proyecto software real

### 1.3. Motivación

En los últimos años se ha producido un incremento de los ataques y amenazas a todo tipo de sectores tanto en cantidad como en variedad. Debido a la necesidad de los negocios, obtener información en tiempo real, accesibilidad, servicios centralizados etcétera y a nuevas tecnologías internet de las cosas (IoT) la tendencia irá en aumento. Todos estos ataques suponen grandes pérdidas económicas para las empresas.[7][8]

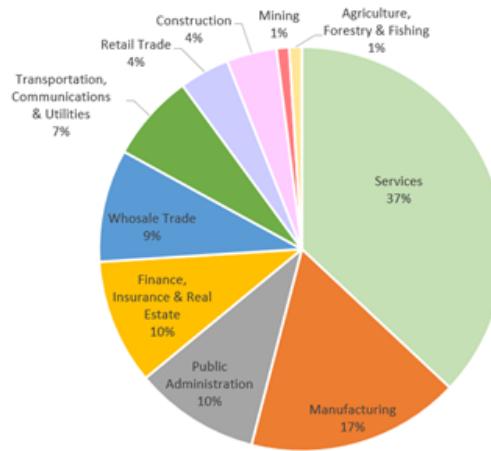
La tipología de los ataques varía mucho, entre los más destacados: errores de configuración; fuga de información metadatos; errores en la gestión de la información; validación inadecuada de datos de entrada; problemas criptográficos; permisos privilegios y control de acceso; versiones obsoletas; buffer overflow; error en la gestión de recursos; manejo de sesiones etc[9]

Figura 1.1: Tipología de ataques



Todos estos posibles ataques afectan a organizaciones de todo tipo, entre las más afectadas se encuentran aquellas que ofrecen servicios, bancos, empresas de manufacturación etc. Pero además se ha podido observar cómo estos ataques influyen también a otro tipo de empresas, que a priori no se encuentran tan informatizadas como son las empresas agrícolas, empresas mineras o de construcción. [10]

Figura 1.2: Sectores atacados



Por tanto, existen múltiples tipos de ataques y amenazas disponibles para todo tipo de empresas, no sólo aquellas que se dedican a la informática. Es por esto por lo que el proyecto actual cobra aún más valor, debido a la necesidad de conocer el estado de los sistemas e infraestructuras de una organización en tiempo real, para poder actuar frente a una alerta notificada con prontitud.

En definitiva, un dispositivo de alerta temprana como este va a permitir conocer el estado de los sistemas e infraestructuras en tiempo real, y así poder detectar vulnerabilidades y fugas de información, convirtiéndose en un sistema preventivo indispensable para una empresa.

## 1.4. Metodología de trabajo

Para el desarrollo de este proyecto se ha seguido una metodología ágil.[11] Este método de trabajo permite “trocear” el proyecto en pequeñas partes que deben irse completando y entregarse en pocas semanas. Esto va a permitir observar el progreso del proyecto y observar el funcionamiento de las nuevas implementaciones que se vayan realizando e ir corrigiendo los errores que se vayan produciendo. El objetivo final es el de poder crear un producto de calidad que responda a las necesidades iniciales del proyecto.

Las características de la metodología de trabajo implementada son las siguientes: [12]

- Iterativo, incremental y evolutivo.
- Retroalimentación corta – “Sprints” cortos.
- Flexible.
- Mejor comunicación.
- Menos riesgo, lo cual supone una mayor calidad.
- Entregas ágiles.

Con todo esto, la idea del proyecto en cada iteración es la de ir implementando una nueva tecnología. Es decir, para cada iteración habrá que seleccionar la herramienta a introducir en el sistema y realizar todo el proceso de diseño, desarrollo, pruebas, mejoras y planificaciones de trabajos futuros. De este modo, siempre se tendría un producto mínimo viable.

## 1.5. Documento y consideraciones previas

A continuación, se describe la estructura de la memoria del proyecto:

- **Capítulo 1. Introducción** se expone el origen, contexto del proyecto, así como la motivación que lleva a hacer dicho proyecto.
- **Capítulo 2. Estado del Arte** se expone el estado actual de la monitorización en las organizaciones, así como las ventajas y diferencias que aporte este proyecto sobre lo ya existente.
- **Capítulo 3. Entorno tecnológico** se expone las herramientas utilizadas en el proyecto, así como las plataformas utilizadas.
- **Capítulo 4. Plan de Desarrollo de Software** Este capítulo está destinado a describir el plan de desarrollo del proyecto y de sus especificaciones técnicas.
- **Capítulo 5. Seguimiento del proyecto** se explican las diferentes fases en las que ha sido dividido el proyecto.
- **Capítulo 6. Plan de Gestión de Riesgos** se explican las responsabilidades y actividades relacionadas con la gestión de riesgos.
- **Capítulo 7. Análisis** se exponen las características operacionales del software.
- **Capítulo 8. Diseño e Implementación** se expone la estructura implementada de CETA así como la configuración de las máquinas empleadas.
- **Capítulo 9. Pruebas** se expone ejemplos de posibles ataques y detecciones por parte de CETA.
- **Capítulo 10. Conclusiones y Líneas futuras** se expone la conclusiones a las que se llegan tras la finalización del proyecto y las posibles continuaciones del proyecto.



# Capítulo 2

## Estado del Arte

### 2.1. Introducción

Las empresas tienen a su disposición diversidad de herramientas para gestionar su seguridad o muchas veces dependen de terceros que realizan esta tarea. Entre estas herramientas podemos encontrar aquellas que se encargan de gestionar activos a otras que monitorean el tráfico de red entrante y saliente de la organización.

Por eso, si nos centramos en las herramientas encargadas de gestionar activos podemos encontrar plataformas como CMDB Aris [13]. Este tipo de herramientas permiten gestionar todos los activos de la organización y poder incluso ver los procesos de negocio, iniciativas estratégicas, servicios de negocio etc.

Por otro lado, existen herramientas de gestión de vulnerabilidades como Rapid7 Nmap [14], que permiten respaldar todo el ciclo de vida de las vulnerabilidades, desde su descubrimiento hasta la mitigación de estas mismas.

Existen también herramientas de monitoreo de red, a través de las cuales se puede supervisar los componentes de la red, como pueden ser los cortafuegos, conmutadores, enruteadores etc. Además de poder monitorizar el tráfico, es una excelente manera de mejorar el rendimiento de la red.[15]

Las organizaciones también disponen de sistemas de gestión de eventos e información de seguridad como puede ser SIEM QRadar [16]. Estos sistemas centralizan el almacenamiento y la interpretación de los datos relevantes de seguridad, proporcionando información inteligente para que los equipos respondan rápidamente y así reducir el impacto de los incidentes.

Por último, hay herramientas externas a una organización que pueden ser consultadas y aportan gran información. Entre estas herramientas destacan la lista de CVEs públicos accesibles por todas las personas, o sondas externas que proporcionan gran cantidad de información como es Shodan.[4][5]

En definitiva, existen múltiples tipos de herramientas distintas que aportan gran valor a

una empresa en lo que respecta a la seguridad de esta misma. El problema existente con estas herramientas es que la mayoría de ellas funcionan de forma independiente unas de otras, por lo que la posible unión de todas ellas aportaría gran valor de cara a la securización de una organización. Por tanto, la posible integración de todas estas herramientas con la adición de mecanismos que relacionen todos los datos, como puede ser la inteligencia artificial, dotaría a una organización de sistema de detección de amenazas temprano totalmente revolucionario.

## 2.2. Monitorización externa

Tan importante es ver cómo es el tráfico de red interno de nuestra organización y los intentos de acceso desde el exterior a nuestros sistemas, como ver cómo nuestros sistemas se ven desde el exterior.

Este tipo de búsquedas deben ir orientadas hacia nuestras preocupaciones:

- Evaluación de una organización específica. Este análisis se puede realizar buscando por dominios o hosts, por rango de IPs o buscando credenciales de la organización.
- Evaluación frente a una amenaza en concreto. Este análisis se puede realizar buscando puertos abiertos a internet, servicios abiertos a internet o vulnerabilidades concretas.

Durante este tipo de búsquedas es importante no aportar mucha información adicional, ya que una búsqueda con mucho detalle puede revelar información a otros.

Por tanto la monitorización externa dota a una organización de las siguientes ventajas:

- "Situational Awareness". Ayuda a los encargados de tomar las decisiones de una organización a tener la información y la comprensión disponibles para tomar buenas decisiones en el curso de su trabajo. Puede centrarse específicamente en ayudar a las personas y las organizaciones a proteger sus activos en el ámbito de la ciberseguridad o puede tener un mayor alcance. Situational Awareness hace posible obtener información relevante de toda la organización, integrar esa información y difundirla para ayudar a las personas a tomar mejores decisiones.
- Servicio de seguridad gestionado o del estado. Permite evaluar actividades de actores importantes (estados, APTs, etc), también permite evaluar tendencias en la actividad tanto por explotación de vulnerabilidades conocidas como por identificación de posibles zero days y por último, permite identificar sectores o empresas más atacadas.
- Casos de servicios propios de empresa. Permite realizar una evaluación de riesgos de la empresa, se puede conocer el perímetro accesible externamente, los puertos accesibles desde el exterior, las vulnerabilidades explotables por terceros y las configuraciones de seguridad deficientes. Por otro lado, permite realizar una evaluación de ataques realizados, entre las que se puede encontrar la búsqueda de credenciales comprometidas, de información filtrada y a la venta, de usurpación de dominios y de dominios fraudulentos.

Para cada una de estas ventajas proporcionadas por la monitorización externa existen diferentes herramientas, entre las que destacan las siguientes: [6]

- "Situational Awareness"
  - Kaspersky: muestra ataques medidos por todas sus herramientas desplegadas. Estadísticas por herramienta y país.[17]
  - Sophos: muestra ataques medidos por su AV, estadísticas malware, web y spam.[18]
  - Akamai: muestra ataques web por países, sectores o por tipo de ataque.[19]
  - Digital Attack Map: mapa que muestra los ataques DDOS a nivel mundial, con los puertos explotados.[20]
  - Talos: mapa mundial que muestra ataques spam o malware.[21]
  - Looking Glass: muestra ataques phising y botnet por países.[22]
- Discovery
  - Google: se puede obtener la dirección de cámaras IP o de listados IP a partir de pastebin, por ejemplo.
  - Nmap: permite la búsqueda por IP de puertos y servicios accesibles.[23]
  - Nexpose: es un nmap automatizado, los resultados son almacenados en la base de datos.[14]
- Vulnerabilidades. Grupo de herramientas que permiten el escaneo y gestión de vulnerabilidades de seguridad de sistemas informáticos.
  - OpenVas[24]
  - Nessus[25]
  - Nexpose[14]
- Riesgo de empresa
  - Bitsight: permite realizar una evaluación de una organización específica. Su pueden realizar búsquedas por dominio o hosts, por rango de IPs o mediante rating.[26]
  - Dominios fraudulentos. Este tipo de fuentes externas permiten la vigilancia de dominios con nuestra marca, para detectar orígenes fraudulentos y webs de la empresa que no se han renovado y han comprado otros aprovechando la oportunidad. Además, permite detectar webs con nombres muy parecidos para hacer fraude.
    - Guardian
    - Recorded Future[27]
  - Bechas de datos. Es importante realizar una búsqueda prudente en este tipo de herramientas, ya que si se busca un usuario y clave se está revelado. Permite tener un control de reseteo de claves frente a fechas de hallazgos.
    - Red Tor[28]
    - Mercados ilegales
    - Have I been pwned?[29]
  - Buscadores multipropósito
    - Shodan[4]
    - Censys[30]

## 2.3. Monitorización interna

La monitorización interna permite a las organizaciones que sean auditadas, que el cumplimiento legal no se vea comprometido, supervisar las acciones realizadas dentro de la organización y realizar análisis forense.

A la hora de realizar esta monitorización, se pueden distinguir dos tipos:

- En tiempo real. Útil para la detección de incidentes críticos y la evolución de potenciales amenazas.
- A demanda. Útil para realizar un análisis causa raíz (que pasó?; qué originó el problema?) o un análisis forense (quién ha sido?; cómo se desarrolló?).

La importancia que tiene este tipo de monitorizaciones es muy amplia, ya que permite velar por la seguridad de las organizaciones. Se puede destacar:

- Violaciones de políticas.
- Detección de intrusiones.

A la hora de realizar una monitorización interna de una organización, es importante tener en cuenta qué se va a monitorizar. Normalmente se suele observar lo siguiente:

- Logs: lo que pasa en mi sistema.
- Tráfico: lo que pasa entre equipos y con el exterior.
- Archivos: detección en detalle.

Por último, es importante saber dónde realizar la monitorización, ya sea en una máquina donde se generan eventos/logs y es posible insertar sensores para la detección y posterior reenvío de logs a través de agentes; en un entorno para detectar el tráfico que hay y añadir analizadores de tráfico; en sistemas de seguridad centralizadas como puede ser un SIEM que muestra la información en tiempo real.

Al igual que ocurría con la monitorización externa, se van a exponer aquí algunas de las herramientas más comunes para realizar monitorizaciones internas: [6]

- Tráfico. Normalmente se utilizan NIDS (Sistemas de detección de intrusión en una red). Se encargan de detectar anomalías que indiquen un riesgo potencial, para ello analizan todos los paquetes tanto entrantes como salientes, ya que puede darse que el ataque se inicie desde el interior de la organización.[31]
  - Snort[33]
  - Suricata[34]
- Host. Normalmente se utilizan HIDS (Sistema de detección de intrusión en un host). Se encarga de detectar anomalías que indiquen un riesgo potencial, revisando las actividades en el host.[32]
  - Ossec[35]

- Archivos
  - YARA: se trata de una herramienta dirigida a ayudar a los investigadores a identificar y clasificar muestras de malware. Es decir, permite detectar malware basado en firmas de manera similar a cómo lo hacen las soluciones de antivirus tradicionales.[36]
- Logs
  - Sigma: su objetivo principal es proporcionar una forma estructurada en la que los investigadores o analistas puedan describir sus métodos de detección, una vez desarrollados, y compartirlos con otros.[37]



# Capítulo 3

## Entorno Tecnológico

En este capítulo se va a explicar la tecnología utilizada durante la realización del proyecto. Por un lado, se van a detallar las herramientas de monitorización interna/externa empleadas para el desarrollo del proyecto. Por otro lado, se van a detallar las diferentes máquinas utilizadas con sus configuraciones sobre las que se alojan las herramientas anteriormente descritas. Por último, se van a exponer el resto de herramientas utilizadas durante el proyecto.

### 3.1. Herramientas

En este apartado se van a exponer las herramientas de monitorización interna y externa utilizadas en el proyecto:

- Shodan
- CVE Search
- Suricata
- Snort
- Ossec
- Wazuh
- AlienVault OSSIM

#### 3.1.1. Shodan

Shodan [4] es el motor de búsqueda que permite a los usuarios encontrar todo tipo de dispositivos (routers, servidores etc.) conectados a internet a través de una variedad de filtros.

Fue lanzado en 2009 por el informático John Matherly que concibió la idea de buscar dispositivos vinculados a internet. Shodan nos muestra puertos, banners, servicios y geolocalización entre otros. Más que nada, Shodan es un buscador de dispositivos IOT.

Entre los datos recogidos por Shodan, se encuentra información de todos los servicios incluyendo HTTP (puertos 80 y 8080), HTTPS (puertos 443 y 8443), FTP (puerto 21), SSH (puerto 22), Telnet (puerto 23) y muchos más.

Figura 3.1: Shodan



### 3.1.2. CVE API

CVE-Search [5] es accesible a través de una interfaz web y una API HTTP. CVE-Search es una interfaz para buscar información públicamente conocida de vulnerabilidades de seguridad en software y hardware junto con sus exposiciones correspondientes. CVE-Search es un servicio público operado por CIRCL.

Entre las fuentes de datos que incluye CVE-Search cabe destacar:

- Bases de datos nacional de vulnerabilidades del NIST
- Enumeración de plataformas común (CPE)
- Enumeración de debilidades comunes (CWE)
- Estadísticas de incidentes CIRCL y clasificación de amenazas
- Toolswatch/vFeed

El código fuente de CVE-Search se encuentra en GitHub. Los principales autores de cve-search son Alexandre Dulaunoy y Pieter-Jan Moreels con el apoyo de la comunidad, incluido CIRCL.

Figura 3.2: CVE-API



### 3.1.3. Suricata

Suricata [34] es un motor de red de código abierto y multiplataforma de alto rendimiento IDS (Sistema de Detección de Intrusos), IPS (Sistema de Prevención de Intrusos) y seguridad en la red, desarrollado por la comunidad OISF (Open Information Security Foundation). Al ser un NIDS puede detectar ataques en todo el segmento de la red sobre el que se ubique Suricata.

El motor Suricata es capaz de detección de intrusiones en tiempo real (IDS), prevención de intrusiones en línea (IPS), monitoreo de seguridad de red (NSM) y procesamiento pcap fuera de línea. Suricata inspecciona el tráfico de la red utilizando un poderoso y extenso lenguaje de firmas y reglas, y tiene un soporte de scripts Lua para la detección de amenazas complejas.

Dispone de formatos de entrada y salida estándar como YAML y JSON, por tanto, las integraciones con herramientas como SIEM existentes, Splunk, Logstash/Elasticsearch, Kibana y otras bases de datos se vuelven fáciles.

Entre las características más relevantes de Suricata destacan:

- Multi-threading
- Detección automática de protocolos
- Descompresión gzip
- Biblioteca HTP independiente
- Métodos de entrada estándar
- Salida unificada2
- Variables de flujo
- Coincidencia de IP rápida
- Módulo de registro HTTP
- Salida estándar JSON
- Binarios de Windows
- Secuencias de comandos de Lua
- Salida de preludio
- Coincidencia de archivos, registros, extracción, cálculo de suma de comprobación md5.
- Reputación de IP
- Registrador de DNS
- Soporte VXLAN desde 4.1.5

Figura 3.3: Suricata



### 3.1.4. Snort

Snort [33] es el sistema de prevención de intrusiones (IPS) de código abierto más importante del mundo. Snort IPS usa una serie de reglas que ayudan a definir la actividad de red maliciosa y usa esas reglas para encontrar paquetes que coincidan con ellas y genera alertas para los usuarios.

Snort también se puede implementar en línea para detener estos paquetes. Snort tiene tres usos principales: como rastreador de paquetes como tcpdump, como registrador de paquetes, que es útil para la depuración del tráfico de red, o puede usarse como un sistema de prevención de intrusiones en la red en toda regla.

Snort tiene una base de datos de ataques que se actualiza constantemente a través de internet. Los usuarios pueden crear firmas basadas en las características de los nuevos ataques de red y enviarlas a la lista de correo de firmas de Snort. Esta ética de comunidad y compartir ha convertido a Snort en uno de los IDS basados en red más populares, actualizados y robustos.

Adicionalmente cada usuario puede añadir sus propias reglas para la detección de amenazas en una organización.

Figura 3.4: Snort



### 3.1.5. Ossec

Ossec [35] es un sistema de detección de intrusiones (HIDS) gratuito y de código abierto. Realiza análisis de registros, verificación de integridad, monitoreo del registro de Windows, detección de rootkit, alertas basadas en el tiempo y respuesta activa. Ossec proporciona detección de intrusiones para la mayoría de los sistemas operativos, incluidos Linux, OpenBSD, FreeBSD, OS X, Solaris y Windows.

Ossec tiene una arquitectura multiplataforma centralizada que permite monitorear y administrar múltiples sistemas fácilmente, como ya se verá más adelante, tiene múltiples agentes y un servidor centralizado que recibe toda la información de los agentes. Además, tiene un motor de análisis de registros que puede correlacionar y analizar registros de múltiples dispositivos y formatos.

Ossec consta de una aplicación principal (servidor), un agente y una interfaz web.

- Manager (servidor)
- Agente, programa instalado en los sistemas a monitorear.
- Modo sin agente, usado para monitorear firewalls, enrutadores o sistemas Unix.

Figura 3.5: Ossec



### 3.1.6. Wazuh

Wazuh [38] es un sistema de detección de intrusos basado en host de código abierto y libre (HIDS). Realiza análisis de registro, comprobación de integridad, supervisión del registro de Windows, detección de rootkits, alertas basadas en el tiempo y respuesta activa. Proporciona detección de intrusiones para la mayoría de los sistemas operativos, incluyendo Linux, AIX, HP-UX, macOS, Solaris y Windows. Wazuh tiene una arquitectura centralizada y multiplataforma que permite que múltiples sistemas sean fácilmente monitoreados y administrados.

Los principales componentes de Wazuh son:

- Agente. El agente ligero de Wazuh está diseñado para realizar una serie de tareas con el objetivo de detectar amenazas y, cuando sea necesario, activar respuestas automáticas.
- Servidor. El servidor de Wazuh se encarga de analizar los datos recibidos de los agentes, procesar los eventos a través de decodificadores y reglas, y utilizar la inteligencia de amenazas para buscar los conocidos IOC (Indicadores de Compromiso). El servidor también se utiliza para gestionar los agentes, configurándolos y actualizándolos a distancia cuando sea necesario.

- Elastic Stack. Las alertas generadas por Wazuh son enviadas a Elasticsearch, donde son indexadas y almacenadas. El plugin Wazuh Kibana proporciona una potente interfaz de usuario para la visualización y el análisis de datos, que también puede utilizarse para gestionar y supervisar la configuración y el estado de los agentes.

Figura 3.6: Wazuh



### 3.1.7. AlienVault OSSIM

OSSIM [39] el producto de seguridad de la información y gestión de eventos (SIEM) de código abierto de AlienVault, proporciona un SIEM completo con recolección de eventos, normalización y correlación.

El objetivo del proyecto es ofrecer una herramienta que ayude a la administración de eventos de seguridad mediante un motor de correlación y una colección detallada de herramientas Open Source las cuales sirven al administrador para tener una vista de todos los aspectos relativos a la seguridad en su infraestructura. OSSIM a su vez provee una fuerte motor de correlación, con detallados niveles, bajos, medianos y altos de interfaces de visualización, como también reportes y herramientas de manejo de incidentes.

Figura 3.7: AlienVault



## 3.2. Entorno de desarrollo

En este apartado se van a exponer los entornos de desarrollo utilizados en el proyecto:

- Centos 8
- Security Onion
- Ubuntu

- Windows 10

### 3.2.1. Red Hat (CENTOS 8)

CentOS [40] es un sistema operativo de tipo Unix de código abierto basado en el kernel de Linux. CentOS se lanzó en 2004 y se distribuye con Licencia Pública General de GNU (GNU GPL o GNU General Public Licence, en inglés). Es una distribución de Linux estable, predecible y fácil de usar, con una creciente comunidad que la respalda.

Al igual que su modelo, CentOS es una plataforma corporativa destinada, esencialmente, a su implementación en empresas y organizaciones de gran tamaño. En principio, la distribución de Linux también se puede utilizar en el sector privado, pero esta no ha sido la prioridad de sus desarrolladores.

Las características principales de CENTOS son las siguientes:

- Estabilidad. CentOS cuenta con una comprometida comunidad de desarrolladores que la mantiene actualizada y garantiza la compatibilidad tanto con software nuevo como con aplicaciones antiguas. Además, los desarrolladores de CentOS cuentan con el soporte de una comunidad activa de usuarios voluntarios de todas partes del mundo.
- Alto rendimiento y disponibilidad. Ofrece un gran rendimiento y alta disponibilidad al usar KVM (Máquina Virtual basada en el kernel o Kernel-based Virtual Machine, en inglés) para la virtualización.
- Seguridad. El equipo de seguridad de Red Hat detecta vulnerabilidades de manera proactiva y garantiza un nivel de seguridad elevado. Asimismo, CentOS incluye la extensión del kernel SELinux (Security Enhanced Linux).
- Actualizaciones regulares y soporte. Las versiones de CentOS se actualizan de forma regular, aproximadamente cada 6 meses, y ofrecen soporte durante 10 años.

Figura 3.8: Centos



### 3.2.2. Security Onion

Security Onion [41] es una distribución de Linux abierta y gratuita para la búsqueda de amenazas, la supervisión de la seguridad empresarial y la gestión de registros. Security Onion proporcionará visibilidad de su tráfico de red y el contexto en torno a alertas y eventos anómalos.

Las herramientas que incluye Security Onion son las siguientes:

- Elasticsearch
- Logstash
- Kibana
- Snort
- Suricata
- Bro
- OSSEC
- Sguil
- Squert
- NetworkMiner
- muchas otras herramientas de seguridad.

Figura 3.9: Security Onion



### 3.2.3. Ubuntu

Ubuntu [42] es un sistema operativo de software libre y código abierto. Es una distribución de Linux basada en Debian. Puede correr en computadores de escritorio y servidores. Está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto.

Figura 3.10: Ubuntu



### **3.2.4. Windows 10**

Windows 10 [43] es el actual sistema operativo desarrollado por Microsoft como parte de la familia de sistemas operativos Windows NT. Fue dado a conocer oficialmente en septiembre de 2014, seguido por una breve presentación de demostración en la conferencia Build 2014.

Figura 3.11: Windows 10



## **3.3. Otras herramientas**

En este apartado se van a exponer el resto de herramientas utilizadas en el proyecto:

- ELK Stack
- Python
- MongoDB
- Overleaf Latex
- Bitbucket
- Github
- SublimeText
- VirtualBox
- Syslog-ng
- Pulled Pork

### **3.3.1. ELK Stack**

“ELK” es la sigla para tres proyectos open source: Elasticsearch, Logstash y Kibana. [44]

#### **ElasticSearch**

Elasticsearch es un motor de búsqueda y analítica.

#### **Logstash**

Logstash es un pipeline de procesamiento de datos del lado del servidor que ingesta datos de una multitud de fuentes simultáneamente, los transforma y luego los envía a un escondite, como Elasticsearch.

## Kibana

Kibana permite a los usuarios visualizar los datos en cuadros y gráficos con Elasticsearch.

Figura 3.12: ELK



### 3.3.2. Pyhton

Python [45] es un lenguaje de programación que surgió por los años 80's en Países Bajos. Se trata de un lenguaje de programación multiplataforma, ya que soporta orientación a objetos, programación interpretativa y programación funcional. Además, es un lenguaje interpretado, dinámico y multiplataforma.

Figura 3.13: Python



### 3.3.3. MongoDB

Base de datos NoSQL de código abierto. MongoDB almacena los datos en estructuras de datos BSON, similares a JSON, con un esquema dinámico que permite la integración de datos de forma fácil y rápida. [46]

Figura 3.14: MongoDB



### 3.3.4. Overleaf

Overleaf [47] es un servicio de LaTeX colaborativo en línea, que se ha utilizado para la realización de esta memoria. Cuenta con una detallada documentación que ha sido consultada de manera frecuente para la estructuración de la memoria del proyecto.

Figura 3.15: Overleaf



### 3.3.5. BitBucket

BitBucket [48] es una plataforma diseñada para la gestión de proyectos y control de versiones de código. Además, permite el trabajo colaborativo entre desarrolladores. En nuestro proyecto, se ha utilizado para gestionar el control de versiones de Git de la aplicación y alojar el código fuente.

Figura 3.16: Bitbucket



### 3.3.6. Github

Github [60] es una plataforma diseñada para la gestión de proyectos y control de versiones de código. Además, permite el trabajo colaborativo entre desarrolladores. En nuestro proyecto, se ha utilizado para gestionar el control de versiones de Git de la aplicación y alojar el código fuente.

Figura 3.17: Github



### 3.3.7. SublimeText

Sublime Text [49] es un editor de texto y editor de código fuente. Durante el proyecto se ha utilizado para modificar y añadir nuevas funcionalidades a la herramienta.

Figura 3.18: SublimeText



### 3.3.8. VirtualBox

Oracle VM VirtualBox [50] es un software de virtualización para arquitecturas x86/amd64. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Para este proyecto, ha sido la herramienta utilizada para alojar todas las máquinas virtuales. Además, se ha creado una red virtual nueva que ha permitido la conexión de todas las máquinas.

Figura 3.19: VirtualBox



### 3.3.9. Syslog-ng

Syslog-ng [51] es una implementación gratuita y de código abierto del protocolo syslog para sistemas Unix y similares a Unix. Durante el proyecto se ha utilizado para conectar las diferentes máquinas virtuales y poder así transportar los logs a la plataforma central CETA.

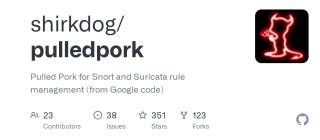
Figura 3.20: Syslog-ng

**syslog-ng**

### 3.3.10. Pulled Pork

Pulled Pork [52] es una herramienta basada en PERL para la gestión de reglas de Suricata y Snort. Puede determinar su versión de Snort y descargar automáticamente las últimas reglas para usted.

Figura 3.21: Pulled Pork





# Capítulo 4

## Plan de Desarrollo de Software

### 4.1. Introducción

Este capítulo está destinado a describir el plan de desarrollo del proyecto y de sus especificaciones técnicas. Este apartado provee una visión global del enfoque de desarrollo propuesto y una descripción abreviada de las especificaciones.

#### 4.1.1. Propósito

El propósito del Plan de Desarrollo de Software es el de proporcionar la información necesaria para tener control sobre el proyecto y describir su funcionalidad. Además, nos permite dotar el proyecto de una base teórica susceptible de revisión, modificaciones y mejoras. El propósito de las Especificaciones Técnicas es el de describir con cierto nivel de detalles las medidas adoptadas para implementar correctamente el Plan de Desarrollo.

#### 4.1.2. Usuarios

El Plan de Desarrollo de Software está concebido como herramienta de trabajo y referencia y no como como texto divulgativo para usuarios finales de la aplicación. La audiencia prevista para el mismo, por tanto, tiene un marcado perfil técnico. Como usuarios de este documento encontramos:

- **Alumno:** es el encargado de analizar, planificar, diseñar, desarrollar, implementar y documentar el proyecto.
- **Tutor:** es el encargado de la revisión periódica del proyecto, así como la solución de problemas que puedan surgir.

#### 4.1.3. Alcance

En esta memoria se incluye una visión general del proyecto, además de identificar los diferentes participantes involucrados y los roles de cada uno de ellos. También, se desarrolla un plan de trabajo en iteraciones y se añadirá más información que pueda ser relevante para el desarrollo del proyecto.

#### **4.1.4. Resumen**

De acuerdo a lo especificado, el Plan de Desarrollo de Software se organiza en secciones de la siguiente manera:

- **Vista general del proyecto.** Proporciona una descripción de los objetivos, suposiciones, características, tipo de desarrollo, entregables del proyecto.
- **Organización del proyecto.** Describe la estructura organizacional que se ha llevado a lo largo del proyecto en base a los roles de los usuarios.
- **Gestión de procesos del proyecto.** Explica los costos temporales y de planificación del proyecto, así como la desviación de estos mismos.

### **4.2. Visión general**

#### **4.2.1. Objetivos**

El objetivo principal del este proyecto consiste en la realización de una plataforma que se encargue de recoger toda la información de las herramientas de seguridad de una organización. Además, con toda la información recogida, será capaz de crear visualizaciones para cada una de las herramientas integradas en la plataforma, así como la correlación de los datos recibidos para realizar una detección temprana basada en eventos.

La plataforma permitirá visualizar el estado de la organización desde el exterior, es decir, se podrán usar herramientas open source para la observar qué activos están expuestos a internet y qué es lo que se ve de cada uno de ellos. Junto con lo anterior mencionado, tendremos una herramienta completa para supervisar y monitorizar el estado de una organización en tiempo real, y así, disponer de una herramienta de detección robusta.

#### **4.2.2. Restricciones y suposiciones**

El proyecto está limitado por las siguientes restricciones:

##### **1. Restricción de presupuesto:**

El proyecto es low cost, por lo que es fundamental el uso de herramientas y software libre.

##### **2. Restricción de personal:**

El proyecto contará con 2 integrantes, el alumnos y el tutor.

##### **3. Restricción temporal:**

El proyecto tendrá un plazo máximo similar a la finalización de las entregas de Trabajo de Fin de Máster establecido por la Universidad Pontificia Comillas.

#### 4.2.3. Características

Las características principales del proyecto son las siguientes:

- La duración estimada del proyecto comprende desde Febrero hasta Julio.
- El proyecto será desarrollado por el alumno y supervisado por el tutor.
- Documentación detallada que incluye la descripción del proceso de desarrollo, diagramas, diseño, planificación y manual de usuario.
- Requisitos susceptibles a cambios, adición de nuevos componentes a la aplicación en función de prioridad y tiempo principalmente.
- Es requisito indispensable la actualización y notificación de alertas en tiempo real, ya que lo más importante en un proyecto como este, es poder el poder detectar de manera temprana los posibles ataques sufridos en una organización.

#### 4.2.4. Metodología de desarrollo

Son muchas las metodologías empleadas para el desarrollo de software, que nos permiten crear un marco de trabajo para estructurar, planificar y controlar el proceso de desarrollo. En la actualidad hay múltiples tipos distintos de metodologías de desarrollo software: modelo en cascada, modelo en esprial, metodología de prototipo, metodología de programación extrema, metodologías ágiles etc.

Durante la realización de este proyecto, se ha determinado que la metodología a seguir es la de métodos ágiles, y más concretamente, el modelo SCRUM. La características [53] que hacen que SCRUM haya sido la metodología elegida son las siguientes:

- **Simplicidad:** el proyecto cuenta con un equipo limitado y tiempo limitado.
- **Rapidez:** tener durante la realización del proyecto un producto mínimo viable, y que sea a través de cada iteración cuando se le vaya añadiendo valor y funcionalidad, atendiendo siempre a las limitaciones de tiempo.
- **Flexibilidad:** adaptación a las diferentes circunstancias que se puedan dar a lo largo de todo el proyecto.
- **Retroalimentación:** obtener en cada una de las etapas la retroalimentación por parte del tutor.
- **Innovación:** en la línea de la características flexibilidad. En cada iteración se puede buscar innovar, sin alejarse de los objetivos finales de la aplicación.
- **Mantenibilidad y productividad:** La adaptación al proyecto hace que con cada iteración se avance a un ritmo mejor.
- **Mitigación de riesgos:** Los requisitos deben ser implementados en cada iteración, sin dejar ninguno de estos para el final y que derive en problemas o fallas.
- **Motivación:** Al estar compuestos de diferentes fases e iteraciones, supone que cada una de estas suponga un reto para el equipo y se afronte con la mayor motivación posible.

#### 4.2.5. Entregables

A continuación se indican y describen cada uno de los artefactos que han sido, son o serán generados y utilizados por el proyecto y que constituyen los entregables. Todos estos entregables estarán sujetos a cambios a lo largo de las diferentes iteraciones hasta alcanzar la iteración final.

- Plan de Desarrollo de Software
- Seguimiento del Proyecto
- Plan de Gestión de Riesgos
- Análisis
- Modelo de Diseño y Arquitectura
- Pruebas
- Versión final del producto
- Manual de usuario

### 4.3. Organización

#### 4.3.1. Roles

Los roles [54] del proyecto, van a ser los roles que conforman Scrum. Scrum está conformado por 3 roles principalmente: Product Owner, Scrum Master y Team.

- **Product Owner:** es el encargado de decidir el trabajo que debe hacerse. Entre sus funciones principales destaca:
  - Gestionar prioridades
  - Representante del negocio
- **Scrum Master:** Es el encargado de ayudar al equipo gestionando el uso correcto de la metodología Scrum. Además, se encarga de recudir los posibles impedimentos que puedan producirse y que, de este modo, no afecten al equipo.
- **Team:** Grupo de personas con los conocimiento técnicos necesarios para la realización del proyecto. Son los responsables de llevar a cabo la historia de los sprints y de la calidad y producción del software.

En este proyecto concreto, los roles se distribuyen de la siguiente manera:

Nombre	Rol
Juan Carlos Cortinas	Product Owner
Gonzalo Herreros Diezhandino	Scrum Master/Team

Cuadro 4.1: Roles en el proyecto

#### 4.3.2. Reuniones del proyecto

Al igual que ocurre con los roles, para las reuniones [55] del proyecto se han seguido los cinco eventos de Scrum para cumplir con el control del proceso: definición de Backlog del Producto, Planificación del Sprint, Scrum diario, Revisión del sprint y Retrospectiva del sprint.

Evento	Descripción
Backlog del Producto	Reunión inicial en la que se plantea la idea inicial y planificación del proyecto
Planificación del sprint	Reunión de trabajo previa al inicio de un sprint, en el que se determinan los objetivos y las tareas que llevan a lograr dichos objetivos.
Scrum diario	Breve reunión diaria del equipo. Se realiza la revisión de tareas del día anterior y tareas a realizar en el día actual.
Revisión del sprint	Reunión a la finalización de cada sprint, en el que se muestra lo realizado durante el sprint.
Retrospectiva del sprint	Reunión donde se analiza lo que se ha realizado correctamente durante el sprint y aquello que puede ser mejorado.

Cuadro 4.2: Eventos scrum

### 4.4. Planificación y estimación

#### 4.4.1. Estimación temporal

El proyecto siguiendo la metodología Scrum, se ha seguido dividiéndolo en diferentes sprints. Los Sprints iniciales tienen una duración mayor que los Sprints siguientes. Aunque no debería ser así, debido que el tiempo por Sprint debe ser similar, en este caso las tareas a realizar en el inicio del proyecto suponen un mayor esfuerzo y una dedicación mayor debido al desconocimiento previo por parte de los integrantes del equipo. Una vez todo se puso en marcha, la velocidad de Sprints se estabilizó, creando en cada iteración producto más completo y mejorado.

A continuación se muestra la planificación de los sprints, el orden de cada uno y las fechas de inicio y fin:

Sprint	Fecha de inicio	Fecha de fin
Sprint 1	22/02/2021	14/03/2021
Sprint 2	15/03/2021	15/04/2021
Sprint 3	15/04/2021	05/05/2021
Sprint 4	05/05/2021	14/05/2021
Sprint 5	15/05/2021	26/05/2021
Sprint 6	27/05/2021	04/06/2021
Sprint 7	05/06/2021	16/06/2021
Sprint 8	17/06/2021	23/06/2021
Sprint 9	24/06/2021	04/07/2021
Sprint 10	05/07/2021	09/07/2021

Cuadro 4.3: Sprints del proyecto

Como se ve el trabajo se ha ido realizando en diferentes fases/sprints, en el cuál cada uno de ellos se ha aportado gran cantidad de valor al proyecto. El tiempo de trabajo en cada uno de los sprints varía de uno a otro, pero se estima un trabajo de 5 horas diarias durante todos los sprints, con una jornada laboral de lunes a viernes.

El cálculo de horas totales del proyecto, es el siguiente:

- 5 días de trabajo a la semana \* 5 horas diarias de trabajo \* 22 semanas trabajadas = 550 horas

#### 4.4.2. Estimación de costes

En este proyecto, pese a ser un trabajo encargado a un alumno y ser un Proyecto de Trabajo de Fin de Máster, sí que existe una remuneración establecida en unos 500 euros mensuales. Si se tratase de un proyecto real, un trabajo de este tipo, estimado en unas 550 horas y, teniendo en cuenta el sueldo [56] por horas de un Ingeniero Informático en España, que se encuentra en torno a los 10 euros la hora, la atribución total a percibir por el ingeniero sería de unos 5500 euros por el desarrollo del proyecto.

En cuanto al coste de las herramientas utilizadas en este proyecto, es nulo. Es un proyecto "Low Cost", en el que se ha tratado de utilizar todas las herramientas necesarias en su versión gratuita, sin coste alguno.

#### 4.4.3. Desviación del proyecto

El proyecto no se ha desviado demasiado de su estimación inicial. Es cierto que en las primeras fases, debido a la inexperiencia con ciertas herramientas y a la necesidad de aprender el funcionamiento de estas mismas, el proceso fue mucho más lento. Pero por eso, la estimación de las primeras fases es mayor en comparación con los sprints finales.

Por todo esto, se puede decir que la estimación ha sido correcta y que se ha completado el proyecto dentro de las fechas límites establecidas.

## Capítulo 5

# Seguimiento del Proyecto

### 5.1. Introducción

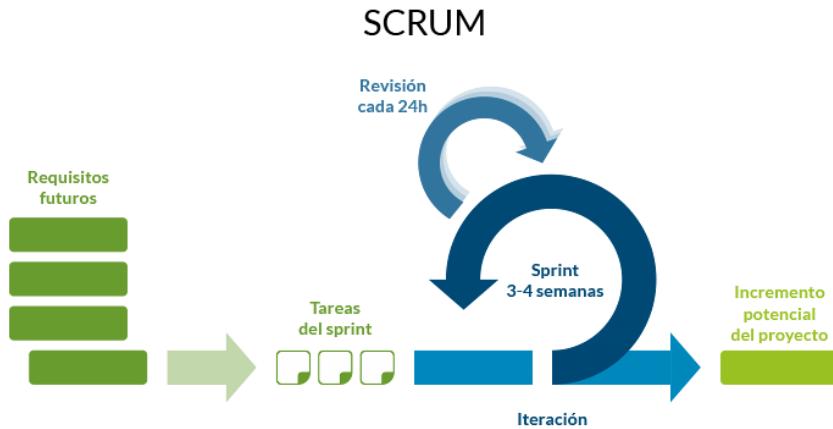
En este capítulo se va a explicar las diferentes fases que ha seguido el proyecto. Como ya se ha comentado en capítulos anteriores, la metodología utilizada para el desarrollo del proyecto es Scrum. Scrum es una metodología ágil de desarrollo software, basada principalmente en la estrategia de desarrollo incremental.

En la Figura 5.1 podemos observar el flujo de trabajo de la metodología Scrum. A continuación procederemos a explicar cada una de las etapas que la componen:

- **Requisitos futuros:** Hay que entender un sprint como un miniproyecto dentro del proyecto global, y cada uno de estos miniproyectos tienen un objetivo concreto. Esta primera etapa se produce al inicio del sprint, y en ella, se definirán aspectos como la funcionalidad, objetivos o riesgos del sprint. Además, en esta etapa, se explicará cómo se desarrollará cada punto del sprint.
- **Tareas del sprint:** En esta etapa el sprint se divide en subtareas, de tal forma que se asigna prioridades a cada una de ellas y se distribuyen para que estas sean asumibles por el equipo del trabajo.
- **Iteración:** Durante el periodo establecido por el sprint, se trabaja sobre las diferentes tareas encomendadas para dicho sprint.
- **Revisión diaria:** Cada día se hace una evaluación del trabajo realizado el día anterior y del trabajo por realizar en el día actual.
- **Incremento potencial del proyecto:** Tras la finalización del miniproyecto, se obtiene una parte funcional del proyecto. Además, durante esta etapa se hace balance del sprint, en definitiva, se analiza y evalua los resultados.

Los entregables tras esta etapa, sirven no sólo para recibir feedback por parte de los desarrolladores, sino que también es utilizado para que los usuarios finales puedan dar su feedback de la iteración.

Figura 5.1: Flujo de Trabajo Scrum



## 5.2. Sprints

A continuación, vamos a ir detallando cada uno de los sprints en los que ha sido dividido el proyecto.

### 5.2.1. Sprint 1: Reuniones - Inicio proyecto

El Sprint 1 supone el comienzo del proyecto. Durante este sprint, se realizan las reuniones iniciales entre Product Owner y Scrum Master, en el que se detallan los objetivos globales del proyecto. Además, esta etapa supone la primera toma de contacto con la herramienta CETA.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Reunión asignación TFM
- Preparación del entorno de trabajo (Máquina virtual Centos)
- Documentación Inicial
- Estudio herramientas

#### Reunión asignación TFM

Reunión entre Product Owner y Scrum Master, para la asignación del proyecto. Además, de la asignación se exponen los objetivos principales del proyecto.

**Estimación:** 1 punto de historia.

## Preparación del entorno de trabajo

Esta tarea consiste en la preparación del entorno que posteriormente se utilizará para la realización del proyecto. Consta de diferentes subtareas, que se detallan a continuación:

- Instalación de máquina virtual CENTOS que contiene el programa previo CETA.
- Crear repositorio en Bitbucket y enlazarlo con el proyecto.

**Estimación:** 2 puntos de historia.

## Documentación inicial

Esta tarea consiste en redactar el Anexo A y B con las especificaciones que servirán de guía para desarrollar el proyecto.

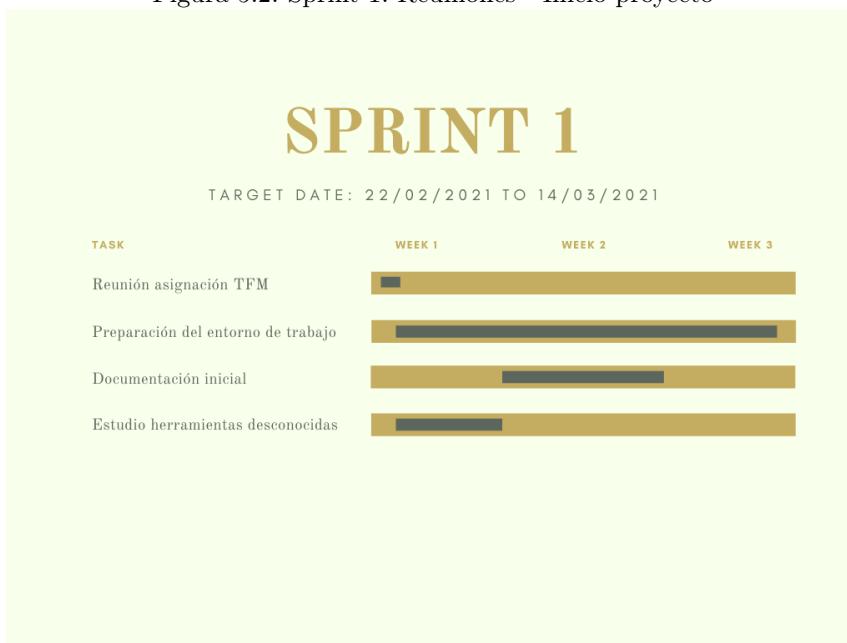
**Estimación:** 2 puntos de historia.

## Estudio de herramientas

Esta tarea consiste en estudiar la herramienta CETA desarrollada por el Product Owner (Juan Carlos Cortinas). Además, es necesario también estudiar las herramientas que se van a incorporar al proyecto, como son: Snort, Suricata, Ossec, Wazuh, AlienVault...

**Estimación:** 3 puntos de historia.

Figura 5.2: Sprint 1: Reuniones - Inicio proyecto



### **5.2.2. Sprint 2: Puesta a Punto CETA**

El Sprint 2 corresponde con la puesta en funcionamiento de CETA, tal cuál como se encontraba anteriormente. Las herramientas que utilizaba CETA se vieron obsoletas algunas de ellas y fue necesario la realización de modificaciones en el código y la descarga de nuevas versiones para poner de nuevo a punto dicha herramienta.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Descarga de nuevas versiones de las herramientas
- Instalación de dependencias
- Modificación de código

#### **Descarga de nuevas versiones de las herramientas**

Esta tarea consiste en la actualización de versión de las herramientas utilizadas en la plataforma. En este caso, fue necesaria la actualización de ELK Stack por completo (ElasticSearch, Logstash y Kibana), así como la plataforma CVE Search.

**Estimación:** 2 puntos de historia.

#### **Instalación de dependencias**

Esta tarea consiste en la instalación de las dependencias instaladas a mano anteriormente por el Scrum Master. De este modo se creó un programa encargado de descargar y mantener actualizado todas estas dependencias al iniciar la herramienta CETA.

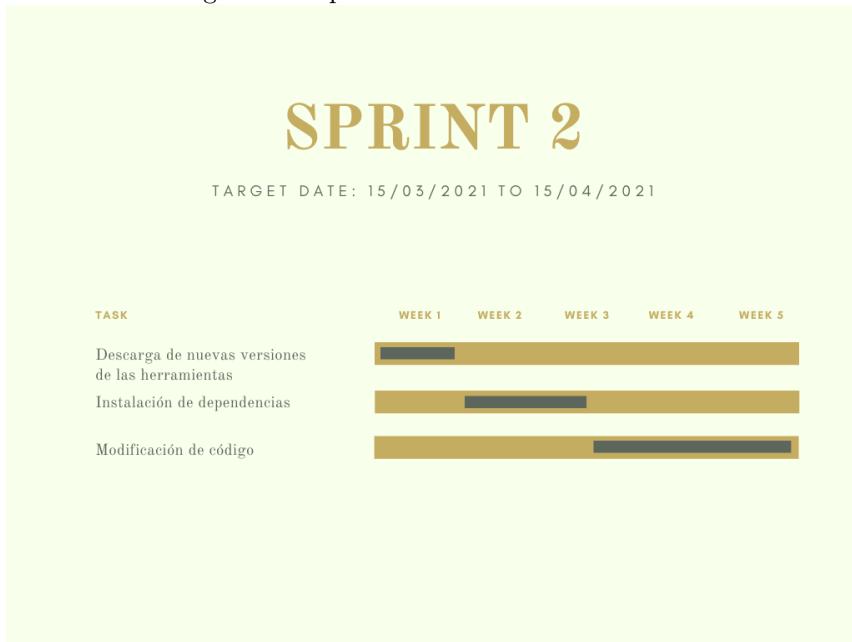
**Estimación:** 2 puntos de historia.

#### **Modificación de código**

Esta tarea consistió en corregir el código para que funcionase correctamente con las nuevas versiones instaladas de las herramientas, ya que existían partes del código obsoletas. Fue una tarea laboriosa debido a la necesidad de ir debbugando el código poco a poco, para poder observar correctamente los diferentes puntos de fallo.

**Estimación:** 5 puntos de historia.

Figura 5.3: Sprint 2: Puesta a Punto CETA



### 5.2.3. Sprint 3: Suricata

El Sprint 3 corresponde con la integración en CETA de la herramienta Suricata.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Estudio de sistema operativo Security Onion
- Instalación de máquina virtual Security Onion
- Configuración de Suricata
- Conexión CENTOS - Security Onion
- Uso de ELK Stack

#### Estudio de sistema operativo Security Onion

Esta tarea consistió en el estudio del sistema operativo Security Onion y todas las ventajas que proporciona a la hora de monitorizar sistemas debido a las herramientas de monitorización que trae incorporado.

**Estimación:** 1 puntos de historia.

## **Instalación de máquina virtual Security Onion**

Esta tarea consistió en la instalación de la máquina virtual que alojaba el sistema operativo Security Onion. Además, fue necesario seleccionar la configuración de Security Onion que mejor se ajustaba al proyecto, en este caso se optó por la que tenía la herramienta Suricata.

**Estimación:** 1 puntos de historia.

## **Configuración de Suricata**

Esta tarea consistió en configurar Suricata. Debido a la selección de configuración de Security Onion, Suricata ya se encontraba instalado por defecto, por lo que únicamente fue necesario revisar que funcionase correctamente y que la configuración de alertas fuese la adecuada.

**Estimación:** 1 puntos de historia.

## **Conexión CENTOS - Security Onion**

Esta tarea consistió en conectar la máquina virtual Security Onion con la máquina CENTOS en la que se aloja CETA. La conexión consistió en facilitar el envío de logs de Suricata de una máquina a otra a través de la herramienta Syslog-ng.

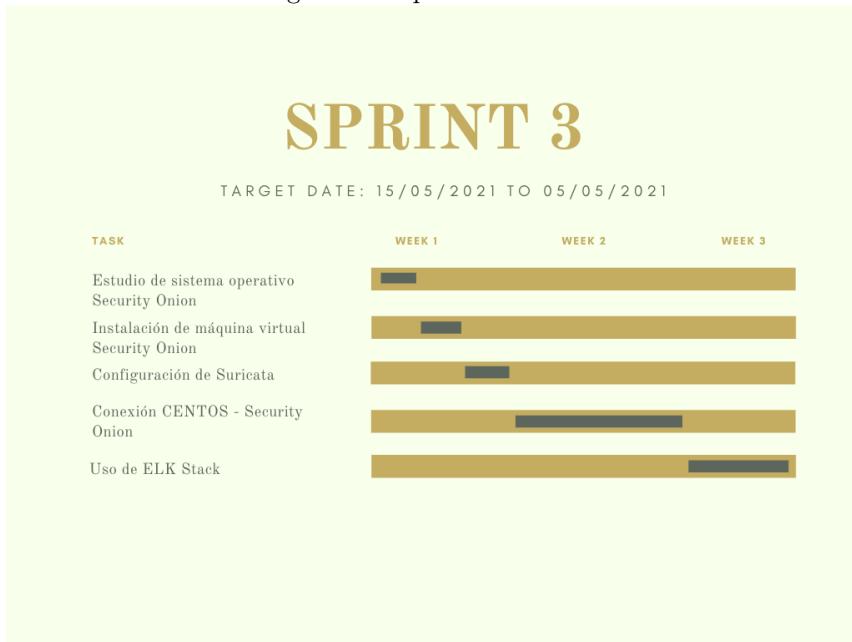
**Estimación:** 2 puntos de historia.

## **Uso de ELK Stack**

Esta tarea consistió en el uso de las herramientas ELK Stack, para la importación y correcta visualización de los logs recibidos de Suricata. Para ello, se utilizaron las 3 herramientas que ELK Stack proporciona, como son: Logstash, ElasticSearch y Kibana.

**Estimación:** 3 puntos de historia.

Figura 5.4: Sprint 3: Suricata



#### 5.2.4. Sprint 4: Snort

El Sprint 4 corresponde con la integración en CETA de la herramienta Snort.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Instalación de máquina virtual Ubuntu
- Configuración de Snort
- Conexión CENTOS - Ubuntu
- Uso de ELK Stack

#### Instalación de máquina virtual Ubuntu

Esta tarea consistió en la instalación de la máquina virtual que alojaba el sistema operativo Ubuntu.

**Estimación:** 1 puntos de historia.

#### Configuración de Snort

Esta tarea consistió en configurar Snort. Fue necesario su instalación, así como la comprobación de que la configuración fuese la correcta.

**Estimación:** 2 puntos de historia.

### Conexión CENTOS - Ubuntu

Esta tarea consistió en conectar la máquina virtual Ubuntu con la máquina CENTOS en la que se aloja CETA. La conexión consistió en facilitar el envío de logs de Snort de una máquina a otra a través de la herramienta Syslog-ng.

**Estimación:** 2 puntos de historia.

### Uso de ELK Stack

Esta tarea consistió en el uso de las herramientas ELK Stack, para la importación y correcta visualización de los logs recibidos de Snort. Para ello, se utilizaron las 3 herramientas que ELK Stack proporciona, como son: Logstash, ElasticSearch y Kibana.

**Estimación:** 3 puntos de historia.

Figura 5.5: Sprint 4: Snort



### 5.2.5. Sprint 5: Ossec

El Sprint 5 corresponde con la integración en CETA de la herramienta Ossec.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Instalación de máquina virtual Ubuntu Agente y Servidor

- Configuración de Ossec
- Conexión CENTOS - Ubuntu Servidor
- Uso de ELK Stack

## **Instalación de máquina virtual Ubuntu Agente y Servidor**

Esta tarea consistió en la instalación de 2 máquinas virtuales Ubuntu, una que actuará como agente de la herramienta Ossec y otra que funcionará como servidor de la herramienta Ossec.

**Estimación:** 1 puntos de historia.

## **Configuración de Ossec**

Esta tarea consistió en configurar Ossec. En una de las máquinas virtuales se instaló Ossec como agente y en la otra como servidor. Posteriormente fue necesario la conexión de agente y servidor. Además se observó la configuración de Ossec y se añadieron nuevas reglas.

**Estimación:** 2 puntos de historia.

## **Conexión CENTOS - Ubuntu Servidor**

Esta tarea consistió en conectar la máquina virtual Ubuntu que aloja el Servidor de Ossec con la máquina CENTOS en la que se aloja CETA. La conexión consistió en facilitar el envío de logs de Ossec de una máquina a otra a través de la herramienta Syslog-ng.

**Estimación:** 2 puntos de historia.

## **Uso de ELK Stack**

Esta tarea consistió en el uso de las herramientas ELK Stack, para la importación y correcta visualización de los logs recibidos de Ossec. Para ello, se utilizaron las 3 herramientas que ELK Stack proporciona, como son: Logstash, Elasticsearch y Kibana.

**Estimación:** 3 puntos de historia.

Figura 5.6: Sprint 5: Ossec



### 5.2.6. Sprint 6: Wazuh

El Sprint 6 corresponde con la integración en CETA de la herramienta Wazuh.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Instalación de máquina virtual Windows 10 Agente y Ubuntu Servidor
- Configuración de Wazuh
- Conexión CENTOS - Ubuntu Servidor
- Uso de ELK Stack

#### Instalación de máquina virtual Windows 10 Agente y Ubuntu Servidor

Esta tarea consistió en la instalación de 2 máquinas virtuales, una que actuará como agente (Windows 10) de la herramienta Wazuh y otra que funcionará como servidor (Ubuntu) de la herramienta Wazuh.

**Estimación:** 1 puntos de historia.

#### Configuración de Wazuh

Esta tarea consistió en configurar Wazuh. En una de las máquinas virtuales se instaló Wazuh como agente y en la otra como servidor. Posteriormente fue necesario la conexión de

agente y servidor. Además se observó la configuración de Wazuh que se encuentra por defecto.

**Estimación:** 2 puntos de historia.

### Conexión CENTOS - Ubuntu Servidor

Esta tarea consistió en conectar la máquina virtual Ubuntu que aloja el Servidor de Wazuh con la máquina CENTOS en la que se aloja CETA. La conexión consistió en facilitar el envío de logs de Wazuh de una máquina a otra a través de la herramienta Syslog-ng.

**Estimación:** 2 puntos de historia.

### Uso de ELK Stack

Esta tarea consistió en el uso de las herramientas ELK Stack, para la importación y correcta visualización de los logs recibidos de Wazuh. Para ello, se utilizaron las 3 herramientas que ELK Stack proporciona, como son: Logstash, ElasticSearch y Kibana.

**Estimación:** 3 puntos de historia.

Figura 5.7: Sprint 6: Wazuh



### 5.2.7. Sprint 7: AlienVault OSSIM

El Sprint 8 corresponde con el intento de incorporación a CETA de la herramienta de monitorización AlienVault. Durante este periodo se estuvo investigando la herramienta y

probando su funcionamiento para su posible futura incorporación al sistema central.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Estudio AlienVault
- Instalación y prueba de conexión

### **Estudio AlienVault**

Esta tarea consistió en estudiar la herramienta AlienVault y sus posibilidades de cara a una futura integración en CETA.

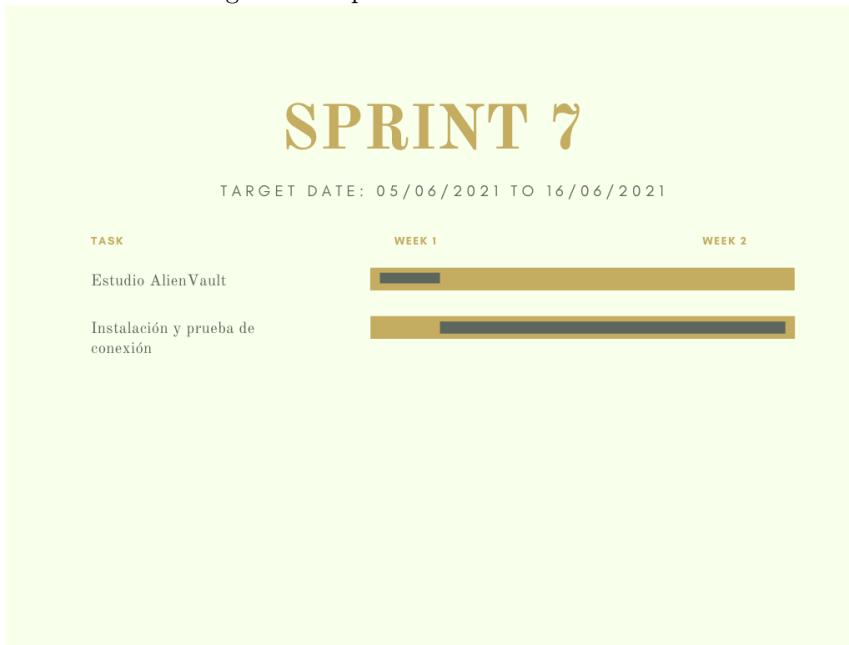
**Estimación:** 1 puntos de historia.

### **Instalación y prueba de conexión**

Esta tarea consistió en la instalación de AlienVault y configuración de la herramienta. Además se probó a conectar a CETA, pero dicha tarea era muy tediosa.

**Estimación:** 3 puntos de historia.

Figura 5.8: Sprint 7: AlienVault OSSIM



### **5.2.8. Sprint 8: Correlación Datos**

El Sprint 8 corresponde con la correlación de todos los logs recogidos por la herramienta CETA. Para realizar la correlación de los datos será necesario crear un dataset que unifique los datos de las diferentes fuentes, para posteriormente poder crear una visualización con estos nuevos datos.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Programa de correlación
  
- Integración en CETA

#### **Programa de correlación**

Esta tarea consistió en la realización del programa encargado de correlacionar los datos y crear un dataset que contuviese la información de las diferentes herramientas en un formato unificado.

**Estimación:** 3 puntos de historia.

#### **Integración en CETA**

Esta tarea consistió en integrar en CETA el programa de correlación para poder visualizar en el dashboard los nuevos datos correlacionados.

**Estimación:** 2 puntos de historia.

Figura 5.9: Sprint 8: Correlación Datos



### 5.2.9. Sprint 9: Clustering

El Sprint 9 corresponde al clusterizado de los datos unificados tras la correlación.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Programa de clusterizado
- Integración en CETA

#### Programa de clusterizado

Esta tarea consistió en la realización del programa encargado de clusterizar los datos unificados tras la correlación. Para ello fue necesario ir comprobando las diferentes opciones e ir observando cuál es el modelo que mejor se ajustaba a los datos.

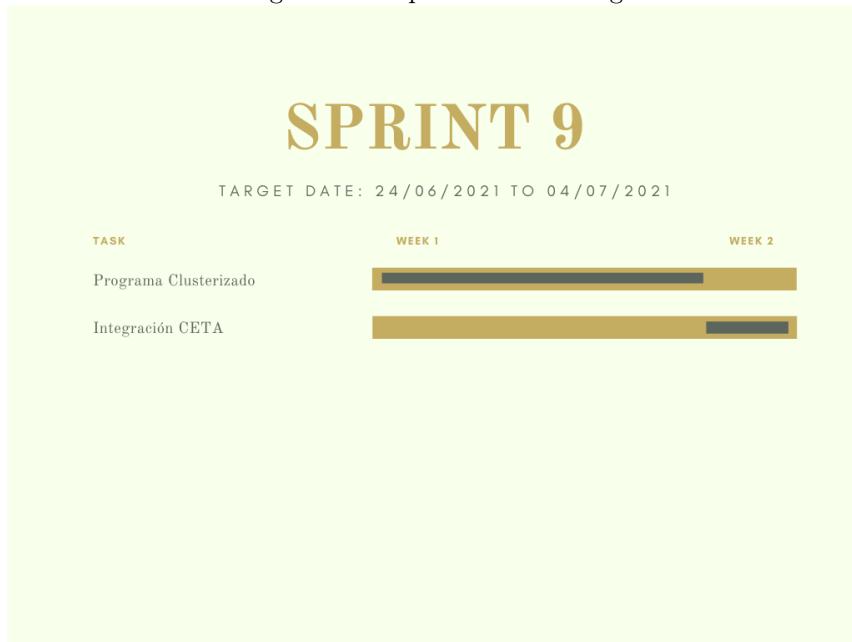
**Estimación:** 3 puntos de historia.

#### Integración en CETA

Esta tarea consistió en integrar en CETA el programa de clusterizado para poderlo visualizar en el dashboard.

**Estimación:** 2 puntos de historia.

Figura 5.10: Sprint 9: Clustering



### 5.2.10. Sprint 10: Documentación y Revisión final

El Sprint 10 consistió en la documentación y revisión del proyecto final.

Se detallan a continuación, las diferentes tareas del sprint, que han de ser estimadas en una escala de 0-5 puntos de historia, en función del tiempo requerido para cada una de ellas.

- Documentación
- Revisión

#### Documentación

Esta tarea consistió la documentación exhaustiva de todos aquellos aspectos que faltaban en el documento.

**Estimación:** 3 puntos de historia.

#### Revisión

Esta tarea consistió en revisar la documentación y el software desarrollado.

**Estimación:** 2 puntos de historia.

Figura 5.11: Sprint 10: Documentación y Revisión final



# Capítulo 6

# Plan de Gestión de Riesgos

## 6.1. Introducción

Los riesgos son eventos o circunstancias cuya probabilidad de incidencia es incierta, pero que en caso de que ocurra tiene un efecto, normalmente negativo, sobre los objetivos del proyecto. La probabilidad de ocurrencia de un riesgo difiere entre la multitud de riesgos posibles que puede tener un proyecto. Por eso, es importante saber que la exposición a un riesgo depende directamente de la probabilidad de que este ocurra y la perdida asociada que se estima a cada riesgo.

El Plan de Gestión de Riesgos tiene como principal finalidad describir las responsabilidades y actividades relacionadas con la gestión de riesgos. Un Plan de Gestión de Riesgos se encarga de definir lo siguiente:

- Organigrama para la gestión de riesgos.
- Proceso de identificación y análisis de riesgos.
- Herramientas y técnicas a utilizar.
- Taxonomía de riesgos a utilizar.
- Plantillas estandarizadas para la identificación y gestión de riesgos.
- Actividades de control de riesgos y periodicidad de las mismas.

En este proyecto se va a realizar un proceso de evaluación de riesgos que contiene las siguientes pautas:

- Identificar riesgos.
- Análisis cualitativo de riesgos.
- Plan de respuesta al riesgo.
- Plan de contingencia.

### **6.1.1. Identificación del riesgo**

Esta es la primera de las tareas a realizar en un plan de gestión de riesgos. Hay múltiples maneras de identificar los riesgos de un proyecto software:

- Revisión de documentación existente sobre riesgos.
- Revisión de planificación y estimaciones.
- Lluvia de ideas.
- Juicio experto: método Delphi. [57]
- Taxonomía de riesgos.
- Análisis SWOT. [58]
- Diagrama de Ishikawa. [59]

En este proyecto se utilizarán principalmente las técnicas de: Revisión de documentación existente y lluvia de ideas.

### **6.1.2. Análisis cualitativo de riesgos**

Consiste en definir de manera cualitativa la importancia de la prioridad de cada riesgo. Las técnicas empleadas para realizar este análisis son las siguientes:

- Juicio experto.
- Tablas de impacto.
- Matrices de probabilidad e impacto.
- Agrupación por causas.
- Agrupación por prioridad temporal.

### **6.1.3. Plan de respuesta al riesgo y contingencia**

La principal finalidad es la de atenuar la probabilidad de o el impacto de los riesgos mediante la inserción de actividades y recursos en la planificación del proyecto. Estas son algunas de las técnicas seguidas para el plan de respuesta al riesgo:

- Evitar riesgos.
- Transferir el riesgo.
- Atenuar el riesgo.
- Aceptación del riesgo.

## 6.2. Gestión de riesgos

A continuación se muestran el listados de riesgos del proyecto:

<b>R01</b>	Inexperiencia alumno
<b>Causa</b>	Alumno sin suficientes conocimientos
<b>Consecuencia</b>	Mal desarrollo del proyecto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	MEDIO
<b>Plan de acción</b>	Consultar documentación o con expertos
<b>Plan de contingencia</b>	Evaluar conocimientos adquiridos

Cuadro 6.1: Riesgo 01

<b>R02</b>	Mala estimación de tiempos
<b>Causa</b>	Estimación no se ciñe a la realidad
<b>Consecuencia</b>	Retraso del proyecto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	MEDIA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Planificación real
<b>Plan de contingencia</b>	Realizar seguimiento durante el proyecto

Cuadro 6.2: Riesgo 02

<b>R03</b>	Escatimar calidad del proyecto
<b>Causa</b>	No realizar las pruebas de calidad correspondientes
<b>Consecuencia</b>	Baja calidad del producto final
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	MEDIA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Realizar pruebas pertinentes
<b>Plan de contingencia</b>	Dedicar un tiempo tras cada iteración a las pruebas

Cuadro 6.3: Riesgo 03

<b>R04</b>	Pérdida de información
<b>Causa</b>	Se pierde código o datos del proyecto
<b>Consecuencia</b>	Rehacer la parte perdida
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Rehacer lo perdido
<b>Plan de contingencia</b>	Realizar copias de seguridad

Cuadro 6.4: Riesgo 04

<b>R05</b>	Esfuerzo mayor que el estimado
<b>Causa</b>	Dedicación en una tarea lleva más de lo estimado
<b>Consecuencia</b>	Retraso en el proyecto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	MEDIA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Reestructurar el sprint
<b>Plan de contingencia</b>	Realizar ajustes de estimación realistas

Cuadro 6.5: Riesgo 05

<b>R06</b>	Costoso aprendizaje de herramientas
<b>Causa</b>	La curva de aprendizaje para la nueva herramienta de desarrollo es más larga de lo esperado
<b>Consecuencia</b>	Retraso de la tarea
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	MEDIA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Reestructurar planificación
<b>Plan de contingencia</b>	Pedir ayuda a expertos

Cuadro 6.6: Riesgo 06

<b>R07</b>	Herramientas de desarrollo inadecuadas
<b>Causa</b>	Las herramientas de desarrollo no funcionan como se esperaba
<b>Consecuencia</b>	Retraso de la tarea
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Cambio de herramienta
<b>Plan de contingencia</b>	Informarse de herramientas previamente

Cuadro 6.7: Riesgo 07

<b>R08</b>	Producto final no se ajusta a usuario
<b>Causa</b>	No se ha solicitado información al usuario, por lo que el producto al final no se ajusta a las necesidades del usuario
<b>Consecuencia</b>	Reestructuración del producto
<b>Prioridad</b>	BAJA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	CRITICO
<b>Plan de acción</b>	Rediseño de producto final
<b>Plan de contingencia</b>	Consultar con usuarios

Cuadro 6.8: Riesgo 08

<b>R09</b>	Aparición de nuevos requisitos
<b>Causa</b>	Tutor exige nuevos requisitos
<b>Consecuencia</b>	Reestructuración del producto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	MEDIA
<b>Impacto</b>	MEDIO
<b>Plan de acción</b>	Rediseño de nuevos requisitos
<b>Plan de contingencia</b>	Requisitos claros desde inicio del proyecto

Cuadro 6.9: Riesgo 09

<b>R10</b>	Seguimiento incorrecto del proyecto
<b>Causa</b>	No se producen reuniones o feedback para el seguimiento del proyecto
<b>Consecuencia</b>	Pueden suponer retrasos en el proyecto o errores en el proyecto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	MEDIO
<b>Plan de acción</b>	Solicitar reunión para seguimiento
<b>Plan de contingencia</b>	Realizar seguimiento del proyecto

Cuadro 6.10: Riesgo 10

<b>R11</b>	Disponibilidad de miembros del proyecto
<b>Causa</b>	Baja disponibilidad de los miembros del proyecto
<b>Consecuencia</b>	Pueden suponer retrasos en el proyecto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	MEDIO
<b>Plan de acción</b>	Ajustar proyecto a disponibilidad del equipo
<b>Plan de contingencia</b>	Consultar disponibilidad

Cuadro 6.11: Riesgo 11

<b>R12</b>	Robo de dispositivos
<b>Causa</b>	Robo de alguna herramienta de trabajo
<b>Consecuencia</b>	Puede suponer pérdida de parte del proyecto
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	BAJA
<b>Impacto</b>	ALTO
<b>Plan de acción</b>	Recuperar dispositivos
<b>Plan de contingencia</b>	Guardar dispositivos a buen recaudo

Cuadro 6.12: Riesgo 12

<b>R13</b>	Inposibilidad de acceder a herramientas
<b>Causa</b>	Recursos humanos no permite el acceso a las herramientas de la organización
<b>Consecuencia</b>	Puede suponer pérdida de parte de la funcionalidad
<b>Prioridad</b>	ALTA
<b>Probabilidad</b>	ALTA
<b>Impacto</b>	MEDIO
<b>Plan de acción</b>	Buscar nuevas herramientas Open Source
<b>Plan de contingencia</b>	Buscar nuevas opciones

Cuadro 6.13: Riesgo 13

### 6.3. Control de riesgos

Uno de los objetivos principales del control de riesgos es el de actualizar el registro de riesgos conforme avanza el proyecto, indentificando y analizando nuevos riesgos que pudieran surgir, y elaborando nuevas respuestas a esos riesgos. Además, otro de los objetivos es el de comprobar si se ha materialziado alguno de los riesgos, y en caso de ser así, ejecutar los planes de acción correspondientes al riesgo. Por último, otra de la finalidades del control de riesgos es la de realizar el seguimiento de los planes de respuesta en ejecución y administrar el fondo de reserva para contingencias.

Durante la realización del proyecto se han materializado los siguientes riesgos potenciales anteriormente mencionados:

- Riesgo 13: Imposibilidad de acceder a herramientas. Inicialmente la idea del proyecto era la de poder acceder a las herramientas internas de la organización Iberdrola, como son el SIEM QRadar, CMDB Aris, Rapid Nexpose. Por circunstancias ajenas al tutor y alumno de este TFM no se ha podido llevar a cabo esta tarea debido a la negativa por parte de recursos humanos de tener acceso a dichas herramientas. Como solución se ha adoptado la mencionada anteriormente de utilizar herramientas de monitorización Open Source como son Snort, Suricata, Ossec, Wazuh etc.
- Riesgo 06: Costoso aprendizaje de herramientas. Ciertas herramientas como es el caso de AlienVault OSSIM su aprendizaje fue costoso, hasta el punto de que su incorporación a

CETA se determinó que era demasiado compleja y no se llegó a producir por su elevado coste de tiempo.

- Riesgo 05: Esfuerzo mayor que el estimado. La puesta a punto de la herramienta CETA supuso un esfuerzo mayor que el estimado inicialmente. La aparición de nuevos errores durante la realización de esta tarea fue la que supuso este esfuerzo mayor que el estimado.



# Capítulo 7

## Análisis

### 7.1. Análisis de requisitos

#### 7.1.1. Introducción

La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software. Cliente y desarrollador desempeñan un papel importante en la ingeniería de requisitos.

El análisis de requisitos es una tarea de ingeniería del software que se encuentra entre la tarea de definición de software a nivel de sistema y el diseño de software. El análisis de requisitos permite al desarrollador del sistema especificar las características operacionales del software.

En cuanto al análisis de requisitos en el desarrollo de metodologías ágiles difiere del análisis tradicional. La principal diferencia radica en el conocimiento por parte de todo el equipo de las necesidades del cliente, al que se le intenta guiar para adaptar el producto final a realizar.

Los requisitos en una metodología ágil se separan en diferentes historias de usuario, agrupadas todas ella en un Backlog que está ordenado prioridad y que irá evolucionando durante el desarrollo del proyecto.

En este apartado trataremos los siguientes requisitos: funcionales, no funcionales y de información.

#### 7.1.2. Requisitos funcionales

Los requisitos funcionales describen el funcionamiento del sistema. A continuación en las **tablas 7.1 y 7.2** se muestra un listado con los requisitos funcionales del proyecto.

ID	Requisito	Descripción
RF01	Buscar en Shodan	El sistema deberá permitir buscar los activos de una organización en Shodan
RF02	Buscar en CVE Search	El sistema deberá permitir buscar las vulnerabilidades existentes para los activos de la organización en CVE Search API
RF03	Conectar Suricata	El sistema deberá permitir conectarse a cualquier sistema que albergue la herramienta Suricata
RF04	Conectar Snort	El sistema deberá permitir conectarse a cualquier sistema que albergue la herramienta Snort
RF05	Conectar Ossec	El sistema deberá permitir conectarse a cualquier sistema que albergue la herramienta Ossec
RF06	Conectar Wazuh	El sistema deberá permitir conectarse a cualquier sistema que albergue la herramienta Wazuh
RF07	Conectar AlienVault OSSIM	El sistema deberá permitir conectarse a cualquier sistema que albergue la herramienta AlienVault OSSIM
RF08	Recibir alertas Suricata	El sistema deberá permitir la recepción de las alertas generadas por la herramienta Suricata
RF09	Recibir alertas Snort	El sistema deberá permitir la recepción de las alertas generadas por la herramienta Snort
RF10	Recibir alertas Ossec	El sistema deberá permitir la recepción de las alertas generadas por la herramienta Ossec
RF11	Recibir alertas Wazuh	El sistema deberá permitir la recepción de las alertas generadas por la herramienta Wazuh
RF12	Recibir alertas AlienVault OSSIM	El sistema deberá permitir la recepción de las alertas generadas por la herramienta AlienVault OSSIM
RF13	Visualizar datos Shodan	El sistema deberá permitir la visualización de los datos recibidos por Shodan tras la búsqueda
RF14	Visualizar datos CVE Search	El sistema deberá permitir la visualización de los datos recibidos por CVE Search API tras la búsqueda
RF15	Visualizar datos Suricata	El sistema deberá permitir la visualización de las alertas recibidas por la herramienta Suricata
RF16	Visualizar datos Snort	El sistema deberá permitir la visualización de las alertas recibidas por la herramienta Snort
RF17	Visualizar datos Ossec	El sistema deberá permitir la visualización de las alertas recibidas por la herramienta Ossec

Cuadro 7.1: Requisitos Funcionales 1

ID	Requisito	Descripción
RF18	Visualizar datos Wazuh	El sistema deberá permitir la visualización de las alertas recibidas por la herramienta Wazuh
RF19	Visualizar datos AlienVault OSSIM	El sistema deberá permitir la visualización de las alertas recibidas por la herramienta Alien-Vault OSSIM
RF20	Correlacionar Datos	El sistema deberá permitir la correlación de los datos recibidos de las diferentes herramientas de monitorización a las que se encuentra conectado
RF21	Visualizar datos correlacionados	El sistema deberá permitir la visualización de los datos correlacionados
RF22	Clusterizar datos correlacionados	El sistema deberá permitir el clusterizado de los datos correlacionados de las diferentes herramientas
RF23	Visualizar clusterizado de datos	El sistema deberá permitir la visualización del clusterizado realizado sobre los datos correlacionados

Cuadro 7.2: Requisitos Funcionales 2

### 7.1.3. Requisitos no funcionales

Los requisitos no funcionales describen propiedades emergentes del sistema, tales como tiempos de respuesta, necesidades de almacenamiento, fiabilidad... Pueden llegar a ser más críticos que los requisitos funcionales. El incumplimiento de un requisito funcional supone el degradado del proyecto software, mientras que el incumplimiento de un requisito no funcional puede suponer la inutilización de la aplicación.

A continuación se detallan los requisitos no funcionales de la aplicación en la **tabla 7.3**, que han sido elaborados siguiendo los siguientes criterios:

- Usabilidad

- Fiabilidad

- Rendimiento

- Soporte

ID	Requisito	Descripción
RNF01	Facilidad de uso	La aplicación deberá ser fácil y sencilla de utilizar
RNF02	Facilidad de instalación	La aplicación debe ser fácil de instalarse
RNF03	Actualización automática	La aplicación debe recibir de forma automática las alertas de las herramientas.
RNF04	Tiempos de respuesta	La aplicación debe garantizar tiempos de respuesta no superiores a 5s una vez que los datos están cargados.
RNF05	Alta disponibilidad	El sistema debe estar altamente disponible, una disponibilidad de 99,99 %
RNF06	Datos fiables	Los datos mostrados en la aplicación deben provenir de fuentes fiables
RNF07	Facilidad de aprendizaje	La aplicación debe ser fácil de aprender su funcionamiento

Cuadro 7.3: Requisitos No Funcionales

#### 7.1.4. Requisitos de información

Los requisitos de información hacen referencia a los datos que el sistema debe almacenar y el contenido de cada uno de ellos. A continuación en la **tabla 7.4** se describen estos requisitos.

ID	Requisito	Descripción
RI01	Shodan	Datos recibidos tras la búsqueda en la API de Shodan
RI02	CVE Search	Datos recibidos tras la búsqueda en la API de CVE Search
RI03	Suricata	Alertas recibidas de la herramienta Suricata
RI04	Snort	Alertas recibidas de la herramienta Snort
RI05	Ossec	Alertas recibidas de la herramienta Ossec
RI06	Wazuh	Alertas recibidas de la herramienta Wazuh
RI07	AlienVault	Alertas recibidas de la herramienta AlienVault OS-SIM
RI08	Correlación	Datos unificados de todas las herramientas
RI09	Clusterizado	Cluster de los datos correlacionados

Cuadro 7.4: Requisitos de Información

## 7.2. Diagrama de actores

Nuestra aplicación consta de un único actor principal que se expone a continuación:

- Responsable de seguridad de la organización: **tabla 7.5**

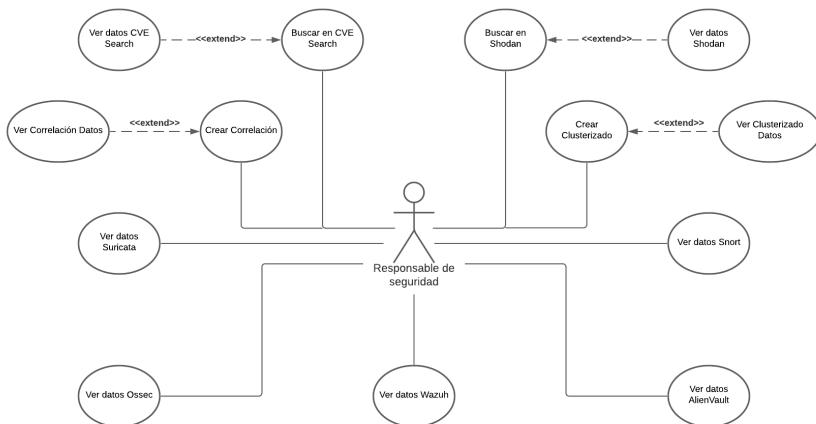
Actor	Responsable de seguridad de la organización
Descripción	Representa a un usuario de una organización responsable de la seguridad de esta misma.
Características	Usuario con conocimientos suficientes para interpretar la información mostrada por la herramienta.

Cuadro 7.5: Actor: Responsable de seguridad de la organización

### 7.3. Diagrama de casos de uso

En este apartado se detallará el diagrama de casos de uso para el actor mencionado en el anterior apartado y se realizará una breve explicación sobre el diagrama.

Figura 7.1: Casos de uso: Responsable de seguridad



La figura 7.1 muestra las distintas posibilidades que se le otorgan al usuario responsable de seguridad de una organización. El usuario podrá acceder a todo el contenido de la herramienta y visualizar todo lo que está contiene. Además, podrá buscar información adicional proporcionada por las fuentes externas a las que se encuentra conectada la herramienta.

## 7.4. Casos de Uso

### 7.4.1. Ver Datos Shodan

CU-01	Ver Datos Shodan
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar los activos expuestos al exterior de una organización
<b>Resumen</b>	El usuario responsable de seguridad de la organización introducirá el nombre de esta misma para realizar la búsqueda en Shodan y visualizará los datos que esta proporciona

Cuadro 7.6: CU-01:Ver Datos Shodan

CU-01	CURSO NORMAL
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario introduce el nombre de la organización	5. El sistema envía los datos a shodan, para realizar la búsqueda de la organización
4. El usuario selecciona la opción de buscar	6. El sistema muestra los datos recibidos por Shodan
7. El usuario visualiza la información en un dashboard	

Cuadro 7.7: CU-01:Curso Normal

#### 7.4.2. Ver Datos CVEs Organización

<b>CU-02</b>	<b>Ver Datos CVEs Organización</b>
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar los CVEs correspondientes a los activos expuestos al exterior de una organización
<b>Resumen</b>	El usuario responsable de seguridad de la organización introducirá el nombre de esta misma para realizar la búsqueda en Shodan y correlacionar los datos recibidos con la lista de CVEs y visualizará los datos

Cuadro 7.8: CU-02:Ver Datos CVEs Organización

<b>CU-02</b>	<b>CURSO NORMAL</b>
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario introduce el nombre de la organización	5. El sistema envía los datos a shodan, para realizar la búsqueda de la organización y correlaciona los datos recibidos con la lista de CVEs pública
4. El usuario selecciona la opción de buscar	6. El sistema muestra los datos de la correlación
7. El usuario visualiza la información en un dashboard	

Cuadro 7.9: CU-02:Curso Normal

#### 7.4.3. Ver Datos Suricata

CU-03	Ver Datos Suricata
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar las alertas mostradas por la herramienta Suricata en tiempo real
<b>Resumen</b>	El usuario responsable de seguridad de la organización podrá visualizar las alertas generadas por la herramienta Suricata disponible en una organización en tiempo real

Cuadro 7.10: CU-03:Ver Datos Suricata

CU-03	CURSO NORMAL
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario visualiza la información en un dashboard	

Cuadro 7.11: CU-03:Curso Normal

#### 7.4.4. Ver Datos Snort

CU-04	Ver Datos Snort
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar las alertas mostradas por la herramienta Snort en tiempo real
<b>Resumen</b>	El usuario responsable de seguridad de la organización podrá visualizar las alertas generadas por la herramienta Snort disponible en una organización en tiempo real

Cuadro 7.12: CU-04:Ver Datos Snort

<b>CU-04</b>	<b>CURSO NORMAL</b>
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario visualiza la información en un dashboard	

Cuadro 7.13: CU-04:Curso Normal

#### 7.4.5. Ver Datos Ossec

<b>CU-05</b>	<b>Ver Datos Ossec</b>
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar las alertas mostradas por la herramienta Ossec en tiempo real
<b>Resumen</b>	El usuario responsable de seguridad de la organización podrá visualizar las alertas generadas por la herramienta Ossec disponible en una organización en tiempo real

Cuadro 7.14: CU-05:Ver Datos Ossec

<b>CU-05</b>	<b>CURSO NORMAL</b>
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario visualiza la información en un dashboard	

Cuadro 7.15: CU-05:Curso Normal

#### 7.4.6. Ver Datos Wazuh

CU-06	Ver Datos Wazuh
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar las alertas mostradas por la herramienta Wazuh en tiempo real
<b>Resumen</b>	El usuario responsable de seguridad de la organización podrá visualizar las alertas generadas por la herramienta Wazuh disponible en una organización en tiempo real

Cuadro 7.16: CU-06:Ver Datos Wazuh

CU-06	CURSO NORMAL
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario visualiza la información en un dashboard	

Cuadro 7.17: CU-06:Curso Normal

#### 7.4.7. Ver Datos AlienVault

CU-07	Ver Datos AlienVault
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar las alertas mostradas por la herramienta AlienVault en tiempo real
<b>Resumen</b>	El usuario responsable de seguridad de la organización podrá visualizar las alertas generadas por la herramienta AlienVault disponible en una organización en tiempo real

Cuadro 7.18: CU-07:Ver Datos AlienVault

<b>CU-07</b>	<b>CURSO NORMAL</b>
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario visualiza la información en un dashboard	

Cuadro 7.19: CU-07:Curso Normal

#### 7.4.8. Ver Correlación Datos

<b>CU-08</b>	<b>Ver Correlación Datos</b>
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar los datos de las diferentes herramientas correlacionados
<b>Resumen</b>	El usuario responsable de seguridad de la organización visualizará los datos correlacionados de las diferentes herramientas trás ejecutar el programa que realiza esta tarea

Cuadro 7.20: CU-08:Ver Correlación Datos

<b>CU-08</b>	<b>CURSO NORMAL</b>
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario pulsa el botón de correlación	4. El sistema ejecuta el programa de correlación de las diferentes fuentes
6. El usuario visualiza la información en un dashboard	5. El sistema muestra los datos recibidos por el programa de correlación

Cuadro 7.21: CU-08:Curso Normal

#### 7.4.9. Ver Clusterizado Datos

CU-09	Ver Clusterizado Datos
<b>Actor</b>	Usuario responsable de seguridad de la organización
<b>Tipo</b>	Primario, Esencial
<b>Precondición</b>	Usuario pertenece al departamento de seguridad de la organización
<b>Postcondición</b>	
<b>Propósito</b>	Visualizar el clusterizado de los datos correlacionados de las diferentes herramientas
<b>Resumen</b>	El usuario responsable de seguridad de la organización visualizará la clusterización de los datos correlacionados de las diferentes fuentes

Cuadro 7.22: CU-09:Ver Clusterizado Datos

CU-09	CURSO NORMAL
Acción del actor	Acción del sistema
1. El usuario inicia la aplicación	2. El sistema muestra la aplicación
3. El usuario pulsa el botón de clusterizado	4. El sistema ejecuta el programa de clusterizado de los datos correlacionados
6. El usuario visualiza la información en un dashboard	5. El sistema muestra los datos recibidos por el programa de clusterizado

Cuadro 7.23: CU-09:Curso Normal

# Capítulo 8

## Diseño e Implementación

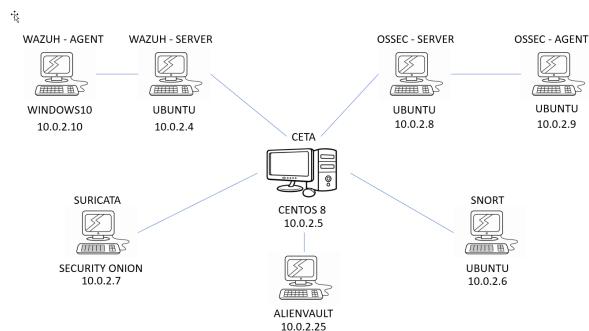
Aclaradas las herramientas a utilizar y los requisitos de desarrollo, se va a explicar ahora el diseño utilizado para la realización de la aplicación. Se expondrá en un primer momento la configuración de los entornos de desarrollo utilizados, así como el esquema de red en el que se alojan las diferentes máquinas virtuales. Por otro lado, se expondrá la configuración que se ha realizado para añadir a la plataforma central como es CETA, el resto de herramientas de monitorización. Por último, se expondrá la solución adoptada para la correlación de los datos y el algoritmo de Machine Learning empleado con estos datos.

### 8.1. Entorno de desarrollo

El entorno de desarrollo está compuesto por diferentes máquinas virtuales, en las cuáles se tiene una máquina central a la que se conectan el resto, y envían la información proporcionada por las herramientas de monitorización que alojan.

A continuación, se muestra el esquema de red de las máquinas virtuales del proyecto. Junto con cada máquina virtual aparece una etiqueta con la herramienta de monitorización que aloja, así como la IP que tiene asignada dentro de la red.

Figura 8.1: Esquema de red



Pese a que más adelante se irá exponiendo en cada caso más concreto la conexión realizada entre todas las máquinas, se va a realizar una breve exposición ahora de esto mismo. En el Capítulo de Tecnología se expuso la herramienta Syslog-*ng*, una implementación gratuita de código abierto del protocolo syslog. Esta implementación es la que se ha utilizado para realizar las conexiones entre las diferentes máquinas virtuales con la máquina virtual central. Con esta herramienta es posible el envío de logs de unas máquinas a otras a través del protocolo syslog.

## 8.2. CETA

CETA es la herramienta principal del proyecto, sobre la que se encuentran conectadas por un lado las herramientas de monitorización externa de una organización, como son las sondas shodan y CVE Search, así como las herramientas de monitorización interna de una organización, como son Suricata, Snort, Wazuh, Ossec y AlienVault. Además, en esta herramienta se ha incorporada la funcionalidad de correlación de los datos proporcionados por las diferentes herramientas, así como la implementación de técnicas de Machine Learning para la predicción y clasificación de la información recibida.

Figura 8.2: CETA: Cyber Early Threat Alarm



En lo que respecta a CETA, se han realizado diferentes implementaciones, desde la puesta a punto de la herramienta debido a la utilización de librerías obsoletas, hasta la adición de nuevas funcionalidades con la incorporación de las herramientas de monitorización de otras máquinas y la adición de funcionalidad como la correlación de datos e implementación de algoritmos de Machine Learning.

En esta sección del capítulo se irá explicando cada uno de los pasos realizados en el proyecto, corresponde con el orden seguido en los Sprints. Se explicará tanto la configuración seguida en CETA, como en las máquinas que alojan las herramientas de monitorización, la descarga de estas herramientas, su configuración y la conexión con la máquina central.

### 8.2.1. Puesta a punto

La primera de las tareas a realizar sobre la herramienta fue la puesta a punto de esta misma. CETA fue implementada por el Scrum Master (Juan Carlos Cortinas) años atrás, por lo que muchas librerías e implementaciones realizadas sobre la herramienta se encontraban obsoletas y necesitaban ser modificadas para poder volver a poner en funcionamiento la

herramienta.

## Dependencias

CETA fue desarrollada en una máquina virtual que no correspondía con la actual (CENTOS 8), por lo que la mayoría de las librerías que requería para su correcto funcionamiento no se encontraban descargadas en la máquina virtual.

Para no tener que descargar a mano todas estas dependencias cada vez que CETA fuese desplazada a otra máquina virtual, se creó un script que contenía todo lo necesario para la descarga de las dependencias. De este modo, cada vez que se iniciase la herramienta, comprobaría que todas las librerías se encuentran correctamente descargadas en sus últimas versiones para poder iniciar sin errores la plataforma.

Para ello, se creó el script denominado initRequirements con toda esta funcionalidad de descarga y actualización de librerías.

## Servicios

Inicialmente CETA antes de la adición de la nueva funcionalidad de este proyecto, contaba con 2 servicios principales para la monitorización externa de organizaciones. Utilizaba la sonda externa mundialmente conocida como es Shodan y el listado de CVEs públicos proporcionados por una API conocida como CVE Search. Todos los datos recibidos por estas 2 herramientas eran almacenados en una base de datos como es MongoDB y posteriormente utilizados por ELK Stack (ElasticSearch y Kibana), para la visualización de los datos en la propia plataforma.

El problema con todos estos servicios es que se encontraban funcionando en versiones antiguas, algunas de ellas también obsoletas o que habían sido modificadas, por lo que el correcto funcionamiento de CETA era imposible a no ser que se realizasen una serie de modificaciones tanto en las versiones descargadas como en las conexiones entre ellas para su correcto funcionamiento.

A continuación, se exponen las tareas realizadas para cada uno de los servicios que se acaban de mencionar:

- MongoDB: se actualizó la versión descargada de MongoDB a una de las últimas versiones más estables como es la 4.4.4.
- Shodan: para este servicio no fue necesario la realización de ninguna tarea.
- CVE Search: la API utilizada era una local que proporcionaba la descarga de dicho servicio. El problema es que las llamadas a la API habían cambiado con el paso de las versiones, por lo que fue necesario cambiar las llamadas a la API.
- ELK Stack: el problema con este servicio es muy similar al de MongoDB, la versión que se estaba utilizando se encontraba completamente obsoleta, por lo que fue necesario descargar una versión estable nueva como es la 7.12.0.

Con todas estas modificaciones a realizar, se creó otro script de arranque de estos servicios. Dicho script comprobaba que los servicios se encontrasen actualizados a su última versión e iba arrancando uno a uno en el orden correcto. Es decir, para poder utilizar los servicios tanto de Shodan como de CVE Search, es necesario tener previamente lanzado la base de datos MongoDB para poder almacenar los datos. Lo mismo ocurre con Elasticsearch, necesita que se encuentre MongoDB ejecutándose para poder conectarse a este y obtener los datos y posteriormente enlazarlos con Kibana para su visualización.

El script se denominó StartServices y el resumen de su funcionalidad es la siguiente: Actualizar los servicios a su última versión estable; Arrancar los servicios en un orden lógico; Mantener los servicios funcionando para el posterior arranque de CETA.

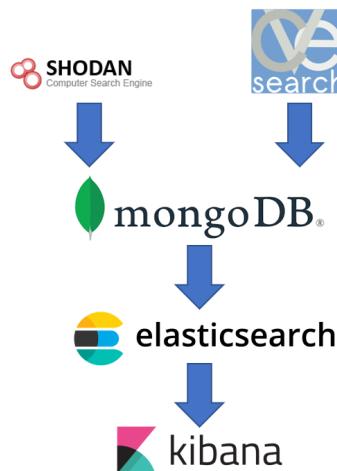
## Visualización

Las modificaciones sufridas en las versiones de los 2 servicios principales que integraba CETA como son Shodan y CVE Search, supuso que los datos recibidos se desviasen un poco de lo que recibía anteriormente. Es por esto, que internamente los datos que se alojan en MongoDB y eran pasados a Elasticsearch se vieron modificados, y como consecuencia los datos de Elasticsearch a Kibana ocurrió lo mismo.

Es por esto, que fue necesario tanto la reestructuración interna como externa (visualización) de los datos de Shodan y CVE Search. Básicamente lo que se realizó fue observar las llamadas a los servicios y los datos que estos recogían. De este modo fue sencillo cambiar unos por otros, ya que las modificaciones de datos entre las diferentes versiones era mínima.

A continuación, se muestra el esquema de flujo de datos desde los servicios que integra CETA hasta la visualización de estos mismos.

Figura 8.3: CETA: Flujo de datos



### **8.2.2. Suricata**

Suricata es la primera herramienta de monitorización que se incorporó a CETA. En este apartado se va a explicar todos los detalles de la configuración de Suricata, así como de la máquina virtual que aloja la herramienta. También es necesario exponer la conexión entre Suricata y CETA para la transmisión de los datos en tiempo real, y la visualización de los datos recibidos en CETA.

Es por esto, que se va a dividir en diferentes apartados esta sección. Se comenzará hablando de la configuración existente de la máquina virtual que aloja la herramienta. A continuación, se hablará de la propia herramienta de monitorización de su descarga y configuración. Lo siguiente será hablar de la conexión entre Suricata y CETA, para ello es necesario explicar la conexión entre las máquinas virtuales en las que se encuentra cada una de las herramientas. Por último, el proceso de almacenamiento y visualización de datos con las herramientas ELK Stack es necesario que sean explicadas.

Con todo esto, tendríamos una visión completa de la herramienta de monitorización NIDS como es Suricata.

#### **Security Onion**

Como ya se ha comentado anteriormente, Suricata se encuentra alojada en una máquina virtual con el sistema operativo Security Onion.

Para la configuración de esta máquina virtual se ha determinado que las siguientes opciones que se van a comentar a continuación son las más óptimas:

- Evaluation mode. Este modelo es recomendado para usuarios inexpertos con este tipo de sistema operativo y para máquinas virtuales. Configura rápida y automáticamente todos los detalles del sistema.
- Custom mode. Permite ver todas las opciones y que no sean las Best Practices las que sean usadas por defecto.
- Emerging Threats Open. Esta opción determina que las reglas IDS que van a ser utilizadas son gratis y no requieren de licencias de pago.
- IDS engine: Suricata. Security Onion permite descargar Suricata o Snort. Esta máquina aloja la herramienta Suricata por lo que se selecciona esta opción.
- Enable network sensor. Para esta máquina no va a influir tanto en el rendimiento ya que es una prueba, por lo que no interesa que el rendimiento mejore.

Estas opciones marcadas son las más importantes, el resto que aparecen a la hora de configurar la máquina son a gusto de cada uno, ya que hace referencia a número de días que se desean almacenar logs o tamaños etc.

#### **Suricata**

En lo que respecta a la herramienta Suricata, debido a la selecciones a la hora de configurar Security Onion esta viene incluida en el sistema operativo, por lo que no será necesario

una descarga adicional.

Adicionalmente, en lo que respecta a la configuración de reglas de Suricata, se permite la opción de utilizar Pulled Pork para determinar la versión de Suricata y descargar las últimas reglas.

## Conexión: Sylog-ng

Para realizar la conexión entre las máquinas ya se ha comentado que la herramienta a utilizar es Syslog-ng. En primer lugar es necesario que Syslog-ng se encuentre descargado en ambas máquinas virtuales, tanto la que aloja Suricata como CETA. A continuación, se explican los detalles de la configuración de esta herramienta.

Desde la máquina virtual Security Onion es necesario especificar el directorio que contiene los logs de Suricata y especificar el programa que genera dichos logs. Por otro lado, es necesario exponer la dirección IP destino de la máquina CENTOS dónde se encuentra CETA, y el puerto y protocolo a través del cuál serán enviados los logs. Configuración Syslog-ng en Security Onion:

```
1 source s_suricata{
2     file("/var/log/suricata/eve.json"
3         program	override("suricata")
4         flags(no-parse));
5 };
6
7 destination d_suricata{
8     syslog("10.0.2.5" transport("tcp") port(5514));
9 }
10
11 log{
12     source(s_suricata);
13     destination(d_suricata);
14 }
```

En el lado de la máquina CENTOS, es necesario determinar el puerto a través del cuál se reciben los logs y el protocolo. Además, es importante especificar el destino de los logs, en este caso se ha creado una carpeta nueva con el nombre de suricata bajo el directorio /var/log, en el cuál es necesario dar los permisos de administrador para poder escribir. Configuración Syslog-ng en CENTOS:

```
1 source s_suricata{
2     syslog(transport("tcp") port(5514));
3 };
4
5 destination d_suricata{
6     file("/var/log/suricata/logs.txt"
7         owner("root")
8         group("root")
9         perm(0777));
10 }
11
12 log{
13     source(s_suricata);
14     destination(d_suricata);
15 }
```

Tambiés es necesario abrir el puerto correspondiente en CENTOS para permitir la conexión. En este caso el puerto que es necesario abrir es el 5514. Para permitir que el firewall de CENTOS acepte paquetes, es necesario especificar lo siguiente:

```
1 # firewall-cmd --zone=public --add-port=5514/tcp --permanent  
2  
3 # firewall-cmd --reload
```

## ELK Stack

Una vez se tienen los logs almacenados en la máquina virtual CENTOS dónde se encuentra CETA, es necesario importar estos datos a la propia máquina. Para ello, se utilizarán las herramientas ELK Stack.

En primer lugar Logstash toma como entrada el fichero txt almacenado en el directorio /var/log/suricata y lo filtra, de este modo cada fila del fichero corresponde a una entrada que es mandada a ElasticSearch, y en la que cada dato del fichero corresponde a una columna diferente. Por ello, es importante especificar que el tipo de fichero leído es syslog y separar de este modo la cabecera de cada entrada de todo fichero syslog y el contenido de este mismo.

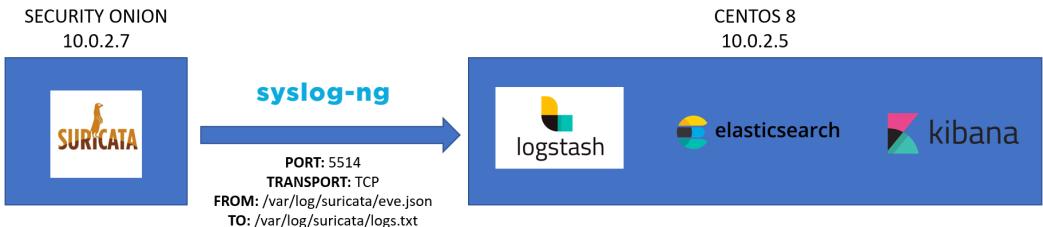
```
1 input{  
2     file{  
3         type => "syslog"  
4         path => "/var/log/suricata/logs.txt"  
5         start_position => beginning  
6     }  
7 }  
8  
9 filter{  
10    if [type] == "syslog"{  
11        grok {  
12            match => {"message" => "%{SYSLOGBASE} %{GREEDYDATA:syslog_message}"  
13        }  
14        json {  
15            source => "syslog_message"  
16        }  
17    }  
18 }  
19  
20 output{  
21    elasticsearch {hosts => ["localhost:9200"] index => "suricata"}  
22    stdout {codec => rubydebug}  
23 }
```

Por último, los datos filtrados son enviados a Elasticsearch, todos ellos bajo el índice denominado suricata. Elasticsearch actuará como buscador, para desde Kibana poder crear otro índice con el mismo nombre (suricata) y poder así obtener los datos para realizar la visualización de los mismos.

## Flujo de datos

A continuación se muestra el flujo de datos desde Suricata hasta su visualización en CETA.

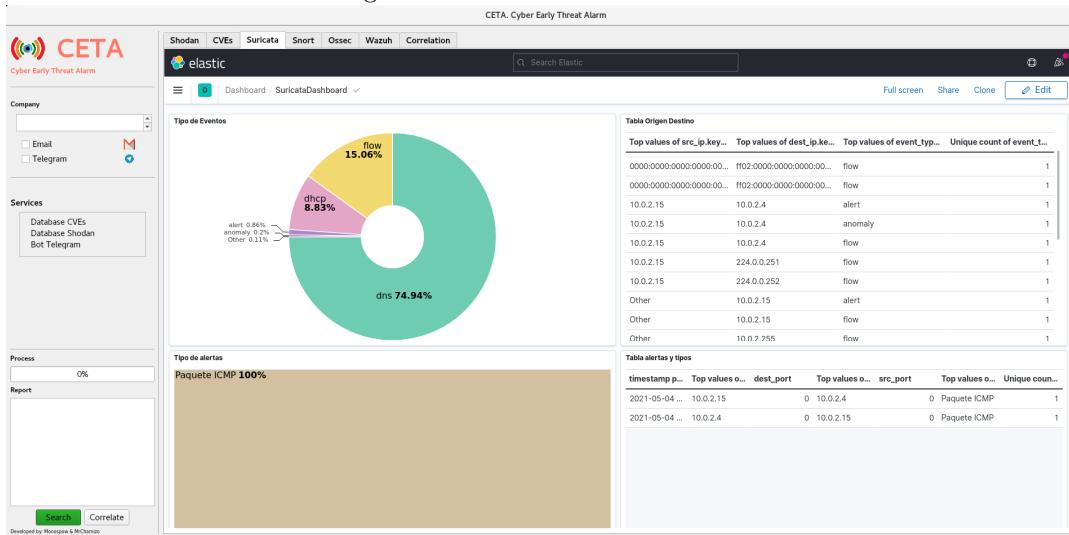
Figura 8.4: Suricata: Flujo de datos



## Visualización

La visualización de forma dinámica de los datos de Suricata en CETA consiste en la integración de la visualización generada en Kibana y reflejada en CETA a través de su enlace HTTP. Con todo esto, se obtiene una visualización como la siguiente.

Figura 8.5: Suricata: Visualización



### 8.2.3. Snort

Snort fue la segunda herramienta que se anexó a CETA. Para completar los NIDS junto a Suricata. De este modo cualquier organización podría conectar a la plataforma cualquiera de las 2 herramientas. En este apartado se van a explicar todos los detalles de la configuración de Snort, así como la de la máquina virtual que aloja la herramienta. Es necesario exponer la conexión entre Snort y CETA para la transmisión de las alertas en tiempo real, y la visualización correspondiente de los datos en CETA.

Por esto, se va a dividir en diferentes apartados esta sección. Se comenzará hablando de la configuración de la máquina virtual Ubuntu donde se encuentra la herramienta de mo-

nitorización Snort. A continuación, se hablará de la propia herramienta Snort, tanto de su descarga como de su configuración. Lo siguiente será hablar de la conexión entre Snort y CETA para la transmisión de datos en tiempo real de una máquina a otra. Por último, es necesario explicar el proceso de almacenado, indexación y visualización de las alertas recibidas en la máquina destino.

Con todo esto, tendríamos una visión completa de la herramienta de monitorización NIDS como es Snort.

## Ubuntu

Snort se encuentra alojado en una máquina virtual con el sistema operativo Ubuntu. La configuración de este sistema operativo no tiene ninguna peculiaridad, ya que se usa con los parámetros por defecto que esta tiene.

Lo único destacable, es que la máquina es importada de la página web de Osboxes, la cuál tiene máquinas preconfiguradas de todo tipo.

## Snort

La instalación de Snort se ha realizado siguiendo los pasos indicados en su propia página web. La versión instalada es la última disponible, que en este caso es Snort 3. Durante la instalación no ocurrió ningún problema. Al igual que ocurre con Suricata, se permite la opción de que Pulled Pork descargue las últimas reglas y mantenga la versión de Snort actualizada.

Adiconalmente en lo que respecta a las reglas, se han añadido 2 para la futura realización de pruebas. A continuación, se muestran las 2 reglas añadidas al fichero local.rules ubicado en el directorio /etc/snort/rules.

```
1 # -----
2 # LOCAL RULES
3 # -----
4
5 alert icmp any any -> $HOME_NET any (msg:"Paquete ICMP"; sid:100000001;)
6 alert tcp $HOME_NET any -> any 80 (msg:"Trafico http"; sid:100000003;)
```

Con estas 2 reglas cualquier ping dentro de la red sobre la que se encuentre Snort quedará reflejado como alerta, al igual que cualquier consulta saliente sobre el puerto 80 se considerará como tráfico http.

Por último, hay que destacar que el formato de salida de Snort no es de tipo json, al igual que ocurre con Suricata y otras herramientas como Ossec o Wazuh. Snort tiene un formato de salida de las alertas denominado Unified2, el cuál no es fácilmente legible. Es por esto, que para poder obtener una salida en el formato json, se ha tenido que utilizar una herramienta de idstools denominada u2json. U2json lee logs en formato unified2 y los guarda en una salida con formato JSON. En la propia página aparecen todos los comandos y opciones necesarias para lanzar el comando.

## Conexión: Sylog-ng

Para realizar la conexión entre las máquinas ya se ha comentado que la herramienta a utilizar es Syslog-ng. En primer lugar es necesario que Syslog-ng se encuentre descargado en ambas máquinas virtuales, tanto la que aloja Snort como CETA. A continuación, se explican los detalles de la configuración de esta herramienta.

Desde la máquina virtual Ubuntu es necesario especificar el directorio que contiene los logs de Snort y especificar el programa que genera dichos logs. Por otro lado, es necesario exponer la dirección IP destino de la máquina CENTOS dónde se encuentra CETA, y el puerto y protocolo a través del cuál serán enviados los logs. Configuración Syslog-ng en Ubuntu:

```
1 source s_snort{
2     file("/var/log/snort/alerts.json"
3         program	override("snort")
4         flags(no-parse));
5 };
6
7 destination d_snort{
8     syslog("10.0.2.5" transport("tcp") port(5515));
9 }
10
11 log{
12     source(s_snort);
13     destination(d_snort);
14 }
```

En el lado de la máquina CENTOS, es necesario determinar el puerto a través del cuál se reciben los logs y el protocolo. Además, es importante especificar el destino de los logs, en este caso se ha creado una carpeta nueva con el nombre de snort bajo el directorio /var/log, en el cuál es necesario dar los permisos de administrador para poder escribir. Configuración Syslog-ng en CENTOS:

```
1 source s_snort{
2     syslog(transport("tcp") port(5515));
3 };
4
5 destination d_snort{
6     file("/var/log/snort/logs.txt"
7         owner("root")
8         group("root")
9         perm(0777));
10 }
11
12 log{
13     source(s_snort);
14     destination(d_snort);
15 }
```

También es necesario abrir el puerto correspondiente en CENTOS para permitir la conexión. En este caso el puerto que es necesario abrir es el 5515. Para permitir que el firewall de CENTOS acepte paquetes, es necesario especificar lo siguiente:

```
1 # firewall-cmd --zone=public --add-port=5515/tcp --permanent
2
3 # firewall-cmd --reload
```

## ELK Stack

Una vez se tienen los logs almacenados en la máquina virtual CENTOS dónde se encuentra CETA, es necesario importar estos datos a la propia máquina. Para ello, se utilizarán las herramientas ELK Stack.

En primer lugar Logstash toma como entrada el fichero txt almacenado en el directorio /var/log/snort y lo filtra, de este modo cada fila del fichero corresponde a una entrada que es mandada a ElasticSearch, y en la que cada dato del fichero corresponde a una columna diferente. Por ello, es importante especificar que el tipo de fichero leído es syslog y separar de este modo la cabecera de cada entrada de todo fichero syslog y el contenido de este mismo.

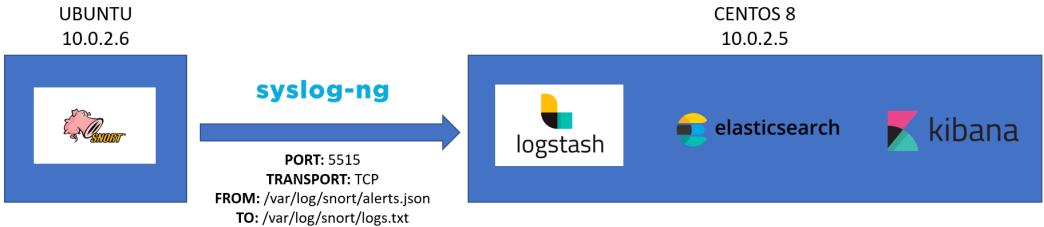
```
1 input{
2     file{
3         type => "syslog"
4         path => "/var/log/snort/logs.txt"
5         start_position => beginning
6     }
7 }
8
9 filter{
10     if [type] == "syslog"{
11         grok {
12             match => {"message" => "%{SYSLOGBASE} %{GREEDYDATA:syslog_message}"}
13         }
14         json {
15             source => "syslog_message"
16         }
17     }
18 }
19
20 output{
21     elasticsearch {hosts => ["localhost:9200"] index => "snort"}
22     stdout {codec => rubydebug}
23 }
```

Por último, los datos filtrados son enviados a Elasticsearch, todos ellos bajo el índice denominado snort. Elasticsearch actuará como buscador, para desde Kibana poder crear otro índice con el mismo nombre (snort) y poder así obtener los datos para realizar la visualización de los mismos.

## Flujo de datos

A continuación, se muestra el flujo de datos desde Snort hasta su visualización en CETA.

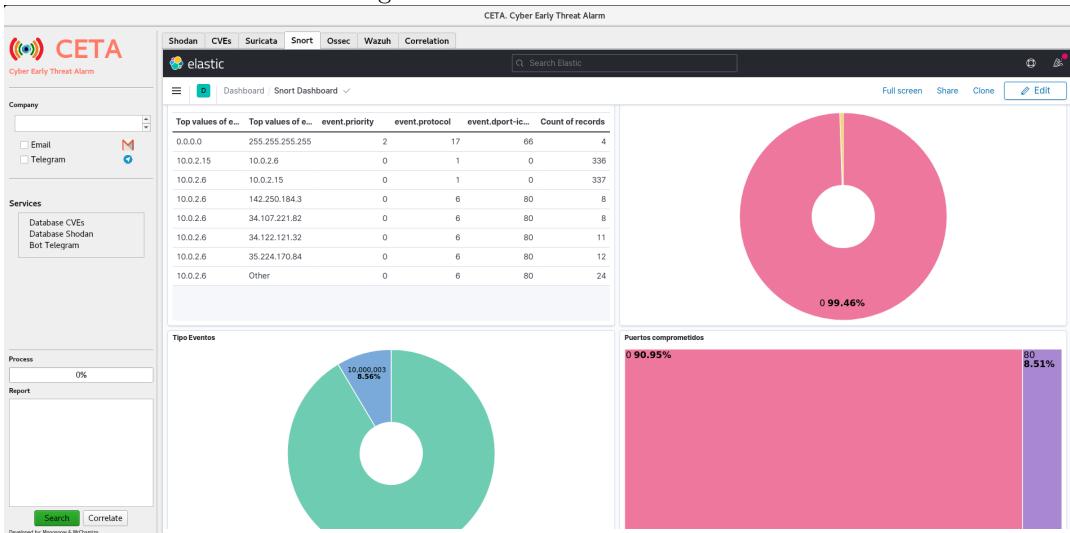
Figura 8.6: Snort: Flujo de datos



## Visualización

La visualización de forma dinámica de los datos de Snort en CETA consiste en la integración de la visualización generada en Kibana y reflejada en CETA a través de su enlace HTTP. Con todo esto, se obtiene una visualización como la siguiente.

Figura 8.7: Snort: Visualización



### 8.2.4. Ossec

Ossec fue la tercera herramienta que se anexó a CETA. Ossec se decidió introducir para tener así un HIDS conectado a la plataforma. En este apartado se van a explicar todos los detalles de la configuración de Ossec tanto del agente como del servidor y su conexión, así como la de las máquinas virtuales que alojan las herramientas. Es necesario exponer la conexión entre Ossec server y CETA para la transmisión de las alertas en tiempo real, y la visualización correspondiente de los datos en CETA.

Por esto, se va a dividir en diferentes apartados esta sección. Se comenzará hablando de la configuración de la máquina virtual Ubuntu donde se encuentra la herramienta de moni-

torización Ossec, ambas máquinas utilizadas para el agente como el servidor son similares. A continuación, se hablará de la propia herramienta Ossec, tanto de su descarga como de su configuración, además de la conexión entre agente y servidor. Lo siguiente será hablar de la conexión entre Ossec server y CETA para la transmisión de datos en tiempo real de una máquina a otra. Por último, es necesario explicar el proceso de almacenado, indexación y visualización de las alertas recibidas en la máquina destino.

Con todo esto, tendríamos una visión completa de la herramienta de monitorización HIDS como es Ossec.

## Ubuntu

Ossec se encuentra alojado en una máquina virtual con el sistema operativo Ubuntu. La configuración de este sistema operativo no tiene ninguna peculiaridad, ya que se usa con los parámetros por defecto que esta tiene.

Lo único destacable, es que la máquina es importada de la página web de Osboxes, la cuál tiene máquinas preconfiguradas de todo tipo. Y que es necesario la creación de 2 máquinas virtuales para poder instalar en una de ellas el agente Ossec y conectarlo a otra dónde se encuentra el servidor Ossec.

## Ossec

Para la descarga de OSSEC se han seguido los pasos descritos en la web de referencia de descarga de Ossec.

Los pasos seguidos han sido los siguientes:

- Instalar las dependencias requeridas posteriores.
- Descargar la última versión de OSSEC.
- Descomprimir.
- Instalar el servidor: en la opción de descarga de que tipo de instalación desea hay que seleccionar la de servidor.
- Instalar el agente: en la opción de descarga de que tipo de instalación desea hay que seleccionar la de agente. Al seleccionar esta opción es necesario introducir la dirección IP del servidor.
- Conectar agente y servidor. Para conectar agente y servidor es necesario realizarlo a través de la herramienta ubicada en la dirección /var/ossec/bin/manage\_agents. Para el enlace es necesario introducir en el servidor la dirección IP del agente, el ID que se va a asignar al agente y el nombre que se le quiere dar. Desde el lado del agente, es necesario introducir una Key generada por el servidor una vez se ha registrado el agente. Con todo esto, se reinicia Ossec y se puede comprobar mediante el comando list\_agents que ambas máquinas se encuentran conectadas.

Además, al igual que ocurre con Snort se han añadido unas reglas adicionales en el fichero local\_rules.xml.

```

1 <group name="local ,syslog ,authentication_success ,">
2   <rule id="10000002" level="13">
3     <if_group>authentication_success</if_group>
4     <time> 10 pm - 9 pm </time>
5     <description>Login fuera de horas</description>
6   </rule>
7 </group>
8
9 <group name="syslog ,adduser ">
10   <rule id="10010" level="8">
11     <pcre2>^new group</pcre2>
12     <description>Grupo anadido al sistema</description>
13   </rule>
14
15   <rule id="10011" level="8">
16     <pcre2>^new user|^new account added</pcre2>
17     <description>Usuario anadido al sistema</description>
18   </rule>
19
20   <rule id="10012" level="2">
21     <pcre2>^delete user|^account deleted|^remove group</pcre2>
22     <description>Grupo o usuario eliminado del sistema</description>
23   </rule>
24
25   <rule id="10013" level="8">
26     <pcre2>^change user</pcre2>
27     <description>Informacion de usuario cambiada</description>
28   </rule>
29
30   <rule id="10014" level="10">
31     <program_name_pcre2>useradd</program_name_pcre2>
32     <pcre2>failed adding user</pcre2>
33     <description>Fallo al anadir usuario</description>
34   </rule>
35 </group>

```

Todas estas reglas tienen que ver con la creación, modificación, eliminación de grupos y usuarios. Además se ha creado una para registrar el login de los usuarios y comprobar si se realiza fuera de la hora establecida.

## Conexión: Sylog-*ng*

Para realizar la conexión entre las máquinas ya se ha comentado que la herramienta a utilizar es Syslog-*ng*. En primer lugar es necesario que Syslog-*ng* se encuentre descargado en ambas máquinas virtuales, tanto la que aloja el servidor Ossec como CETA. A continuación, se explican los detalles de la configuración de esta herramienta.

Desde la máquina virtual Ubuntu es necesario especificar el directorio que contiene los logs de Ossec y especificar el programa que genera dichos logs. Por otro lado, es necesario exponer la dirección IP destino de la máquina CENTOS dónde se encuentra CETA, y el puerto y protocolo a través del cuál serán enviados los logs. Configuración Syslog-*ng* en Ubuntu:

```

1 source s_ossec{
2   file("/var/ossec/logs/alerts/alerts.json"
3   program-override("ossec")
4   flags(no-parse));
5 };

```

```

6
7 destination d_ossec{
8     syslog("10.0.2.5" transport("tcp") port(5516));
9 }
10
11 log{
12     source(s_ossec);
13     destination(d_ossec);
14 }
```

En el lado de la máquina CENTOS, es necesario determinar el puerto a través del cuál se reciben los logs y el protocolo. Además, es importante especificar el destino de los logs, en este caso se ha creado una carpeta nueva con el nombre de ossec bajo el directorio /var/log, en el cuál es necesario dar los permisos de administrador para poder escribir. Configuración Syslog-ng en CENTOS:

```

1 source s_ossec{
2     syslog(transport("tcp") port(5516));
3 };
4
5 destination d_ossec{
6     file("/var/log/ossec/logs.txt"
7         owner("root")
8         group("root")
9         perm(0777));
10 }
11
12 log{
13     source(s_ossec);
14     destination(d_ossec);
15 }
```

También es necesario abrir el puerto correspondiente en CENTOS para permitir la conexión. En este caso el puerto que es necesario abrir es el 5516. Para permitir que el firewall de CENTOS acepte paquetes, es necesario especificar lo siguiente:

```

1 # firewall-cmd --zone=public --add-port=5516/tcp --permanent
2
3 # firewall-cmd --reload
```

## ELK Stack

Una vez se tienen los logs almacenados en la máquina virtual CENTOS dónde se encuentra CETA, es necesario importar estos datos a la propia máquina. Para ello, se utilizarán las herramientas ELK Stack.

En primer lugar Logstash toma como entrada el fichero txt almacenado en el directorio /var/log/ossec y lo filtra, de este modo cada fila del fichero corresponde a una entrada que es mandada a ElasticSearch, y en la que cada dato del fichero corresponde a una columna diferente. Por ello, es importante especificar que el tipo de fichero leído es syslog y separar de este modo la cabecera de cada entrada de todo fichero syslog y el contenido de este mismo.

```

1 input{
2     file{
3         type => "syslog"
4         path => "/var/log/ossec/logs.txt"
```

```

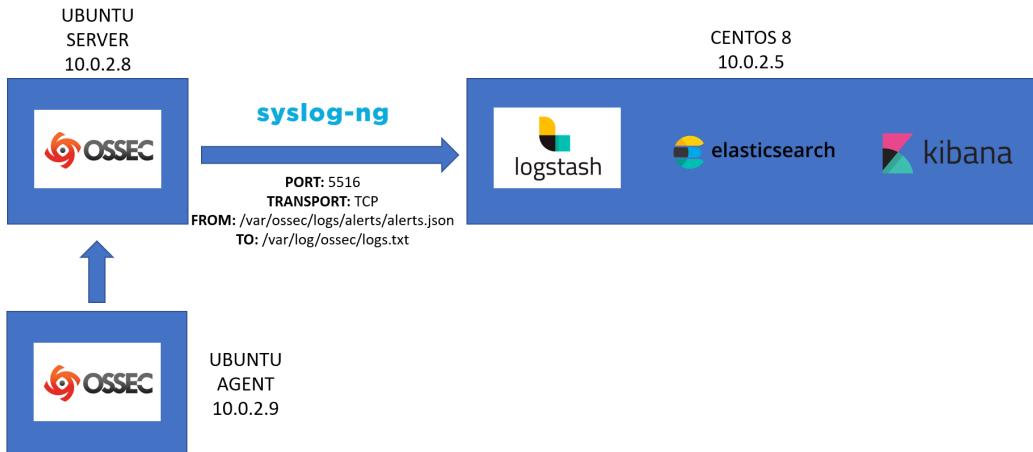
5     start_position => beginning
6   }
7 }
8
9 filter{
10   if [type] == "syslog"{
11     grok {
12       match => {"message" => "%{SYSLOGBASE} %{GREEDYDATA:syslog_message}"}
13     }
14     json {
15       source => "syslog_message"
16     }
17   }
18 }
19
20 output{
21   elasticsearch {hosts => ["localhost:9200"] index => "ossec"}
22   stdout {codec => rubydebug}
23 }
```

Por último, los datos filtrados son enviados a Elasticsearch, todos ellos bajo el índice denominado ossec. Elasticsearch actuará como buscador, para desde Kibana poder crear otro índice con el mismo nombre (ossec) y poder así obtener los datos para realizar la visualización de los mismos.

## Flujo de datos

A continuación, se muestra el flujo de datos desde Ossec hasta su visualización en CETA.

Figura 8.8: Ossec: Flujo de datos

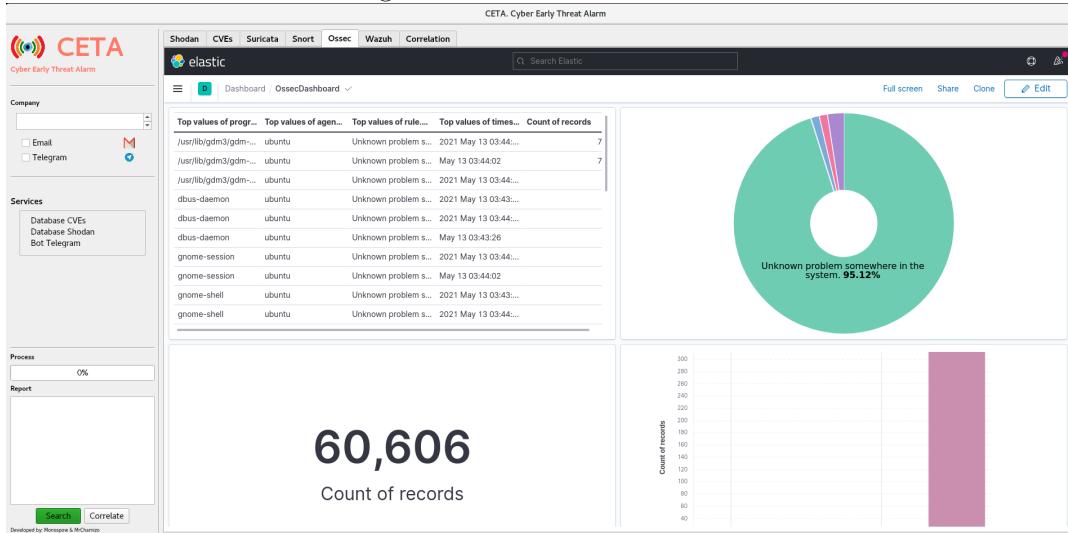


## Visualización

La visualización de forma dinámica de los datos de Ossec en CETA consiste en la integración de la visualización generada en Kibana y reflejada en CETA a través de su enlace

HTTP. Con todo esto, se obtiene una visualización como la siguiente.

Figura 8.9: Ossec: Visualización



### 8.2.5. Wazuh

Wazuh fue la cuarta herramienta que se anexó a CETA. Wazuh se decidió introducir para tener así un HIDS conectado a la plataforma junto Ossec, aunque en este caso la idea es que Wazuh funcione como EDR (Endpoint Detection and Response). En este apartado se van a explicar todos los detalles de la configuración de Wazuh tanto del agente como del servidor y su conexión, así como la de las máquinas virtuales que alojan las herramientas. Es necesario exponer la conexión entre Wazuh server y CETA para la transmisión de las alertas en tiempo real, y la visualización correspondiente de los datos en CETA.

Por esto, se va a dividir en diferentes apartados esta sección. Se comenzará hablando de la configuración de la máquina virtual Ubuntu dónde se encuentra la herramienta de monitorización Wazuh Server y la máquina virtual Windows dónde se encuentra la herramienta de monitorización Wazuh Agent. A continuación, se hablará de la propia herramienta Wazuh, tanto de su descarga como de su configuración, además de la conexión entre agente y servidor. Lo siguiente será hablar de la conexión entre Wazuh server y CETA para la transmisión de datos en tiempo real de una máquina a otra. Por último, es necesario explicar el proceso de almacenado, indexación y visualización de las alertas recibidas en la máquina destino.

Con todo esto, tendríamos una visión completa de la herramienta de monitorización HIDS/EDR como es Wazuh.

#### Ubuntu - Windows

Wazuh Server se encuentra alojado en una máquina virtual con el sistema operativo Ubuntu. La configuración de este sistema operativo no tiene ninguna peculiaridad, ya que se

usa con los parámetros por defecto que esta tiene. Por otro lado, Wazuh Agent se encuentra alojado en una máquina virtual con el sistema operativo Windows 10. Al igual que la máquina Ubuntu, esta máquina Windows no tiene ninguna peculiaridad ya que usa los parámetros por defecto.

Lo único destacable, es que las máquinas son importadas de la página web de Osboxes, la cuál tiene máquinas preconfiguradas de todo tipo. Y que es necesario la creación de 2 máquinas virtuales para poder instalar en una de ellas el agente Wazuh y conectarlo a otra dónde se encuentra el servidor Wazuh.

## Wazuh

Wazuh ha sido instalado siguiendo los pasos encontrados en su página web oficial. Tanto el agente como el servidor para la distribución seleccionada tiene su guía de instalación. [61]

### Conexión: Syslog-*ng*

Para realizar la conexión entre las máquinas ya se ha comentado que la herramienta a utilizar es Syslog-*ng*. En primer lugar es necesario que Syslog-*ng* se encuentre descargado en ambas máquinas virtuales, tanto la que aloja el servidor Wazuh como CETA. A continuación, se explican los detalles de la configuración de esta herramienta.

Desde la máquina virtual Ubuntu es necesario especificar el directorio que contiene los logs de Wazuh y especificar el programa que genera dichos logs. Por otro lado, es necesario exponer la dirección IP destino de la máquina CENTOS dónde se encuentra CETA, y el puerto y protocolo a través del cuál serán enviados los logs. Configuración Syslog-*ng* en Ubuntu:

```
1 source s_wazuh{  
2     file("/var/ossec/logs/alerts/alerts.json"  
3     program_override("wazuh")  
4     flags(no_parse));  
5 };  
6  
7 destination d_wazuh{  
8     syslog("10.0.2.5" transport("tcp") port(5517));  
9 }  
10  
11 log{  
12     source(s_wazuh);  
13     destination(d_wazuh);  
14 }
```

En el lado de la máquina CENTOS, es necesario determinar el puerto a través del cuál se reciben los logs y el protocolo. Además, es importante especificar el destino de los logs, en este caso se ha creado una carpeta nueva con el nombre de wazuh bajo el directorio /var/log, en el cuál es necesario dar los permisos de administrador para poder escribir. Configuración Syslog-*ng* en CENTOS:

```
1 source s_wazuh{  
2     syslog(transport("tcp") port(5517));  
3 };  
4
```

```

5 destination d_wazuh{
6   file("/var/log/wazuh/logs.txt"
7   owner("root")
8   group("root")
9   perm(0777));
10 }
11
12 log{
13   source(s_wazuh);
14   destination(d_wazuh);
15 }
```

También es necesario abrir el puerto correspondiente en CENTOS para permitir la conexión. En este caso el puerto que es necesario abrir es el 5517. Para permitir que el firewall de CENTOS acepte paquetes, es necesario especificar lo siguiente:

```

1 # firewall-cmd --zone=public --add-port=5517/tcp --permanent
2
3 # firewall-cmd --reload
```

## ELK Stack

Una vez se tienen los logs almacenados en la máquina virtual CENTOS dónde se encuentra CETA, es necesario importar estos datos a la propia máquina. Para ello, se utilizarán las herramientas ELK Stack.

En primer lugar Logstash toma como entrada el fichero txt almacenado en el directorio /var/log/wazuh y lo filtra, de este modo cada fila del fichero corresponde a una entrada que es mandada a ElasticSearch, y en la que cada dato del fichero corresponde a una columna diferente. Por ello, es importante especificar que el tipo de fichero leído es syslog y separar de este modo la cabecera de cada entrada de todo fichero syslog y el contenido de este mismo.

```

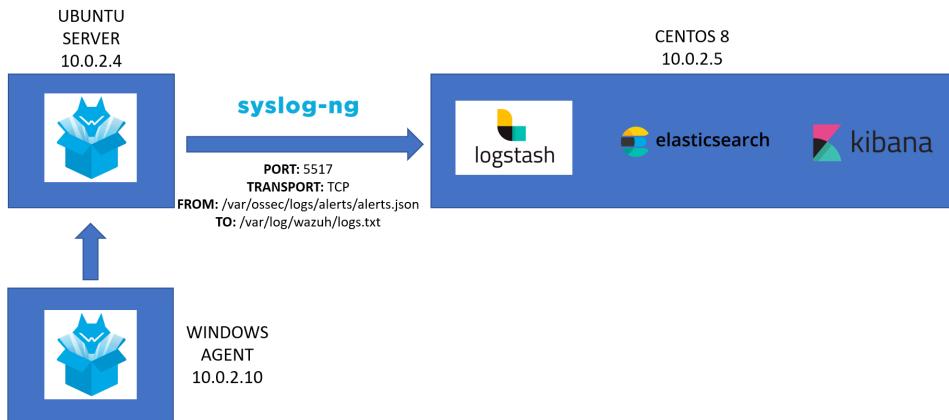
1 input{
2   file{
3     type => "syslog"
4     path => "/var/log/wazuh/logs.txt"
5     start_position => beginning
6   }
7 }
8
9 filter{
10   if [type] == "syslog"{
11     grok {
12       match => {"message" => "%{SYSLOGBASE} %{GREEDYDATA:syslog_message}"}
13     }
14     json {
15       source => "syslog_message"
16     }
17   }
18 }
19
20 output{
21   elasticsearch {hosts => ["localhost:9200"] index => "wazuh"}
22   stdout {codec => rubydebug}
23 }
```

Por último, los datos filtrados son enviados a Elasticsearch, todos ellos bajo el índice denominado wazuh. Elasticsearch actuará como buscador, para desde Kibana poder crear otro índice con el mismo nombre (wazuh) y poder así obtener los datos para realizar la visualización de los mismos.

## Flujo de datos

A continuación, se muestra el flujo de datos desde Wazuh hasta su visualización en CETA.

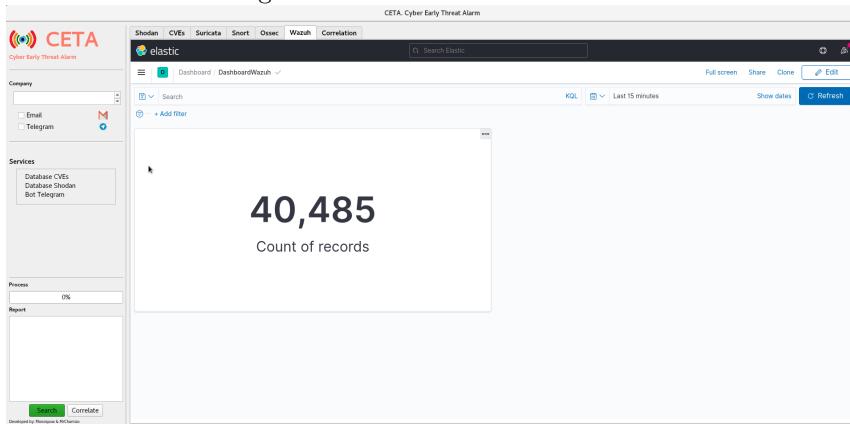
Figura 8.10: Wazuh: Flujo de datos



## Visualización

La visualización de forma dinámica de los datos de Wazuh en CETA consiste en la integración de la visualización generada en Kibana y reflejada en CETA a través de su enlace HTTP. Con todo esto, se obtiene una visualización como la siguiente.

Figura 8.11: Wazuh: Visualización



### **8.2.6. AlienVault**

La herramienta AlienVault se ha llegado a testear pero no ha sido posible conectarla a CETA debido a la complejidad que esto supone. A diferencia de las otras herramientas, no es de tan fácil acceso los logs generados por dicha herramienta. De hecho, la única posibilidad que se observó durante su testeo era la de crear un script que a través del acceso a su interfaz web descargase periódicamente la lista de alertas generadas.

Esto suponía una gran complejidad, además de distanciarse de la idea del proyecto que es la de recibir alertas de forma dinámica y automática. Para la solución adoptada tendría que crearse un temporizador que periódicamente descargase los datos para poder mostrarlos en CETA.

En definitiva una herramienta que de cara a una futura mejora sería muy útil.

## **8.3. Correlación de datos**

Una vez se tiene conectadas las diferentes herramientas de monitorización de organizaciones, es interesante la correlación de los datos proporcionados por las diferentes plataformas. La idea principal que se ha adoptado para crear dicha correlación, es la de crear un dataset unificado con los datos de todas las herramientas extrayendo aquellos campos más importantes de cada una de ellas. Una vez se tuviese dicho dataset, para poder correlacionar diferentes tipos de eventos lo interesante sería crear una visualización de los datos y poder así hacer análisis comparativos y de tendencias.

Con esta introducción se va a entrar ahora más en detalle en lo que respecta a la creación del dataset unificado y la visualización de dichos datos en CETA.

### **8.3.1. Creación Dataset**

Para la creación del dataset se toma como entrada las 4 fuentes de logs de las herramientas Suricata, Snort, Ossec y Wazuh. Estos logs se encuentran ubicados en los directorios comentados en el anterior apartado bajo la ruta /var/log.

El programa se encarga de crear el nuevo dataset extrayendo los datos más importantes de las diferentes fuentes. A continuación, se explica los datos extraídos de cada herramienta.

- Suricata. Los datos extraídos de esta herramienta son los siguientes:

- Source IP
- Destination IP
- Source Port.
- Destination Port
- Time
- Alert id
- Alert severity

- Alert signature
- Snort. Los datos extraídos de esta herramienta son los siguientes:
- Source IP
  - Destination IP
  - Time
  - Priority
- Ossec. Los datos extraídos de esta herramienta son los siguientes:
- Time
  - Alert
  - Rule Level
  - Hostname
  - Full log
- Wazuh. Los datos extraídos de esta herramienta son los siguientes:
- Time
  - Rule description
  - Rule level
  - Agent name
  - Full log

Todos estos datos se organizan de la siguiente manera en el nuevo dataset (la primera columna corresponde a los nombres de los datos del nuevo dataset).

<b>DATASET</b>	<b>SURICATA</b>	<b>SNORT</b>	<b>OSSEC</b>	<b>WAZUH</b>
platform	suricata	snort	ossec	wazuh
source-ip	src_ip	source-ip		
destination-ip	dest_ip	destination-ip		
source-port	src_port			
destination-port	dest_port			
time	timestamp	event-second	timestamp	timestamp
alert	alert signature_id	event signature_id	rule comment	rule description
priority	alert severity	event priority	rule level	rule level
hostname		hostname		agent name
full_log	alert signature		full_log	full_log

Cuadro 8.1: Correlation Dataset

Como se puede observar hay una clara distinción entre los datos procedentes de un NIDS y de un HIDS. Los que provienen de un NIDS tiene una dirección de origen y destino, mientras los que provienen de un HIDS tan sólo tienen el host dónde se produce la alerta.

En cuanto a las cosas en común que tienen los datos de las diferentes fuentes, podemos destacar la fecha y la prioridad de la alerta. Como se comentará más adelante en el clusterizado estas 2 variables serán las utilizadas para esta acometida.

A pesar de todo, con esta información obtenemos una única fuente de datos que contiene la información de todas las herramientas y que aporta gran valor de cara a realizar tanto las correlaciones pertinentes, como la futura implementación de un algoritmo de Machine Learning para clasificar el tipo de alertas.

### 8.3.2. Correlación

El siguiente paso una vez creado el dataset, es poder importarlo y visualizarlo en la herramienta CETA. Al igual que ocurre con el resto de herramientas, se va a utilizar ELK Stack para realizar esta tarea.

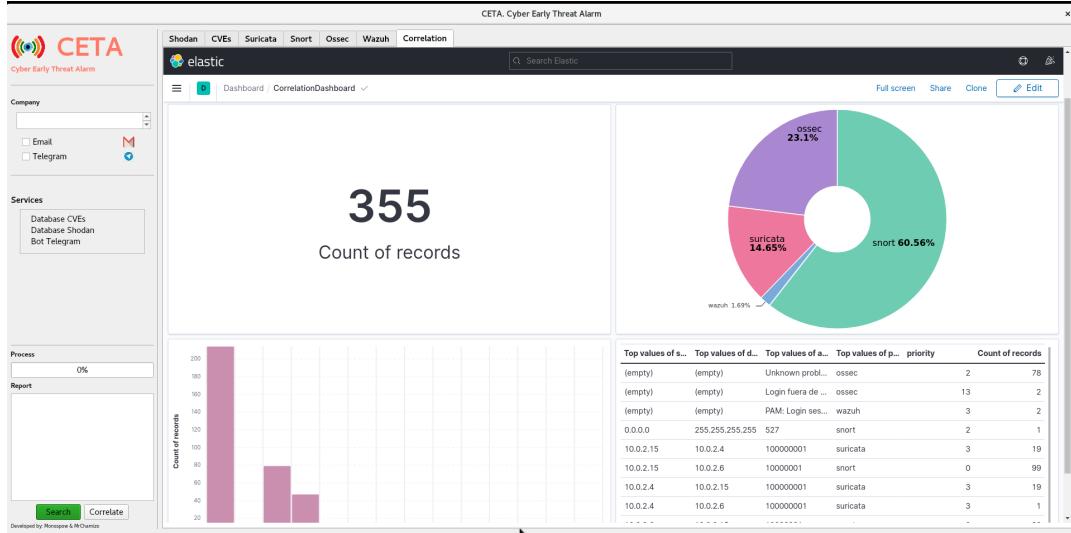
En un primer momento, Logstash va a filtrar los datos para posteriormente cargarlos a ElasticSearch. Es por esto, que es necesario añadir un nuevo fichero de configuración a logstash como el siguiente que se muestra a continuación.

```
1 input{
2     file{
3         type => "json"
4         path => "/home/U406577/git/alertatemprana/2.Alertatemprana/dataset/
5         logs.json"
6         start_position => beginning
7     }
8 }
9 filter{
10    json {
11        source => "message"
12    }
13 }
14
15 output{
16     elasticsearch {hosts => ["localhost:9200"] index => "correlation"}
17     stdout {codec => rubydebug}
18 }
```

Es importante especificar el fichero de entrada, así como el índice que se desea que reciba en ElasticSearch que actuará como buscador. Desde Kibana poder crear otro índice con el mismo nombre (correlation) y poder así obtener los datos para realizar la visualización de los mismos.

Lo interesante es que se podrá hacer todo tipo de visualizaciones, mostrando por ejemplo la comparativa de alertas entre plataformas. O se podría ver en una tabla la lista de direcciones origen y destino de todas las herramientas sobre las que se ha producido una alerta o incluso ver por fecha y hora en qué momento hay más alertas en todas las herramientas para detectar un posible ataque potencialmente peligroso.

Figura 8.12: Correlación: Visualización



## 8.4. Clusterizado

El último paso consiste en clasificar las diferentes alertas mediante el uso e implementación de un algoritmo de Machine Learning. Para este caso concreto el modelo que mejor se ajusta para la clasificación es el clusterizado.

La elección de este tipo de algoritmo se debe a que los datos son de tipo no supervisado, es decir, se tiene un set de datos X sin etiqueta, no se sabe lo que se está buscando. Este algoritmo agrupará los objetos similares en clusters, y a diferencia de la clasificación los clusters no son conocidos al comienzo del experimento.

### 8.4.1. Algoritmo

Entrando un poco más en detalle en lo que respecta al algoritmo, el clusterizado utilizado en este experimento es el llamado K-means. La idea consiste en calcular en base a los datos que se tienen el número de clusters óptimo sobre el que luego se tendrá que realizar el clusterizado.

Para esto, ya se ha comentado en la parte de correlación, pero los datos que mejor encajan para las 4 herramientas son el del tiempo y prioridad de la alerta. El algoritmo devuelve el número de cluster óptimos que son mandados al método Kmeans que se encarga de realizar el clusterizado. Finalmente se obtiene un dataset con el cluster correspondiente a cada entrada.

Otro dato importante, es que este tipo de algoritmos no funciona cuando estamos tratando con caracteres que no son numéricos, es decir con strings. Por eso, previamente a la elección del número de clusters y clusterizado, ha sido necesario la transformación de los strings a

enteros. Para ello, se ha convertido la cadena de strings a bytes y su correspondiente valor numérico. Así, el algoritmo funciona correctamente ya que los datos recibidos son de tipo entero.



# Capítulo 9

## Pruebas

En este capítulo se van a detallar las diferentes pruebas que se han realizado sobre CETA, para comprobar que tanto la comunicación, como la generación de alertas adicionales funciona correctamente. Básicamente en este apartado se van a simular una serie de 'posibles ataques' y la comprobación de que esta alerta es enviada a CETA en tiempo real y es visible para el experto de seguridad responsable de una organización.

En este caso, las pruebas se van a realizar sobre las herramientas Suricata, Snort y Ossec a las cuáles como ya se ha comentado en el anterior capítulo se las ha añadido unas reglas adicionales para probar la seguridad y en este caso, para probar la generación de alertas en tiempo real.

En cada una de las secciones de cada herramienta se volverá a explicar brevemente la alerta creada para la realización de la prueba, así como la simulación del 'ataque' y la respuesta mostrada en la visualización de CETA.

### 9.1. Suricata

La primera de las herramientas que se va a probar es Suricata. Como ya se ha dicho anteriormente, comenzaremos exponiendo la nueva regla generada en suricata en el fichero local rules, posteriormente la simulación del ataque y finalmente la visualización.

#### 9.1.1. Regla

Se ha añadido la siguiente regla al fichero local rules de la herramienta Suricata ubicado en /etc/suricata/rules.

```
1 alert icmp any any -> $HOME_NET any (msg:"Paquete ICMP"; sid:100000001;)
```

#### 9.1.2. Ataque

La simulación del ataque en este caso es muy simple. Tan solo consiste en la realización de un ping de una máquina a otra dentro de la red sobre la que se encuentra Suricata observando. El comando sería algo como lo siguiente:

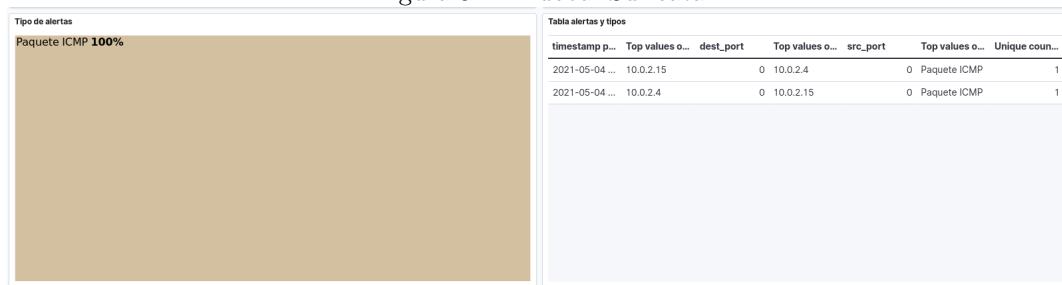
```
1 ping 10.0.2.4
```

Con la ejecución de este comando Suricata debería generar una alerta que es mandada a través de la trama de conexión y visualización de CETA.

### 9.1.3. Visualización

Realizando la comprobación en CETA se observa que se ha realizado correctamente el ataque y por consiguiente el equipo ha respondido en consecuencia, generando una alerta en la plataforma central sobre la que el responsable de seguridad de una organización debe operar.

Figura 9.1: Prueba: Suricata



## 9.2. Snort

La segunda herramienta que se va a probar es Snort. Se comenzará exponiendo las nuevas reglas generadas en snort en el fichero local rules, posteriormente la simulación del ataque y finalmente la visualización.

### 9.2.1. Regla

Se han añadido las siguientes reglas al fichero local rules de la herramienta Snort ubicado en /etc/snort/rules.

```
1 # -----
2 # LOCAL RULES
3 # -----
4
5 alert icmp any any -> $HOME_NET any (msg:"Paquete ICMP"; sid:100000001;)
6 alert tcp $HOME_NET any -> any 80 (msg:"Trafico http"; sid:100000003;)
```

### 9.2.2. Ataque

La simulación del ataque en este caso es muy simple. Tan solo consiste en la realización de un ping de una máquina a otra dentro de la red sobre la que se encuentra Snort observando. El comando sería algo como lo siguiente:

```
1 ping 10.0.2.4
```

La simulación del otro ataque también es sencillo. Tan solo basta con conectarse a una web http a través del siguiente comando:

```
1 curl http://google.es
```

Con la ejecución de estos comandos Snort debería generar una alerta que es mandada a través de la trama de conexión y visualización de CETA.

### 9.2.3. Visualización

Realizando la comprobación en CETA se observa que se ha realizado correctamente el ataque y por consiguiente el equipo ha respondido en consecuencia, generando una alerta en la plataforma central sobre la que el responsable de seguridad de una organización debe operar.

Se observa que existen 2 tipos de alertas generadas, una con el identificador 10.000.003 y otra con el identificador 10.000.001, que corresponden con los identificadores de las alertas creadas previamente.

Figura 9.2: Prueba: Snort



## 9.3. Ossec

La tercera y última herramienta que se va a probar es Ossec. Se comenzará exponiendo las nuevas reglas generadas en ossec en el fichero local.rules, posteriormente la simulación del ataque y finalmente la visualización.

### 9.3.1. Regla

Se han añadido las siguientes reglas al fichero local.rules.xml de Ossec.

```
1 <group name="local ,syslog ,authentication_success ,">
2   <rule id="10000002" level="13">
3     <if_group>authentication_success</if_group>
4     <time> 10 pm - 9 pm </time>
5     <description>Login fuera de horas</description>
6   </rule>
7 </group>
```

### 9.3.2. Ataque

La simulación del ataque en este caso es muy simple. Tan solo consiste en el login como super usuario de la máquina agente fuera de las horas marcadas por la regla. El comando a utilizar sería el siguiente:

```
1 sudo su
```

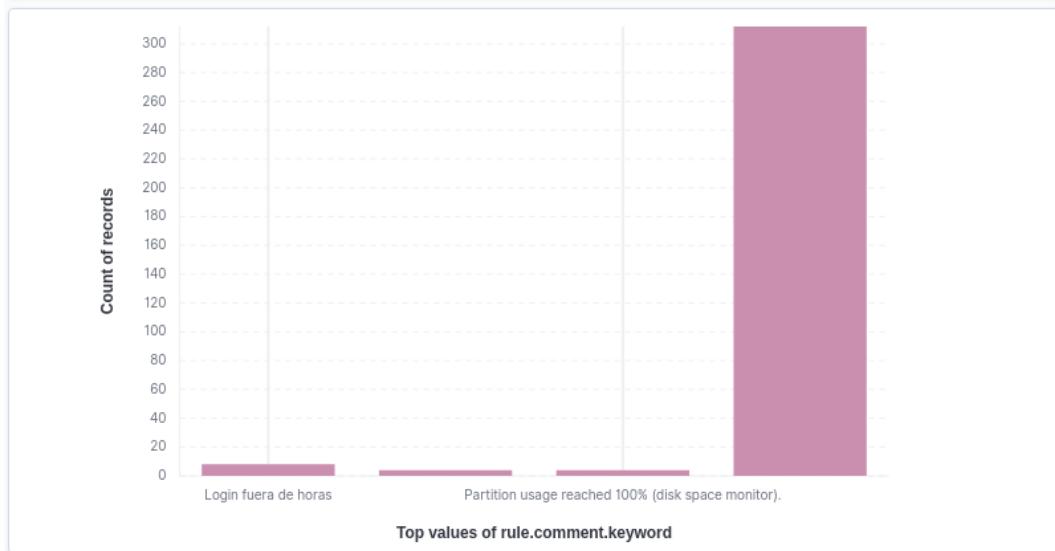
Con la ejecución de este comando Ossec debería generar una alerta que es mandada a través de la trama de conexión y visualización de CETA. Hay que tener en cuenta que la alerta sólo será generada si el login se produce fuera del horario marcado.

### 9.3.3. Visualización

Realizando la comprobación en CETA se observa que se ha realizado correctamente el ataque y por consiguiente el equipo ha respondido en consecuencia, generando una alerta en la plataforma central sobre la que el responsable de seguridad de una organización debe operar.

Se observa que existe una alerta con el nombre de login fuera de horas que se ha producido en el dispositivo agente que tiene Ossec.

Figura 9.3: Prueba: Ossec



# Capítulo 10

## Conclusiones y Líneas Futuras

En este capítulo se expondrán los motivos por los que se considera que se han alcanzado los objetivos marcados al comienzo del trabajo y se indicarán las mejoras que podrían realizarse en un futuro.

### 10.1. Conclusiones

Uno de los objetivos de este TFM era la capacidad de atender un proyecto real de una empresa como es Iberdrola con el renombre que esta tiene. Durante este proyecto, se ha experimentado ese proceso de trabajo en equipo mediante la comparación y exposición de resultados día a día tras las sucesivas reuniones y, la dificultad que puede suponer el no disponer de las herramientas correctas o de la información de estas mismas debido al desconocimiento a lo largo del proyecto. Pese a las dificultades iniciales con la tecnología que un primer momento se iba a utilizar, se ha conseguido desarrollar un proyecto muy completo y con la posibilidad de seguir siendo mejorado de cara a futuro, pero que incluso en la versión actual podría funcionar de manera óptima para su utilización dentro de una organización. Por todo esto, considero que los objetivos marcados en este aspecto se han cumplido.

La monitorización de sistemas y organizaciones me despertaba gran interés, ya que tras haber podido estudiar la asignatura de Monitorización de Sistemas del Máster en Ciberseguridad en la Universidad Pontificia Comillas, alcancé unos conocimientos básicos que fueron puestos en práctica durante el proyecto. A lo largo del proyecto he podido ampliar y mejorar estos conocimiento previos en lo que respecta a la monitorización, llegando a alcanzar y estudiar aspectos completamente nuevos. Cabe destacar también el uso del conocimiento recibido en otras asignaturas del máster como son Inteligencia Artificial y Machine Learning, en la que se ha podido aplicar parte de lo aprendido en el algoritmo de clusterizado creado en el proyecto. Aún así, hay muchas cosas aún por aprender y mejorar de cara al futuro.

En lo que respecta a la propia herramienta de la que tanto se ha hablado durante los diferentes capítulos como es CETA, fue algo totalmente novedoso. Nunca antes se había trabajado con herramientas que integra CETA como son las correspondientes a ELK Stack. Hasta ahora había trabajado con bases de datos para el almacenamiento de la información y posterior visualización en herramienta como pueden ser Power BI o Tableu. El uso de las he-

rramientas ELK Stack fue algo totalmente novedoso a lo que me tuve que adaptar y aprender a funcionar, en un primer momento fue algo complejo, pero con la dedicación, investigación, pruebas y trabajo se consiguieron los objetivos marcados, pero aún así quedan muchas cosas por aprender.

Otro aspecto importante del proyecto tenía que ver con la adición al proyecto de herramientas de monitorización de todo tipo, desde HIDS a NIDS, incluyendo incluso EDR. Por suerte para la realización de este proyecto se han utilizado herramientas que ya conocía su funcionamiento previamente y su integración ha resultado ser más sencilla. Por otro lado, se han utilizado otras herramientas que eran totalmente desconocidas ya que nunca antes se habían trabajado con ellas, por lo que gracias a la información encontrada en diferentes repositorios y la ayuda proporcionada por el tutor, también se convirtió en una tarea sencilla.

Por último, el objetivo final es la realización de un proyecto software completo. Un gran reto, ya que pese a haber realizado previamente uno durante el desarrollo del trabajo de fin de grado y la realización de un proyecto de investigación para la Universidad de Valladolid, este es completamente diferente debido al perfil de seguridad que se quiere dotar a la herramienta y la importancia que puede llegar a tener esta dentro de una organización. Esto supone un desafío, ya que el proyecto consta de muchas partes distintas, desde las reuniones iniciales para poner en marcha el proyecto, hasta las diferentes implementaciones que se han ido realizando sobre este mismo añadiendo más funcionalidad. Reto completado con creces, debido a la dedicación y trabajo realizado sobre este mismo.

Por todo esto anteriormente mencionado, creo que los objetivos que se marcaron en un principio a la hora de asignación del proyecto se han completado correctamente y que el motivo por el que decidí elegir este TFM para alcanzar dichos conocimientos se han logrado.

### 10.1.1. Conclusiones CETA

Como se comentó en los primeros capítulos del proyecto, el número de amenazas y ataques recibido por parte de las organizaciones cada vez es mayor, y es aquí dónde disponer de mecanismos de seguridad y herramientas de monitorización que determinen el estado de la organización pasan a ser fundamentales.

Durante el desarrollo del proyecto he ido observando el potencial que puede llegar a tener una herramienta como esta y más concretamente la aquí desarrollada como es CETA. La visibilidad que aporta para una organización tanto desde el punto de vista externo, mostrando la visibilidad de sus activos desde el exterior, como el punto de vista interno, mostrando los activos internos y las alertas generadas por cada uno de ellos, aporta gran valor de cara a tener un control total de la seguridad de la organización.

Además de la gran ventaja que supone tener una herramienta centralizada la visualización de los datos generados por las diferentes herramientas de monitorización de una empresa, la posibilidad de unificar todos esos datos para ser analizados de forma conjunta aporta aún más valor. Por eso, en este proyecto se ha tratado de observar y generar ese valor adicional basado en la integración de los resultados mostrados por todas las herramientas. Como ya

se ha dicho en capítulos anteriores, el poder tener los datos unificados permite ver en que franjas de tiempo se producen más ataques, poder relacionar los ataques generados por unas herramientas por los generados por otras, comparar el nivel de prioridad generado por cada herramienta etc.

Por último, una vez que se tienen estos datos unificados se pueden utilizar algoritmos de machine learning para la predicción y clasificación de amenazas. De este modo, una organización basada en los datos que tiene podría simular y predecir ataques futuros y así estar preparados para este futuro ataque, o incluso lo que es en este proyecto desarrollado como es la clasificación de las alertas, para poder asociar cada una de ellas a un tipo diferente y poder así hacer análisis varios.

En definitiva una herramienta totalmente necesaria de cara al futuro de las organizaciones para tener el control y manejo absoluto de la seguridad.

## 10.2. Líneas futuras

En este apartado se comentarán todas aquellas ideas que pueden ser implementadas de cara al futuro desarrollo del proyecto.

### 10.2.1. Adición de nuevas herramientas de monitorización

Actualmente CETA dispone de 4 herramientas de monitorización conectadas como son: Suricata, Snort, Ossec y Wazuh. La idea inicial del proyecto era la de añadir herramientas de las que disponen las organizaciones como son CMDB, SIEM o herramientas de vulnerabilidades.

De cara al futuro por tanto la incorporación de CMDB Aris, SIEM QRadar o Nexplore aportaría más valor a la plataforma y la convertiría así en algo aún más indispensable para una organización, junto con las herramientas que ya lleva integradas.

### 10.2.2. Adición de nuevas sondas externas

Actualmente CETA incorpora 2 sondas externas como son Shodan y CVE Search. Con ambas se obtiene una información de los activos expuestos al exterior de la organización. Si además se incorporasen nuevas sondas externas en relación con otras organizaciones podríamos detectar y prevenir ataques futuros a nuestra empresa.

Es decir, la adición de sondas en las cuales una empresa que se ve atacada exponga el ataque y cómo fue a través de qué activo, puede ser muy útil para mi organización de cara a saber por dónde empezar a protegerme y ver si ese ataque me puede influenciar a mí en base a los datos proporcionados por el resto de sondas externas y los datos proporcionados por las herramientas de monitorización interna.

### **10.2.3. Mejora algoritmo de clasificación**

Acutalmente CETA incorpora un algoritmo de clasificación basado en clusterizado mediante KMeans. Este algoritmo implementado es muy básico, ya que simplemente toma 2 valores del dataset unificado para calcular el número de clusters y posteriormente realizar el clusterizado.

De cara al futuro con la incorporación de más herramientas,, sería interesante la mejora del clusterizado a ser posible añadiendo más campos para calcular el número de clusters óptimos y mejorando los parámetros del algoritmo. Por otro lado, sería posible la implementación de otro tipo de algoritmos de clasificación como puede ser el agrupamiento jerárquico, que igual mejora lo realizado con KMeans.

### **10.2.4. Implementación de algoritmo para detección de anomalías**

De cara al futuro con la cantidad de información que aportan las diferentes herramientas integradas en CETA, sería interesante la implementación de un algortimo que se encargase de detectar las anomalías.

La clave para el correcto funcionamiento de este algoritmo sería la detección de desviaciones a partir de un modelo de normalidad. Un ejemplo posible sería la detección de intrusión. Posiblemente para poder realizar esta tarea tendríamos que recoger todos los logs de las herramientas, no sólo las alertas para poder así detectar correctamente cuando se produce una anomalía.

### **10.2.5. Implementación algoritmo de predicción**

En base al histórico de alertas generadas, sería posible determinar ataques futuros. Mediante la implementación de algoritmos basados en machine learning de predicción con las alertas generadas por las herramientas, sería posible determinar ataques futuros y estar así preparados frente a ellos.

### **10.2.6. Mejora de rendimiento**

Otro de los aspectos a mejorar de cara al futuro es la mejora de rendimiento de la herramienta CETA. Actualmente su uso en una máquina virtual y el poco aprovisionamiento de recursos para dicha máquina hace que vaya un poco lenta la herramienta. La posible integración en un servidor o dispositivo físico puede incrementar el rendimiento y hacer así que funcione con fluidez.

# Apéndice A

# Manual de Usuario

## A.1. Introducción

El objetivo de este manual, es el de mostrar, mediante una serie de pasos, el funcionamiento de esta aplicación, con las diferentes opciones que ofrece.

Cabe mencionar que las imágenes mostradas están tomadas a través de la propia máquina virtual CENTOS 8, y que es posible que el aspecto en otros dispositivos varíe, sobre todo dependiendo de las dimensiones de la pantalla. No obstante, el funcionamiento será el mismo.

## A.2. Inicio

El proyecto dispone de 3 carpetas:

- Scripts: Contiene los Scripts de inicio.
- AlertaTemprana: Contiene el ejecutable de inicio de CETA.
- AlertaTempranaBot: Contiene todo lo necesario para la comunicación mediante el bot de Telegram.

Figura A.1: Manual de usuario: Carpetas



Para una inicialización correcta, es necesario ejecutar el programa dentro de la carpeta Scripts que se denomina initRequirements. Este programa se encarga de comprobar que todas

las dependencias y herramientas se encuentre correctamente descargadas antes de ejecutar CETA.

Figura A.2: Manual de usuario: initRequirements



Una vez ejecutado este programa estaremos en disposición de iniciar CETA. Para ello, es necesario acceder ahora a la carpeta AlertaTemprana y ejecutar el programa AlertSystemMain, que será el encargado de hacer las llamadas pertinentes a otros scripts y arrancar todos los servicios y la herramienta.

Figura A.3: Manual de usuario: AlertSystemMain



Este programa se encarga de iniciar los servicios MongoDB, Kibana, ElasticSearch y TelegramBot. Esta comprobación de que los servicios se han iniciado correctamente se puede observar a través de los terminales que aparecen en pantalla para cada servicio.

Figura A.4: Manual de usuario: Inicio Servicios

Tres capturas de terminal separadas por espaciadores. La primera, titulada 'MongoDB', muestra una línea de código JSON. La segunda, titulada 'Elastic', muestra un mensaje de warning sobre memoria. La tercera, titulada 'Kibana', muestra un log de inicio de monitoreo de estadísticas.

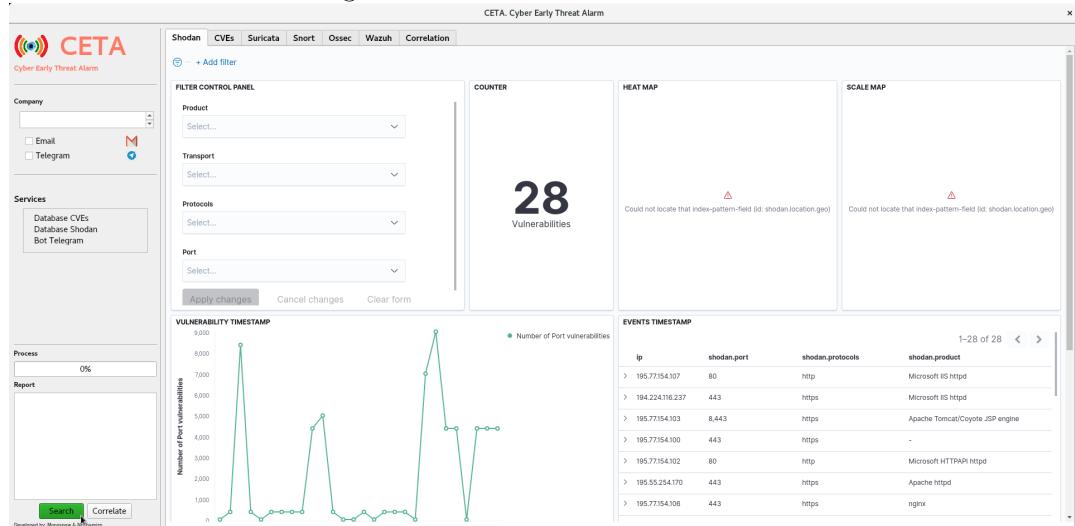
Además también inicia el buscador de CVEs y los carga en MongoDB. En este caso, es el propio terminal dónde se puede observar esta tarea.

Figura A.5: Manual de usuario: Inicio CVE Search

```
-> [Updating cve-search database]
2021-06-21 13:06:09,210 - DBUpdater - INFO - Starting cpe
2021-06-21 13:06:09,251 - CPEDownloads - INFO - - CPE database update started
2021-06-21 13:06:09,251 - CPEDownloads - INFO - - CPE's are not modified since the last update
Download files: 100%
2021-06-21 13:06:13,019 - CPEDownloads - INFO - - Duration: 0:00:03.769402
2021-06-21 13:06:13,022 - CPEDownloads - INFO - - Finished CPE database update
2021-06-21 13:06:13,022 - DBUpdater - INFO - - cpe has 344571 elements (0 update)
2021-06-21 13:06:13,022 - DBUpdater - INFO - - Starting cve
2021-06-21 13:06:13,022 - CPEDownloads - INFO - - CVE database update started
Download files: 0%
2021-06-21 13:06:13,829 - CPEDownloads - INFO - - CVE's are not modified since the last update
2021-06-21 13:06:14,238 - CPEDownloads - INFO - - CVE's are not modified since the last update
2021-06-21 13:06:14,238 - CPEDownloads - INFO - - Duration: 0:00:01.260876
2021-06-21 13:06:14,238 - CPEDownloads - INFO - - Finished CVE database update
2021-06-21 13:06:14,238 - DBUpdater - INFO - - cve has 1234 elements (0 update)
2021-06-21 13:06:14,238 - DBUpdater - INFO - - Starting cwe
2021-06-21 13:06:14,316 - CPEDownloads - INFO - - CWE database update started
Download files: 0%
2021-06-21 13:06:14,316 - CPEDownloads - INFO - - CWE's are not modified since the last update
2021-06-21 13:06:14,316 - CPEDownloads - INFO - - Duration: 0:00:01.260876
2021-06-21 13:06:14,316 - CPEDownloads - INFO - - Finished CWE database update
2021-06-21 13:06:14,316 - DBUpdater - INFO - - cwe has 1234 elements (0 update)
2021-06-21 13:06:14,316 - DBUpdater - INFO - - Starting capec
2021-06-21 13:06:16,010 - CAPECDownloads - INFO - - CAPEC database update started
Download files: 100%
2021-06-21 13:06:17,953 - CAPECDownloads - INFO - - CAPEC's are not modified since the last update
Download files: 100%
2021-06-21 13:06:16,006 - CAPECDownloads - INFO - - Duration: 0:00:01.949479
2021-06-21 13:06:16,006 - CAPECDownloads - INFO - - Finished CAPEC database update
2021-06-21 13:06:16,006 - DBUpdater - INFO - - capec has 527 elements (0 update)
2021-06-21 13:06:16,006 - DBUpdater - INFO - - Starting redis-cache-cpe
2021-06-21 13:06:16,006 - DBRedisBrowser - INFO - - Redis CPE database update started
Inserting CPE's in redis: 100% [344571/344571] | 344571/344571 [10:24:00:00, 551.851t/s]
2021-06-21 13:19:08,064 - CPERedisBrowser - INFO - - Duration: 0:10:27.798402
2021-06-21 13:19:08,141 - DBUpdater - INFO - - redis-cache-cpe updated
2021-06-21 13:19:08,141 - DBUpdater - INFO - - Starting via4
2021-06-21 13:19:10,183 - VIADownloads - INFO - - VIA4 database update started
Download files: 0%
2021-06-21 13:19:24,988 - VIADownloads - INFO - - VIA4's are not modified since the last update
Download files: 100%
```

Por último, aparece la propia vista de CETA. Esta vista tiene las siguientes peculiaridades, en la parte izquierda es posible introducir el nombre de una compañía y marcar que los resultados desean ser enviados a través del email o de telegram. Existen dos botones, uno para realizar la búsqueda de la compañía a través de las fuentes externas y otro para correlacionar los datos de las diferentes herramientas internas de monitorización. En la parte superior se pueden ver diferentes apartados que corresponden a cada una de las herramientas integradas en la plataforma, se pueden ver los datos de cada una de ellas.

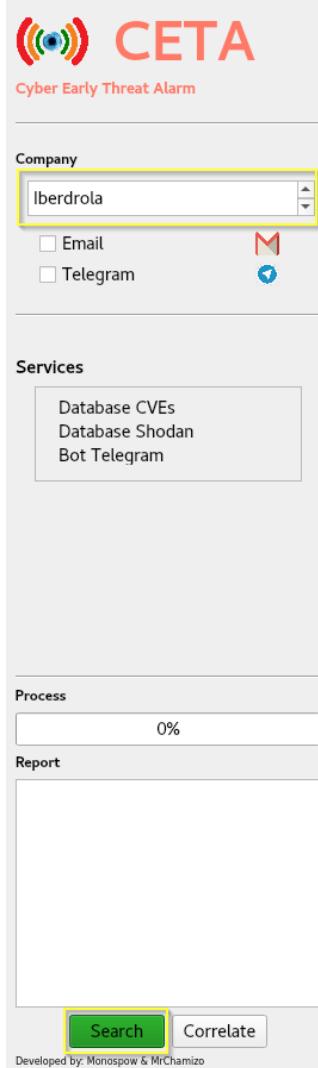
Figura A.6: Manual de usuario: CETA



### A.3. Shodan

Para hacer uso de la sonda Shodan es necesario introducir el nombre de la compañía y pulsar el botón de Search. Esta ventana tarda un poco en cargar debido a la búsqueda que esta realizando internamente sobre la sonda y muestra la información recibida por pantalla.

Figura A.7: Manual de usuario: Buscar Shodan



En cuanto a la visualización, se muestra los datos más relevantes proporcionado por Shodan de forma gráfica y resumida. A continuación, se muestra la información proporcionada por la búsqueda anterior.

Figura A.8: Manual de usuario: CETA Shodan

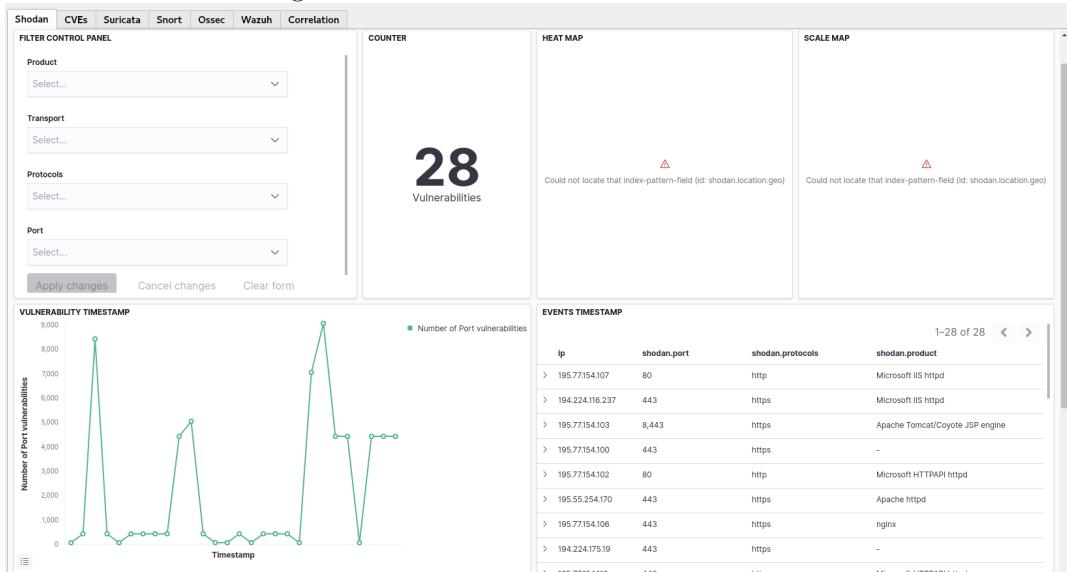
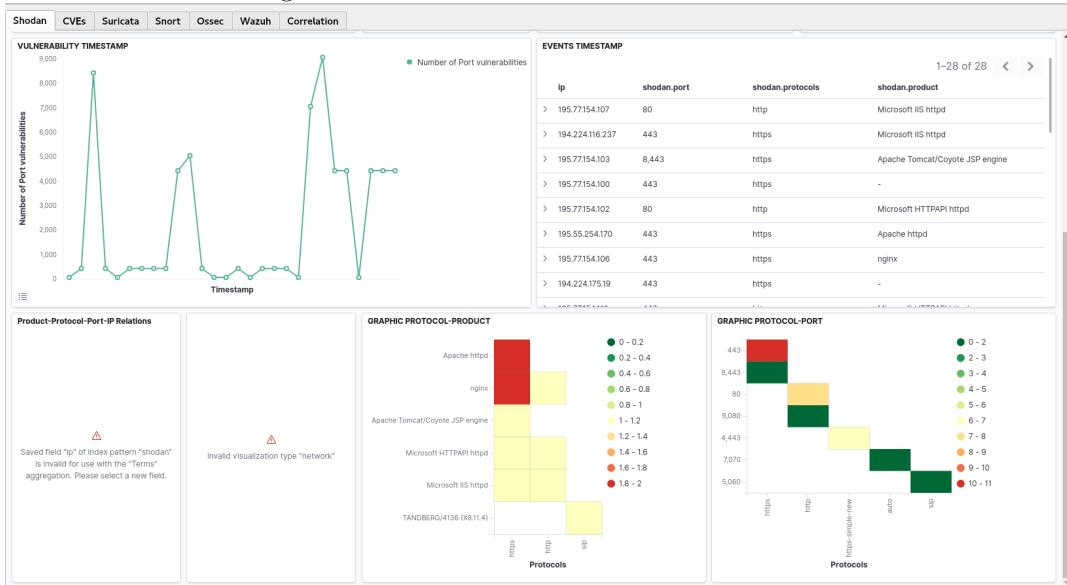


Figura A.9: Manual de usuario: CETA Shodan 1

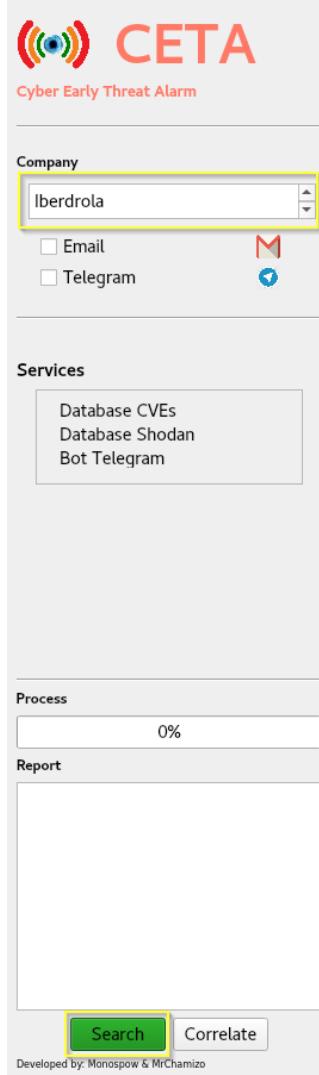


## A.4. CVE Search

Para hacer uso de la sonda CVE Search es necesario introducir el nombre de la compañía y pulsar el botón de Search. Esta ventana tarda un poco en cargar debido a la búsqueda que

esta realizando internamente sobre la sonda y muestra la información recibida por pantalla.

Figura A.10: Manual de usuario: Buscar CVE Search



En cuanto a la visualización, se muestra los datos más relevantes proporcionado por CVE Search de forma gráfica y resumida. A continuación, se muestra la información proporcionada por la búsqueda anterior. Lo interesante es que trata de cruzar los datos recibidos por Shodan con la lista de CVEs y así obtener los CVEs de los activos expuestos al exterior por una organización.

Figura A.11: Manual de usuario: CETA CVE Search

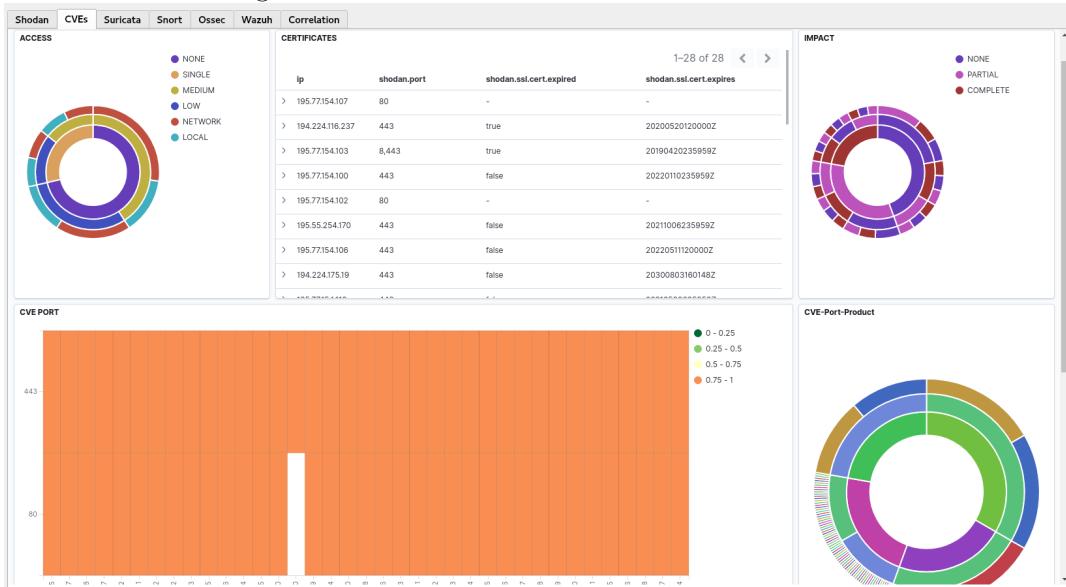
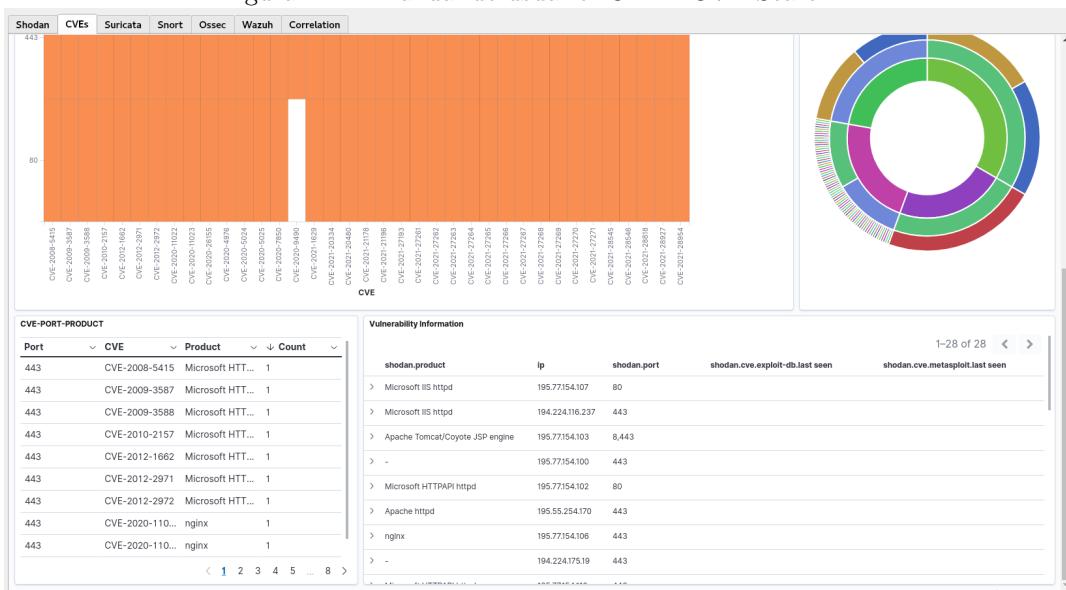


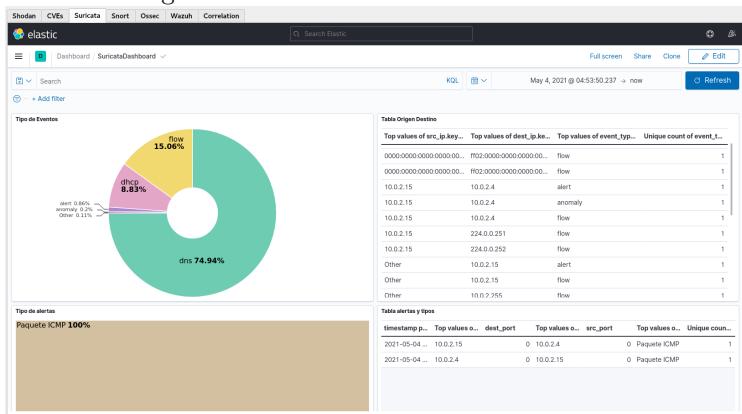
Figura A.12: Manual de usuario: CETA CVE Search 1



## A.5. Suricata

La visualización de las alertas producidas por suricata es muy simple. Tan solo hay que acceder a su campo correspondiente y se tendrá acceso a dicha visualización. Una de las ventajas que proporciona este dashboard frente al de Shodan y CVEs es que es posible modificarlo en tiempo real y añadir y quitar gráficas a gusto del usuario. Para ello tan sólo es necesario pulsar el botón de editar.

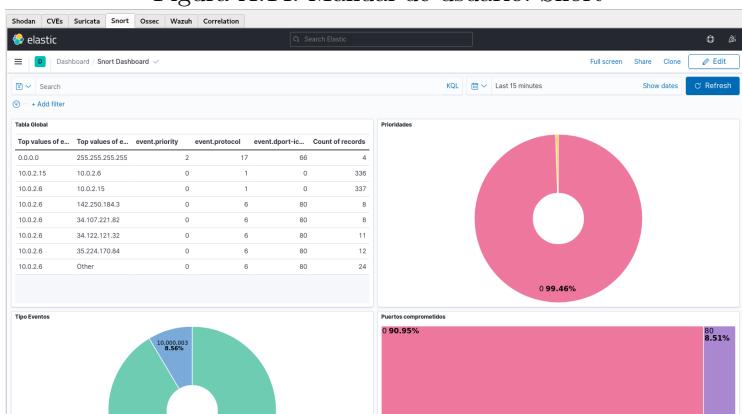
Figura A.13: Manual de usuario: Suricata



## A.6. Snort

La visualización de las alertas producidas por snort es muy simple. Tan solo hay que acceder a su campo correspondiente y se tendrá acceso a dicha visualización. Una de las ventajas que proporciona este dashboard frente al de Shodan y CVEs es que es posible modificarlo en tiempo real y añadir y quitar gráficas a gusto del usuario. Para ello tan sólo es necesario pulsar el botón de editar.

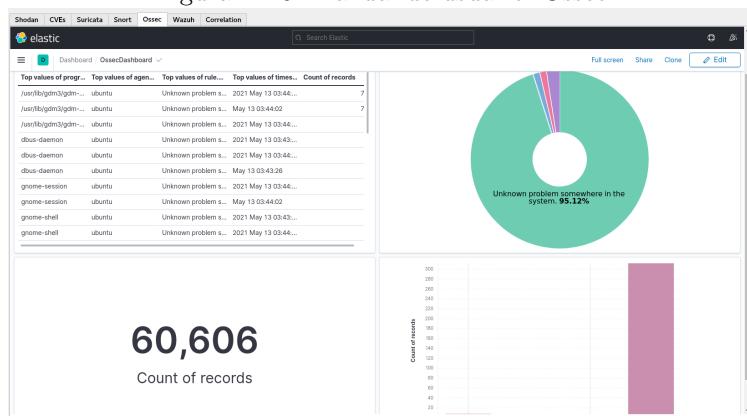
Figura A.14: Manual de usuario: Snort



## A.7. Ossec

La visualización de las alertas producidas por ossec es muy simple. Tan solo hay que acceder a su campo correspondiente y se tendrá acceso a dicha visualización. Una de las ventajas que proporciona este dashboard frente al de Shodan y CVEs es que es posible modificarlo en tiempo real y añadir y quitar gráficas a gusto del usuario. Para ello tan sólo es necesario pulsar el botón de editar.

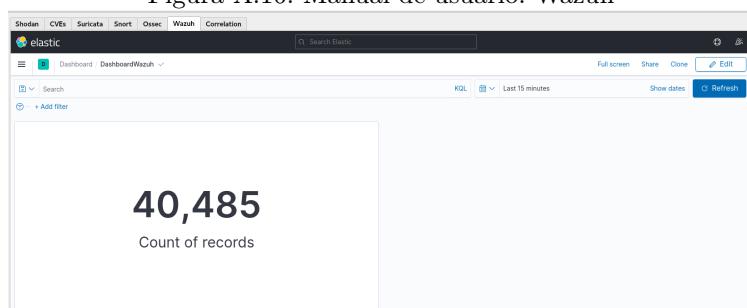
Figura A.15: Manual de usuario: Ossec



## A.8. Wazuh

La visualización de las alertas producidas por wazuh es muy simple. Tan solo hay que acceder a su campo correspondiente y se tendrá acceso a dicha visualización. Una de las ventajas que proporciona este dashboard frente al de Shodan y CVEs es que es posible modificarlo en tiempo real y añadir y quitar gráficas a gusto del usuario. Para ello tan sólo es necesario pulsar el botón de editar.

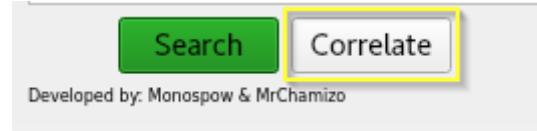
Figura A.16: Manual de usuario: Wazuh



## A.9. Correlation

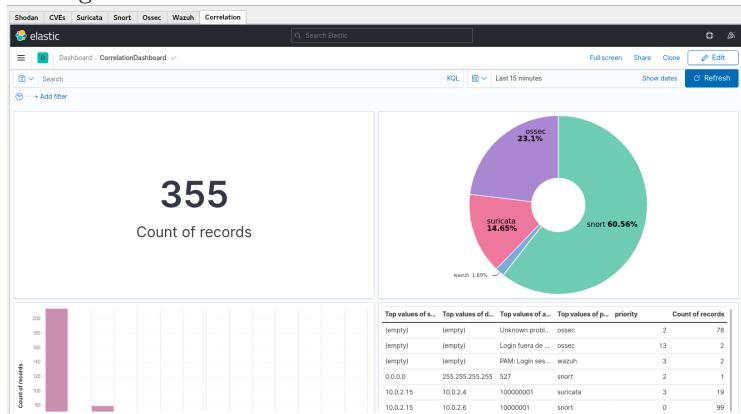
Para realizar la correlación de los datos de las diferentes herramientas de monitorización, basta simplemente con pulsar el botón de Correlate y este cargará automáticamente la vista de correlación de forma automática.

Figura A.17: Manual de usuario: Correlation



En cuanto a la visualización, ocurre lo mismo que con el resto de herramientas. Es posible modificar en tiempo real el dashboard añadiendo y suprimiendo gráficas. Además, como se acaba de comentar, tras pulsar el botón la vista se carga automáticamente con los últimos datos recibidos de las herramientas.

Figura A.18: Manual de usuario: Correlation Dashboard



## Apéndice B

# Soporte Digital

Una breve explicación del contenido almacenado en el soporte digital:  
<https://bitbucket.org/mrchamizo/alertatemprana2021/src/master/>

- Carpeta Scripts: contiene los ejecutables correspondientes para la descarga y comprobación de que todas las dependencias están instaladas. Además, incorpora los scripts de ejecución de los servicios y de inicialización como son ELK Stack, MongoDB. Por último, contiene scripts de finalización de los servicios.
- Carpeta AlertaTemprana: contiene toda la información y el código necesario para que funcione correctamente la herramienta CETA. En su interior se encuentra el ejecutable inicial para arrancar CETA, así como los scripts para la correlación y clusterizado de los datos.
- Carpeta AlertaTempranaBot: contiene el código necesario para poner en funcionamiento el bot de telegram.

El repositorio oficial con toda la información del proyecto es el siguiente:  
<https://github.com/MrChamizo98/Detecci-n-y-Alerta-Temprana-GEM>



# Bibliografía

- [1] *Cada vez más empresas llevan a cabo su transformación tecnológica,*  
HTTPS://NOTICIASDELACIENCIA.COM/.  
Recuperado a 08/07/2021,  
de <https://noticiasdelaciencia.com/art/28985/cada-vez-mas-empresas-llevan-a-cabo-su-transformacion-tecnologica>
- [2] *Un informe de INTERPOL muestra un aumento alarmante de los ciberataques durante la epidemia de COVID-19,*  
HTTPS://WWW.INTERPOL.INT.  
Recuperado a 08/07/2021,  
de <https://www.interpol.int/es/Noticias-y-acontecimientos/Noticias/2020/Un-informe-de-INTERPOL-muestra-un-aumento-alarmante-de-los-ciberataques-durante-la-epidemia-de-COVID-19>
- [3] *CiberSOC: Gestión y monitorización de la seguridad informática en las empresas,*  
HTTPS://WWW.INFORGES.ES.  
Recuperado a 08/07/2021,  
de <https://www.inforges.es/post/cibersoc-gestion-monitorizacion-seguridad-informatica-en-las-empresas>
- [4] *Shodan,*  
HTTPS://WWW.SHODAN.IO/.  
Recuperado a 08/07/2021,  
de <https://www.shodan.io/>
- [5] *CVE Search,*  
HTTPS://CVE.MITRE.ORG/.  
Recuperado a 08/07/2021,  
de [https://cve.mitre.org/cve/search\\_cve\\_list.html](https://cve.mitre.org/cve/search_cve_list.html)  
<https://www.circl.lu/services/cve-search/>
- [6] *Temario Asignatura Monitorización y Forense Máster Ciberseguridad,*  
UNIVERSIDAD PONTIFICIA COMILLAS.  
Recuperado a 08/07/2021
- [7] *Fuerte crecimiento de los ciberataques en España en los últimos dos años,*  
HTTPS://FUTURE.INESE.ES/.  
Recuperado a 08/07/2021,

- de <https://future.inese.es/fuerte-crecimiento-de-los-ciberataques-en-espana-en-los-ultimos-dos-anos/>
- [8] *Los delitos informáticos ocasionaron en 2019 pérdidas superiores al 1 % del PIB mundial, por encima de los 800.000 millones de euros,*  
HTTPS://WWW.BUSINESSINSIDER.ES/.  
Recuperado a 08/07/2021,  
de <https://www.businessinsider.es/impacto-ciberdelitos-ya-superior-1-pib-mundial-768519>
- [9] *Los tipos de ciberataques a empresas más frecuentes,*  
HTTPS://WWW.VIEWNEXT.COM/.  
Recuperado a 08/07/2021,  
de <https://www.viewnext.com/tipos-de-ciberataques-a-empresas/>
- [10] *El sector financiero es el más afectado por los ciberataques,*  
HTTPS://WWW.EXPANSION.COM/.  
Recuperado a 08/07/2021,  
de <https://www.expansion.com/economia-digital/2020/10/25/5f956b50468aebf4638b467c.html>
- [11] *Desarrollo ágil de software,*  
HTTPS://ES.WIKIPEDIA.ORG/.  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
- [12] *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa,*  
HTTPS://WWW.IEBSCHOOL.COM/.  
Recuperado a 08/07/2021,  
de <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- [13] *CMDB (Base de datos de la gestión de configuración),*  
HTTPS://ES.WIKIPEDIA.ORG/.  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/Base\\_de\\_datos\\_de\\_la\\_gesti%C3%B3n\\_de\\_configuraci%C3%B3n](https://es.wikipedia.org/wiki/Base_de_datos_de_la_gesti%C3%B3n_de_configuraci%C3%B3n)
- [14] *Rapid7 Nexpose),*  
HTTPS://WWW.RAPID7.COM.  
Recuperado a 08/07/2021,  
de <https://www.rapid7.com/products/nexpose/>
- [15] *Sistemas IDS, IPS, HIDS, NIPS, SIEM ¿Qué son?,*  
HTTPS://WWW.A2SECURE.COM.  
Recuperado a 08/07/2021,  
de <https://www.a2secure.com/blog/ids-ips-hids-nips-siem-que-es-esto/>
- [16] *SIEM,*  
HTTPS://ES.WIKIPEDIA.ORG/.

Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/Gesti%C3%B3n\\_de\\_informaci%C3%B3n\\_y\\_eventos\\_de\\_seguridad](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_informaci%C3%B3n_y_eventos_de_seguridad)

[17] *Kaspersky*,  
<HTTPS://WWW.KASPERSKY.ES/>.  
Recuperado a 08/07/2021,  
de <https://www.kaspersky.es/>

[18] *Sophos*,  
<HTTPS://WWW.SOPHOS.COM/>.  
Recuperado a 08/07/2021,  
de <https://www.sophos.com/>

[19] *Akami*,  
<HTTPS://WWW.AKAMAI.COM/>.  
Recuperado a 08/07/2021,  
de <https://www.akamai.com/es/es/resources/cyber-security.jsp>

[20] *Digital Attack Map*,  
<HTTPS://WWW.DIGITALATTACKMAP.COM/>.  
Recuperado a 08/07/2021,  
de <https://www.digitalattackmap.com/>

[21] *Talos*,  
<HTTPS://TALOSINTELLIGENCE.COM/>.  
Recuperado a 08/07/2021,  
de <https://talosintelligence.com/>

[22] *Looking Glass*,  
<HTTPS://LOOKINGGLASSCYBER.COM/>.  
Recuperado a 08/07/2021,  
de <https://lookingglasscyber.com/>

[23] *Nmap*,  
<HTTPS://NMAP.ORG/>.  
Recuperado a 08/07/2021,  
de <https://nmap.org/>

[24] *OpenVas*,  
<HTTPS://WWW.OPENVAS.ORG/>.  
Recuperado a 08/07/2021,  
de <https://www.openvas.org/>

[25] *Nessus*,  
<HTTPS://ES-LA.TENABLE.COM/>.  
Recuperado a 08/07/2021,  
de <https://es-la.tenable.com/products/nessus/nessus-professional>

[26] *Bitsight*,  
<HTTPS://WWW.BITSIGHT.COM/>.

Recuperado a 08/07/2021,  
de <https://www.bitsight.com/>

[27] *Recorded Future*,  
[HTTPS://WWW.RECORDED FUTURE.COM/](https://WWW.RECORDED FUTURE.COM/).  
Recuperado a 08/07/2021,  
de <https://www.recordedfuture.com/>

[28] *Red Tor*,  
[HTTPS://WWW.TORPROJECT.ORG/ES/](https://WWW.TORPROJECT.ORG/ES/).  
Recuperado a 08/07/2021,  
de <https://www.torproject.org/es/>

[29] *Have I been Pwned*,  
[HTTPS://HAVEIBEENPWNED.COM/](https://HAVEIBEENPWNED.COM/).  
Recuperado a 08/07/2021,  
de <https://haveibeenpwned.com/>

[30] *Censys*,  
[HTTPS://CENSYS.IO/](https://CENSYS.IO/).  
Recuperado a 08/07/2021,  
de <https://censys.io/>

[31] *NIDS*,  
[HTTPS://ES.WIKIPEDIA.ORG/](https://ES.WIKIPEDIA.ORG/).  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/NIDS#:~:text=NIDS%20\(Network%20Intrusion%20Detection%20System,de%20intrusos%20en%20una%20Red.&text=Los%20NIDS%20no%20s%C3%B3lo%20vigilan,desde%20el%20propio%20sistema%20prottegido.](https://es.wikipedia.org/wiki/NIDS#:~:text=NIDS%20(Network%20Intrusion%20Detection%20System,de%20intrusos%20en%20una%20Red.&text=Los%20NIDS%20no%20s%C3%B3lo%20vigilan,desde%20el%20propio%20sistema%20prottegido.)

[32] *HIDS*,  
[HTTPS://ES.WIKIPEDIA.ORG/](https://ES.WIKIPEDIA.ORG/).  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/Host-based\\_intrusion\\_detection\\_system#:~:text=HIDS%20es%20tambi%C3%A9n%20conocido%20como,a%20las%20de%20los%20IDS.](https://es.wikipedia.org/wiki/Host-based_intrusion_detection_system#:~:text=HIDS%20es%20tambi%C3%A9n%20conocido%20como,a%20las%20de%20los%20IDS.)

[33] *Snort*,  
[HTTPS://WWW.SNORT.ORG/](https://WWW.SNORT.ORG/).  
Recuperado a 08/07/2021,  
de <https://www.snort.org/>

[34] *Suricata*,  
[HTTPS://SURICATA.IO/](https://SURICATA.IO/).  
Recuperado a 08/07/2021,  
de <https://suricata.io/>

[35] *Ossec*,  
[HTTPS://WWW.OSSEC.NET/](https://WWW.OSSEC.NET/).

Recuperado a 08/07/2021,  
de <https://www.ossec.net/>

<https://www.cytomic.ai/es/tendencias/reglas-yara-detectar-malware/>

- [36] *Reglas YARA*,  
<HTTPS://WWW.CYTOMIC.AI/>.  
Recuperado a 08/07/2021,  
de <https://www.cytomic.ai/es/tendencias/reglas-yara-detectar-malware/>
- [37] *Reglas SIGMA*,  
<HTTPS://CIBERSEGURIDAD.BLOG/>.  
Recuperado a 08/07/2021,  
de <https://ciberseguridad.blog/que-son-las-reglas-sigma-y-por-que-las-necesitas/>
- [38] *Wazuh*,  
<HTTPS://WAZUH.COM/>.  
Recuperado a 08/07/2021,  
de <https://wazuh.com/>
- [39] *AlienVault OSSIM*,  
<HTTPS://CYBERSECURITY.ATT.COM/>.  
Recuperado a 08/07/2021,  
de <https://cybersecurity.att.com/products/ossim>
- [40] *CENTOS*,  
<HTTPS://WWW.IONOS.ES>.  
Recuperado a 08/07/2021,  
de <https://www.ionos.es/digitalguide/servidores/know-how/que-es-centos-versiones-y-requisitos-del-sistema/>  
[https://www.stackscale.com/es/blog/centos-linux/#Caracteristicas\\_principales\\_de\\_CentOS](https://www.stackscale.com/es/blog/centos-linux/#Caracteristicas_principales_de_CentOS)
- [41] *Security Onion*,  
<HTTPS://SECURITYONIONSOLUTIONS.COM/>.  
Recuperado a 08/07/2021,  
de <https://securityonionsolutions.com/>
- [42] *Ubuntu*,  
<HTTPS://UBUNTU.COM/DOWNLOAD>.  
Recuperado a 08/07/2021,  
de <https://ubuntu.com/download>
- [43] *Windows10*,  
<HTTPS://ES.WIKIPEDIA.ORG/>.  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/Windows\\_10](https://es.wikipedia.org/wiki/Windows_10)
- [44] *ELK Stack*,  
<HTTPS://WWW.ELASTIC.CO/>.

Recuperado a 08/07/2021,  
de <https://www.elastic.co/es/what-is/elk-stack>

- [45] *Python*,  
<HTTPS://WWW.PYTHON.ORG/>.  
Recuperado a 08/07/2021,  
de <https://www.python.org/>
- [46] *MongoDB*,  
<HTTPS://WWW.MONGODB.COM/ES>.  
Recuperado a 08/07/2021,  
de <https://www.mongodb.com/es>  
<https://es.overleaf.com/project>
- [47] *Overleaf Latex*,  
<HTTPS://ES.OVERLEAF.COM>.  
Recuperado a 08/07/2021,  
de <https://es.overleaf.com>
- [48] *Bitbucket*,  
<HTTPS://BITBUCKET.ORG/PRODUCT/>.  
Recuperado a 08/07/2021,  
de <https://bitbucket.org/product/>
- [49] *SublimeText*,  
<HTTPS://WWW.SUBLIMETEXT.COM/>.  
Recuperado a 08/07/2021,  
de <https://www.sublimetext.com/>
- [50] *VirtualBox*,  
<HTTPS://WWW.VIRTUALBOX.ORG/>.  
Recuperado a 08/07/2021,  
de <https://www.virtualbox.org/>
- [51] *Syslog-*ng**,  
<HTTPS://WWW.SYSLOG-NG.COM/>.  
Recuperado a 08/07/2021,  
de <https://www.syslog-ng.com/>
- [52] *Pulled Pork*,  
<HTTPS://GITHUB.COM/SHIRKDOG/PULLEDPORK>.  
Recuperado a 08/07/2021,  
de <https://github.com/shirkdog/pulledpork>
- [53] *Ventajas Scrum*,  
<HTTPS://PROYECTOSAGILES.ORG/>.  
Recuperado a 08/07/2021,  
de <https://proyectosagiles.org/beneficios-de-scrum/#gestion-riesgos>

- [54] *Roles Scrum*,  
HTTPS://INTEGRAIT.COM.MX/.  
Recuperado a 08/07/2021,  
de <https://integrait.com.mx/blog/roles-de-scrum/>
- [55] *Eventos Scrum*,  
HTTPS://WWW.SCRUMMANAGER.NET/.  
Recuperado a 08/07/2021,  
de <https://www.scrummanager.net/bok/index.php?title=Eventos>
- [56] *Salario medio Ingeniero Informático*,  
HTTPS://WWW.INDEED.ES/.  
Recuperado a 08/07/2021,  
de <https://www.indeed.es/salaries/ingeniero-inform%C3%A1tico-Salaries?period=monthly>
- [57] *Método Delphi*,  
ES.WIKIPEDIA.ORG.  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/M%C3%A9todo\\_Delphi](https://es.wikipedia.org/wiki/M%C3%A9todo_Delphi)
- [58] *Análisis SWOT*,  
PMFARMA.ES.  
Recuperado a 08/07/2021,  
de <http://pmfarma.es/articulos/1253-analisis-swot.html>
- [59] *Diagrama de Ishikawa*,  
ES.WIKIPEDIA.ORG.  
Recuperado a 08/07/2021,  
de [https://es.wikipedia.org/wiki/Diagrama\\_de\\_Ishikawa](https://es.wikipedia.org/wiki/Diagrama_de_Ishikawa)
- [60] *GitHub*,  
HTTPS://GITHUB.COM/.  
Recuperado a 08/07/2021
- [61] *Descarga Wazuh*,  
HTTPS://DOCUMENTATION.WAZUH.COM/.  
Recuperado a 08/07/2021  
de <https://documentation.wazuh.com/current/getting-started/index.html>
- [62] *Información TFG Training For Trainers Gonzalo Herreros Diezhandino*

