SIEM en ELK i Kibana

Pere Garcia Sanahuja

Màster en Ciberseguretat i Privadesa
Anàlisi de dades

Tutor/a de TF
Joan Caparrós Ramírez
Professor/a responsable de l'assignatura
Andreu Pere Isern Deyà

10 de gener de 2023

Universitat Oberta de Catalunya

FITXA DEL TREBALL FINAL

Títol del treball:	*SIEM en ELK i Kibana*
Nom de l'autor:	*Pere Garcia Sanahuja*
Nom del consultor/a:	*Joan Caparrós Ramírez*
Nom del PRA:	*Andreu Pere Isern Deyà*
Data de lliurament (mm/aaaa):	*01/2023*
Titulació o programa:	*Màster en Ciberseguretat i Privadesa*
Àrea del Treball Final:	*Anàlisi de dades*
Idioma del treball:	*Català*
Paraules clau	*SIEM, ELK, IOC*

Resum del Treball

La finalitat del treball és implementar un SIEM mitjançant la pila ELK de codi obert. Per fer-ho s'ha fet servir l'eina Docker que permet virtualitzar els serveis. Un cop implementada la pila i per tal de simular un funcionament real, s'han afegit serveis que generen dades i d'altres que permeten capturar-les, d'aquesta manera la implementació és totalment funcional i amb les dades es podrà generar informació útil pel SIEM.

Aquesta informació permetrà a l'equip de seguretat respondre a les amenaces de manera més eficient localitzant-les un cop es produeixen o obtenint possibles evidències del que hagi pogut succeir.

Per dur a terme la implementació s'ha optat per l'ús d'una metodologia àgil que permeti més facilitat a l'hora d'adaptar-se a possibles canvis i els resultats d'aquesta han permès implementar el sistema en la seva totalitat en un entorn controlat, però escalable a altres entorns.

Finalment, s'ha pogut comprovar el funcionament de tot el SIEM i el seu funcionament amb les dades capturades.

Abstract

The target of this work is to implement a SIEM using the open source ELK stack. To do this, the Docker tool has been used, which allows the services to be virtualised. Once the stack has been implemented and in order to simulate real operation, services have been added. Some of them generate data and others allow data to be captured, in this way the implementation is fully functional and the data can be used to generate useful information for the SIEM.

This information will allow the security team to respond to threats as efficiently as possible while they are occurring or to obtain possible evidence of what has happened.

In order to carry out the implementation, an agile methodology has been chosen to allow an easier adaptation to possible changes, and the results have allowed the system to be implemented in its entirety in a controlled environment, but scalable to others.

Finally, it has been possible to test the functions of the entire SIEM and its operation with the captured data.

Índex

Llista de figures

1. Introducció

1.1. Context i justificació del Treball

Actualment les amenaces i atacs als sistemes d'informació són molts freqüents i les empreses que ho pateixen, saben de bona mà la necessitat que hi ha de protegir-se d'aquests atacs i de desenvolupar una bona estratègia per tal de sortir-ne un cop han sigut atacades.

Aquests atacs i amenaces no es poden preveure, però si es pot minimitzar el seu impacte o fins i tot mitigar-lo mitjançant un SIEM que permeti donar més visibilitat al centre de resposta d'incidents, ajudant-los a prendre decisions durant el transcurs d'un atac o per aconseguir respostes un cop aquest ha estat detectat.

Un SIEM és un sistema que gestiona la informació i els esdeveniments de seguretat. S'utilitza per capturar dades i analitzar-les, ja sigui en temps real o no, per posteriorment extreure conclusions o fer-ne informes per l'equip de TI. D'aquesta manera, l'equip pot monitoritzar diferents fonts de dades sense necessitat de tenir personal en cada una de les fonts i alhora, obtindre resultats per fer més fàcil la presa de decisions.

Les fonts de dades solen ser registres d'aplicacions, paquets de xarxa o fins i tot esdeveniments que s'envien expressament al sistema per generar una alerta.

Aquest tipus de sistemes són costos d'implementar i de mantenir, ja que involucren a moltes parts d'una companyia i la implementació inicial sòl durar molt de temps. A més, si les empreses que l'instal·len no tenen clar el seu funcionament, el sistema pot quedar-se sense recursos o directament ser eliminat.

Per aquest motiu, desenvolupar un SIEM open source[1] que sigui relativament senzill d'instal·lar, mantenir i alimentar, pot ser una solució viable per algunes petites empreses que necessitin aquest sistema o per altres que no vulguin destinar-hi gaires recursos d'entrada.

Si bé és cert que hi ha algun SIEM gratuït, cap et permet implementar-lo directament als propis sistemes ni personalitzar-lo gaire, ja que els sistemes actuals gratuïts tenen versions molt limitades de les seves funcions i no permeten modificar-les.

Finalment, l'objectiu principal del treball és utilitzar la pila ELK com a base per implementar el SIEM i aprofitar aquesta pila per poder capturar dades de diferents fonts utilitzant les pròpies aplicacions que ofereix.

1.2. Objectius del Treball

[1] S'anomena open source a tot aquell software compost de codi obert.

L’objectiu final del treball és implementar un SIEM amb les eines “open source” de la pila ELK. Per arribar a aquest objectiu farà falta una base de coneixement d’aquesta pila i entendre el funcionament d’una plataforma SIEM.

Per aquest motiu, els objectius del treball els podem dividir en tres categories:

Objectius de coneixement:

- Entendre el funcionament d’un SIEM.
- Aprendre a utilitzar les eines del stack ELK.
- Investigar els diferents formats disponibles per compartir indicadors de compromís (IOC).
- Gestionar diferents fonts d’ingesta de dades per alimentar el SIEM.
- Investigar com analitzar i utilitzar la informació extreta de les dades per ajudar a l’equip de resposta d’incidents.

Objectius del sistema SIEM:

- Gestionar la infraestructura i requeriments d’un SIEM.
- Monitoritzar i gestionar la infraestructura des del punt de vista d’un SIEM.
- Aconseguir analitzar les dades utilitzant indicadors de compromís (IOC).

Objectius del TFM:

- Memòria escrita del TFM.
- Presentació final en vídeo.
- Repositori de codi font per poder executar el SIEM.

Finalment, en el repositori de codi font relacionat amb el treball, hi trobarem l’entregable que permetrà instal·lar el SIEM i aquest serà capaç de:

- Visualitzar la informació dels equips disponibles a la xarxa.
- Donar servei amb un cost molt reduït (infraestructura).
- Agilitzar la presa de decisions de l’equip de resposta.

1.3. Enfocament i mètode seguit

L’enfoc del treball consistirà en implementar el SIEM utilitzant la pila ELK i per fer-ho haurem de realitzar una primera part teòrica que consistirà en la cerca d’informació tant de la pila com del sistema SIEM, i una segona part més pràctica on s’implementarà el sistema i es farà una demostració.

En la primera part, s’obtindrà coneixement de tot el ecosistema que conforma un SIEM, les eines utilitzades per construir-lo i alimentar-lo. A la segona part, es configurarà el SIEM i s’utilitzaran sistemes externs com ara servidors NGINX o llocs web per poder alimentar-lo. Com explorar totes les opcions i obtindre tot el coneixement desitjat sobre els àmbits d’aquest treball implicaria moltes hores, hi hauran parts que es duran a terme conjuntament, és a dir, un cop obtingut el coneixement, s’utilitzarà per fer la implantació.

Un cop conegut l'enfocament del treball, aquest s'ha de desenvolupar com a projecte i del desenvolupament es podran crear tasques i fites, que en el cas del treball, seran les entregues PAC.

Per dur a terme aquestes tasques i arribar a les fites, s'utilitzarà la metodologia Agile SCRUM. Aquesta metodologia permet definir sprints[2] que contenen diferents tasques i el resultat de cada sprint és una fita que inclourà un o varis artefactes. Utilitzar SCRUM ajudarà a reconduir les tasques en cas de trobar algun inconvenient durant l'execució de cada una d'elles.

Finalment l'autor del treball té experiència com a desenvolupador i arquitectura del software, fet que permetrà agilitzar algunes tasques de la part pràctica del treball al tenir coneixements previs del funcionament i instal·lació de la pila ELK.

1.4. Impacte en sostenibilitat, ètic-social i de diversitat

Degut a la naturalesa del treball, implementar un SIEM comporta que aquest analitzi diferents fonts d'informació. Aquestes fonts poden ser públiques o bé privades i en ambos casos poden contenir informació confidencial que pot arribar a identificar persones reals. Per aquest motiu s'identifica un impacte negatiu derivat de la implementació del projecte en un entorn real, la identificació d'usuaris reals.

Degut al impacte, es plantegen les següents mesures preventives:

- Anonimitzar les dades de caràcter privat o que continguin informació que pugui identificar a un usuari.
- Xifrar totes les dades privades utilitzant mecanismes que n'extreguin el hash[3] i enviant-les al SIEM posteriorment.
- Evitar l'enviament de les dades sensibles al SIEM utilitzant alguna eina que permeti evitar-les.
- Tractar-les en el moment de la ingesta al SIEM, posant una capa de software per sobre que permeti detectar aquest tipus de dades i les faci opaques al sistema SIEM per tal que cap usuari malintencionat les pugui veure o fer-ne qualsevol ús.

1.5. Planificació del Treball

El treball ha estat planificat per dur-se a terme dins les dates establertes per la UOC, tenint en compte dies festius tant nacionals com cap de setmana, nadal, etc.

El total d'hores dedicades al treball es poden veure desglossades per tasques a continuació:

---

[2] Un sprint és un concepte de la metodologia àgil que fa referència a una finestra de temps en la qual s'està desenvolupant un projecte.

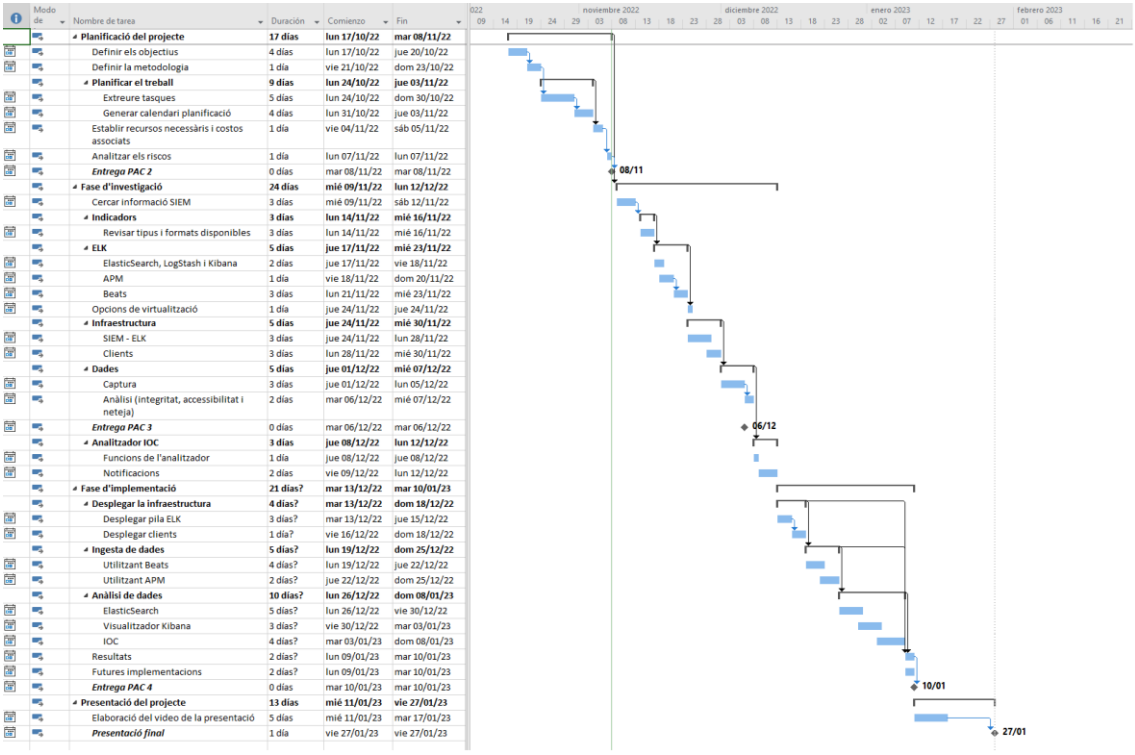
[3] El hash és una funció matemàtica que converteix blocs de bytes en una cadena alfanumèrica de longitud fixe.

Tasca	Hores dedicades
Planificació del projecte	**60**
Planificar el treball	45
Recursos i riscos	15
Fase d'investigació	**120**
SIEM	20
Indicadors	20
ELK	20
Infraestructura	20
Dades	20
Analitzador IOC	20
Fase d'implementació	**90**
Desplegament de la infraestructura	20
Ingesta de dades	20
Anàlisis de dades	40
Resultats	10
Presentació final	**30**
Revisió de la memòria	20
Presentació audiovisual	10
Total:	**300**

A més, s'exposa una taula detallada amb totes les tasques:

Modo de	Nombre de tarea	Trabajo	Duración	Comienzo	Fin	Hito
	Planificació del projecte	**0 horas**	**17 días**	**lun 17/10/22**	**nar 08/11/22**	**No**
	Definir els objectius	0 horas	4 días	lun 17/10/22	jue 20/10/22	No
	Definir la metodologia	0 horas	1 día	vie 21/10/22	om 23/10/22	No
	Planificar el treball	**0 horas**	**9 días**	**lun 24/10/22**	**jue 03/11/22**	**No**
	Extreure tasques	0 horas	5 días	lun 24/10/22	om 30/10/22	No
	Generar calendari planificació	0 horas	4 días	lun 31/10/22	jue 03/11/22	No
	Establir recursos necessàris i costos associats	0 horas	1 día	vie 04/11/22	sáb 05/11/22	No
	Analitzar els riscos	0 horas	1 día	lun 07/11/22	lun 07/11/22	No
	Entrega PAC 2	0 horas	0 días	nar 08/11/22	nar 08/11/22	Sí
	Fase d'investigació	**0 horas**	**24 días**	**nié 09/11/22**	**lun 12/12/22**	**No**
	Cercar informació SIEM	0 horas	3 días	nié 09/11/22	sáb 12/11/22	No
	Indicadors	**0 horas**	**3 días**	**lun 14/11/22**	**nié 16/11/22**	**No**
	Revisar tipus i formats disponibles	0 horas	3 días	lun 14/11/22	nié 16/11/22	No
	ELK	**0 horas**	**5 días**	**jue 17/11/22**	**nié 23/11/22**	**No**
	ElasticSearch, LogStash i Kibana	0 horas	2 días	jue 17/11/22	vie 18/11/22	No
	APM	0 horas	1 día	vie 18/11/22	om 20/11/22	No
	Beats	0 horas	3 días	lun 21/11/22	nié 23/11/22	No
	Opcions de virtualització	0 horas	1 día	jue 24/11/22	jue 24/11/22	No
	Infraestructura	**0 horas**	**5 días**	**jue 24/11/22**	**nié 30/11/22**	**No**
	SIEM - ELK	0 horas	3 días	jue 24/11/22	lun 28/11/22	No
	Clients	0 horas	3 días	lun 28/11/22	nié 30/11/22	No
	Dades	**0 horas**	**5 días**	**jue 01/12/22**	**nié 07/12/22**	**No**
	Captura	0 horas	3 días	jue 01/12/22	lun 05/12/22	No
	Anàlisi (integritat, accessibilitat i neteja)	0 horas	2 días	nar 06/12/22	nié 07/12/22	No
	Entrega PAC 3	0 horas	0 días	nar 06/12/22	nar 06/12/22	Sí
	Analitzador IOC	**0 horas**	**3 días**	**jue 08/12/22**	**lun 12/12/22**	**No**
	Funcions de l'analitzador	0 horas	1 día	jue 08/12/22	jue 08/12/22	No
	Notificacions	0 horas	2 días	vie 09/12/22	lun 12/12/22	No
	Fase d'implementació	**0 horas**	**21 días?**	**nar 13/12/22**	**nar 10/01/23**	**No**
	Desplegar la infraestructura	**0 horas**	**4 días?**	**nar 13/12/22**	**om 18/12/22**	**No**
	Desplegar pila ELK	0 horas	3 días?	nar 13/12/22	jue 15/12/22	No
	Desplegar clients	0 horas	1 día?	vie 16/12/22	om 18/12/22	No
	Ingesta de dades	**0 horas**	**5 días?**	**lun 19/12/22**	**om 25/12/22**	**No**
	Utilitzant Beats	0 horas	4 días?	lun 19/12/22	jue 22/12/22	No
	Utilitzant APM	0 horas	2 días?	jue 22/12/22	om 25/12/22	No
	Anàlisi de dades	**0 horas**	**10 días?**	**lun 26/12/22**	**om 08/01/23**	**No**
	ElasticSearch	0 horas	5 días?	lun 26/12/22	vie 30/12/22	No
	Visualitzador Kibana	0 horas	3 días?	vie 30/12/22	nar 03/01/23	No
	IOC	0 horas	4 días?	nar 03/01/23	om 08/01/23	No
	Resultats	0 horas	2 días?	lun 09/01/23	nar 10/01/23	No
	Futures implementacions	0 horas	2 días?	lun 09/01/23	nar 10/01/23	No
	Entrega PAC 4	0 horas	0 días	nar 10/01/23	nar 10/01/23	Sí
	Presentació del projecte	**0 horas**	**13 días**	**nié 11/01/23**	**vie 27/01/23**	**No**
	Elaboració del video de la presentació	0 horas	5 días	nié 11/01/23	nar 17/01/23	No
	Presentació final	0 horas	1 día	vie 27/01/23	vie 27/01/23	Sí

I finalment podem veure la planificació mitjançant un diagrama de Gantt del projecte:



Modo de	Nombre de tarea	Duración	Comienzo	Fin
	Planificació del projecte	**17 días**	**lun 17/10/22**	**mar 08/11/22**
	Definir els objectius	4 días	lun 17/10/22	jue 20/10/22
	Definir la metodologia	1 día	vie 21/10/22	dom 23/10/22
	Planificar el treball	**9 días**	**lun 24/10/22**	**jue 03/11/22**
	Extreure tasques	5 días	lun 24/10/22	dom 30/10/22
	Generar calendari planificació	4 días	lun 31/10/22	jue 03/11/22
	Establir recursos necessàris i costos associats	1 día	vie 04/11/22	sáb 05/11/22
	Analitzar els riscos	1 día	lun 07/11/22	lun 07/11/22
	Entrega PAC 2	0 días	mar 08/11/22	mar 08/11/22
	Fase d'investigació	**24 días**	**mié 09/11/22**	**lun 12/12/22**
	Cercar informació SIEM	3 días	mié 09/11/22	sáb 12/11/22
	Indicadors	**3 días**	**lun 14/11/22**	**mié 16/11/22**
	Revisar tipus i formats disponibles	3 días	lun 14/11/22	mié 16/11/22
	ELK	**5 días**	**jue 17/11/22**	**mié 23/11/22**
	ElasticSearch, LogStash i Kibana	2 días	jue 17/11/22	vie 18/11/22
	APM	1 día	vie 18/11/22	dom 20/11/22
	Beats	3 días	lun 21/11/22	mié 23/11/22
	Opcions de virtualització	1 día	jue 24/11/22	jue 24/11/22
	Infraestructura	**5 días**	**jue 24/11/22**	**mié 30/11/22**
	SIEM - ELK	3 días	jue 24/11/22	lun 28/11/22
	Clients	3 días	lun 28/11/22	mié 30/11/22
	Dades	**5 días**	**jue 01/12/22**	**mié 07/12/22**
	Captura	3 días	jue 01/12/22	lun 05/12/22
	Anàlisi (integritat, accessibilitat i neteja)	2 días	mar 06/12/22	mié 07/12/22
	Entrega PAC 3	0 días	mar 06/12/22	mar 06/12/22
	Analitzador IOC	**3 días**	**jue 08/12/22**	**lun 12/12/22**
	Funcions de l'analitzador	1 día	jue 08/12/22	jue 08/12/22
	Notificacions	2 días	vie 09/12/22	lun 12/12/22
	Fase d'implementació	**21 días?**	**mar 13/12/22**	**mar 10/01/23**
	Desplegar la infraestructura	**4 días?**	**mar 13/12/22**	**dom 18/12/22**
	Desplegar pila ELK	3 días?	mar 13/12/22	jue 15/12/22
	Desplegar clients	1 día?	vie 16/12/22	dom 18/12/22
	Ingesta de dades	**5 días?**	**lun 19/12/22**	**dom 25/12/22**
	Utilitzant Beats	4 días?	lun 19/12/22	jue 22/12/22
	Utilitzant APM	2 días?	jue 22/12/22	dom 25/12/22
	Anàlisi de dades	**10 días?**	**lun 26/12/22**	**dom 08/01/23**
	ElasticSearch	5 días?	lun 26/12/22	vie 30/12/22
	Visualitzador Kibana	3 días?	vie 30/12/22	mar 03/01/23
	IOC	4 días?	mar 03/01/23	dom 08/01/23
	Resultats	2 días?	lun 09/01/23	mar 10/01/23
	Futures implementacions	2 días?	lun 09/01/23	mar 10/01/23
	Entrega PAC 4	0 días	mar 10/01/23	mar 10/01/23
	Presentació del projecte	**13 días**	**mié 11/01/23**	**vie 27/01/23**
	Elaboració del video de la presentació	5 días	mié 11/01/23	mar 17/01/23
	Presentació final	1 día	vie 27/01/23	vie 27/01/23

1.6. Recursos necessaris i pressupost del projecte

Degut a que la implementació del sistema serà feta sobre contenidors, tenim la opció de fer servir un ordinador, ja sigui personal o un servidor físic, o utilitzar un proveïdor de "Cloud computing". En el cas de fer la implementació a un d'aquests proveïdors, haurem de tenir en compte els costos associats a la infraestructura[4].

Un cop triada la ubicació de la instal·lació, la implementació del SIEM necessitarà com a mínim un ordinador o servidor que disposi de 8 GB de memòria RAM, una CPU de 4 nuclis de 64 bits que tingui suport natiu per tasques de virtualització i 50GB d'espai d'emmagatzematge.

En el cas que la implementació del SIEM es dugui a terme sobre el sistema operatiu Windows, serà necessari activar el sistema "WSL 2" i el suport a la virtualització a nivell de "BIOS". A més, es necessitarà accés a internet per poder descarregar les imatges base dels contenidors.

A continuació, s'exposa un exemple de costos del projecte utilitzant un ordinador personal:

Recurs	Descripció	Cost
Ordinador	Plataforma on implementar el SIEM i els clients de prova	500€
Connexió a internet	Necessària per descarregar les imatges dels contenidors	20€/mes

[4] Calculadora de preus dels serveis disponibles al proveïdor cloud AWS https://calculator.aws/

1.7. Estat de l'art

Fent una cerca per internet s'ha revisat el funcionament de la pila ELK i com aquesta pot fer-se servir com a solució SIEM. Actualment existeixen altres solucions SIEM d'altres proveïdors, alguns amb llicència i d'altres sense. Més endavant, durant la fase d'investigació se'n nombraran algunes, però donat que aquest treball correspon únicament a la implementació del SIEM mitjançant la pila ELK, no s'entrarà en molt detall.

Respecte a la implementació del SIEM, és a dir, la motivació del treball, ve donada per l'interès d'aprendre com funciona un sistema d'aquestes dimensions i com pot ser implementat amb software de codi obert.

1.8. Anàlisi de riscos

El projecte té en compte els següents riscos a l'hora de dur-se a terme i tots tenen una mitigació que s'ajusta als temps d'entrega del treball:

1.8.1. Falta de temps per acabar el treball

Risc alt

Definició
És possible que durant el temps que dura el desenvolupament del treball pugui sorgir algun imprevist que faci trontollar la previsió inicial. Aquest podria ser un problema de salut, laboral o fins i tot familiar, que impedeixi continuar amb la programació del calendari.

Mitigació
S'ha de tenir les tasques realitzades segons la planificació del calendari i en cas de no aconseguir-ho, s'haurà de reduir algun objectiu o part del projecte per tal de poder arribar al conjunt a temps.

1.8.2. Problemes amb la implementació del sistema SIEM

Risc alt

Definició
Durant la fase d'implementació és possible que aparegui algun inconvenient amb el funcionament de la pila ELK o alguna incompatibilitat del sistema que no permeti dur a terme la implementació correctament.

Mitigació
Per mitigar aquest risc, es documentarà el problema i s'intentarà resoldre utilitzant alternatives. Si bé és cert que l'aparició d'un problema durant la implementació és beneficiós pel projecte al cobrir possibles casos d'ús, donat que el temps és ajustat el risc de sortir-se de les dates del calendari és alt.

1.8.3. Treball sobredimensionat

Risc alt

Definició

És possible que a mida que es vagin resolen les tasques, el projecte vagi agafant unes dimensions molt més àmplies del que s'havia previst en un principi durant la fase de planificació. El fet de redimensionar el projecte durant l'execució pot ser molt perjudicial ja que allargarà el calendari del mateix i el farà incomplir.

Mitigació

És important centrar-se en les dates del calendari de tasques, ja que no fer-ho i destinar més temps del establert a algunes tasques, pot portar a una falta de temps per resoldre'n d'altres. En cas de sobredimensió del projecte, s'ha de fer l'exercici de resumir al màxim la part en qüestió i seguir amb la previsió per tal d'evitar el risc.

1.8.4. Incompatibilitat del format de les dades amb els mètodes de captura pel SIEM

Risc mitjà

Definició

Un com implementat el SIEM, s'ha d'alimentar de dades per poder dur a terme la seva funcionalitat. Aquestes es capturen utilitzant diferents mètodes o serveis com ara un APM o els paquets Beats de Elastic. Tots ells tenen una capa de personalització que ajuda a llegir les dades en diferents formats, però cada aplicació és diferent i no totes es desenvolupen tenint en compte qui llegirà els registres un cop estigui funcionant.

Mitigació

En el cas de sorgir la situació, s'intentarà resoldre mitjançant alguna aplicació o es reeditaran per tal de poder alimentar el SIEM, ja que resoldre diferents formats de dades o unificar un flux de captura de dades queda fora de l'àmbit del projecte.

1.8.5. Utilitzant un proveïdor cloud, pèrdua de les credencials o errors de configuració de la infraestructura

Risc baix

Definició

El SIEM està pensat per poder instal·lar-se a qualsevol ordinador que compleixi les necessitats de hardware del sistema, però si durant el desenvolupament del treball apareix algun problema amb el meu ordinador personal, s'haurà de traspassar tota la part d'implementació al cloud per poder seguir endavant. Això provocarà la implementació de la infraestructura al proveïdor escollit i el manteniment d'aquesta.

En el cas que hi haguessin errors o una pèrdua de les credencials, el projecte patiria un retràs sobre les tasques previstes i podria fer trontollar l'entrega final.

Mitigació

Una solució és tenir un altre ordinador a mà que compleixi les necessitats del SIEM. En cas de no tenir-lo, sempre es pot optar per utilitzar alguna eina IaC[5] com Terraform[6] per tal de poder reinstal·lar tot l'entorn del proveïdor cloud de manera senzilla i eficient.

2. Fase d'investigació

La primera part del treball consistirà en una investigació sobre tots els conceptes relacionats amb un SIEM i quins són els requeriments necessaris per poder implementar-lo correctament utilitzant la pila ELK.

2.1. Definició d'un sistema SIEM

SIEM és l'acrònim en anglès de "Security Information and Event Management", és a dir, un sistema que gestiona tant els esdeveniments com la informació de seguretat del seu entorn dins una xarxa en temps real.

El propòsit del sistema és donar visibilitat i resposta als responsables de seguretat per tal que aquests puguin prendre decisions ràpidament a l'hora de detectar i resoldre amenaces informàtiques que puguin estar succeint a qualsevol dispositiu de la xarxa. A més, vist des del punt de la organització, el sistema garanteix uns mínims de seguretat i fortalesa respecte les possibles incidències.

Generalment aquests tipus de sistemes s'utilitzen dins d'un àmbit empresarial, per tal que el SIEM tingui un control absolut dels esdeveniments de l'organització i pugui respondre d'una manera més concreta a qualsevol amenaça. El fet de poder controlar tot l'ecosistema informàtic l'ajuda a poder aprendre els comportaments dels sistemes als que està connectat i a preveure certes situacions o patrons fora de la normalitat.

El sistema SIEM és l'evolució de dues tecnologies que anteriorment tenien funcions clarament diferents, el SEM i el SIM.

2.1.1. SEM

SEM és l'acrònim en anglès de "Security Events Management", és a dir, gestor d'esdeveniments de seguretat. La seva funció és detectar patrons irregulars en temps real, de tal manera que es puguin monitoritzar i emetre notificacions en cas oportú.

[5] IaC és l'acrònim de Infrastructure as Code
[6] Terraform IaC https://www.terraform.io/

Els sistemes SEM agrupaven tots els registres de tots aquells dispositius que els hi enviaven dades, però el seu problema era que els administrador o desenvolupadors d'aplicacions no tenien clar quin era el format necessari per enviar els esdeveniments amb tota la informació. Això va provocar que s'intentés arribar a un consens per desenvolupar un marc de treball i un protocol d'enviament de les dades genèric, però finalment no es va dur a terme.

2.1.2. SIM

SIM també és un acrònim en anglès, de "Security Information Management", que vindria a ser un gestor de la seguretat de la informació.

La seva funció és guardar dades per poder analitzar-les. Aquestes dades corresponen a registres de seguretat de diversos dispositius com ara dispositius d'accés biomètric, accessos a documents o dades dins un servidor, etc.

Tots els registres són enviats al SIM per tal que aquest els pugui mostrar mitjançant informes al responsable de seguretat i en cas necessari poder generar un incident de seguretat.

Cal recalcar que un sistema SIM només s'encarrega de descobrir possibles incidents de seguretat utilitzant l'anàlisi de les dades rebudes, no de bloquejar-ne l'accés o gestionar-lo.

2.1.3. Dades

Com s'ha exposat, el sistema SIEM utilitza l'anàlisi de dades en temps real per donar resposta a incidents o visibilitat al equip de seguretat. Per poder fer-ho necessita ingerir dades i aquestes poden provenir de diferents fonts:

- Servidors
- Aplicacions
- Analítiques de sistemes
- Sistemes d'autorització
- Serveis externs
- Dispositius físics

Tots aquests dispositius tenen en comú que guarden les seves execucions o els registres en un suport informàtic. Ho poden fer a un arxiu o enviant-los a una aplicació que sigui capaç d'utilitzar-los, però sempre se'n pot fer un ús.

El SIEM centralitza tots aquests registres en un una base de dades, la qual posteriorment utilitza per analitzar-les.

2.1.4. Característiques

Un cop explicat el funcionament del SIEM, podem identificar a grans trets les següents característiques:

- Monitoritzar les potencials amenaces en temps real i de manera centralitzada
- Emetre notificacions per alertar l'equip de seguretat
- Complir amb la normativa vigent en matèria de protecció de dades i seguretat (important per la certificació ISO 27001)
- Aportar més informació a l'hora de la detecció d'incidents per poder respondre d'una manera més adequada i eficient

A més d'aquestes característiques, en el cas d'implementar un SIEM en una organització obtindríem les següents avantatges:

- Una base de coneixement per conèixer en tot moment el que es fa dins l'organització
- Evitar possibles atacs o minimitzar-ne els riscs amb una resposta en temps real
- Reduir costos al automatitzar processos d'anàlisi

Com es pot veure, un SIEM té una visió global de tot el que succeeix dins una xarxa o una organització. Anteriorment a l'apartat d'implicacions ètiques, es feia èmfasis en un aspecte molt rellevant un cop implementat un SIEM dins una organització, la identificació dels usuaris i la seva activitat. Aquest aspecte s'ha de tenir en compte ja que la privacitat de les dades ha de ser un dels punts a tenir en compte per no guardar ni mostrar informació que perjudiqui a algun individu.

2.1.5. Software disponible al mercat

Actualment existeixen diverses eines que permet la implementació d'un SIEM. Algunes són de codi obert i permet fer la instal·lació i configuració manualment, però sempre oferint una versió de pagament que normalment inclou assistència tècnica, ús del software com a SaaS[7] i alguna funció extra que no és inclosa dins la versió gratuïta.

Algunes d'aquestes opcions són:

- Pila ELK (gratuïta i opció amb llicència de pagament)
- IBM QRadar Security (pagament)
- AlienVault OSSIM (gratuïta i opció amb llicència de pagament)
- OSSEC, fa ús de la pila ELK internament (gratuïta i opció amb llicència de pagament)
- Wazuh (gratuïta)
- Prometheus (gratuïta)
- TheHive – StrangeBee (gratuïta i opció amb llicència de pagament)

[7] SaaS és l'acrònim de Software as a Service

Cal dir que algunes d'aquestes eines no conformen un SIEM per si soles. Per exemple, la pila ELK que s'implementarà en aquest treball està composta de tres eines (Elasticsearch, Logstash i Kibana) que poden ser útils per separat en altres aplicacions informàtiques.

2.2. Indicadors

Per tal de poder garantir la seguretat d'un sistema, es necessiten indicadors que actuïn a mode d'intel·ligència o coneixement per poder identificar i actuar sobre una amenaça. La implementació del SIEM passa per entendre correctament aquests indicadors, ja que són la font d'informació indispensable per poder entendre el comportament i el procés pel qual s'ha produït una amenaça.

Existeixen dos tipus, els indicadors de compromís i els d'atac. Cada un té un objectiu ben definit i una utilitat en concret, i el SIEM permet generar el coneixement que els conforma.

2.2.1. De compromís – IOC

Un indicador de compromís és una evidència d'un possible atac, obtingut analitzant els esdeveniments històrics. La informació extreta permet descobrir possibles patrons o processos que han desencadenat en un esdeveniment maliciós i un cop ha succeït, es recapitula la informació i es defineix com un indicador de compromís.

L'indicador ha de definir quins són els motius que han compromès el sistema i aquests poden ser canvis en el registre del sistema, aparició d'arxius desconeguts, processos desconeguts, peticions rebudes o enviades a llocs web desconeguts, entre d'altres.

2.2.2. D'atac - IOA

Els indicadors d'atac són aquells que permeten descriure l'atac utilitzant la informació que succeeix en temps real. A diferència dels indicadors de compromís, no utilitzen la informació històrica per generar coneixement, sinó que ells mateixos extreuen el coneixement segons la situació.

Per exemple, si dos servidors estan connectats mitjançant el port 443 i mentre es monitoritza els registres dels accessos als dos servidors, es detecta un intent d'accés al port 444, es produeix un indicador d'atac, el qual revela en temps real que s'està utilitzant el port 444 en comptes del 443. Un cop vist l'indicador, l'equip de contingència ho revisarà i actuarà en conseqüència.

Altres exemples d'indicadors d'atac:

- Peticions a destins fora de la xarxa
- Servidors públics accedint a serveis interns de la xarxa
- Tràfic alt en horaris de lleure

- Escaneigs de xarxa des de serveis interns fora dels habituals
- Inicis de sessió des de diferents ubicacions
- Moltes alarmes consecutives d'ús de recursos d'un mateix node o servei

La manera que tenen els indicadors d'atac de localitzar una possible amenaça és revisar totes les activitats i registres de la xarxa i nodes que la componen en temps real. D'aquesta manera pot preveure certs atacs i bloquejar-los abans que succeeixin.

Per detectar les amenaces, els indicadors d'atac utilitzen unes regles definides que els permeten classificar els registres que van rebent dins d'uns llindars i si aquests són sobrepassats fora dels seus límits, el indicador genera la alerta. En el cas del exemple del port 443, la regla seria que aparegués qualsevol registre d'un servidor a l'altre utilitzant un port diferent del 443.

2.3. ELK

La implementació del SIEM es farà mitjançant la pila ELK[8], que inclou les eines Elasticsearch, Logstash i Kibana. Com ja s'ha comentat en altres seccions, l'acrònim ELK el conformen les inicials de les tres eines de la pila.

En aquest apartat s'exposarà cada una de les eines que es faran servir per implementar el SIEM, algunes de les quals no formen part de la pila ELK directament, però si del seu ecosistema.

2.3.1. El stack

Originalment la pila la conformaven l'eina Elasticsearch i Kibana. Més tard s'hi va afegir Logstash que permet capturar dades per enviar-les a Elasticsearch i així englobar tot el flux de captura i anàlisi de dades. Aquest flux està compost d'una part de captura, una de ingesta i una última de visualització i anàlisi. Com es pot observar l'última part conté dues accions, visualitzar i analitzar, això és degut a que l'eina Kibana permet visualitzar les dades i a la vegada utilitzar aquest visualitzador com un "dashboard" on poder filtrar i analitzar les dades.

La pila disposa de dues opcions a l'hora de ser utilitzada, una gratuïta auto gestionable i una altra de pagament amb diferents nivells de llicència. La versió gratuïta és la de codi obert i és la que s'utilitzarà en aquest treball per implementar el SIEM. Per més informació sobre les llicències consultar la seva web[9].

A continuació es farà una introducció de cada una de les eines disponibles al ecosistema de la pila.

2.3.2. ElasticSearch

[8] Disponible a Elastic (https://www.elastic.co/es/)
[9] Preus a Elastic (https://www.elastic.co/es/pricing/)

Il·lustració 1 - Logotip de l'eina Elasticsearch [1]

L'eina Elasticsearch és el nucli de la pila basada en l'api Lucene[10]. És una base de dades de tipus documental i distribuïda, la qual la primera versió va ser publicada pel seu autor Shay Banon l'any 2010.

Desenvolupada utilitzant Java com a llenguatge de programació, disposa d'una interfície RESTful que permet fer consultes mitjançant el format JSON. Això fa que sigui compatible amb la integració d'altres llenguatges de programació.

Està enfocada sobretot a la cerca de texts, però permet qualsevol tipus. La velocitat de resposta és molt alta i la seva naturalesa distribuïda permet la resiliència al adaptar-se a possibles caigudes de nodes.

2.3.3. LogStash

Il·lustració 2 - Logotip de l'eina Logstash [1]

Logstash és l'eina que s'encarrega gestionar les dades dels registres, un servidor que s'encarrega de gestionar la informació disponible que té a l'abast. Està basada en Java, concretament desenvolupada en JRuby.

Principalment la seva funció és la d'enviar les dades a un repositori central, en el cas de la pila ELK, a la base de dades Elasticsearch. Per fer-ho, primer llegeix les dades directament des dels registres, desprès les codifica per poder-les gestionar tant al propi servidor com al node on seran enviades. Un cop codificades, les dades poden ser filtrades per gestionar i processar esdeveniments tals com la detecció de certes IPs dins el text o accés a certs links. Finalment les dades s'envien a un node extern.

2.3.4. Kibana

[10] API Lucene (https://es.wikipedia.org/wiki/Lucene)

Il·lustració 3 - Logotip de l'eina Kibana [1]

Kibana és un visualitzador desenvolupat en Java que forma part de la pila ELK que permet visualitzar les dades indexades a Elasticsearch mitjançant la creació de gràfics de barres, lineals, filtres per dimensions i dates, etc.

Disposa d'una funció que permet crear un dashboard utilitzant els diferents filtres i gràfics generats per tal de poder visualitzar la informació de manera molt més ràpida.

Per últim, inclou una funció que permet inicialitzar el projecte del servidor APM abans de poder començar a registrar les dades de traçabilitat d'una aplicació.

2.3.5. Servidor APM

Il·lustració 4 - Logotip de l'eina Elastic APM [1]

Una altre eina que queda fora de la pila ELK, però forma part del seu ecosistema és el servidor APM. APM és l'acrònim de "Application Performance Monitoring" i com es pot entreveure la seva funció és la de observar el funcionament de les aplicacions per tal de veure els temps de latència en certes operacions, els errors generats, la memòria utilitzada pel servidor, etc.

Forma part de la solució Elastic Observability, la qual inclou el servidor APM, la base de dades Elasticsearch i el visualitzador Kibana.

En aquest treball s'implementarà el servidor APM utilitzant una aplicació desenvolupada amb Python per tal de veure el seu funcionament i com es relaciona amb la ingesta de dades dels registres.

2.3.6. Beats

Il·lustració 5 - Logotip del paquet d'eines Elastic Beats [1]

Els beats són agents que capturen dades i les envien a Elasticsearch o a Logstash en funció de si les dades es volen filtrar o es volen enviar directament a la base de dades. Hi ha diferents tipus de beats, i tots ells estan pensats per poder facilitar la captura i, transformació o mapeig de dades segons la font.

Per exemple, existeix un beat anomenat Metricbeat que permet enviar les mètriques dels serveis que s'executen en un sistema de tal manera que a Kibana es pugui veure la quantitat de memòria utilitzada per cada servei, ús de cpu, de disc dur, etc.

Aquests agents permeten agilitzar la integració dels processos d'ingesta de dades i en la implementació del SIEM se n'utilitzaran uns en concret per poder monitoritzar la seguretat de la informació dels dispositius i dels arxius disponibles al sistema.

2.3.7. Elastic Agent

Els Elastic Agents són clients que monitoritzen la màquina on són instal·lats així com els serveis, però també són capaços de rebre dades de serveis externs i reenviar-les a ElasticSearch. Venen a ser com un grup de beats que en un sol agent permeten capturar dades de múltiples fonts. Per fer-ho utilitza un servidor Fleet que s'encarrega de gestionar les connexions entre Kibana i els Elastic Agents. Un cop connectats els agents, Kibana permet monitoritzar-los externament i actualitzar-los quan sigui necessari.

L'avantatge d'aquest tipus d'agents és que un cop instal·lats i connectats amb Kibana, no tenen gaire cost de manteniment ja que tot es pot dur a terme des de Kibana. Per contra, només es pot instal·lar un agent en cada màquina i en casos en que es volen rebre dades de molts serveis externs, s'haurà de fer una instal·lació en un altre node o dispositiu.

Finalment, en el moment de fer aquest treball el Elastic Agent es troba en fase d'ús públic, però té pendents moltes funcionalitats dels altres beats. Això indica que es probable que en el futur només es faci ús d'aquest tipus d'agents en comptes dels Beats.

2.4. Virtualització

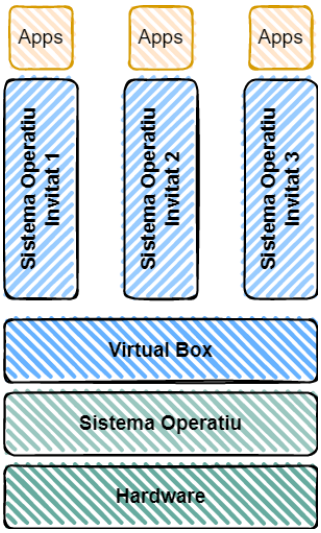
La implementació del SIEM del projecte està pensada per ser escalable i poder-se instal·lar a un servidor, ja sigui físic o en el núvol utilitzant algun proveïdor

cloud com Amazon Web Services o Google Cloud. Degut a aquest requeriment és important destacar que el SIEM utilitzarà la virtualització com a tecnologia principal pel seu funcionament.

La virtualització és una tecnologia que permet utilitzar els recursos d'una màquina amfitriona per allotjar múltiples sistemes operatius o serveis. Aquests serveis o sistemes operatius compartiran els recursos de la màquina amfitriona en funció de les seves necessitats segons el tipus de virtualització.

Per implementar el SIEM s'ha investigat la virtualització completa i la parcial, o com s'anomenarà en aquest treball, de sistema operatiu. Apart d'aquestes dues, existeixen altres tipus de virtualitzacions, però queden fora de l'abast del treball.

2.4.1. Completa



Il·lustració 6 - Exemple de virtualització completa amb Virtual Box

La virtualització completa és un tipus de virtualització que permet replicar virtualment un sistema complert i que aquest estigui totalment aïllat del sistema de la màquina amfitriona.

Per dur a terme aquesta virtualització, les aplicacions encarregades de fer-ho reserven els recursos necessaris de la màquina amfitriona per tal que el sistema en faci un ús dedicat. D'aquesta manera l'aïllament entre els dos sistemes és complert.

Exemples d'aplicacions que permeten la virtualització completa:

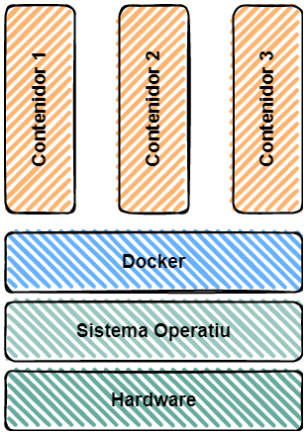
- Virtual Box
- VMWare

Aquest tipus de virtualització permet un manteniment molt més senzill, ja que si existeix qualsevol inconvenient amb el sistema virtualitzat, es pot apagar o recuperar utilitzant una còpia d'aquest i restaurar-la ràpidament.

Un dels usos més habituals és el poder tenir diferents arquitectures de sistemes operatius en una sola màquina, ja que això permet desenvolupar o testejar components en altres entorns sense haver d'adquirir el hardware.

Com es pot veure, si es vol tindre una gran quantitat de sistemes virtuals dins d'una sola màquina, aquesta hauria de tindre molts recursos disponibles i contextualitzant el projecte d'implementació del SIEM, no és el que es pretén. A més a més, recordar que la pila ELK es compon de tres peces de software essencials que funcionen entre elles com un mecanisme complet i necessiten estar constantment connectades, així que utilitzar la virtualització complerta obligaria a instal·lar les aplicacions en un mateix sistema, de manera que hauríem de reservar molts recursos pel seu funcionament més el del sistema virtual, o instal·lar diferents sistemes, un per cada peça del software i reservar els recursos necessaris a més de controlar la xarxa que les uneixi per tal que puguin ser visibles.

2.4.2. Sistema Operatiu



Il·lustració 7 - Exemple de virtualització a nivell de SO mitjançant Docker

El tipus de virtualització a nivell de sistema operatiu és més flexible que el complet, ja que permet compartir els recursos del sistema amfitrió i no requereix tenir instal·lat tot el sistema per fer-ne ús, ja que també comparteix la arquitectura del sistema amfitrió. A diferència de la virtualització completa, aquest tipus provoca un requeriment sobre aquest tipus de virtualització, i és que el sistema que es vol virtualitzar ha d'utilitzar la mateixa arquitectura del sistema operatiu amfitrió, ja que sinó no podria funcionar degut a que no disposaria de les eines per fer-ho.

Tot i així, aquest tipus no s'apodera dels recursos de la màquina amfitriona ni ocupa tant espai al disc dur, ja que al compartir la base del sistema, només necessita els arxius per fer funcionar el servei. Aquesta característica permet que el sistema tingui una ràpida posada en marxa i rapidesa en iniciar-se, però no permet fer canvis a les àrees primàries del sistema operatiu, ja que el comparteix amb la màquina amfitriona.

Algunes aplicacions que permeten virtualitzar i fer ús dels serveis d'aquest tipus són:

- OpenVZ
- Docker
- LXC
- Kubernetes

2.4.3. Ús per la implementació del SIEM

Un cop revisades les dues principals opcions a l'hora d'implementar el SIEM, es decideix fer ús de la virtualització a nivell de sistema operatiu.

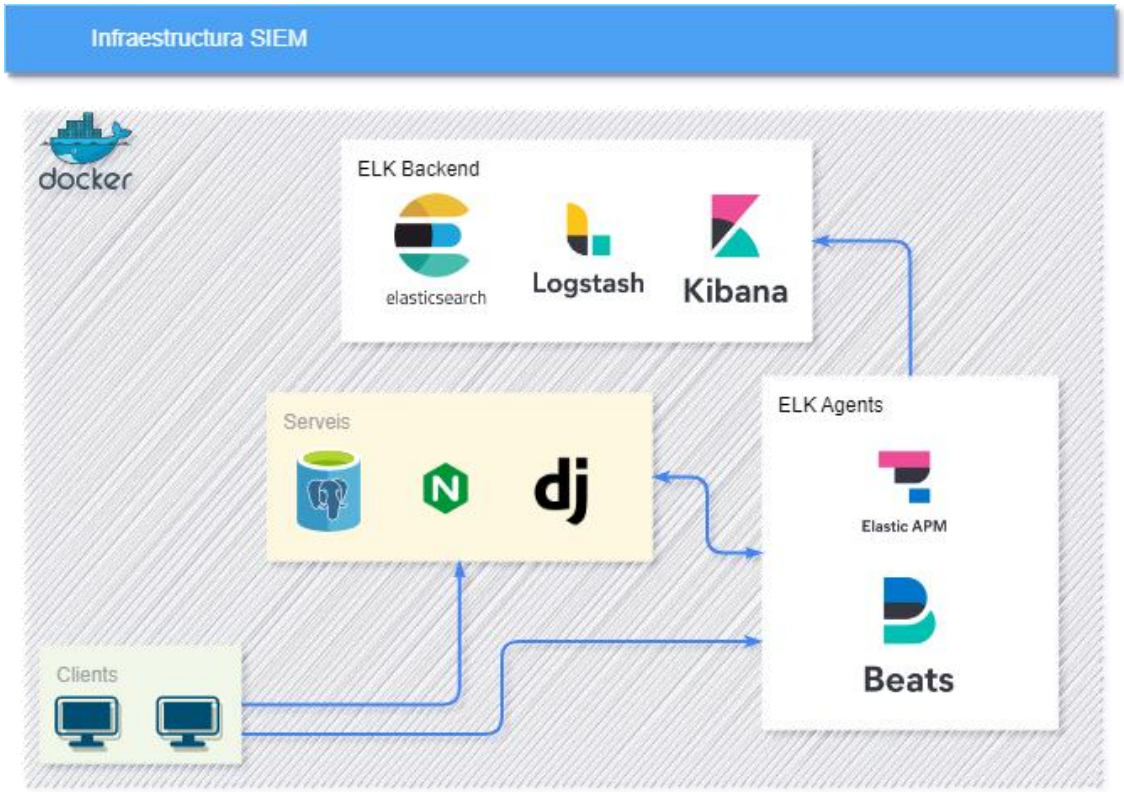
La decisió ve degut a la necessitat de diferents serveis que proporcionin registres per enviar al SIEM. A més, el propi SIEM necessitarà recursos per poder funcionar i en conjunt provocarà que el treball requereixi gestionar múltiples serveis.

Per tal d'evitar el màxim possible invertir temps en gestionar els recursos i configurar els sistemes virtualitzats, s'utilitzarà Docker[11] com a eina per implementar el SIEM. Aquest sistema permetrà una implementació més ràpida i escalable a més sistemes que necessitin monitoritzar-se.

Per executar serveis en l'aplicació Docker primer s'ha de escriure les instruccions que necessita el servei en qüestió i desprès iniciar-lo. Les instruccions s'escriuen en un arxiu i aquest arxiu es compila per generar una "imatge". Aquesta imatge immutable és la que s'executarà des de l'aplicació Docker.

[11] Pàgina web de l'aplicació Docker https://www.docker.com/

2.5. Infraestructura



Il·lustració 8 - Infraestructura SIEM [1] [2] [3] [4] [5]

La implementació del SIEM utilitzant contenidors de Docker permetrà vincular diversos serveis a la mateixa xarxa del SIEM per tal d'enviar-hi dades. L'entorn disposarà d'una infraestructura composta per la pròpia pila ELK, serveis com ara una aplicació en Python[12] o una base de dades, uns clients que facin peticions a aquests serveis i les mètriques i ús del propi sistema on s'executa l'aplicació Docker.

Aquest conjunt formarà la infraestructura utilitzada per aquest treball a fi de mostrar la implementació del SIEM utilitzant la pila ELK i mostrar tots els processos necessaris per capturar o enviar-hi dades.

2.5.1. ELK

La pila ELK està composta de tres principals serveis: ElasticSearch, Logstash i Kibana. Tots hauran de estar disponibles dins la mateixa xarxa per poder-se visualitzar i es comunicaran mitjançant unes credencials d'accés. Aquestes s'han de generar i permetran que la comunicació entre ells pugui dur-se a terme.

Tal com es comentava a l'apartat 2.3 ELK, els serveis estan escrits en Java i s'executen amb la màquina virtual de Java JVM[13], fet que implica que es tinguin en compte aquests requeriments a l'hora d'executar-los. El mateix succeeix amb

[12] Python és un llenguatge de programació https://www.python.org/
[13] JVM és l'acrònim de Java Virtual Machine

l'espai al disc dur. Al utilitzar Docker s'haurà de virtualitzar els volums de dades que faran servir aquests serveis i un d'ells és una base de dades, per tant el volum que utilitzarà haurà de ser més gran que el de Kibana o Logstash.

Atesos els requeriments, també s'han de tenir en compte de exposar els ports necessaris a la màquina amfitriona. Com s'ha comentat els serveis es comuniquen entre ells i utilitzen uns ports concrets per fer-ho. Això implica que si es fa alguna modificació als noms dels serveis o als ports, s'haurà de replicar la configuració als altres per tal que puguin comunicar-s'hi, sempre respectant la necessitat que té la màquina amfitriona de connectar-se a aquests serveis, ja que Docker permet exposar els ports dins la seva pròpia xarxa virtual i si es desitja, exposar-los també a la xarxa local de la màquina amfitriona.

Per últim, ElasticSearch és una base de dades documental que permet tenir un conjunt de rèpliques actuant a mode de clúster[14]. Per dur a terme la implementació no és necessari ja que amb una sola instància del servei és suficient, però en un entorn real on hi ha molts nodes o dispositius a la xarxa del SIEM, sí que és important plantejar la necessitat d'utilitzar un clúster de ElasticSearch per tal d'evitar pèrdua de dades o augmentar-ne la capacitat d'emmagatzematge.

2.5.2. Serveis

El SIEM necessita dades per poder dur a terme les seves funcions. Aquestes poden enviar-se des de diferents fonts, inclosos serveis web com ara aplicacions o altres bases de dades.

Com la implementació es durà a terme mitjançant contenidors de Docker, s'afegiran altres serveis a la mateixa xarxa on resideix la pila ELK. Els serveis es poden dividir en dos grups:

- Serveis d'aplicació
- Serveis de captura de dades

Els serveis d'aplicació són aquells que utilitzaran els clients per consumir un servei tal com una base de dades, una aplicació web que serveix HTML i un servidor web Nginx.

D'altra banda els serveis de captura de dades són aquells que com el seu nom indica, capturen dades de l'entorn on s'executen aquests serveis d'aplicació o dels propis clients.

En aquest projecte es capturaran les mètriques del sistema de la màquina amfitriona i dels serveis, així com els registres d'aquests últims, així que qualsevol esdeveniment que succeeixi s'emmagatzemarà al SIEM.

[14] Un clúster és una agrupació de elements. En aquest cas, una agrupació de nodes que executen el servei ElasticSearch.

Cal fer èmfasis en el ús de Docker, ja que tal com s'ha comentat és una aplicació encarregada de gestionar contenidors virtuals dins la pròpia màquina amfitriona. Això comportarà que les mètriques són comunes per tots els contenidors i només caldrà instal·lar un sol contenidor per cada màquina, que en aquest cas, en serà un de sòl. També s'ha de tindre en compte la gestió dels logs[15], que en aquest cas es pot dur a terme de dues maneres:

- Fent un mapeig de la carpeta de registres de cada servei i contenidor
- Capturant les sortides de cada contenidor utilitzant un Elastic Agent o Beat

Ambdues maneres són funcionals, però els Agents de Elastic o els Beats permeten l'autodescobriment dels contenidors en temps real i la captura dels registres que generen.

Així doncs, s'utilitzarà els Elastic Agents per capturar les dades dels contenidors dels serveis i dels clients.

Per últim el servei Elastic APM permetrà traçar l'aplicació web en cas que hi hagi qualsevol error o excepció, així com el seu rendiment. Aquesta traçabilitat es durà a terme implementant un agent dins el codi del servei web i fent que aquest enviï les dades a un Agent que un cop processades, acabarà enviant-les al servidor principal de Elastic Agents i aquest a ElasticSearch.

2.5.3. Clients

Per utilitzar els serveis de demostració per generar dades, concretament el servidor web i l'aplicació, s'utilitzaran dos clients que aniran fent peticions cada cinc i deu segons. Aquest sistema permetrà capturar les dades i enviar-les a ElasticSearch per tal de visualitzar-les posteriorment.

Cada client serà un contenidor basat en el sistema operatiu Linux Ubuntu i executarà un script en bash que simularà les peticions. Aquest script s'executarà ininterrompudament mentre el contenidor estigui operatiu.

[15] Els logs són els registres.

3. Fase d'implementació

La segona part del treball consistirà en implementar el SIEM utilitzant el hardware i la infraestructura definida. Per fer-ho, s'utilitzarà Docker i es construiran les imatges dels contenidors que ho necessitin o es descarregaran directament del repositori de contenidors de Elastic[16].

Un cop obtingudes les imatges es desplegarà la infraestructura utilitzant Docker, concretament l'eina Docker-Compose[17] que permet generar una composició de tots els serveis en un sòl arxiu.

Quan la infraestructura estigui aixecada, es capturaran les dades mitjançant els Agents o Beats necessaris i les dades generades seran analitzades mitjançant Kibana.

Finalment, es crearà un dashboard a Kibana per poder visualitzar la informació de l'aplicació web, les mètriques d'ús de la màquina amfitriona i els contenidors disponibles d'aquesta.

3.1. Desplegament de la infraestructura

La infraestructura del SIEM estarà composta per la pila ELK, un servei web que inclou NGINX + PostgreSQL + Python (Django Framework), un servidor Elastic Fleet encarregat d'enviar les dades a ElasticSearch, un Agent Elastic, un Agent FileBeat encarregat de supervisar els contenidors de la màquina amfitriona i dos clients que faran peticions al servei web.

Com a hardware utilitzat es farà servir un PC amb sistema operatiu Windows 10 Pro de 64 bits que té una CPU AMD FX-8350 de 8 nuclis a 4GHz amb 16 GB de memòria RAM i un disc dur SSD de 250 GB.

Degut a la decisió d'utilitzar Docker com a aplicació encarregada de gestionar els serveis que conformen el SIEM, el sistema operatiu ha de comptar amb el subsistema Linux Ubuntu WSL instal·lat, així com l'aplicació d'escriptori de Docker. Aquest subsistema no està disponible a totes les versions de Windows 10 o superior, així que en cas d'instal·lar-lo en un altre sistema operatiu Windows, cal comprovar la seva existència. En el cas de Windows 10 Pro sí que ve instal·lat per defecte.

Finalment, al subsistema Linux Ubuntu se li assignaran 8GB de RAM i l'ús total de CPU i espai de disc dur, així que la única limitació serà la memòria.

3.1.1. Preparació de l'entorn

[16] Repositori de contenidors de Docker de Elastic https://www.docker.elastic.co/

[17] L'eina Docker-compose forma part de Docker i permet executar les instruccions d'aquest mitjançant un document en format YAML.

La instal·lació del subsistema Linux Ubuntu pot dur-se a terme simplement accedint a la Microsoft Store i cercant per "Ubuntu"[18]. Un cop instal·lat, apareixerà com una aplicació més dins el programari del sistema i si s'executa s'obrirà un terminal dins el subsistema el qual permetrà executar scripts o funcions com si es tractés d'un sistema Linux Ubuntu complert.

L'aplicació de Docker, també haurà de ser instal·lada al sistema. Per fer-ho cal dirigir-se a la seva pàgina web i descarregar la versió corresponent[19].

Un cop descarregada, s'haurà d'accedir a les seves opcions i assegurar-se de tenir activades les següents funcions:

- General -> "Use the WSL 2 based engine"
- General -> "Use Docker Compose V2"
- Resources -> WSL Integration -> "Enable Integration with my default WSL distro"
- Resources -> WSL Integration -> Activar el subsistema Ubuntu

Activar les funcions nombrades permetrà que Docker pugui fer ús del subsistema Linux i utilitzar la virtualització de manera nativa, fet que ajudarà a l'eficiència de l'entorn.

En el cas que s'utilitzés un sistema operatiu Linux com Ubuntu en comptes de Windows, el procediment hagués quedat reduït a instal·lar únicament l'aplicació Docker mitjançant la següent comanda:

sudo apt install -y docker.io

En aquest punt l'entorn ja permet utilitzar contenidors de Docker així que es pot començar a desplegar l'entorn del SIEM. Primer es crearà un arxiu de Docker Compose amb el format YML que permetrà descriure tota la infraestructura en codi i desprès s'executarà aquest arxiu per desplegar l'entorn.

3.1.2. ELK

Abans de començar, es genera un arxiu anomenat "siem.yml" que serà el que contindrà tota la infraestructura del SIEM. Dins l'arxiu es definirà la versió del parser[20] de l'aplicació Docker compose que es voldrà fer ús.

```
version: '3.8'
```

[18] Enllaç d'instal·lació del subsistema Linux Ubuntu a Windows https://www.microsoft.com/store/productId/9PDXGNCFSCZV
[19] Enllaç a la pàgina web de l'aplicació Docker Desktop https://www.docker.com/products/docker-desktop/
[20] Un parser és un intèrpret

Per desplegar la pila ELK es descriuran els tres serveis en l'arxiu generat. Cada un d'ells anirà agrupat sota la paraula clau "services" i per aquest treball s'utilitzarà la versió 8.5.3 de la pila.

Servei ElasticSearch:

```
services:
  es01:
    build: ./elasticsearch/
    hostname: es01
    container_name: es01
    environment:
      ES_JAVA_OPTS: -Xms512m -Xmx512m
    restart: on-failure
    volumes:
      - data01:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
    networks:
      - elk

volumes:
  data01:
    driver: local

networks:
  elk:
    driver: bridge
    driver_opts:
      com.docker.network.bridge.enable_icc: "true"
```

Com es pot veure el servei ElasticSearch té assignat el nom "es01" dins la xarxa Docker i el contenidor també té el mateix nom. D'aquesta manera quan es vulgui identificar el contenidor per analitzar dades es podrà fer servir aquest nom com a identificador a la cerca.

Per generar la imatge, s'utilitzarà el context dins la carpeta "./elasticsearch", on s'inclou un arxiu Dockerfile[21] i la configuració del servei. El arxiu Dockerfile conté les instruccions que permetran copiar la configuració del servei a la imatge i aquesta serà generada mitjançant el contenidor oficial de Elastic.

```
FROM docker.elastic.co/elasticsearch/elasticsearch:8.5.3

COPY --chmod=0644 --chown=elasticsearch:elasticsearch elasticsearch.yml
/usr/share/elasticsearch/config/elasticsearch.yml
```

[21] Un arxiu Dockerfile conté les instruccions necessàries per generar la imatge virtual que conté tots els processos necessàries per executar un servei com a contenidor.

D'altra banda el arxiu de configuració del servei contindrà les configuracions més bàsiques per iniciar-lo:

```
cluster.name: es-cluster
network.host: 0.0.0.0
node.name: es01
discovery.type: single-node
xpack.security.enabled: true
```

Tal com es va comentar ElasticSearch s'executa utilitzant Java, per tant s'ha de controlar l'entorn que executa el servei i això es fa mitjançant la variable "ES_JAVA_OPTS" que permet definir la memòria RAM assignada a la JVM. A més es defineixen dues configuracions pel servei, una per assignar el mateix nom que el del contenidor i l'altre per informar a ElasticSearch que no hi ha més nodes que ell mateix, és a dir, que no es crearà cap clúster amb diferents serveis ElasticSearch, sinó que únicament hi haurà un node. Aquesta configuració evitarà que el servei faci una sèrie de controls inicials per buscar altres nodes i s'iniciarà més ràpid.

Finalment, s'obrirà el port 9200 a la màquina amfitriona per si aquesta vol fer alguna petició a la API REST del servei, s'assignarà el contenidor a una xarxa virtual (nombrada elk en aquest cas) i es generarà un volum per guardar les dades.

Un cop escrites les instruccions, és el moment de construir la imatge i posar-la en marxa, per fer-ho s'haurà d'executar la següent comanda:

docker-compose -f .\siem.yml build --no-cache es01

Aquesta comanda construirà la imatge i un cop hagi acabat, es podrà veure-la executant la comanda:

docker images

Un cop obtinguda la imatge, es procedirà a iniciar el servei:

docker-compose -f .\siem.yml up -d es01

I es podrà comprovar el seu funcionament mitjançant:

docker ps

```
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS                                   NAMES
b2c671f22db8   tfm-es01   "/bin/tini -- /usr/l…"   44 minutes ago   Up 44 minutes   0.0.0.0:9200->9200/tcp, 9300/tcp   es01
```

Com es pot veure el servei ja s'ha iniciat i té disponible el port 9200 per la màquina amfitriona i el port 9300 per altres contenidors dins la xarxa.

A continuació, s'haurà de generar les credencials necessàries per poder relacionar els altres serveis mitjançant el propi contenidor del servei.

Primer s'accedirà al contenidor:

docker exec -it es01 bash

Executada aquesta comanda, la terminal mostrarà l'usuari elasticsearch al host es01, fet que confirmarà l'accés. Un cop dins, s'haurà d'iniciar els usuaris per poder utilitzar Kibana:

./bin/elasticsearch-reset-password -u elastic

./bin/elasticsearch-reset-password -u kibana_system

Cadascuna de les comandes retornarà una contrasenya que s'haurà de copiar a la configuració del servei Kibana, així que caldrà guardar-les per més endavant. En acabar caldrà escriure "*exit*" per sortir del contenidor i tornar a la terminal de la màquina amfitriona.

<u>Servei Kibana:</u>

```
  kib01:
    build: ./kibana/
    hostname: kib01
    container_name: kib01
    restart: on-failure
    ports:
      - 5601:5601
    networks:
      - elk
    depends_on:
      - es01
```

En el cas del servei Kibana, s'indicarà la xarxa i la dependència del servei ElasticSearch, així com el context de construcció de la imatge que inclou el Dockerfile:

```
FROM docker.elastic.co/kibana/kibana:8.5.3

COPY --chmod=0644 --chown=kibana:kibana kibana.yml
/usr/share/kibana/config/kibana.yml
```

I la configuració del servei.

```
server.name: kib01
server.host: 0.0.0.0
elasticsearch.hosts: [ http://es01:9200 ]
monitoring.ui.container.elasticsearch.enabled: true
```

```
monitoring.ui.container.logstash.enabled: true
elasticsearch.username: kibana_system
elasticsearch.password: KIBANA_SYSTEM_PASSWORD
```

La variable “KIBANA_SYSTEM_PASSWORD” s’haurà de sobreescriure utilitzant la contrasenya generada anteriorment per aquest usuari i en acabat construir la imatge i iniciar-la:

docker-compose -f .\siem.yml build --no-cache kib01
docker-compose -f .\siem.yml up -d kib01

Al iniciar el contenidor, es podrà accedir al servidor web que inclou Kibana mitjançant l’adreça http://localhost:5601 i utilitzant l’usuari “elastic” i la contrasenya generada per l’usuari anteriorment dins el contenidor del servei ElasticSearch.

Per poder capturar les dades i utilitzar-les es necessitarà generar una clau de xifratge que permetrà a Kibana xifrar les dades enviades des dels Fleets i beats. Per generar-la s’haurà d’accedir al contenidor i executar la següent comanda que un cop seguides les instruccions retornarà tres claus que s’hauran de escriure a la configuració del servei Kibana, regenerar la imatge i reiniciar el contenidor.

./bin/kibana-encryption-keys generate -i

Les claus correspondran a les següents opcions de la configuració:

```
## X-Pack security encrypt
xpack.encryptedSavedObjects.encryptionKey: CLAU_1
xpack.reporting.encryptionKey: CLAU_2
xpack.security.encryptionKey: CLAU_3
```

Un cop iniciada la sessió, dirigir-se a http://localhost:5601/app/management/security/api_keys/ i crear una nova API key ja que s’utilitzarà per consumir la API REST del servei ElasticSearch i generar l’usuari pel servei LogStash, d’aquesta manera es pot explorar una altra via per generar usuaris que no sigui la de l’ús del CLI de ElasticSearch.

Primer s’haurà de crear el rol per logstash que s’anomenarà “logstash_writer” i només tindrà accés als índexs que de nom comencin amb “logstash-”. Es farà amb cURL i s’utilitzarà la API Key generada en el pas anterior:

```
curl --location --request POST
'localhost:9200/_security/role/logstash_writer' \
--header 'Authorization: ApiKey API_KEY_GENERADA_AMB_KIBANA' \
--header 'Content-Type: application/json' \
--data-raw '{
  "cluster": ["manage_index_templates", "monitor", "manage_ilm"],
  "indices": [
    {
```

```
      "names": [ "logstash-*" ],
      "privileges":
["write","create","create_index","manage","manage_ilm"]
    }
  ]
}'
```

Com es pot veure la petició apunta a la màquina amfitriona, però en el port 9200 que és el que ha de tenir obert el servei ElasticSearch. Si tot ha anat bé, s'obtindrà el retorn que informarà de la seva correcta creació.

```
{
    "role": {
        "created": true
    }
}
```

A continuació s'haurà de generar el usuari i relacionar-lo amb el rol creat:

```
curl --location --request POST
'localhost:9200/_security/user/logstash_internal' \
--header 'Authorization: ApiKey API_KEY_GENERADA_AMB_KIBANA' \
--header 'Content-Type: application/json' \
--data-raw '{
  "password" : "CONTRASENYA_DEL_USUARI_LOGSTASH_INTERNAL",
  "roles" : [ "logstash_writer"],
  "full_name" : "Internal Logstash User"
}'
```

L'anterior comanda crearà un usuari "logstash_internal" amb la contrasenya que s'hagi escollit. El retorn de la API serà el següent:

```
{
    "created": true
}
```

Ara ja es tindrà tot el necessari per poder iniciar el servei LogStash.

<u>Servei Logstash:</u>

```
  log01:
    build: ./logstash/
    hostname: log01
    container_name: log01
    restart: on-failure
    ports:
      - 5044:5044
      - 50000:50000/tcp
```

```
      - 50000:50000/udp
      - 9600:9600
    networks:
      - elk
    depends_on:
      - es01
```

En el cas de Logstash es pot veure que disposa de diferents ports per comunicar-se amb la màquina amfitriona, això és degut a que els farà servir com a inputs a l'hora de capturar dades. Els inputs i outputs es defineixen a la configuració del que s'anomena Pipeline [22]de captura:

```
input {
    beats {
        port => 5044
    }

    tcp {
        port => 50000
    }
}

output {
    elasticsearch {
        hosts => "es01:9200"
        user => "logstash_internal"
        password => "CONTRASENYA_DEL_USUARI_LOGSTASH_INTERNAL"
    }
}
```

I la configuració general del servei serà mínima:

```
http.host: 0.0.0.0
node.name: log01
```

En aquest cas es necessitarà copiar aquests arxius al contenidor per tal que el servei en faci ús:

```
FROM docker.elastic.co/logstash/logstash:8.5.3

COPY --chmod=0644 --chown=logstash:logstash logstash.yml
/usr/share/logstash/config/logstash.yml
COPY --chmod=0644 --chown=logstash:logstash ./conf
/usr/share/logstash/pipeline
```

Finalment, construir el contenidor i aixecar-lo:

[22] Un Pipeline de captura és un fluxe d'entrada i sortida per on circulen les dades.

docker-compose -f .\siem.yml build --no-cache log01
docker-compose -f .\siem.yml up -d log01

Com s'ha pogut veure, els tres serveis conviuen dins la mateixa xarxa virtual anomenada "elk". Sinó fos així els contenidors no serien visibles entre ells i no es podrien comunicar. Aquest exemple serviria també per un clúster Docker Swarm[23].

En el cas que cada servei estigués en una màquina separada o en proveïdors cloud diferents, caldria modificar els noms dels hosts[24] amb els DNS[25] de cada màquina.

El següent pas serà construir els serveis necessaris de l'aplicació web, els agents i posteriorment, els clients.

3.1.3. Serveis

Ara que el SIEM es funcional, necessitarà dades per poder mostrar i utilitzar. Aquestes dades poden ser capturades utilitzant els Elastic Agents, els Elastic Beats o ser enviades directament a ElasticSearch amb LogStash.

En aquest cas, es voldrà capturar les mètriques de la màquina amfitriona i els registres de cada un dels contenidors que s'estan executant en temps real. D'aquesta manera a Kibana es visualitzarà aquesta informació facilitant qualsevol intervenció al equip de resposta de seguretat.

Primer caldrà construir el Elastic Fleet Server, que serà l'encarregat de gestionar tots els possibles Elastic Agents que gestioni la pila ELK. El Fleet Server no deixa de ser un Agent, però amb un procés intern que permet la comunicació interna amb el servei ElasticSearch, per aquest motiu s'ha de tindre en compte que la pròpia instal·lació del servidor Fleet ja comporta tenir un Agent capturant dades.

En aquest treball s'executaran dos Elastic Agents per mostrar el seu funcionament, un amb rol de Servidor i l'altre com a Agent, però com no té cap sentit enviar les mètriques de la mateixa màquina, esdedicarà el segon agent com a servidor APM per enviar les dades i mètriques de traçabilitat de l'aplicació web.

Abans de començar, com s'ha comentat el servidor Fleet necessitarà comunicar-se amb el servei ElasticSearch i per fer-ho necessitarà unes credencials que poden ser un usuari i contrasenya o un token de servei. En aquest treball s'ha optat per fer servir un token per tal de no haver de copiar les credencials i facilitar

[23] Docker Swarm és un concepte que fa referència a un grup de nodes o servidors que es comuniquen entre ells com un clúster.
[24] S'utilitza la paraula host per fer referència a la màquina amfitriona.
[25] DNS és l'acrònim de Domain Name Service.

la seva gestió. Per generar el token s'executarà la comanda següent tenint en compte que el contenidor del servei ElasticSearch està en funcionament:

docker exec -it es01 ./bin/elasticsearch-service-tokens create elastic/fleet-server fleet-server-token

Aquesta comanda crida al contenidor del servei ElasticSearch i executa l'aplicació per gestionar els tokens de servei. En aquest cas se li passen els arguments usuari (elàstic/fleet-server) i el nom del token. El resultat de l'execució serà un token de servei que utilitzarem posteriorment durant la instal·lació del servidor Fleet.

El següent pas serà afegir la configuració de polítiques dels Elastic Agents. Aquests es poden instal·lar directament utilitzant Kibana (Manage -> Fleet) dins de cada contenidor, però com la implementació del SIEM és en Docker s'afegiràn directament com un contenidor més i la configuració de la política serà afegida a la del servei Kibana que s'ha construït anteriorment. Per fer-ho primer s'haurà d'aturar el contenidor, esborrar la imatge per si de cas, afegir la configuració i tornar a aixecar el servei mitjançant les següents comandes:

docker stop kib01
docker imatge rm -f kib01

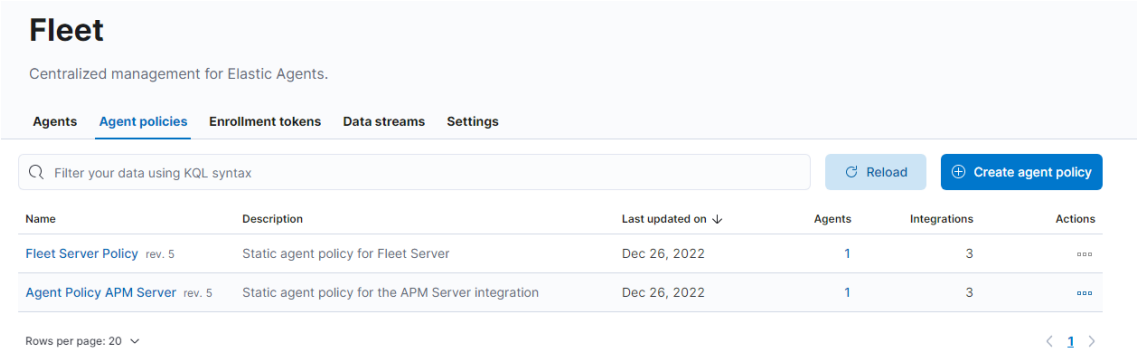
Afegir la configuració de l'annex a l'arxiu Kibana.yml

Aquesta configuració inicialitzarà el servidor Fleet i en derivarà el output cap al servei d'ElasticSearch. També s'afegiran les dues polítiques que es necessitaran. Una pel servidor Fleet i l'altre pel servidor APM que com es pot veure aquest últim defineix el seu host segons el nom que se li posarà al servei APM del SIEM. En canvi en el cas del servidor Fleet es tindrà en compte l'identificador de la política per auto assignar-la al iniciar el servei i aquest es pugui comunicar amb Kibana.

Un cop modificada la configuració, es procedirà a construir i iniciar la imatge de nou:

docker-compose -f .\siem.yml build --no-cache kib01
docker-compose -f .\siem.yml up -d fb01

Per assegurar que les polítiques hauran estat afegides al servei Kibana, es pot revisar el enllaç http://localhost:5601/app/fleet/policies que mostrarà una pantalla semblant a aquesta:



Il·lustració 9 - Polítiques dels agents a Kibana

Un cop afegida la nova configuració, s'haurà d'afegir els agents i l'aplicació web seguint els serveis dins l'arxiu siem.yml, en aquest cas el del servidor Fleet:

```
fleet-server:
  build: ./fleet-server/
  container_name: fleet-server
  hostname: fleet-server
  restart: on-failure
  environment:
    FLEET_SERVER_ENABLE: 1
    FLEET_SERVER_INSECURE_HTTP: 1
    FLEET_SERVER_POLICY_ID: fleet-server-policy
    FLEET_SERVER_ELASTICSEARCH_HOST: "http://es01:9200"
    KIBANA_HOST: "http://kib01:5601"
    KIBANA_FLEET_SETUP: 1
    FLEET_SERVER_SERVICE_TOKEN: "TOKEN_ELASTICSEARCH_GENERAT"
  ports:
    - 8220:8220
  networks:
    - elk
  depends_on:
    - es01
    - kib01
```

La configuració del contenidor és semblant a les anteriors, però la diferència recau en la quantitat de variables d'entorn que s'utilitzen. Aquestes fan referència a:

- FLEET_SERVER_ENABLE:
  Habilitar el mode servidor de l'agent.

- FLEET_SERVER_INSECURE:
  Activat ja que no s'estan utilitzant certificats SSL per connectar els serveis.

- FLEET_SERVER_POLICY_ID:
  El identificador de la política de gestió de dades que farà aquest Agent com a servidor. Ha d'incloure el paquet de polítiques "Fleet Server". Serà el que s'ha definit a la configuració del Kibana modificada.

- FLEET_SERVER_ELASTICSEARCH_HOST:
  El host del servei ElasticSearch.

- KIBANA_HOST:
  El host del servei Kibana.

- KIBANA_FLEET_SETUP:
  Activar en el cas de carregar les polítiques del agent directament al iniciar el servei Kibana. Té relació amb la configuració FLEET_SERVER_POLICY_ID.

- FLEET_SERVER_SERVICE_TOKEN:
  Un token que fa de credencial i permet comunicar l'agent amb el servei ElasticSearch. S'haurà de copiar el valor del token de servei generat anteriorment al contenidor del servei ElasticSearch.

Les anteriors variables permetran enviar les dades al servei aixecat ElasticSearch i comunicar-se amb Kibana per afegir el Agent dins la configuració.

En aquest cas el arxiu Dockerfile només crea una carpeta on recaurà l'estat de sincronització de dades entre el servidor i el servei ElasticSearch.

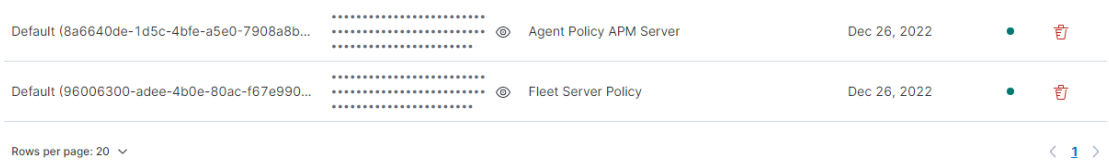
```
FROM docker.elastic.co/beats/elastic-agent:8.5.3
RUN mkdir state
```

Finalment es construeix la imatge amb la comanda següent:

docker-compose -f .\siem.yml build --no-cache fleet-server

Un cop construïda la imatge del servidor Fleet, faltarà la de l'agent destinat a ser un APM, però com es tracta d'un agent se li haurà d'assignar un token per tal que Kibana el relacioni amb una política i a més definir-li el host d'aquest últim i del servidor Fleet. Ambdues adreces dels hosts es tenen disponibles, per una banda l'adreça del servei Kibana és "http://kib01:5601" i per l'altre la del servidor Fleet és "http://fleet-server:8220". Ambdues ja s'han vist anteriorment.

Per aconseguir el token que relacionarà l'agent amb Kibana s'ha de navegar a http://localhost:5601/app/fleet/enrollment-tokens . Al haver afegit en els passos anteriors les polítiques dels agents, la pantalla mostrarà dos tokens, un pel servidor Fleet i l'altre pel servidor APM.



Il·lustració 10 - Tokens pels agents Fleet a Kibana

Caldrà copiar el token assignat a l'agent APM i reescriure'l a la variable corresponent de l'arxiu siem.yml:

```
apm02:
  build: ./fleet-agent/
  container_name: apm02
  hostname: apm02
  restart: on-failure
  environment:
    FLEET_ENROLL: 1
    FLEET_INSECURE: 1
    FLEET_URL: http://fleet-server:8220
    KIBANA_HOST: "http://kib01:5601"
    FLEET_ENROLLMENT_TOKEN: "TOKEN_POLITICA_AGENT_APM"
  depends_on:
    - fleet-server
  networks:
    - elk
```

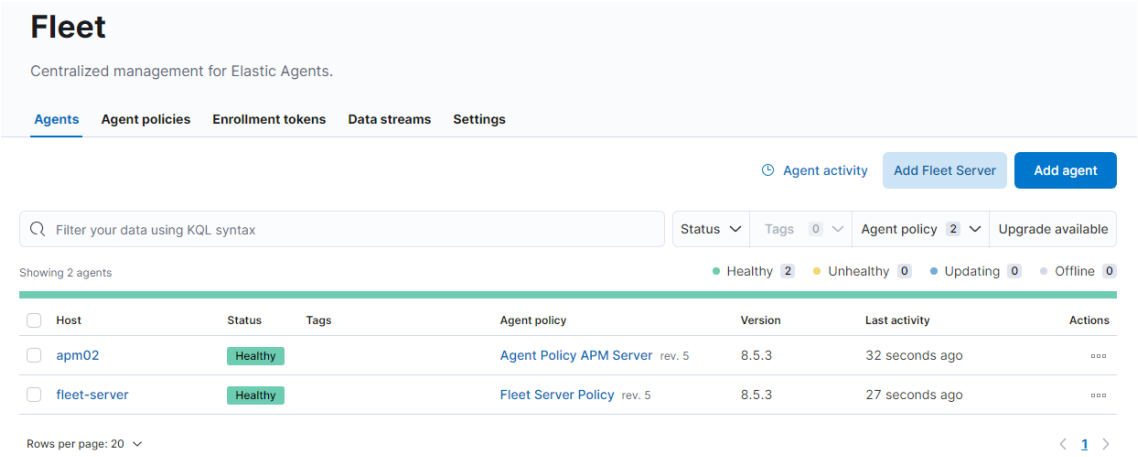
Com l'anterior agent, que tot i ser servidor també ho és, el Dockerfile conté la creació de la carpeta "estat" per mantenir la sincronització de dades aquest cop però, amb el propi servidor Fleet.

Es construeix la imatge i s'inicialitza:

docker-compose -f .\siem.yml build --no-cache apm02
docker-compose -f .\siem.yml up -d apm02

Degut a que les instruccions del contenidor "apm02" contenen una clàusula "depends_on" que fa referencia al servidor Fleet, quan s'executi la comanda automàticament s'iniciarà el contenidor del servidor abans que la de l'agent.

Un cop iniciats els dos serveis, es podrà visualitzar el seu estat a Kibana mitjançant l'adreça http://localhost:5601/app/fleet/agents



Il·lustració 11 - Fleet Agents disponibles a Kibana

Un cop instal·lats els agents, es podrà veure les mètriques del sistema mitjançant l'apartat "Discover" dins Kibana, però encara no es tindran les dades dels registres i ús dels contenidors de Docker de la màquina amfitriona ni tampoc la traçabilitat de l'aplicació web per donar-li un ús l'agent destinat com APM.

Per poder visualitzar els registres dels contenidors s'utilitzarà un Elastic Beat anomenat FileBeat que s'encarregarà de controlar els registres que els contenidors van escrivint als arxius de la màquina amfitriona. Això ho aconseguirà obtenint accés al procés de l'aplicació Docker i a la ubicació dels seus contenidors que són dins la màquina amfitriona. Cal recordar que Docker no permet modificar la configuració dels serveis de la màquina amfitriona al utilitzar la virtualització del sistema operatiu, per tant s'haurà de fer un mapeig del procés Docker de la màquina amfitriona al contenidor i accedir-hi amb un usuari amb permisos, com podria ser l'usuari "root".

Per tant, les instruccions del contenidor del servei FileBeat seran les següents:

```
  fb01:
    build: ./filebeat/
    container_name: fb01
    hostname: fb01
    restart: on-failure
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /var/lib/docker/containers:/var/lib/docker/containers:ro
    depends_on:
      - es01
      - kib01
    networks:
      - elk
```

Si bé és cert que es podria instal·lar FileBeat directament a la màquina amfitriona i resoldre el problema de la virtualització i mapeig de procesos, en aquest treball tot es farà utilitzant contenidors.

El servei de FileBeat no només permet reenviar els registres dels contenidors sinó que també permet enviar registres que s'escriguin en arxius, però per fer-ho primer necessitarà un usuari que tingui permisos al servei ElasticSearch. Per crear-lo, primer necessitarà crear un rol i desprès l'usuari en qüestió.

Utilitzant la petició anterior de LogStash i la API Key:

```
curl --location --request POST
'localhost:9200/_security/role/filebeat_writer' \
--header 'Authorization: ApiKey API_KEY_GENERADA_AMB_KIBANA' \
--header 'Content-Type: application/json' \
--data-raw '{
  "cluster": ["read_ilm", "monitor", "read_pipeline"],
  "indices": [
    {
```

```
      "names": [ "filebeat-*" ],
      "privileges":
["write","create","create_index","manage","manage_ilm", "create_doc"]
    }
  ]
}'
```

La resposta serà que el rol ha estat creat i a continuació es crearà l'usuari assignant-li una contrasenya i el rol creat:

```
curl --location --request POST
'localhost:9200/_security/user/filebeat_internal' \
--header 'Authorization: ApiKey API_KEY_GENERADA_AMB_KIBANA' \
--header 'Content-Type: application/json' \
--data-raw '{
  "password" : "CONTRASENYA_TRIADA",
  "roles" : [ "filebeat_writer"],
  "full_name" : "Internal Filebeat User"
}'
```

Si tot ha sigut correcte la resposta serà que l'usuari haurà estat creat i l'usuari i contrasenya s'hauran de copiar a la configuració de FileBeat juntament amb els registres que es vulgui reenviar les dades:

```
filebeat.autodiscover: # auto-discover tagged docker container
   providers:
    - type: docker
      hints.enabled: true
      templates:
        - condition:
            contains:
              docker.container.image: nginx
          config:
            - module: nginx
              access:
                enabled: true
                input:
                  type: container
                  paths:
                    -
/var/lib/docker/containers/${data.docker.container.id}/*.log
                  stream: stdout
              error:
                enabled: true
                input:
                  type: container
                  paths:
                    -
/var/lib/docker/containers/${data.docker.container.id}/*.log
```

```
                    stream: stderr

setup:
  kibana.host: "http://kib01:5601"
  dashboards.enable: true

output.elasticsearch:
  hosts: [ "http://es01:9200" ]
  username: 'filebeat_internal'
  password: 'CONTRASENYA_TRIADA'
```

Apart de definir la sortida de les dades cap al servei ElasticSearch, també s'activaran els "dashboards" i per fer-ho s'haurà de definir l'adreça de Kibana. Aquests "dashboards" predefinits són per algunes aplicacions en concret com ara Nginx, PHP, Kafka, entre d'altres, i inclouen visualitzacions enfocades a l'ús de cada aplicació.

D'una altra banda, també s'activarà "l'autodiscover" de FileBeat per l'aplicació Docker, que no deixa de ser un supervisor dels contenidors que permet reenviar els registres de les sortides d'aquests cap al servei ElasticSearch. A més, en aquest cas concret s'ha afegit la configuració de lectura de accessos i errors pels contenidors que continguin "nginx" al seu nom ja que aquest té dos fluxos de sortida pels registres en funció de la categoria d'aquests.

Com a la implementació del treball serà contemplada una aplicació web utilitzant Nginx, s'ha cregut oportú desenvolupar una mica aquest cas en concret.

Finalment es construirà la imatge i s'iniciarà el servei:

docker-compose -f .\siem.yml build --no-cache fb01
docker-compose -f .\siem.yml up -d fb01

Un cop iniciats els agents per capturar les dades, s'afegirà a la xarxa del SIEM un servei de PostgreSQL, un de Python que contindrà l'aplicació web i un Nginx que servirà de "proxy" per l'aplicació web i el servei de Kibana.

Les instruccions per aquests serveis seran exposades l'annex d'aquest treball.

3.1.4. Clients

Per tal de generar registres s'afegiran dos contenidors amb un sistema Linux Ubuntu que simularan peticions a l'aplicació web. En rebre-les, l'aplicació web generarà registres al contenidor de Nginx i al de Python, a més de les mètriques que aquest últim enviarà de les peticions a l'agent APM iniciat anteriorment.

Les peticions s'enviaran mitjançant un script que tindrà un bucle infinit i anirà fent peticions cada cinc segons. El codi es troba a l'annex.

3.2. Captura i anàlisi de dades

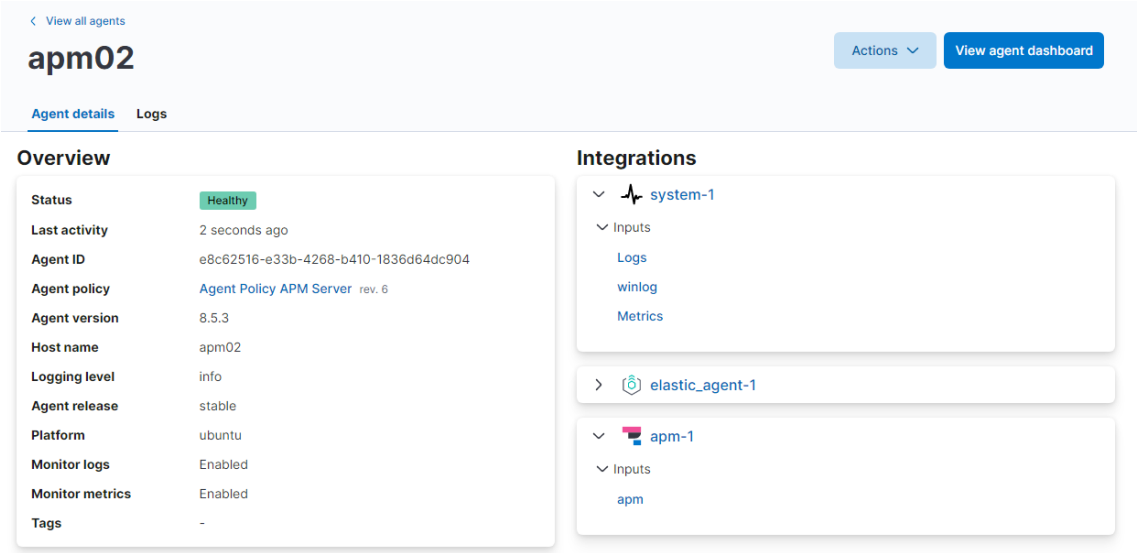
3.2.1. Índexs de ElasticSearch

ElasticSearch utilitza índexs per emmagatzemar les dades. Cada índex té uns permisos associats i normalment s'identifiquen en funció de les dades que hi seran guardades.

Durant la implementació, el servei haurà generat uns índexs per defecte en funció de la necessitat de cada agent, essent així que un cop posat en marxa el sistema hi hauran disponibles els següents:

- logs-*
- metrics-*
- filebeat-*
- APM

Cada un d'ells pertany a les funcions establertes a les seves polítiques. Per exemple, l'agent dedicat com a APM té els inputs de logs i mètriques així com el de APM que aniran al respectiu índex per defecte.

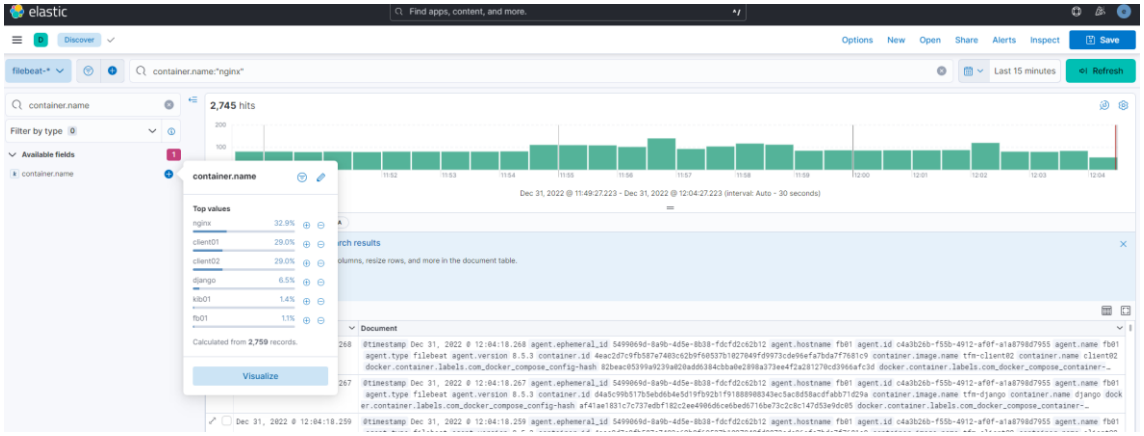


Il·lustració 12 - Exemple de integracions de l'agent APM

També hi ha la possibilitat de modificar l'índex on es volen emmagatzemar les dades a través de la configuració de l'agent, però s'ha de tindre en compte de modificar també la configuració de Kibana ja que aquest també utilitza uns noms d'índexs per defecte.

Cada índex proporciona la informació que és capaç d'enviar l'agent en funció de la política, és a dir, en funció de les integracions que tingui disponibles. Les dades es mostraran a l'apartat "Discover" de la categoria "Analytics".

Al haver implementat els serveis amb l'agent FileBeat per monitoritzar els contenidors de Docker, es podrà veure tota la informació relacionada amb els contenidors disponibles. Per exemple, si es vol revisar les dades del contenidor "Nginx" és tant senzill com seleccionar l'índex i cercar el nom del contenidor per la paraula "nginx".



Il·lustració 13 - Cerca dels registres del contenidor Nginx a Kibana

Com es pot veure a la captura, es pot veure un recompte a mode de previsualització al seleccionar el camp "container.name".

Un cop revisades les dades i assegurat que es reben dades de tots els contenidors, s'haurà de generar el dashboard per l'equip de seguretat.

3.2.2. Dashboard

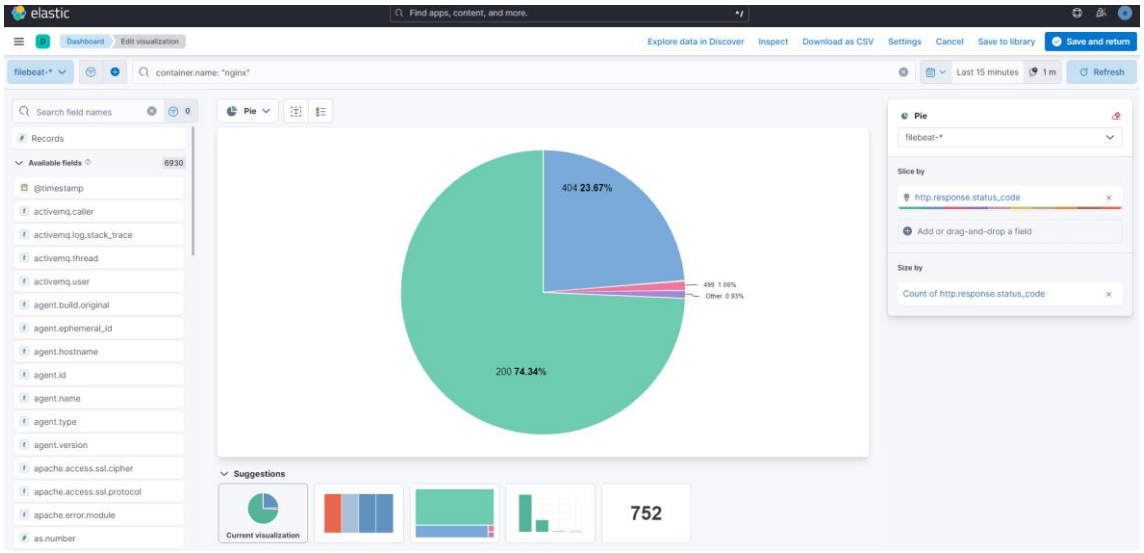
En aquest apartat es crearà un dashboard per poder visualitzar la informació del sistema, és a dir, de la màquina amfitriona i de l'aplicació web. Per fer-ho s'haurà d'anar a l'apartat "Dashboard" dins la categoria "Analytics" i es seleccionarà "Create Dashboard" per començar amb l'assistent.

Cal dir que es poden obtindre exemples a la documentació de la pila ELK de la web de Elastic.

Es podrà anar col·locant la informació per apartats i blocs. Cada bloc és una cerca independent, però es poden col·locar filtres globals que permetin cercar per determinats aspectes. Per exemple, es podria crear un dashboard on poder visualitzar les mètriques i ús de cada contenidor i tenir disponible un filtre global que fos el identificador del contenidor o el seu nom.

Per aquest treball s'ha pensat en fer un dashboard que mostri les mètriques i ús del sistema on s'executa l'aplicació Docker, és a dir, al subsistema Linux de Windows, així com un breu resum de l'aplicació web.

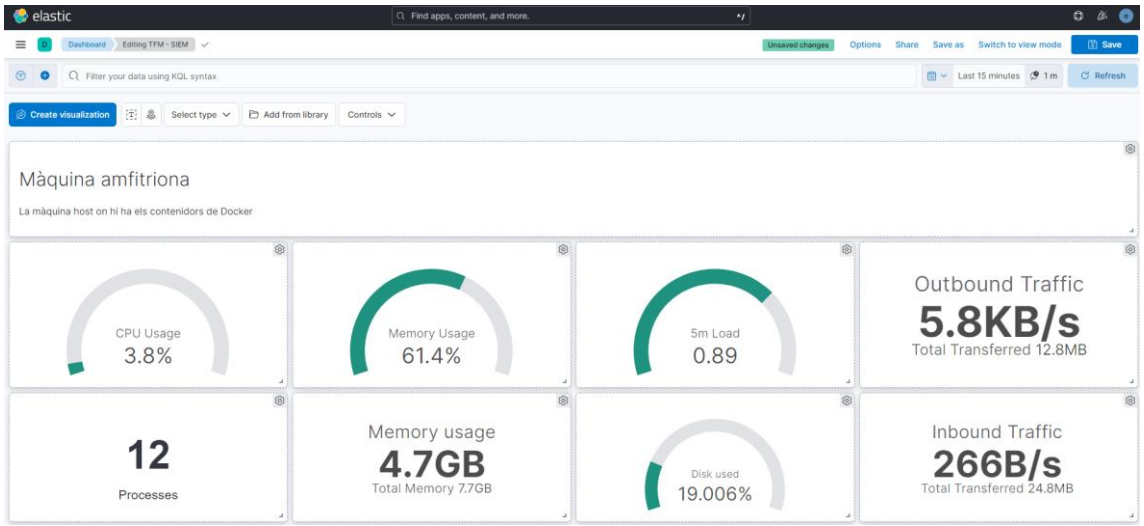
Com a exemple per compondre un bloc d'informació, es farà servir el contenidor del servei Nginx per saber quants i quins codis HTTP ha retornat als clients.



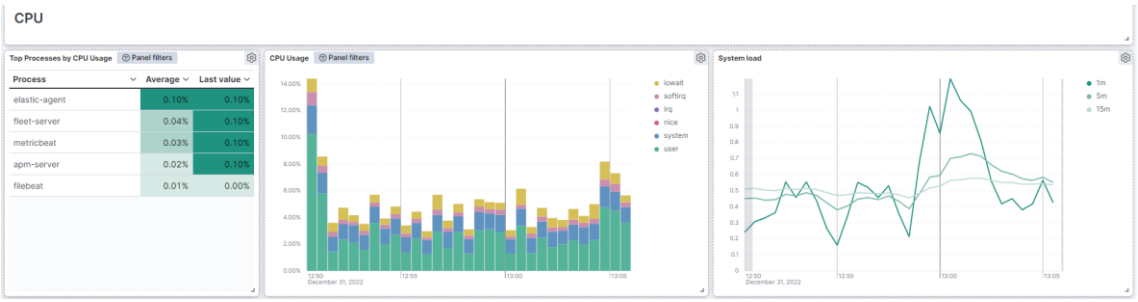
Il·lustració 14 - Exemple d'un bloc del dashboard a Kibana

Com es pot veure a la captura, s'utilitzarà l'índex "filebeat-*" que és el que conté les dades dels contenidors i s'agafarà el camp "http.response.status_code" tant per mostrar el codi, com per fer el recompte.

Un cop s'han acabat de col·locar tots els blocs, el dashboard mostrarà tota la informació necessària per tal que l'equip de seguretat sigui capaç de veure ràpidament l'estat del sistema.



Il·lustració 15 - Dashboard per l'equip de seguretat a Kibana part 1



Il·lustració 16 - Dashboard per l'equip de seguretat a Kibana part 2

Il·lustració 17 - Dashboard per l'equip de seguretat a Kibana part 3

Amb la informació dels blocs, l'equip de seguretat pot veure si hi ha un major consum en algun dels contenidors, una major quantitat de codis HTTP que puguin indicar un possible atac de denegació de servei o fins i tot una possible descàrrega de dades si veuen que el tràfic de sortida és més alt del normal.

3.3. Detecció i resposta del SIEM

El dashboard creat a l'apartat anterior serà molt útil per l'equip, però té l'inconvenient que si no s'està les vint-i-quatre hores pendent, no es pot aprofitar tot el seu potencial. Per aquest motiu es necessitarà un sistema d'alertes que enviï notificacions a l'equip de seguretat tant bon punt com el sistema detecti un indicador d'atac.

Aquestes alertes, posteriorment poden relacionar-se amb casos, els quals un cop resolts poden utilitzar-se com a indicador de compromís, ja que la informació que els composa provindrà d'evidències proporcionades per les dades.

Finalment, el servei Kibana conté un parell de dashboards a la capa gratuïta relacionats amb les alertes i els casos oberts que seran útils a l'hora de fer el seguiment de qualsevol possible incident.

3.3.1. Alertes

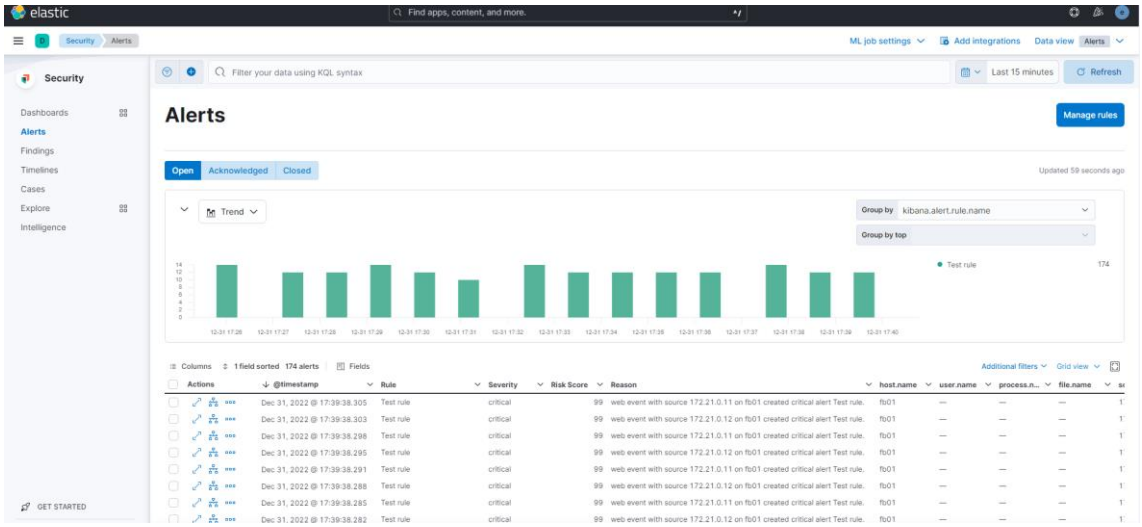
Les alertes es regeixen per regles que utilitzen les dades capturades per controlar condicions. Per defecte Kibana permet instal·lar-ne unes per defecte, però com el projecte s'implementa en un ordinador local i s'utilitza Nginx com a servidor web, es crearà una regla que generi una alerta cada cop que el servidor Nginx retorni de resposta un codi HTTP 404.

Per crear-la s'haurà d'escollir el tipus de regla com "Custom query" i utilitzar l'índex "filebeat-*". Per la cerca serà suficient en seleccionar el contenidor que té el nom Nginx i el codi de resposta HTTP 404.

container.name : "nginx" and http.response.status_code: 404

Com s'ha comentat, quan una regla s'activa fa saltar una alerta i al fer-ho es pot enviar una notificació a diverses plataformes, enviar un email o escriure un document a un índex del servei ElasticSearch. Com la capa gratuïta només permet escriure el document, es seleccionarà aquest i només caldrà estar pendent de comprovar-lo cada N segons amb algun script programat.

Un cop guardada la regla començaran a aparèixer alertes ja que els clients envien peticions a URLs que no existeixen.

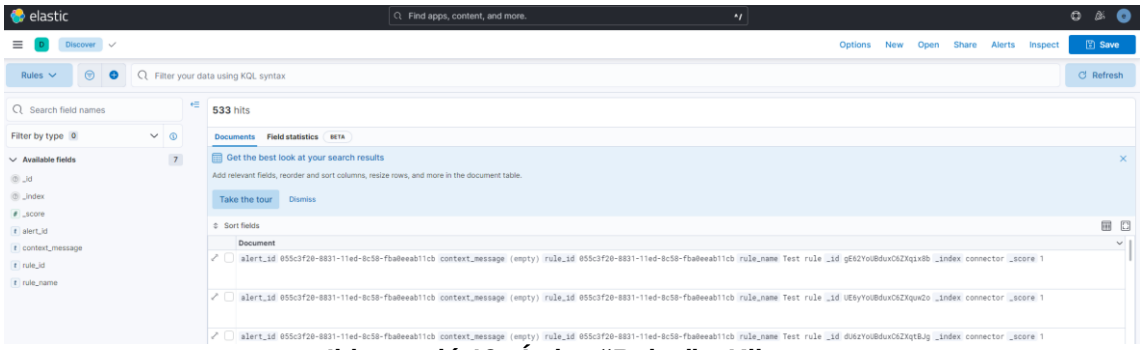


Il·lustració 18 - Alertes a Kibana

Cada alerta permet revisar les dades que l'han feta activar i una sèrie de informació extra relacionada amb la pròpia alerta. A més també permet assignar-la a un cas existent o crear-ne un de nou per ella.

Un cop tractada l'alerta, aquesta pot ser modificada d'estat per tal de donar-la per tancada, en procés o deixar-la oberta per més endavant. Aquí l'equip de seguretat serà l'encarregat de gestionar-les.

En el cas que es volgués revisar les alertes disponibles des d'un sistema extern, només caldria revisar l'índex "Rules", que es genera un cop s'envia la primera alerta.

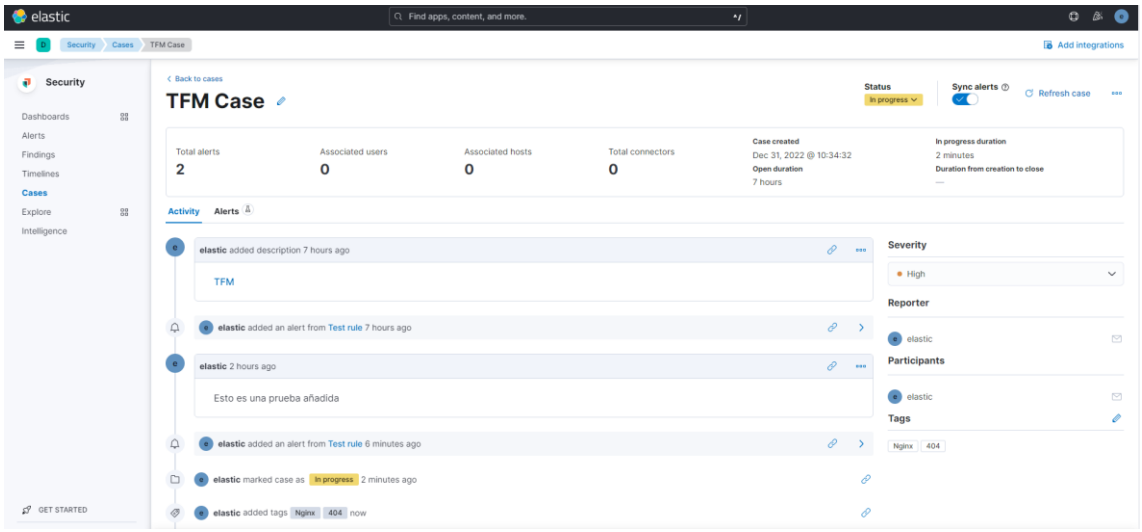


Il·lustració 19 - Índex "Rules" a Kibana

3.3.2. Casos

Els casos són objectes que tenen associades alertes i inclouen informació com ara qui l'ha reportat, el que se n'encarrega, esdeveniments que han anat succeint i s'han anat apuntant, etc. Tota aquesta informació conforme un cas, que un cop tancat podria ser un indicador de compromís, ja que contindria les evidències de com s'ha produït una incidència i com s'ha mitigat.

Pel projecte s'ha creat un cas i se li han assignat dues alertes:



Il·lustració 20 - Exemple de cas de seguretat a Kibana

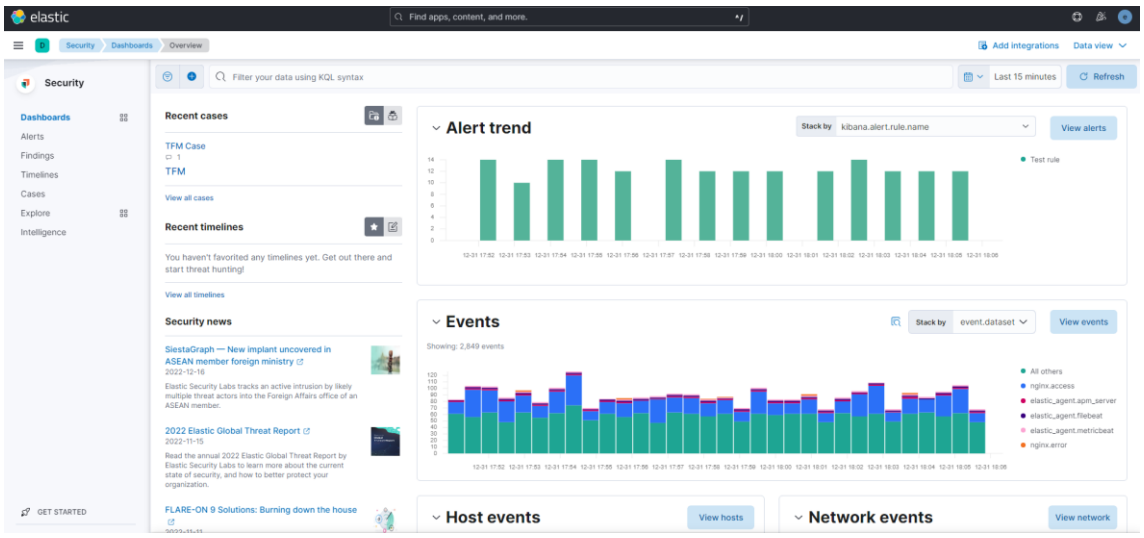
Per últim, quan es genera una alerta aquesta pot associar-se al cas, però si es necessita adjuntar més evidències, Kibana permet generar un fragment temporal on mitjançant una cerca es poden utilitzar els resultats com a tal.

3.3.3. Dashboards

En el moment de la implementació d'aquest projecte (versió 8.5.3) hi ha dos dashboards disponibles que es puguin utilitzar amb l'aplicació Docker:

- Overview o visió global: Presenta un resum d'alertes i esdeveniments que succeeixen als sistemes.

- Detecció i resposta: Mostra un resum de les alertes en curs tant de forma global com per sistema o usuari assignat.



Il·lustració 21 - Dashboard Overview de seguretat a Kibana

Els dos serveixen per donar a l'equip de seguretat una visió general de l'estat del sistema.

4. Conclusions i treballs futurs

Un cop implementat el SIEM i capturant les dades dels serveis, queda clar que la pila ELK és un software molt potent a l'hora de crear una solució. Si bé és cert que al mercat hi ha organitzacions amb software específic per sistemes SIEM, d'opcions gratuïtes n'hi ha ben poques i que a l'hora sigui tant versàtil menys encara.

En aquest treball s'ha optat per implementar el SIEM mitjançant Docker, fet que ha provocat que la instal·lació sigui més senzilla, però a l'hora utilitzi més recursos. Aquest punt és molt interessant de valorar ja que en el cas d'utilitzar aquest SIEM a producció en instàncies al cloud públic, s'hauria de revisar l'estratègia de desplegament i les necessitats que haurien de tenir aquestes instàncies. D'altra banda cal valorar també l'estalvi de cost en manteniment que implica implementar el SIEM tal com s'ha dut a terme.

Si bé és cert que la implementació ha donat resultats molt positius, seria molt interessant poder fer proves amb més nodes i serveis per obtenir moltes més dades ja que com s'ha comentat, Kibana permet integracions amb molts serveis de tercers mitjançant els agents i si no s'utilitzen, els camps es generen igualment al sistema, però queden marcats pel propi sistema com a buits. No és que sigui un problema, però sí que genera soroll.
Tot i així, aquest petit inconvenient es pot resoldre modificant la vista de dades de Kibana que utilitzi el índex amb les dades.

Tot i haver implementat el SIEM amb software de codi obert, Elastic es reserva unes quantes funcionalitats per fer-les de pagament, sobretot les relacionades amb la connectivitat amb altres sistemes. S'ha pogut veure com en el treball s'ha hagut de emmagatzemar el resultat de les alertes a un nou índex, però el cas ideal seria enviar-ho directament a l'equip de resposta per agilitzar tot el procés.

Finalment, un dels temes més importants en un sistema com aquest serà la privacitat de les dades capturades. En la posada en marxa s'han capturat dades del servidor NGINX i entre elles hi ha les IPs de les connexions, les cookies de les peticions HTTP i altres dades sensibles en cas d'enviar-les mitjançant aquest servei. Això provoca que s'hagi d'establir una política de govern de dades específicament pel SIEM ja que aquest inclou molta informació que pot identificar tant les pròpies persones de la organització on és instal·lat com clients que accedeixin a serveis monitoritzats per aquest.

Repositori de codi on trobar la implementació del SIEM d'aquest treball:
https://github.com/peregarcia/tfm-siem

4.1. Assoliment d'objectius

Els objectius del treball es van dividir en dos grans grups, els relacionats amb la investigació i els del propi SIEM un cop implementat.

Els objectius d'investigació han estat assolits ja que eren indispensables abans de dur a terme la implementació: com funciona un SIEM, comprovar que la pila ELK compleixi tots els requeriments per donar una solució en aquesta necessitat, com gestiona els indicadors de compromís, les diferents fonts de dades i captura d'aquestes i com lligar-ho tot per donar la informació a l'equip de resposta.

Per part dels objectius del SIEM, s'ha implementat la infraestructura i assolit els requeriments pel seu funcionament en un entorn virtual utilitzant Docker que pot ser escalat o portat a altres entorns. A més en cas de necessitar més recursos la pila ELK permet crear clústers del servei ElasticSearch per augmentar la capacitat del sistema.

Finalment l'objectiu d'analitzar les dades en temps real per captar indicadors d'atac i posteriorment generar indicadors de compromís també ha estat dut a terme amb les funcionalitats proposades.

4.2. Seguiment de la planificació

S'ha arribat a l'objectiu final d'implementar el SIEM, però hi han hagut problemes que tot i haver-se tingut en compte en la planificació, han afectat al treball. A més la planificació no s'ha pogut seguir com es va definir inicialment degut a un inconvenient personal que va provocar un retràs a la planificació i finalment va ser resolt durant els dies festius següents.

4.2.1. Càrrega de dades amb Logstash

Si bé el servei Logstash s'ha implementat correctament, a l'hora de capturar les dades i generar el pipeline, el servei encarregat de la captura no arribava a funcionar bé del tot. És possible que fos alguna limitació del sistema o recursos, però com les dades també es capturaven des dels agents, es va optar per continuar descartant qualsevol captura de dades amb Logstash.

4.2.2. Xifrat a l'intercanvi de dades entre la pila ELK i serveis

Les dades capturades i enviades al SIEM permeten identificar dispositius i en última instància individus. Per aquest motiu el govern de la dada dins l'organització ha de ser molt estricte en el cas del transport i emmagatzematge d'aquesta.
Durant la implementació es va intentar utilitzar certificats per assolir aquest fet, però va ser impossible. Quan s'utilitzaven, els serveis no aconseguien connectar-se entre ells provocant que la implementació del projecte no fos possible.
Finalment, s'ha deixat com a possible treball futur i el SIEM es connecta mitjançant peticions no xifrades HTTP.

4.2.3. Tractament de les dades

L'impacte ètics-social del projecte és la privacitat de les dades capturades ja que aquestes es poden fer servir per identificar individus. Es va plantejar resoldre l'impacte mitjançant xifratges, filtres i l'ús de hash per emmascarar les dades més

sensibles, però per falta de temps no ha estat possible i no s'ha fet cap tractament a aquestes.

4.3. Impactes ètics-socials

Tal i com es va definir inicialment, la protecció de les dades és un pilar fonamental per aquest tipus de sistemes, però no s'han pogut dur a terme. S'ha deixat exposat com a possible treball futur.

4.4. Treballs futurs

Un cop implementat el SIEM i en funcionament, es plantegen una sèrie de possibles millores pel projecte:

4.4.1. Captura de dades a fonts de dades externes mitjançant Logstash

Com s'ha comentat, ha quedat pendent aprofitar el servei Logstash.

4.4.2. Implementació de privadesa de les dades i política de govern de la dada per una organització que faci servir un SIEM

Per tal de millorar la gestió de les dades i evitar la identificació d'individus mitjançant aquestes, es proposa dur a terme un pla de govern de la dada i mecanismes per implementar la privacitat dels individus a les dades capturades o enviades al SIEM.

4.4.3. Afegir seguretat amb certificats a les connexions al SIEM

Un dels punts conflictius del projecte ha de corregir-se i utilitzar els certificats per xifrar les connexions entre els diferents serveis que componen el SIEM.

4.4.4. Utilitzar Kubernetes permonitoritzar els contenidors

Docker és una de les diferents aplicacions que permeten virtualitzar serveis a nivell de Sistema Operatiu. A més d'altres, existeix Kubernetes que és àmpliament utilitzat i també permet implementar un SIEM amb la pila ELK ja que disposa de suport natiu per capturar les dades de la màquina on s'instal·len els agents.

4.4.5. Comparar la versió open source amb Wazuh

Wazuh és una solució SIEM de codi obert que inclou moltes funcionalitats i utilitza OpenSearch[26] com a base de dades.

4.4.6. Utilitzar STIX[27] per descriure els IOCs generats a Kibana

STIX és un llenguatge d'intercanvi d'informació estructurada pensat en descriure esdeveniments relacionats amb la ciberseguretat mitjançant la relació entre objectes del domini.

[26] Projecte OpenSearch https://opensearch.org/
[27] STIX https://oasis-open.github.io/cti-documentation/stix/intro

5. Bibliografia

- IOCs V/S IOAs – Diferencias claves para la ciberseguridad [20-10-2022] (https://www.cronup.com/iocs-v-s-ioas-diferencias-claves-para-la-ciberseguridad/)
- ¿QUÉ SON LOS INDICADORES DE ATAQUE (IOA)? DIFERENCIAS CON LOS IOC [21-10-2022] (https://ciberseguridad.com/guias/prevencion-proteccion/indicadores-ataque-ioa/)
- ¿Qué es el ELK Stack? [1-11-2022] (https://www.elastic.co/es/what-is/elk-stack)
- What is Elastic Observability? [1-11-2022] (https://www.elastic.co/guide/en/observability/8.5/observability-introduction.html)
- Docker Docs [1-11-2022] (https://docs.docker.com/)
- Docker Hub [1-11-2022] (https://hub.docker.com/)
- Kibana [2-11-2022] (https://www.elastic.co/guide/en/kibana/8.5/introduction.html)
- ElasticSearch [2-11-2022] (https://www.elastic.co/guide/en/elasticsearch/reference/8.5/elasticsearch-intro.html)
- LogStash [2-11-2022] (https://www.elastic.co/guide/en/logstash/8.5/introduction.html)
- Beats [2-11-2022] (https://www.elastic.co/es/beats/)
- APM Python Agent [2-11-2022] (https://www.elastic.co/guide/en/apm/agent/python/current/getting-started.html)
- Elastic Fleet [2-11-2022] (https://www.elastic.co/guide/en/fleet/current/fleet-overview.html)
- Elastic APM [2-11-2022] (https://www.elastic.co/guide/en/apm/guide/8.5/apm-quick-start.html)
- Create an agent policy without using the UI [2-11-2022] (https://www.elastic.co/guide/en/fleet/current/create-a-policy-no-ui.html)
- Writing your first Django app, part 1 [23-11-2022] (https://docs.djangoproject.com/en/4.1/intro/tutorial01/)
- Monitoring Python Applications with Elastic APM [12-12-2022] (https://towardsdatascience.com/monitoring-flask-fastapi-python-applications-with-elastic-apm-33237a39d7b6)
- Elastic APM Server/Agent Scale Model With Docker Compose And Python DJango [13-12-2022] (https://l10nn.medium.com/elastic-apm-server-agent-scale-model-with-docker-compose-and-python-django-2e8b0eccf183)
- Threat Hunting for IOCs with the Elastic Stack [27-12-2022] (https://www.youtube.com/watch?v=yY24yGEe01g)
- Elastic Security para protección contra amenazas automatitzada [27-12-2022] (https://www.elastic.co/es/security/automated-threat-protection)

5.1. Referències

[1] Icones productes Elastic. “So iconic: A story about a team, a dream, and a visual hierarchy”: https://www.elastic.co/es/blog/redesigning-product-logos-and-icons-while-building-a-design-hierarchy-at-elastic

[2] Icona Docker: https://www.docker.com/

[3] Icona Django Project: https://www.djangoproject.com/

[4] Icona PostgreSQL: https://www.postgresql.org/

[5] Icona Nginx: https://www.nginx.com/

6. Annexos

6.1. Configuració de Kibana per connectar el servidor Fleet i l'agent APM:

```yaml
xpack.fleet.agents.fleet_server.hosts: [ http://fleet-server:8220 ]

xpack.fleet.outputs:
  - id: fleet-default-output
    name: default
    type: elasticsearch
    hosts: [ http://es01:9200 ]
    is_default: true
    is_default_monitoring: true

xpack.fleet.packages:
  - name: fleet_server
    version: latest
  - name: system
    version: latest
  - name: elastic_agent
    version: latest
  - name: apm
    version: latest

xpack.fleet.agentPolicies:
  # Política Servidor Fleet
  - name: "Fleet Server Policy"
    id: fleet-server-policy
    description: "Política per Fleet Server"
    monitoring_enabled:
      - logs
      - metrics
    package_policies:
      - name: fleet_server-1
        package:
          name: fleet_server
      - name: system-1
        package:
          name: system
      - name: elastic_agent-1
        package:
          name: elastic_agent
  # Política APM
  - name: "Agent Policy APM Server"
    id: agent-policy-apm-server
    description: "Política per servei APM"
```

```
    monitoring_enabled:
      - logs
      - metrics
    package_policies:
      - name: system-1
        package:
          name: system
      - name: elastic_agent-1
        package:
          name: elastic_agent
      - name: apm-1
        package:
          name: apm
        inputs:
          - type: apm
            vars:
              - name: host
                value: 0.0.0.0:8200
              - name: url
                value: http://apm02:8200
```

6.2. Script de peticions a l'aplicació web dels clients

Instruccions contenidor docker:

```
client01:
    container_name: client01
    restart: always
    build: ./client
    command: /bin/bash client_demo.sh
    depends_on:
      - nginx
    networks:
      - elk
```

Arxiu Dockerfile:

```
FROM ubuntu

USER root
RUN apt update && apt upgrade -y
RUN apt install curl -y
COPY client_demo.sh client_demo.sh
```

Script:

```
#!/bin/bash

counter=0
while true
do
    counter=$(($counter + 1))

    # Home
    if [[ $(($counter % 5)) == 0 ]]
    then
        curl http://nginx/python_demo/ >> home.html
    fi

    # Error
    if [[ $(($counter % 10)) == 0 ]]
    then
        curl http://nginx/python_demo/tfm-pere >> error404.html
    fi

    counter=$(($counter + 1))
    sleep 1
done
```