Universitat Oberta de Catalunya

uoc.edu

Desplegar l'eina "Zeek IDS" i la seva posterior explotació per a l'anàlisi d'activitats sospitoses a la xarxa

Adrià Adell Barbarà

Màster Universitari en Seguretat de les TIC

Anàlisi de dades

Borja Guaita Perez

Victor Garcia Font

2 de juny de 2020

FITXA DEL TREBALL FINAL

Títol del treball:	*Desplegar l'eina "Zeek IDS" i la seva posterior explotació per a l'anàlisi d'activitats sospitoses a la xarxa*
Nom de l'autor:	*Adrià Adell Barbarà*
Nom del consultor/a:	*Borja Guaita Perez*
Nom del PRA:	*Victor Garcia Font*
Data de lliurament (mm/aaaa):	*06/2020*
Titulació o programa:	*Màster Universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions*
Àrea del Treball Final:	*Anàlisi de dades*
Idioma del treball:	*Català*
Paraules clau	*Anàlisi de xarxa,* *IDS,* *Seguretat*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Amb la creixent necessitat de protegir la informació (confidencialitat, integritat i disponibilitat), la finalitat d'aquest treball és detectar diferents problemes de seguretat d'una xarxa local mitjançant un sistema de detecció d'intrusos (IDS) i mostrar-los en uns panells de control (dashboards) per permetre una reacció precoç a les amenaces detectades.

En aquest treball s'estudien no solament les capacitats de detecció que ofereix Zeek IDS (atacs de força bruta, injecció SQL, etc.), sinó que juntament amb la integració amb ELK Stack (Elasticsearch, Logstash i Kibana), som capaços d'enriquir les dades, permetent la detecció de connexions a llocs web fraudulents o la descàrrega de fitxers perillosos en temps real, amb fonts de detecció actualitzades gràcies a les integracions amb Intel Critical Stack i les consultes in-line a VirusTotal. Per a l'elaboració del projecte, s'ha realitzat la implantació de Zeek IDS en una Raspberry Pi 4 i la instal·lació d'ELK Stack en un servidor virtual Ubuntu Server, a més de la creació de quatre dashboards de seguretat.

Mitjançant unes proves de funcionament, es demostra que la instal·lació d'un IDS conjuntament amb ELK, eleva considerablement el nivell de seguretat de qualsevol xarxa. Finalment, les conclusions han sigut satisfactòries, ja que s'assoleix la detecció en temps real de diferents activitats sospitoses a la xarxa, i que gràcies als dashboards de seguretat permetran veure les amenaces i realitzar una intervenció en el casos necessaris.

Abstract (in English, 250 words or less):

With the growing need to protect information (confidentiality, integrity and availability), the purpose of this work is to detect different security problems in a local network by using an intrusion detection system (IDS) and display them in dashboards to allow an early reaction to detected threats.

In this work we study not only the detection capabilities offered by Zeek IDS (brute force attacks, SQL injection, etc.), but together with the integration with ELK Stack (Elasticsearch, Logstash and Kibana), we are also able to enrich the data, in order to detect the connections to fraudulent websites or downloads of dangerous files in real time, with up-to-date detecting sources thanks to the integrations with Intel Critical Stack and in-line queries to VirusTotal. To develop this project, the implementation of Zeek IDS was made on a Raspberry Pi 4 and the installation of ELK Stack was on a virtual Ubuntu Server, additionally the creation of four security dashboards.

Through performance tests, it is shown that the installation of an IDS combined with ELK, significantly raises the level of security in any network. Finally, the results have been satisfactory, as it achieves the detection in real time of different suspicious activities in the network, and that, thanks to the security dashboards, will allow us to see the threats and carry out an intervention when necessary.

Índex

Llista de figures

1. Introducció

1.1. Context i justificació del treball

Sovint, les propietats físiques es protegeixen per evitar robatoris, intrusions o altres activitats potencialment destructives. Algunes llars, per exemple, estan equipades amb alarmes que poden detectar intrusos i notificar (a l'empresa del servei d'alarmes, al propietari, a les autoritats...) quan ocorre una entrada il·legal o hi ha qualsevol altre incidència. Aquestes mesures són necessàries per assegurar la integritat de les llars i, per tant, la seguretat dels seus propietaris.

I amb aquesta mateixa mentalitat, perquè no assegurar la integritat i la seguretat als sistemes de informació i comunicacions? Gràcies al flux de dades i informació, i a la xarxa que proporciona Internet, existeixen usuaris maliciosos que busquen objectius vulnerables com sistemes no actualitzats, sistemes infectats o xarxes executant serveis no segurs.

Si una organització només disposa d'un programari d'antivirus en els equips clients, l'usuari i l'organització molt probablement sabran que han tingut un incident de seguretat quan ja sigui tard. Una situació comú és quan un usuari vol obrir un document que està e la xarxa i aquest no es pot llegir perquè està xifrat. Això indica que aquesta organització ha patit un atac *ransomware*. Cap sistema de seguretat entre Internet i l'equip de l'usuari ha funcionat. Així que, com en el món físic, cal implementar els les xarxes algun tipus d'alarmes per notificar als responsables del sistemes (administradors, equip de seguretat...) que ha passat una activitat sospitosa com per exemple, connexions a dominis amb mala reputació, descàrrega d'arxius maliciosos o possibles atacs de força bruta.

Per intentar minimitzar els riscos en la seguretat de la informació, com són la integritat, la confidencialitat i la disponibilitat, és molt important poder detectar activitats sospitoses i amenaces que passin a la xarxa en temps real, les quals puguin ser les desencadenants d'un ciberincident. Els sistemes dissenyats per a la detecció d'intrusos s'anomenen IDS (*Intrusion Detection System*).

Per donar resposta a aquesta necessitat, la finalitat d'aquest Treball de Fi de Màster és poder disposar en una organització d'una solució capaç d'analitzar el trànsit de la xarxa, que actuï com un monitor de seguretat per a la inspecció en profunditat, a la recerca d'activitats sospitoses, i que aquestes puguin ser representades de forma gràfica en un panell de control per poder actuar ràpidament davant un incident de seguretat.

1.2. Objectius del treball

L'objectiu principal d'aquest Treball de Final de Màster és desplegar l'eina de detecció d'intrusos "Zeek IDS" per a poder detectar possibles activitats sospitoses en una xarxa, i la posterior visualització dels esdeveniments i les activitats potencialment malicioses en un dashboard de forma clara mitjançant amb els components de "ELK Stack". El monitoratge de la xarxa serà en temps real (encara que serà possible de forma offline mitjançant *dumps* de tràfic de xarxa), per tal de poder analitzar aquestes activitats i prendre les mesures

oportunes el més aviat possible. La informació proporcionada pel sistema ajudarà a detectar quins equips realitza aquestes activitats, possibles víctimes o què intentava.

Per aconseguir aquest objectiu, prèviament cal haver assolit una sèrie d'objectius específics o fites durant el seu desenvolupament:

- Realitzar una anàlisi de requeriments per a la implantació dels equips maquinari (físics, virtuals o híbrid) necessaris per a la instal·lar de la solució.
    - En cas que part de la implementació sigui virtual, estudiar i aprendre el funcionament del software “Oracle VirtualBox”.
    - En cas que part de la implementació sigui física, estudiar i aprendre el funcionament del hardware “Raspberry Pi 4”.
- Estudiar i aprendre el funcionament de l'eina de detecció d'intrusos "Zeek IDS": instal·lació, regles, etc.
- Estudiar i aprendre el funcionament de "ELK Stack", formant per:
    - Elasticsearch: motor de cerca i analítica.
    - Logstash: pipeline de processament de dades que recull dades, les transforma i després les envia a Elasticsearch.
    - Kibana: permet visualitzar les dades en quadres i gràfics amb Elasticsearch.
- Instal·lar "Zeek IDS" i "ELK Stack" en el maquinari escollit, segons el disseny.
- Integrar satisfactòriament les solucions "Zeek IDS" i "ELK Stack", per poder visualitzar les amenaces en un entorn gràfic.
- Realitzar un pla de proves per comprovar el correcte funcionament.
- Realitzar la memòria del TFM.

1.3. Enfocament i mètode seguit

L’estratègia per dur a terme el treball consisteix en fer una instal·lació de dos programes existents en el mercat. Per una banda, un programa amb funcionalitat de IDS, capaç d’inspeccionar la xarxa i, per l’altra un programa amb funcionalitats de SIEM, capaç de recollir *logs* del IDS, transformar-los i mostrar-los gràficament. S’hauran d’integrar i adaptar-los a les necessitats específiques d’aquest projecte.

No es realitza un estudi de mercat sobre les diferents solucions d'IDS i SIEM existents, ja que els programes proposats i en els quals es basa aquest treball ja estan definits, com són "Zeek IDS" per a l'anàlisi de xarxa i "ELK Stack" per a visualitzar les alertes.

La metodologia que se seguirà durant el desenvolupament per a poder complir els objectius proposats i validar el resultar obtingut, serà la distribució de la feina en quatre etapes principalment. Les etapes per al desenvolupament del TFM, són:

- Primera etapa:
    - Realitzar la introducció i els objectius de treball.
    - Realitzar una planificació de la feina al llarg del semestre.
- Segona etapa:
    - Investigar les eines que s’utilitzararan per dur a terme el projecte i la interacció entre elles.
        - Zeek IDS.

        - ELK Stack: Elasticsearch, Logstash i Kibana.
    - Realitzar un disseny de la solució a implementar, com l'arquitectura, tipus d'equips a utilitzar (físics, virtuals o ambdós), etc.
        - Estudi de la eina Oracle VirtualBox.
        - Estudi de Raspberry Pi.
    - Recopilar la informació obtinguda.
- Tercera etapa:
    - Implantar la solució a partir del disseny realitzat en l'etapa anterior.
    - Recopilar la documentació necessària sobre la implementació.
    - Realitzar i documentar les diferents proves per provar la solució.
- Quarta etapa:
    - Elaborar la memòria del treball.
    - Preparar una presentació en dispositives sobre el treball.
    - Realitzar la presentació virtual del TFM (vídeo).
    - Defensa del TFM de forma virtual.

1.4. Planificació del treball

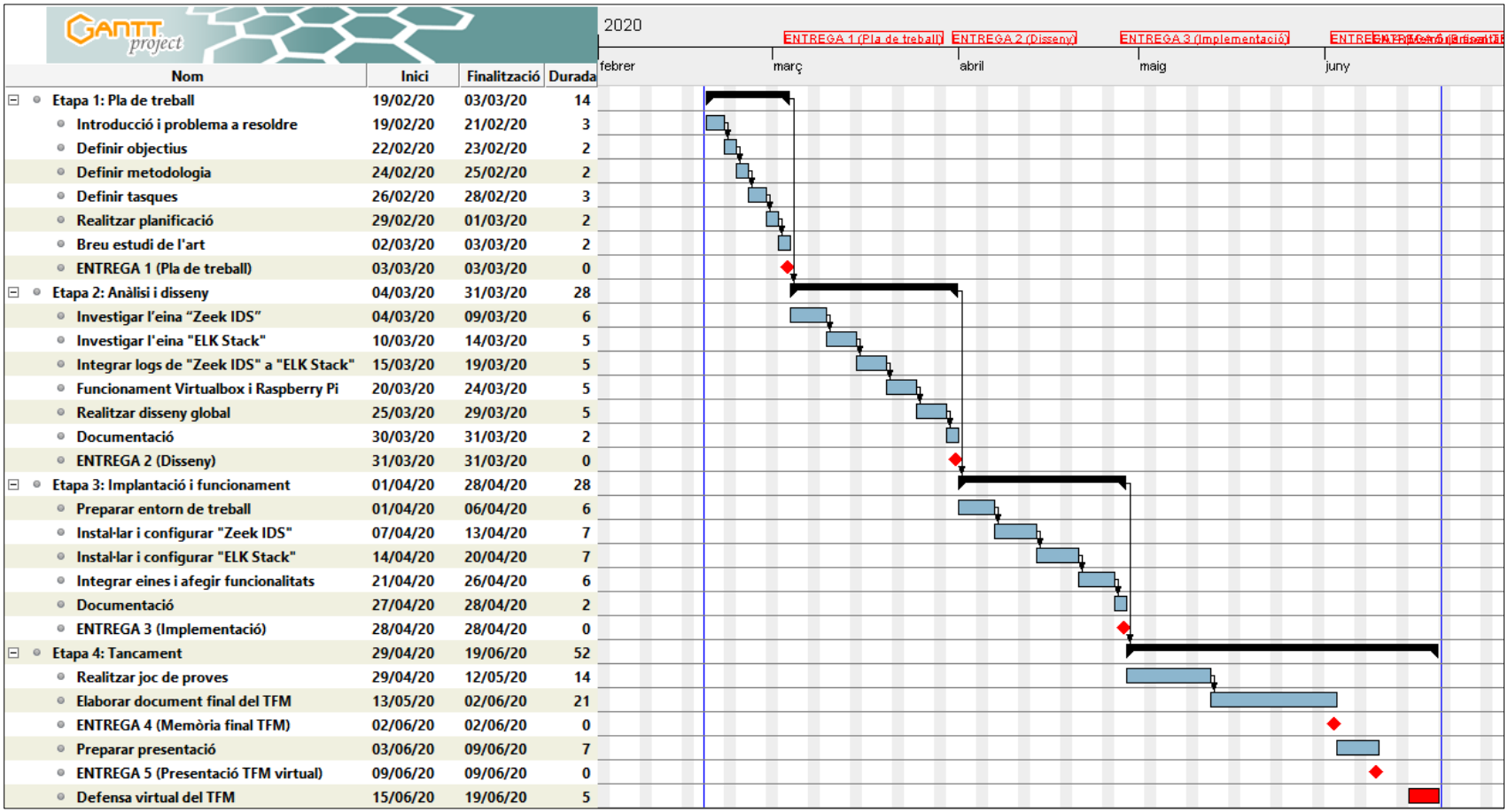
Serà necessari disposar d'un equip informàtic capaç de poder suportar la instal·lació de N màquines virtuals per la realització del treball. Si és el cas, també serà necessària l'adquisició d'un dispositiu hardware per connectar-lo a la xarxa amb capacitat d'executar "Zeek IDS", com per exemple una Raspberry Pi 4. En aquest últim cas, segurament sigui necessari adquirir electrònica de xarxa per tal de que el dispositiu pugui estar a la xarxa en mode promiscu, per tal de capturar tot el trànsit que hi circula.

El llistat de tasques que s'han de realitzar per assolir els objectius descrits, per cada etapa, són:

- Primera etapa: Pla de treball.
    - Estudiar i explicar el problema a resoldre.
    - Definir els objectius.
    - Definir la metodologia a seguir.
    - Definir un llistat de tasques.
    - Realitzar la planificació temporal.
    - ➢ **FITA (PAC 1)**: **Elaborar i entregar un document on reculli la feina realitzada.**
- Segona etapa: Anàlisi i disseny.
    - Investigar l'eina "Zeek IDS".
        - Estudiar el funcionament general.
        - Estudiar els requisits.
        - Investigar si l'eina pot satisfer els requeriments, i com fer-ho.
            - Estudiar com personalitzar l'eina.
            - Estudiar i realitzar scripts en Python.
    - Investigar l'eina "ELK Stack"
        - Estudiar el funcionament d'Elasticsearch.
        - Estudiar el funcionament de Logstash.
        - Estudiar el funcionament de Kibana.
    - Investigar el funcionament de Oracle VirtualBox i/o Raspberry Pi.
    - Investigar interacció entre els elements "Zeek IDS" i "ELK Stack".

    - Disseny global de la solució.
    - Recopilar la informació obtinguda.
    - **FITA (PAC 2): Elaborar i entregar un document on reculli la feina realitzada fins al moment.**
- Tercera etapa: Implantació i funcionament.
    - Implantar la solució a partir del disseny realitzat en l'etapa anterior.
        - Muntar l'entorn de treball
        - Instal·lació de programes "Zeek IDS" i "ELK Stack".
    - Recopilar la documentació necessària sobre la implementació.
    - **FITA (PAC 3): Elaborar i entregar un document on reculli la feina realitzada fins al moment.**
- Quarta etapa: Tancament del projecte.
    - Realitzar i documentar las diferents proves per comprovar la instal·lació.
    - Elaborar document on reculli la feina realitzada en tot el treball.
    - **FITA (PAC 4): Entregar la memòria final del TFM.**
    - Crear document PowerPoint per la presentació en diapositives del treball.
    - Preparar exposició en vídeo per realitzar la presentació virtual.
    - **FITA: Realitzar la presentació virtual del TFM.**
    - **FITA: Realitzar la defensa virtual del TFM.**

En la Figura 1-1 es mostra la planificació temporal de les taques i fites descrites, mitjançant un diagrama de Gantt.



Nom	Inici	Finalització	Durada
Etapa 1: Pla de treball	19/02/20	03/03/20	14
Introducció i problema a resoldre	19/02/20	21/02/20	3
Definir objectius	22/02/20	23/02/20	2
Definir metodologia	24/02/20	25/02/20	2
Definir tasques	26/02/20	28/02/20	3
Realitzar planificació	29/02/20	01/03/20	2
Breu estudi de l'art	02/03/20	03/03/20	2
ENTREGA 1 (Pla de treball)	03/03/20	03/03/20	0
Etapa 2: Anàlisi i disseny	04/03/20	31/03/20	28
Investigar l'eina "Zeek IDS"	04/03/20	09/03/20	6
Investigar l'eina "ELK Stack"	10/03/20	14/03/20	5
Integrar logs de "Zeek IDS" a "ELK Stack"	15/03/20	19/03/20	5
Funcionament Virtualbox i Raspberry Pi	20/03/20	24/03/20	5
Realitzar disseny global	25/03/20	29/03/20	5
Documentació	30/03/20	31/03/20	2
ENTREGA 2 (Disseny)	31/03/20	31/03/20	0
Etapa 3: Implantació i funcionament	01/04/20	28/04/20	28
Preparar entorn de treball	01/04/20	06/04/20	6
Instal·lar i configurar "Zeek IDS"	07/04/20	13/04/20	7
Instal·lar i configurar "ELK Stack"	14/04/20	20/04/20	7
Integrar eines i afegir funcionalitats	21/04/20	26/04/20	6
Documentació	27/04/20	28/04/20	2
ENTREGA 3 (Implementació)	28/04/20	28/04/20	0
Etapa 4: Tancament	29/04/20	19/06/20	52
Realitzar joc de proves	29/04/20	12/05/20	14
Elaborar document final del TFM	13/05/20	02/06/20	21
ENTREGA 4 (Memòria final TFM)	02/06/20	02/06/20	0
Preparar presentació	03/06/20	09/06/20	7
ENTREGA 5 (Presentació TFM virtual)	09/06/20	09/06/20	0
Defensa virtual del TFM	15/06/20	19/06/20	5

FIGURA 1-1: DIAGRAMA DE GANTT DEL TREBALL

1.5. Estat de l'art

Encara que les dues eines en les que es basa aquest treball (Zeek IDS i ELK Stack) són un requeriment, es fa una cerca dels principals productes existents en els sistemes IDS i SIEM (de codi obert).

Un sistema de detecció d'intrusions (IDS) és un programa de detecció d'accessos no autoritzats a una xarxa o en un equip. A continuació es mostren els tres principals productes destinats a la monitorització d'una xarxa:

- **Snort**: Es tracta d'un sistema de detecció d'intrusions (IDS) i un sistema de protecció contra intrusions (IPS) creat el 1998. És senzill d'utilitzar ja que la instal·lació no es complicada, i es basa en l'escriptura de regles per a detectar les amenaces. En les regles es defineixen quines característiques úniques del trànsit de xarxa es vol detectar per a desencadenar una alerta. Té una comunitat molt gran.
- **Suricata**: Es podria considerar una evolució de Snort que va néixer el 2009 creat per l'OISF, ja que ofereix una compatibilitat amb les regles Snort, però és capaç de processar més regles en xarxes amb molt més tràfic, ja que introdueix la capacitat de crear molts fils d'execució (*multi-threading*). La instal·lació és més complicada que Snort i no té una comunitat tant gran.
- **Zeek IDS** (abans, conegut com Bro, desenvolupat el 1995): És un sistema de detecció d'intrusos enfocat a l'anàlisi de la xarxa que busca amenaces específiques i pot desencadenar alertes. No funciona com Snort i Suricata, ja que aquests es basen en regles per detectar excepcions. El funcionament més freqüent és utilitzar-lo per guardar regitres a llarg termini de peticions i resultats HTTP, etc.

Un sistema de gestió d'informació i esdeveniments de seguretat (SIEM) és un sistema que centralitza l'emmagatzematge i la interpretació de les dades rellevants de seguretat. En realitat no es tracta d'una sola eina, sinó que es composa de múltiples components d'agregació, processament, anàlisi i presentació. A continuació es mostren els tres principals productes destinats a la gestió d'informació i esdeveniments de seguretat:

- **ELK Stack**: Compost per Elasticsearch, Logstash i Kibana, és l'eina d'agregació més popular del mercat. L'empresa Elastic treballa i manté els tres components. Elasticsearch és un motor de cerca basat en NoSQL. Logstash és un sistema de pipeline de registre que ingereix les dades, les transformar i les carrega a Elasticsearch. Kibana és la capa superior per poder visualitzar les dades contingudes a l'Elasticsearch. L'utilitzen empreses com Microsoft, Netflix, Facebook, LinkedIn o Cisco.
- **Graylog**: Va ser creat el 2010, però ha sigut recentment quan ha augmentat la seva popularitat, encara que està molt per darrera de ELK Stack. Graylog utilitza Elasticsearch, MongoDB i Graylog Server per funcionar. El seu punt fort és que redueix la latència en l'agragació de logs gràcies al seu flux de dades. També admet la representació d'adreces IP en un mapa (característica també disponible a Kibana).
- **Fluentd**: És un software de recollida de dades desenvolupat per Treasure Data. Està escrit en el llenguatges de programació C i Ruby. Recentment s'ha convertit en un reemplaçament habitual de Logstash en moltes instal·lacions i és habitual als entorns de Kubernetes a causa dels seus baixos requeriments de memòria.

L'elecció de les eines "Zeek IDS" i "ELK Stack" per a realització del projecte són adients, gràcies a totes les funcionalitats que ofereixen, que són de codi obert, i disposen de dues grans comunitats d'usuaris.

1.6. Breu sumari de productes obtinguts

Els productes obtinguts del Treball de Final de Màster són, per una banda la memòria final del projecte i una presentació en format vídeo de la defensa del projecte. En aquests documents s'explica en què consisteix el treball i com s'ha dut a terme. Amb la realització del treball s'ha pogut obtenir una visió global del funcionament d'un IDS de xarxa com és Zeek, d'un dispositiu hardware com la Raspberry Pi, la virtualització de servidors, la recol·lecta i visualització de dades amb diferents productes del ELK Stack, i les integracions amb solucions de tercers per enriquir la detecció d'anomalies a la xarxa.

Per la realització d'una avaluació continua, s'han realitzat un seguit d'entregables, consistents en:

- PAC 1: Introducció i pla de treball
- PAC 2: Implantació i funcionament
- PAC 3: Investigació i disseny
- PAC 4: Memòria final
- PAC 5: Presentació virtual

1.7. Breu descripció dels capítols de la memòria

Al capítol 1, es realitza una introducció al treball, exposant-ne el context i la seva justificació, així com els objectius, l'enfocament, el mètode seguit, la planificació temporal i un breu estat de l'art.

Al capítol 2, es parteix d'un esquema d'alt nivell de la solució que es vol aconseguir en aquest treball, i el seu desenvolupament (com poder capturar el tràfic, la gestió dels logs, l'enriquiment de la informació obtinguda i la seva visualització) fins arribar a una arquitectura final en baix nivell.

Al capítol 3, es preparar l'entorn de treball (laboratori) per realitzar la instal·lacions del productes necessaris per la realització del treball (com Zeek, Filebeat o ELK Stack) i la detecció d'activitats sospitoses a la xarxa mitjançant els propis frameworks de Zeek, la incorporació d'Intel Critical Stack com fonts d'intel·ligència de Zeek, i la integració amb VirusTotal per l'anàlisi dels arxius. Finalment, es mostra la configuració dels Dashboards i les proves de funcionament realitzades.

Al capítol 4, es troben les conclusions del treball segons els resultats obtinguts.

Al capítol 5, es troba el glossari.

Al capítol 6, hi ha el recull de la bibliografia utilitzada per la realització del treball.

Al capítol 7, es troben el annexes. En ells s'hi troben els detalls de les diferents instal·lacions i/o configuracions que s'han dut a terme durant el treball.

2. Investigació i disseny

2.1. Escenari del TFM

Existeix la necessitat de detectar possibles activitats sospitoses en una xarxa i poder veure-les en uns panells de controls (dashboards) de manera clara i concisa.

Les eines capaces per a detectar el tràfic d'una xarxa i inspeccionar-lo son els IDS i, per tal de poder analitzar aquestes activitats es desplegarà l'eina de detecció d'intrusos "Zeek IDS".

Per a poder tractar, enriquir i realitzar una visualització dels esdeveniments i les activitats potencialment malicioses en un o diversos dashboards, s'utilitzaran diverses eines que formen part del "ELK Stack":

- Beats (Filebeat)
- Logstash
- Elasticsearch
- Kibana

En la Figura 2-1, es mostra un esquema d'alt nivell la solució a assolirr en aquest treball.

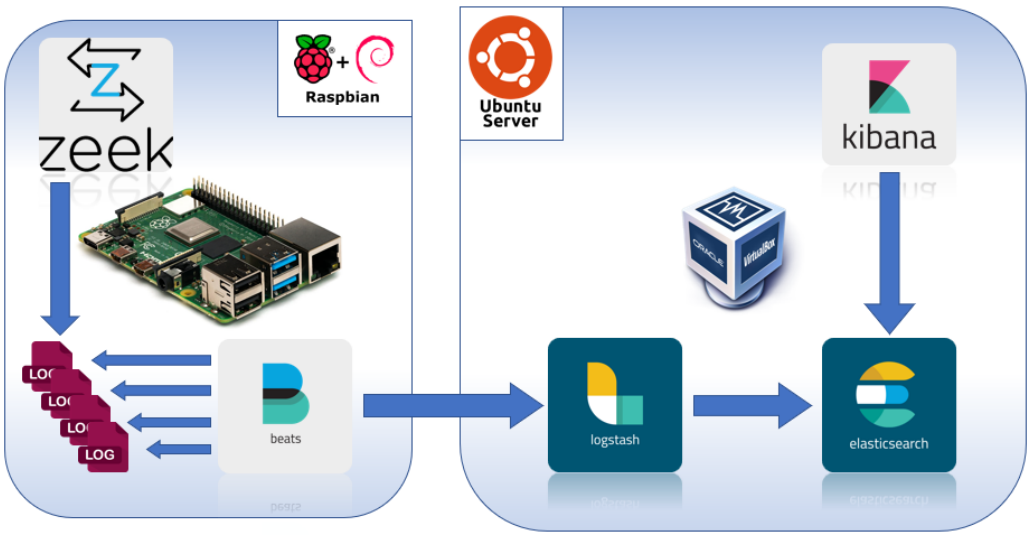


FIGURA 2-1: ESCENARI (DISSENY ALT NIVELL - HLD)

2.2. Components hardware i software

Es realitza una recerca per determinar els components formaran part del treball, tant hardware com software, això com la infraestructura que cal per dur-lo a terme.

El punt de partida és un Router, en aquest cas del model "Mitrastar HGU GPT-2541GNAC", amb capacitat de suportar 2 adreçaments IP, per poder fer de porta d'enllaç entre les dues xarxes que existiran:

- Xarxa de gestió (*management*):
- Xarxa d'equips del projecte: Principalment contindrà els elements de xarxa (switch), equips virtuals (clients i servidors per simular tràfic) i un equip físic per captar el tràfic de xarxa.

L'ordinador personal serà un equip amb capacitat de virtualització. El software de gestió de màquines virtuals serà "Oracle VirtualBox 6.1". El HOST, estarà connectat a les dues xarxes anteriors, una mitjançant l'enllaç Ethernet, i l'altre amb l'enllaç Wi-Fi. Algunes de les màquines virtuals que contindrà són:

- Servidor amb ELK Stack: Logstash, Elasticsearch i Kibana.
- Equips personal clients.
- Servidor web.

Per tal de poder captura el tràfic de xarxa, es connectarà al router, el switch "TP-Link TL-SG105E", amb capacitat de realitzar "port mirroring", per a enviar tot el tràfic de la xarxa a un port en concret. En el port destí de l'enviament del tràfic, hi haurà el hardware "Raspberry Pi 4" connectat en mode promiscu, per poder analitzar totes les dades que hi circulin.

El dispositiu "Raspberry Pi 4", és un hardware que equiparà el sistema operatiu Raspbian (basat en Debian), el software analitzador de trànsit de xarxa de codi obert "Zeek IDS" i un petit mòdul del ELK Stack anomenat Beats, per l'extracció dels logs de Zeek i el posterior enviament a Logstash. Aquest dispositiu tindrà dues interfícies de xarxa, la de monitorització i la de gestió.

El servidor que contindrà Logstash, Elasticsearch i Kibana, serà virtual i amb el sistema operatiu Ubuntu Server (basat en Debian) i estarà allotjat en la màquina HOST. També serà necessària la instal·lació d'un proxy invers, com Nginx, per a poder mostrar els Dashboards a altes màquines de la xarxa, com a la de l'analista de Ciberseguretat.

S'escull la instal·lació de Filebeat a la Raspberry Pi en comptes d'una instal·lació de Logstash, ja que els requeriments en quant a còmput i memòria són inferiors, ja que Logstash precisa de la instal·lació de Java perquè s'executa sobre una màquina virtual Java. I a més, la connexió Filebeat-Logstash és tolerable a errors en la xarxa, ja que en cas de no establir una connexió correcta, els diferents events es mantenen a la cua per a ser enviats i processats.

2.3. Capturar el tràfic de la xarxa: Zeek IDS

La captura de tràfic es realitzarà amb Zeek, fins fa poc conegut com a Bro. Zeek és un analitzador de trànsit de xarxa de codi obert. Principalment es tracta d'un monitor de seguretat que inspecciona tot el trànsit d'una interfície de xarxa, en profunditat, per detectar signes d'activitat sospitosa.

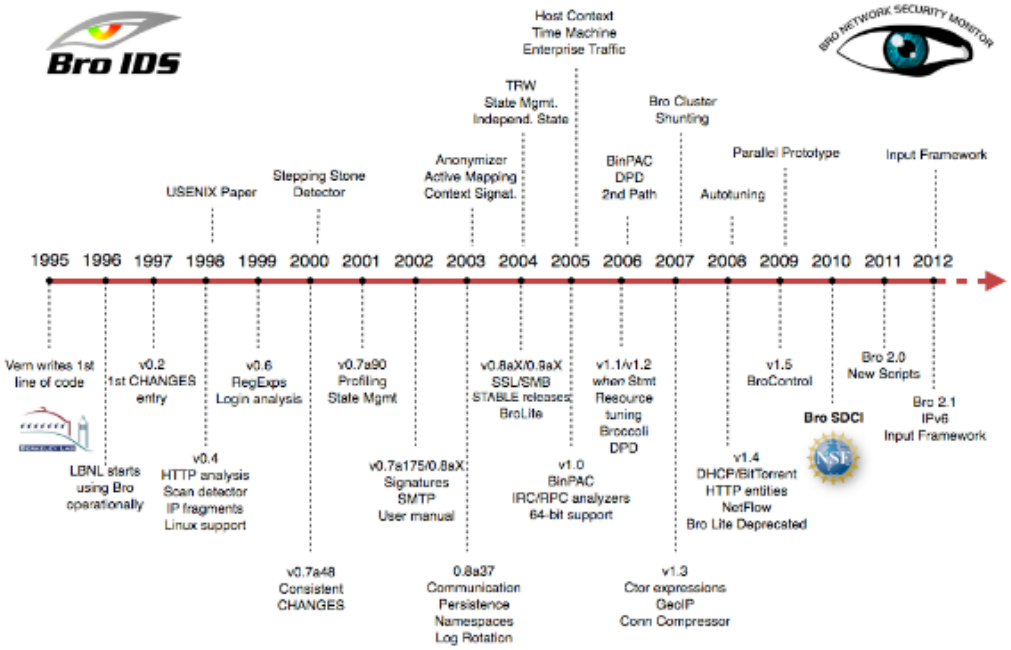


FIGURA 2-2 HISTÒRIA DE ZEEK (BRO)

D'aquesta manera, amb la captura del tràfic de la xarxa amb Zeek, s'obtenen un conjunt registres (logs) de totes les activitats que es realitzen a la xarxa. A més, també queden registrades les connexions a nivell d'aplicació, com per exemple totes les sessions HTTP (amb els URI, capçaleres, tipus MIME, respostes del servidor o, sol·licituds DNS i les seves respostes, certificats SSL, etc). Automàticament, Zeek escriu la informació obtinguda als fitxers de registre (logs) corresponents (estructurats i separats per tipus de connexions), el que facilita la gestió pel post-processament amb altres programes.

A més dels registres de connexions, que posteriorment en serviran per prendre altres decisions, es pretén analitzar o capturar amb Zeek activitats que es podrien considerar sospitoses, com la detecció de programari maliciós, la detecció d'atacs de força bruta, atacs d'injecció SQL, problemes amb les cadenes de certificats SSL, etc.

A continuació es mostren els fitxers de registre generats per Zeek que es configuraran per l'enregistrament en aquest treball:

- Protocols de xarxa:
  - **connection.log**: Connexions TCP/UDP/ICMP realitzades a la xarxa.
  - **dns.log**: Activitat DNS realitzada a la xarxa

    - **files.log**: Activitat relacionada amb els fitxers que circulen per la xarxa. Realitza un hash de cada un d'ells.
    - **ftp.log**: Activitat FTP realitzada a la xarxa.
    - **http.log**: Activitat HTTP realitzada a la xarxa
    - **ssh.log**: Activitat SSH realitzada a la xarxa.
- Detecció avançada:
    - **intel.log**: Match de dades d'intel·ligència.
        - S'afegirà la intel·ligència amb la integració de l'agregador gratuït de fonts d'intel·ligència d'amenaça "**Intel Critical Stack**". Es tracta d'una integració per obtenir informació, com ara adreces Tor, IPs malicioses conegudes o dominis de phishing.
    - **notice.log**: Events importants de Zeek (atacs de força bruta, problemes amb certificats SSL, SQL Injection, etc).

La llista completa de fitxers de registre que Zeek és capaç de detectar i generar es pot trobar en el seu lloc web[1].

Zeek s'instal·larà en el dispositiu hardware "Raspberry Pi 4".

2.4. Recol·lectar fitxers de registre: Beats (Filebeat)

Els arxius de registre (logs) que Zeek emmagatzema, seran capturats per Filebeat, un mòdul del software del ELK Stack, Beats.

Filebeat disposa de diferents mòduls per a diferents programes, entre ells el mòdul per a Zeek. Aquests mòduls, prèvia configuració, són capaços de recollir els logs generats i estructurar-los amb els camps corresponents de cada una de les aplicacions (realitza un *parse* de forma automàtica), per a que automàticament es puguin enviar a la base de dades Elasticsearch (sense utilitzar Logstash); encara que aquest no serà el cas, ja que Filebeat es configurarà per a enviar els logs al Logstash (s'instal·larà al servidor ELK), amb la finalitat d'enriquir-los, per després si ser enviats a Elasticsearch.

Filebeat (amb el mòdul de Zeek) s'instal·larà en el dispositiu hardware "Raspberry Pi 4". Oficialment la plataforma ARM64 no està suportada i s'ha d'instal·lar des de les fonts, compilar i segons el cas, fer una actuació manual, com la copia de certs fitxers.

La "Raspberry Pi 4" rebrà les dades de la xarxa ja que estarà en un port del switch "TP-Link TL-SG105E" configurat per reenviar-hi tot el tràfic (mitjançant port mirroring) que hi circuli. El sistema operatiu de la "Raspberry Pi 4" també s'haurà de configurar el mode promiscu.

[1] https://docs.zeek.org/en/current/script-reference/log-files.html

2.5. Processar i enriquir informació: Logstash

Els arxius de registre es processaran i enriquiran mitjançant Logstash, un component del ELK Stack. Els logs provindran de l'agent Filebeat, instal·lat en la "Raspberry Pi" i Logstash s'instal·larà en la màquina virtual "ELK Stack". Tenint Logstash en execució en un servidor, permet disposar d'un punt central de recepció de logs, en el cas que es disposes de més d'un dispositu d'adquisició de dades.

Logstash té un *pipeline* de processament d'esdeveniments distribuït en tres etapes:

FIGURA 2-3 PIPELINE DE PROCESSAMENT DE LOGSTASH

Les entrades generen esdeveniments, els filtres els modifiquen i les sortides els envien a qualsevol altre lloc. Les entrades i sortides admeten diferents tipus de còdecs per codificar o descodificar les dades de l'entrada cap a la sortida, sense haver de fer servir cap filtre en aquest apartat, per exemple, en l'entrada es podria tenir una entrada de Beats i la sortida deixar-la en un fitxer log del servidor segons certs paràmetres d'entrada.

En aquest cas, l'entrada o input, serà a través de Beats. Logstash crea un procés d'escolta en el port 5504 del servidor ELK. I la sortida o output, es configurarà una connexió amb la base de dades Elasticsearch (instal·lat al mateix servidor, i publicat en el port 9200). Totes les dades, segons el tipus d'esdeveniment es guardaran en índex diferents.

Pel tractament i enriquiment des les dades s'utilitzaran una sèrie de filtres disponibles a Logstash. Un filtre realitza un processament intermediari en un esdeveniment, abans de que aquest s'enviï a Elasticsearch. Els filtres es poden aplicar en funció de les característiques de l'esdeveniment.

El filtre que en un principi havia de tenir més rellevància havia de ser l'anomenat "translate" [2], que a partir d'un camp d'entrada com per exemple, el nom DNS d'una petició, podríem realitzar una cerca en un llista de dominis fraudulents (en un fitxer prèviament preparat) i si és el cas, afegir un camp al log, per tal de després poder filtrar aquests resultats. Però finalment aquest no s'utilitza, ja que per la classificació i alertes de dominis i URLs malicioses, s'utilitza el mòdul d'intel·ligència que es configura a Zeek (al fitxer *intel.log*). El funcionament que primerament s'havia pensat per crear el fitxer amb la llista de dominis i IPs

[2] https://www.elastic.co/guide/en/logstash/current/plugins-filters-translate.html

malicioses, estava compost per dues fonts com “MalwareDomainList” [3] i “URLhaus Database” [4].

Per la recerca de fitxers maliciosos, serà de molta utilitat el filtre “Http” [5], capaç de realitzar consultes REST API a serveis web, en aquest cas d’un proveïdor de deteccions d’antivirus, a través del hash MD5 dels fitxers vistos a la xarxa per Zeek (les dades dels fitxers es buscaran al fitxer *files.log*).

Un altre filtre utilitzat serà el filtre “geoip” [6], que afegeix informació sobre la ubicació geogràfica de les adreces IP a partir de les bases de dades de “Maxmind GeoLite2”, que serà útil per la creació de visualitzacions sobre un mapa a Kibana i enriquir els dashboards. La bases de dades GeoLite2 és una base de dades de GEO localització IP gratuïta.

Altres filtres que seran necessaris pel tractament de dades en l’àmbit de Logstash podran ser:

- Mutate[7]: Permet realitzar mutacions generals als camps (canvis de nom, suprimir-los, substituir-los, crear-ne de nous, etc).
- Split[8]: Permet tractar camps amb informació en format JSON.

2.6. Emmagatzemar la informació: Elasticsearch

La informació s’emmagatzema en la base de dades d’Elasticsearch. Elasticsearch s’instal·larà en la màquina virtual “ELK Stack”, el cual crea un servei al port 9200.

Totes les dades provinents de Logstash s’emmagatzemen en diferents índex, segons configuració d’enviament de Logstash (segons els tipus d’esdeveniment). Per exemple:

- filebeat-*entorn*-connection
- filebeat-*entorn*-dns
- filebeat-*entorn*-files
- filebeat-*entorn*-http
- filebeat-*entorn*-intel
- filebeat-*entorn*-notice
- filebeat-*entorn*-ssh
- filebeat-*entorn*-ssl

El paràmetre “*entorn*”, s’estableix en Logstash i pot ser “*test*” o “*real*” en funció de si les dades d’entrada són perquè s’estan provant certes funcions (*test*), o són dades d’un entorn productiu/real (*real*). Per exemple, uns índex de prova podria ser “filebeat-test-dns”, i un amb dades productictives, “filebeat-real-dns”.

---

[3] MalwareDomainList: http://www.malwaredomainlist.com/hostslist/ip.txt
[4] URLhaus Database: https://urlhaus.abuse.ch/downloads/text_online/
[5] Filtre Http: https://www.elastic.co/guide/en/logstash/current/plugins-filters-http.html
[6] Filtre GeoIP: https://www.elastic.co/guide/en/logstash/current/plugins-filters-geoip.html
[7] Filtre mutate: https://www.elastic.co/guide/en/logstash/current/plugins-filters-mutate.html
[8] Filtre split: https://www.elastic.co/guide/en/logstash/current/plugins-filters-split.html

2.7. Visualitzar les dades: Kibana

Kibana permet una exploració visual de les dades emmagatzemades a l'Elasticsearch, i poder-hi realitzar un anàlisi en temps real mitjançant la creació de dashboard o panells de control personalitzats.

Kibana s'instal·larà en la màquina virtual "ELK Stack", amb la resta de components.

Totes dades a visualitzar en Kibana estaran arxivades a Elasticsearch en diferents índex. Per a realitzar les consultes, es crearan dos patrons d'índex (*index patterns*). Aquest seran del tipus "filebeat-*entorn*-*", ja que així serà capaç d'agrupar totes les fonts de dades entrants que comencen amb el nom corresponent, de manera que es podran realitzar consultes de tots els registres del servidor de l'entorn corresponent de Filebeat (dades del connection, dns, files, etc).

- filebeat-test-*
- filebeat-real-*

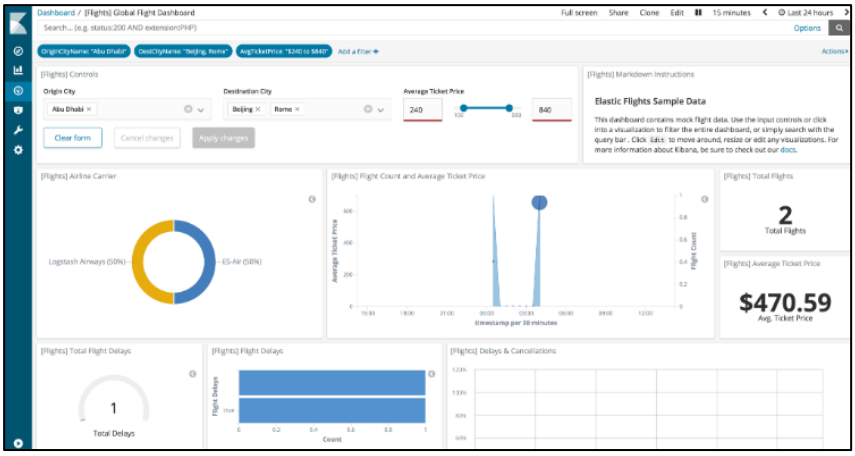


FIGURA 2-4 EXEMPLE D'UN DASHBOARD EN KIBANA

Per aquest projecte es crearan els dashboards necessaris per la monitorització de la xarxa i veure-hi els comportaments sospitosos, com:

- **Dashboard general de seguretat**
- **Dashboard de fitxers i deteccions de virus**
- **Dashboard de connexions malicioses**
- **Dashboard d'altres activitats sospitoses**

2.8. Arquitectura final

En la Figura 2-5 és mostra l'arquitectura final de la solució per a la detecció de comportaments sospitosos amb Zeek i, la representació en Dashboards mitjançant el conjunt d'eines Elasticsearch-Logstash-Kibana.

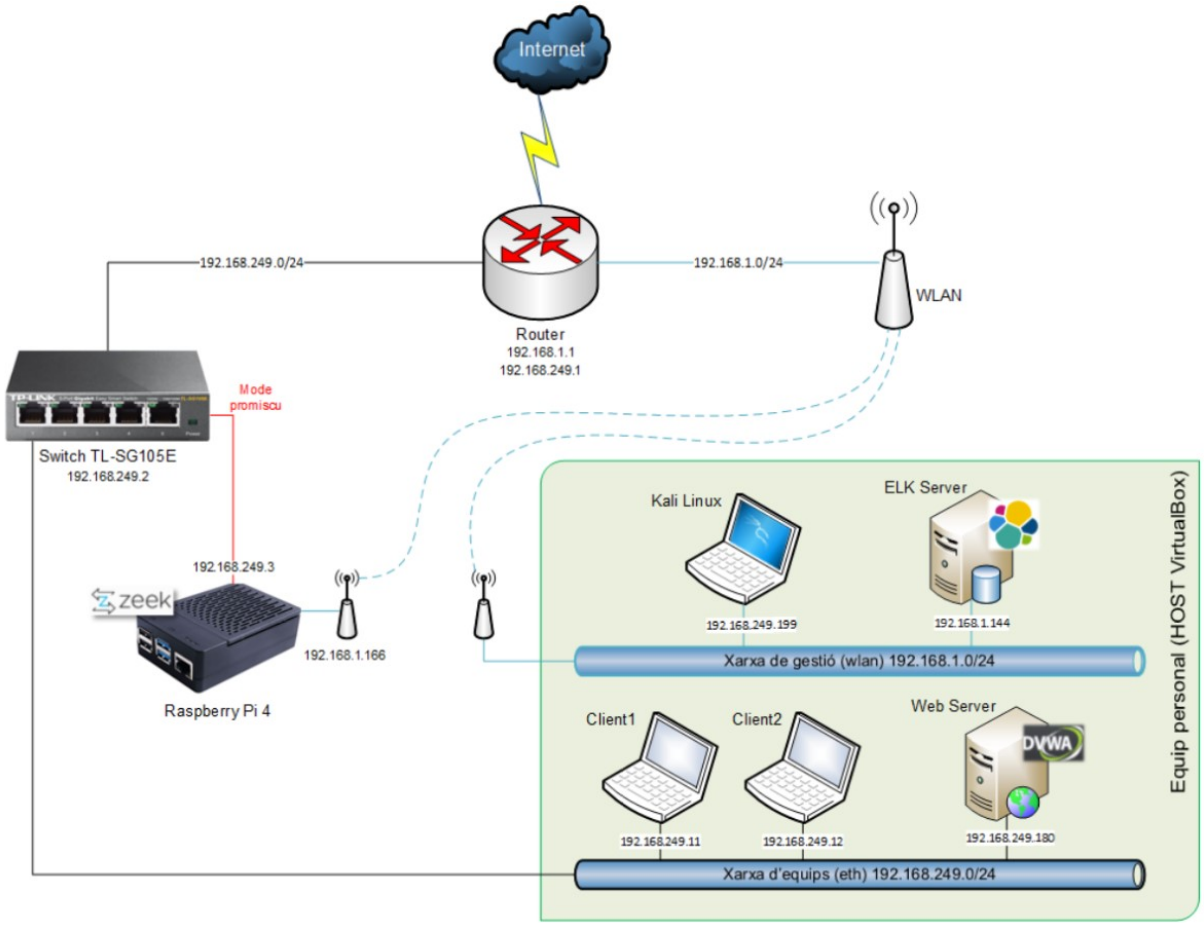


FIGURA 2-5: ARQUITECTURA FINAL (DISSENY BAIX NIVELL - LLD)

2.9. Valoració econòmica

En aquest apartat s'indiquen les despeses associades al desenvolupament i manteniment del treball:

Concepte	Preu
Raspberry Pi 4 Model B (4GB)	64,99 €
Kit Raspberry Pi 4: Caixa amb ventilació activa, font d'alimentació, cable HDMI. *Opcional*	18,99 €
Targeta de memòria microSDXC Classe 10 (64 GB)	11,69 €
Switch TP-Link TL-SG105E	20,99 €
Ordinador personal	0 €
Total:	**116,66 €**

FIGURA 2-6: VALORACIÓ ECONÒMICA DEL PROJECTE

Com es pot observar en la Figura 2-6, totes les despeses fan referència a equipament hardware. Per altra banda, el productes software que formen part de la sol·lució, com Zeek o ELK Stack, estan basats en codi obert, pel que no comporten despeses associades. De la mateixa manera, les fonts externes per l'enriquiment de dades per al Logstash com Intel Critical Stack i VirusTotal, tampoc. En un entorn empresarial el qual pot precisar de centenars de consultes a VirusTotal per comprovar els fitxers de la xarxa, caldria sumar-hi una subscripció Premium (al voltant d'uns 10.000€ anuals).

El producte que es vol obtenir és viable que per una mínima inversió, associada majoritàriament a equips de maquinari, es podran esbrinar comportaments en una xarxa que poden arribar a comportar grans pèrdues de informació, privacitat, segrests d'equips, pèrdua de diners, entre d'altres.

2.10. Joc de proves

Es defineixen un conjunt de proves per validar el funcionament de la solució. Les proves de funcionament consistiran en la generació de tràfic a la xarxa mitjançant dues màquines client i simulacions de tràfic, tant a llocs web maliciosos com no, o simulacions d'atac per part d'un equip maliciós de la xarxa.

Bàsicament, es tractarà de realitzar les següents proves:

- **Generar tràfic des dels clients**: Per poder comprovar:
  - Zeek és capaç de detectar tràfic i guardar els logs
  - Integració de Zeek amb ELK Stack (extracció de logs)
  - Detectar connexions a URL/IP malicioses
  - Detectar fitxers maliciosos
- **Realitzar algun atac amb Kali Linux**: Per poder comprovar altres comportaments sospitosos a la xarxa com podrien ser: SQL Injection, XSS o atacs de força bruta en protocol FTP.

3. Implantació i funcionament

3.1. Preparar entorn de treball

3.1.1. Preparar la xarxa

La funció de *port mirror* consisteix en monitoritzar i fer una copia (mirall) del trànsit de xarxa, reenviant còpies dels paquets entrants i sortints d'un o més ports (*mirrored ports*), a un port específic (*mirroring port*). Normalment, el port destí del mirall està connectat a un dispositiu de diagnòstic de dades, que s'utilitza per analitzar els paquets que circulen per la xarxa, per controlar i solucionar problemes de xarxa. En aquest cas, el port destí serà el dispositiu Raspberry Pi 4, en el qual hi haurà Zeek IDS instal·lat.

Un cop s'endolla el switch "TL-SG105E" al router, li assigna una adreça IP per DHCP. Ja es pot configurar a través de l'administració web. El switch té 5 ports i la configuració de ports es configura segons la Figura 3-1 (a l'Annex 1 es detalla la configuració al switch TL-SG-105E).

Port	Estat	Connectat a	Observacions
Port 1	Habilitat	Router	-
Port 2	-	-	-
Port 3	Habilitat	HOST VirtualBox (xarxa clients)	Mirrored port
Port 4	-	-	-
Port 5	Habilitat	Raspberry Pi 4 (Zeek IDS)	Mirroring port

FIGURA 3-1: CONFIGURACIÓ DEL SWITCH (PORT MIRROR)

3.1.2. Preparar dispositiu Raspberry Pi per a Zeek

A la Raspberry Pi, suport hardware on anirà el software Zeek IDS, s'instal·la el sistema operatiu oficial suportat en Raspberry Pi 4, el **Raspbian GNU/Linux 10 (buster)**.

3.1.2.1. Instal·lar i configurar Raspbian

En l'Annex 1 es detalla la instal·lació del sistema operatiu Raspbian i la configuració de la Raspberry Pi 4.

3.1.2.2. Deshabiltar Generic Receive Offload

Es deshabilita la característica “Generic Receive Offload” (GRO[9]) de la targeta de xarxa (eth0), ja que sinó Zeek no és capaç de veure tot el tràfic de xarxa. El motiu és perquè el GRO, és una tècnica molt utilitzada basada en software per reduir el cost general de processament per paquet (tornant a muntar els paquets petits en els més grans, GRO permet a les aplicacions processar directament menys paquets grans, reduint així el nombre de paquets a processar).

Per aplicar-ho sempre que es reinicia el dispositiu (es perdria al reiniciar-lo), cal posar-ho en el fitxer “rc.local”.

```
sudo nano /etc/rc.local
```

Afegir la següent línia al fitxer (abans del “exit 0”):

```
ethtool -K eth0 gro off
```

Es guarda la configuració i es reinicia el dispositiu per aplicar els canvis.

La Raspberry queda llesta amb el sistema operatiu “Raspbian GNU/Linux 10 (buster)”, per instal·lar el Zeek i Beats.

3.1.3. Preparar servidor ELK Stack

En l’Annex 1 es detalla la instal·lació i configuració del servidor pel ELK Stack.

3.2. Instal·lar i configurar Zeek

La versió de Zeek que s’instal·larà és la 3.1.1[10], alliberada el 10 de març de 2020, que és la última versió estable disponible actualment (Figura 3-2).

En l’Annex 2 es detalla la instal·lació i configuració de Zeek mitjançant les diferents captures de pantalla.

[9] https://doc.dpdk.org/guides/prog_guide/generic_receive_offload_lib.html
[10] Finalment quedarà instal·lada la versió 3.1.2, en l’apartat 3.2.7 es mostra el procés d’actualització.

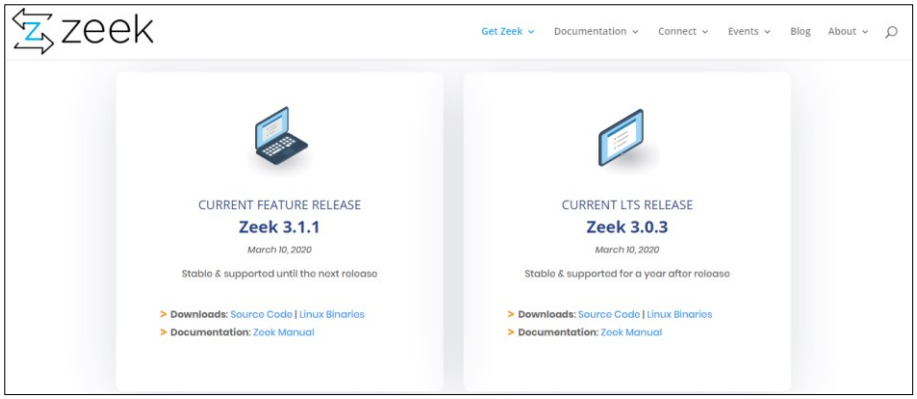


FIGURA 3-2: LLOC WEB DE ZEEK

3.2.1. Instal·lar prerequisits de Zeek

Abans d'instal·lar Zeek, es requereix que s'instal·lin els següents requisits:

- Libpcap (http://www.tcpdump.org)
- Lliberies OpenSSL (http://www.openssl.org)
- Lliberia BIND8
- Libz
- Bash (pel ZeekControl)
- Python 2.6 or superior (pel ZeekControl)

A més, si la instal·lació es fes des de les fonts, caldrien (no aplica en aquest cas):

- CMake 3.0 o superior (http://www.cmake.org)
- Make
- Compilador C/C++
- SWIG (http://www.swig.org)
- Bison 2.5 o superior (https://www.gnu.org/software/bison/)
- Flex (lexical analyzer generator) (https://github.com/westes/flex)
- Capçaleres Libpcap (http://www.tcpdump.org)
- Capçaleres OpenSSL (http://www.openssl.org)
- Capçaleres zlib (https://zlib.net/)
- Python (https://www.python.org/)

Per instal·lar els requisits necessaris, executar en un terminal amb permís d'administrador:

```
sudo apt-get install cmake make gcc g++ flex bison libpcap-dev
libssl-dev python-dev swig zlib1g-dev
```

3.2.2. Instal·lar Zeek

Per la instal·lació de Zeek per a Raspbian 10 (arquitectura ARM), cal afegir el repositori corresponent, i la clau del repositori per a poder confiar-hi. La finalitat és poder distribuir paquets i actualitzacions.

Cal executar la primera comanda com a root, per a poder afegir el repositori:

```
sudo -s

echo 'deb
http://download.opensuse.org/repositories/security:/zeek/Raspbi
an_10/ /' > /etc/apt/sources.list.d/security:zeek.list
```

Es descarrega la clau i s'afegeix al repositori de claus:

```
wget -nv
https://download.opensuse.org/repositories/security:zeek/Raspbi
an_10/Release.key -O Release.key

sudo apt-key add - < Release.key
```

S'actualitzen els repositoris i s'instal·la Zeek:

```
sudo apt-get Update

sudo apt-get install zeek
```

Zeek queda instal·lat, la relació de carpetes que crea al sistema són:

El paquet "deb" de la versió de Zeek instal·lada, també està disponible per a descarregar[11].

3.2.3. Crear un usuari sense privilegis per a Zeek

Es crea un usuari i un grup anomenat "zeek", amb la finalitat de no executar el programa com a root. Aquest, en serà el propietari i l'usuari que executi el programa Zeek/Zeekctl. D'aquesta manera, només se li assignaran els permisos necessaris per al seu funcionament.

```
sudo groupadd zeek

sudo useradd zeek -g zeek  -m -d /home/zeek
```

Es crea una contrasenya per a l'usuari "zeek" (*z33k*):

```
sudo passwd zeek
```

Es canvia el propietari i els permisos de la ubicació "*/opt/zeek*" (instal·lació del programa) a l'usuari/grup zeek:

```
sudo chown -R zeek:zeek /opt/zeek

sudo chmod 740 /opt/zeek
```

Es dona als binaris de Zeek necessaris (zeek i capstats), permís per a poder capturar els paquets de xarxa:

---

[11] Instal·lador ·.deb" de Zeek: https://download.opensuse.org/repositories/security:/zeek/Raspbian_10/armhf/zeek_3.1.1-0_armhf.deb

```
sudo setcap cap_net_raw=eip /opt/zeek/bin/zeek

sudo setcap cap_net_raw=eip /opt/zeek/bin/capstats
```

3.2.4. Configurar variables d'entorn

Es configura al PATH de l'usuari zeek, la ruta dels binaris de Zeek, al fitxer ./bashrc. Per editar el fitxer .bashrc de l'usuari zeek, que està al seu *home directory*:

```
su zeek

nano $HOME/.bashrc
```

Afegir la següent línia, per posar la ruta dels binaris de Zeek al PATH:

```
export PATH=/opt/zeek/bin:$PATH
```

3.2.5. Configuració de Zeek

Abans d'executar Zeek, cal configurar-lo, i es fa mitjançant la configuració de Zeek Control (també conegut com Zeekctl).

ZeekControl és el programa de control per a Zeek. Es tracta d'una shell interactiva per operar la instal·lació de Zeek tant en el mode *standalone*, com serà en aquest cas, o també en el cas que disposi de múltiples nodes.

3.2.5.1. Establir interfície de xarxa a monitoritzar

En l'arxiu "*/opt/zeek/etc/node.cfg*" es configuren el nombre de nodes de l'arquitectura de Zeek. En aquest cas es tracta de configurar un sol node en mode "standalone", i establir la interfície de xarxa que es vol monitoritzar. En el dispositiu Raspberry Pi, la interfície és la connexió per cable Ethernet, per tant, la "eth0".

Editar l'arxiu de configuració:

```
nano /opt/zeek/etc/node.cfg
```

Establir el valor **eth0** en l'apartat **interface** de l'apartat [zeek], corresponent a l'únic node existent en mode standalone.

```
[zeek]
type=standalone
host=localhost
interface=eth0
```

3.2.5.2. Establir el direccionament de xarxa a monitoritzar

En l'arxiu de configuració "*/opt/zeek/etc/networks.cfg*", és on s'especifica a Zeek les xarxes que s'han de monitoritzar.

Es comenten les tres predeterminades i s'afegeix la xarxa 192.168.249.0/24. Per a fer-ho, editar l'arxiu:

```
sudo nano /opt/zeek/etc/networks.cfg
```

Eliminar/comentar línies predeterminades i afegir la xarxa corresponent (i un comentari):

```
192.168.249.0/24        Ethernet Raspberry network
```

3.2.5.3. Configuració de Zeekctl

A l'arxiu de configuració "*/opt/zeek/etc/zeekctl.cfg*", es manté el valor per defecte de LogRotationInterval a 3600 segons, corresponent a la freqüència desitjada en la rotació de logs. La opció de correus electrònics no es configura, però el programa seria capaç d'enviar e-mails per diferents tipus d'events.

```
sudo nano /opt/zeek/etc/zeekctl.cfg
```

Des d'aquest fitxer també es poden establir les diferents rutes on guardar els logs generats, la ruta de la configuració o del spool. La configuració establerta és:

```
MailTo = root@localhost
MailConnectionSummary = 0
MinDiskSpace = 5
MailHostUpDown = 0
LogRotationInterval = 3600
LogExpireInterval = 0
StatsLogEnable = 1
StatsLogExpireInterval = 0
StatusCmdShowAll = 0
CrashExpireInterval = 0
SitePolicyScripts = local.zeek
LogDir = /opt/zeek/logs
SpoolDir = /opt/zeek/spool
CfgDir = /opt/zeek/etc
```

3.2.5.4. Scritps de detecció

A l'arxiu de configuració local.zeek es poden habilitar o deshabilitar els scripts que carrèga Zeek a l'inici. Per editar el fitxer:

```
sudo nano /opt/zeek/share/zeek/site/local.zeek
```

La configuració establerta és per defecte, tret de la configuración afegida per configurar el framework d'intel·ligència[12], una fragment del fitxer és:

```
@load misc/loaded-scripts
@load tuning/defaults
@load misc/capture-loss
@load misc/stats
@load frameworks/software/vulnerable
@load frameworks/software/version-changes
@load-sigs frameworks/signatures/detect-windows-shells
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software
@load protocols/dns/detect-external-names
@load protocols/ftp/detect
@load protocols/conn/known-hosts
@load protocols/conn/known-services
@load protocols/ssl/known-certs
@load protocols/ssl/validate-certs
@load protocols/ssh/geo-data
@load protocols/ssh/detect-bruteforcing
...
```

3.2.5.5. Canviar format dels logs

Per defecte, Zeek emmagatzema els logs en format ASCII. Per l'extracció dels logs de Zeek mitjançant Beats/Filebeat, cal configurar el format de sortida com a JSON (fitxer lleuger d'intecanvi de dades). Per a realitzar aquesta configuració, editar el següent fitxer de configuració:

```
sudo nano
/opt/zeek/share/zeek/base/frameworks/logging/writers/ascii.zeek
```

Canviar el valor de **LogAscii::use_json** de F (false) a T (true).

```
redef LogAscii::use_json = T;
```

3.2.5.6. Tasca de manteniment Zeek

Es configura una tasca programa en el *CRON* per a que executi les tasques de manteniment necessàries del programa ZeekControl.

Editar el crontab de l'usuari zeek:

```
sudo crontab -u zeek -e
```

Afegir la següent tasca cron (s'executa cada 5 minuts la tasca corresponent):

---

[12] Configurat en l'apartat Instalar client Intel Stack

```
*/5 * * * * /opt/zeek/bin/zeekctl cron
```

3.2.6. Executar Zeek

ZeekControl és el programa de control per a Zeek. Es tracta d'una shell interactiva per operar l'instal·lació de Zeek. El primer cop, i cada vegada que hi existeixi un canvi en els arxius de configuració, cal executar:

```
Zeekctl
```

```
Check
Deploy
Status
Start
```

En la columna *Status* s'observa que el programa ja està en execució si apareix l'estat "running".

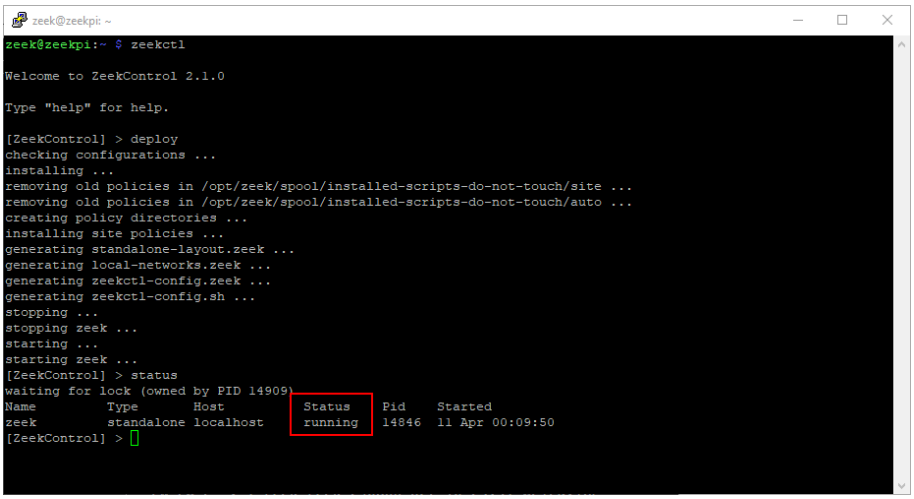


FIGURA 3-3: PRIMERA EXECUCIÓ DE ZEEK

3.2.7. Actualitzacions de Zeek

Durant la redacció de la memòria ha sortit la versió 3.1.2 (el 14 d'abril de 2020). Per tal d'anar actualitzant a les noves versions de Zeek, cal executar:

```
sudo apt-get update
```

Posteriorment, és necessari tornar a realitzar les següents tasques de configuració, documentades en aquesta memòria en capítols anteriors:

- Aplicar permisos a les carpetes per l'usuari i grup "zeek", ja que torna a assignar com a propietari l'usuari root.
- Permisos per llegir tràfic de xarxa als executables corresponents.
- Canviar el format de sortida dels logs a JSON.

3.3. Instal·lar i configurar Filebeat

S'instal·la Filebeat corresponent a Elastic Beats v7.6.2[13] (alliberada el 31 de març de 2020). Per la instal·lació de Filebeat, s'utilitza un script anomenat easyBEATS[14], de Josh Thurston, pensat per la instal·lació de qualsevol dels diferents mòduls de Beats (filebeat, metricbeat…) per a dispositius ARM, com la Raspberry Pi (ja que Elastic no allibera ni manté Beats per a l'arquitectura ARM).

L'script d'instal·lació easyBEATS, el que realitza és:

- Instal·lar els requisits necessaris
- Descarregar-se els fonts de Beats necessàries
- Fer la compilació per a ARM
- Crear estructura de directoris

Per a fer la instal·lació mitjançant easyBEATS, cal:

- Clonar el repositori
- Configurar els paràmetres de l'script
- Executar l'script

3.3.1. Clonar el repositori

Per clonar el repositori de easyBEATS, executar:

```
sudo apt-get install git -y

cd ~

git clone https://github.com/RaoulDuke-Esq/easyBEATS.git

cd easyBEATS

sudo chmod 755 easyBEATS
```

3.3.2. Configurar script d'instal·lació

Editar l'arxiu de configuració per establir els paràmetres d'instal·lació de Beats:

```
nano easyBEATS
```

Aqui, s'estableixen els paràmetres de versió a instal·lar, si es desitja actualitzar el sistema o instal·lar les dependències necessàries, si cal usar el SWAP (es desactiva perquè la Raspberry Pi disposa de més de 2GB de RAM) o els mòduls Beats que es volen instal·lar (només Filebeat, no se'n instal·len altres com Metricbeat, Packetbeat o Auditbeat).

[13] https://github.com/elastic/beats/releases/tag/v7.6.2
[14] https://github.com/josh-thurston/easyBEATS

L'arxiu de configuració, en l'apartat de les variables queda de la següent forma:

```
#!/bin/bash

# Script variables
UPDATE_SYSTEM=true #change to false if you don't want to
upgrade your whole system
INSTALL_DEPS=true #change to false if you have already run this
script successfully before
USE_SWAP=false #change to fales if you're using a Pi4 with 2GB
of RAM or more
WORKING_DIR="beat-factory" #this directory will be created in
/home/pi
#visit https://github.com/elastic/beats/releases to find other
version numbers and commit numbers
BEAT_VERSION_NUM="7.6.2" #the version number of the Beats
release you want to use
BEAT_VERSION="d57bcf8" #the commit number of the Beats release
you want to use
#add as many beats as you want to BEAT_NAME separated by a
space
BEAT_NAME=( filebeat ) #metricbeat filebeat packetbeat
auditbeat heartbeat
INSTALL_LOCAL=true #set to false if you only want to compile
without installing
CLEAN_UP=true #set to false if you want to keep the source
files on your Pi
```

3.3.3. Executar script d'instal·lació

Per instal·lar Filebeat 7.6.2 s'executa l'script d'instal·lació :

```
./easyBEATS
```

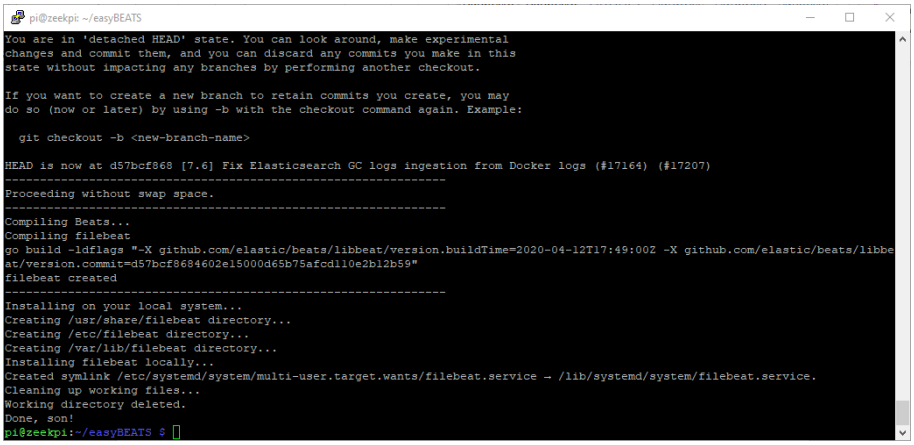


FIGURA 3-4: SCRIPT D'INSTAL·LACIÓ DE FILEBEAT

Filebeat s'ha instal·lat però no hi ha el mòdul per al Zeek. Cal descarregar-se les fonts de la versió Filebeat 7.6.2 per a Linux x86, i copiar a la ubicació del programa de forma manual els arxius necessaris.

Es descarrega la versió de Filebeat corresponent i s'extreu el contingut:

```
wget
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-
7.6.2-linux-x86.tar.gz

tar xzvf filebeat-7.6.2-linux-x86.tar.gz
```

Es canvien els permisos dels arxius a root, perquè són els que han de tenir a la destinació, un cop copiats:

```
cd filebeat-7.6.2-linux-x86.tar.gz

sudo chown -R root:root *

sudo chmod -R 755 *
```

Es copien els arxius necessaris per a disposar del mòdul Zeek de Filebeat en la instal·lació local a la Raspberry Pi:

```
sudo cp -R module/zeek/ /usr/share/filebeat/module/

sudo cp modules.d/zeek.yml.disabled /etc/filebeat/modules.d/

sudo cp fields.yml /etc/filebeat
```

En l'arxiu de configuració "filebeat.yml", especificar la ruta d'on estan els mòduls:

```
sudo nano /etc/filebeat/filebeat.yml
```

Configurar el **path** a l'apartat "filebeat.config.modules" i habilitar l'actualització periòdica (**reload.enabled**):

```
filebeat.config.modules:
        path: /etc/filebeat/modules.d/*.yml
        reload.enabled: true
```

Filebeat queda instal·lat, però aquest no s'inicia automàticament. L'arrencada o parada de Filebeat és diferent segons el sistema operatiu, ja que es pot realitzar de dues maneres, depenent si aquest utilitza SysV init o systemd. En el cas de Raspbian és systemd (es pot saber executant la comanda "*ps -p 1*").

Per tant, es configura Filebeat perquè s'inicií automàticament quan s'inicia el sistema, executant:

```
sudo /bin/systemctl daemon-reload

sudo systemctl enable filebeat.service
```

Per a iniciar manualment el servei Filebeat en aquest moment, executar la següent comanda:

```
sudo systemctl start filebeat.service
```

I en cas que sigui necessari, es pot aturar manualment mitjançant la següent comanda:

```
sudo systemctl stop filebeat.service
```

El fitxer de la definició del servei de Filebeat és */lib/systemd/system/filebeat.service* .

Per comprovar que la instal·lació de Filebeat és correcte, executar:

```
sudo /usr/share/filebeat/bin/filebeat test config -c
/etc/filebeat/filebeat.yml
```

El resultat de la comanda és "**Config OK**".

3.3.4. Habilitar el mòdul Zeek de Filebeat

Executar la següent comanda per habilitar el mòdul de Zeek del Filebeat:

```
sudo /usr/share/filebeat/bin/filebeat -c
/etc/filebeat/filebeat.yml modules enable zeek
```

El resultat que ens retorna és **Enabled zeek**.

Per comprovar els mòduls habilitats i deshabilitats de Filebeat, es pot fer utilitzant la següent comanda:

```
sudo /usr/share/filebeat/bin/filebeat -c
/etc/filebeat/filebeat.yml modules list
```

3.3.5. Configurar el mòdul Zeek de Filebeat

Es modifiquen les opcions del mòdul de Zeek de Filebeat, per a que només es tinguin en compte, i s'enviïn a Logstash, alguns dels logs generats per Zeek.

Editar l'arxiu de configuració "zeek.yml", ja que per defecte està tot en true. També caldrà especificar la ubicació dels logs, ja que per defecte, els busca en la carpeta equivocada (a */var/log/bro/current/* en comptes de */opt/zeek/logs/current/).*

Per editar l'arxiu de configuració:

```
sudo nano /etc/filebeat/modules.d/zeek.yml
```

Queden habilitats els següents logs per ser extrets de Zeek mitjançant Filebeat:

- Connection
- DNS
- Files
- FTP
- HTTP

- Intel
- Notice
- SSH
- SSL

De tots els logs configurats (en **true**), cal especificar el fitxer concret de cada un dels logs. La ruta correcte de cada un d'ells es troba a:

```
/opt/zeek/logs/corrent/fitxer_de_log.log
```

Un exemple de configuració, pel tipus "files", és:

```
files:
        enabled: true
        var.paths: ["/opt/zeek/logs/current/files.log"]
```

3.3.6. Configurar Filebeat

Cal modificar el fitxer "filebeat.yml" per realitzar la configuració de Filebeat. La opció més important és afegir l'apartat de sortides/outputs. Cal deshabilitar la sortida a Elasticsearch i Kibana, i habilitar la sortida a Logstash.

Editar el fitxer de configuració:

```
sudo nano /etc/filebeat/filebeat.yml
```

Afegir els següents paràmetres[15]:

```
output.logstash:
hosts: ["192.168.1.144:5044"]
```

L'script també ha preparat el sistema per arrencar el servei automàticament. Només ens cal iniciar-lo en aquest moment:

```
sudo systemctl restart filebeat.service
```

3.4. Instal·lar i configurar ELK Stack

Els productes que composen l'Elastic Stack, han de ser instal·lats en el següent ordre:

1. Elasticsearch
2. Kibana
3. Logstash

[15] En l'apartat Securitzar comunicació Filebeat-Logstash, es realitza la configuració SSL en la connexió amb Logstash.

3.4.1. Elasticsearch

El paquet Debian per Elasticsearch es pot descarregar des del lloc web de Elastic[16] o a través dels repositoris APT. Com s'instal·la en el servidor Ubuntu, es pot utilitzar el repositori APT per a instal·lar Elasticsearch.

S'instal·la la versió més recent estable d'Elasticsearch, corresponent a la versió 7.6.2.

Per la instal·lació d'Elasticsearch cal afegir el repositori corresponent, i la clau del repositori per a poder confiar-hi. La finalitat és poder distribuir paquets i actualitzacions.

Es descarrega la clau i s'afegeix al repositori de claus (en la pròxima instal·lació de Kibana i Logstash ja no caldrà fer-ho):

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo apt-key add -
```

Instal·lar el requisit "apt-transport-https" i afegir la definició del respositori d'Elastic 7.6 (en la pròxima instal·lació de Kibana i Logstash ja no caldrà fer-ho):

```
sudo apt-get install apt-transport-https

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable
main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
```

S'actualitzen els repositoris i s'instal·la Elasticsearch (la descàrrega són aproximadament uns 300MB):

```
sudo apt-get update && sudo apt-get install elasticsearch
```

Elasticsearch queda instal·lat, però aquest no s'inicia automàticament. L'arrencada o parada d'Elasticsearch és diferent segons el sistema operatiu, ja que es pot realitzar de dues maneres, depenent si aquest utilitza *SysV init* o *systemd*. En el cas de Ubuntu Server és **systemd** (es pot saber executant la comanda "*ps -p 1*").

Per tant, es configura Elasticsearch perquè s'iniciï automàticament quan s'inicia el sistema, executant:

```
sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable elasticsearch.service
```

Per a iniciar manualment el servei d'Elasticsearch en aquest moment, executar la següent comanda:

```
sudo systemctl start elasticsearch.service
```

[16] Instal·lador Elasticsearch per a Debian:
https://www.elastic.co/guide/en/elasticsearch/reference/7.6/deb.html#install-deb

I en cas que sigui necessari, es pot aturar manualment mitjançant la següent comanda:

```
sudo systemctl stop elasticsearch.service
```

El fitxer de la definició del servei de Kibana és */usr/lib/systemd/system/elasticsearch.service* .

El logs d'Elasticsearch es troben a /var/log/Elasticsearch.

Per a comprovar la correcte instal·lació del node Elasticsearch, es pot fer enviant una petició HTTP al port 9200 local:

```
curl -X GET "localhost:9200"
```

La resposta d'Elasticsearch ha de ser en aquest format JSON, on mosta la informació bàsica de la instal·lació:

```
{
  "name" : "elkserver",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "FCOreGqgR2aom3voUHC77A",
  "version" : {
    "number" : "7.6.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef48eb35cf30adf4db14086e8aabd07ef6fb113f",
    "build_date" : "2020-03-26T06:34:37.794943Z",
    "build_snapshot" : false,
    "lucene_version" : "8.4.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

3.4.2. Kibana

El paquet Debian per Kibana es pot descarregar des del lloc web de Elastic[17] o a través dels repositoris APT. Com s'instal·la en el servidor Ubuntu, es pot utilitzar el repositori APT per a instal·lar Kibana.

S'instal·la la versió més recent estable de Kibana, corresponent a la versió 7.6.2.

Cal descarregar la clau del repositori i afegir-la el repositori de claus, així com afegir la definició d'Elastic 7.6 dels repositoris; però com prèviament ja s'ha instal·lat Elasticsearch en aquest servidor, no cal tornar a fer-ho.

[17] Instal·lador Kibana per a Debian:
https://www.elastic.co/guide/en/kibana/current/deb.html

Així que s'actualitzen els repositoris i s'instal·la Kibana (la descàrrega són aproximadament uns 250MB):

```
sudo apt-get update && sudo apt-get install kibana
```

Kibana queda instal·lat, però aquest no s'inicia automàticament. L'arrencada o parada de Kibana és diferent segons el sistema operatiu, ja que es pot realitzar de dues maneres, depenent si aquest utilitza SysV init o systemd. En el cas de Ubuntu Server és systemd (es pot saber executant la comanda "*ps -p 1*").

Per tant, es configura Kibana perquè s'iniciï automàticament quan s'inicia el sistema, executant:

```
sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable kibana.service
```

Per a iniciar manualment el servei de Kibana en aquest moment, executar la següent comanda:

```
sudo systemctl start kibana.service
```

I en cas que sigui necessari, es pot aturar manualment mitjançant la següent comanda:

```
sudo systemctl stop kibana.service
```

El fitxer de la definició del servei de Kibana és */etc/systemd/system/kibana.service* .

Editar l'arxiu de configuració "kibana.yml", per configurar els diferents paràmetres sobre Kibana, com el host, el nom, ubicació d'Elasticsearch, etc.

Per editar l'arxiu de configuració:

```
sudo nano /etc/kibana/kibana.yml
```

Establir els següents paràmetres, referents al nom del servidor i el host d'Elastisearch on realitzarà les consultes (és al propi servidor del ELK Stack, i el port 9200):

```
server.host: "localhost"
server.name: "ELKserver"
elasticsearch.hosts: ["http://localhost:9200"]
```

Reiniciar el servei de Kibana per aplicar els canvis:

```
sudo systemctl restart kibana.service
```

3.4.2.1. Visualitzar Kibana des d'un host extern

Per permetre visualitzar els Dashboards de Kibana des d'equips externs al servidor cal la instal·lació d'un proxy invers, que mitjançant la connexió HTTP redirigeixi la consulta al port de Kibana intern, el 5601.

Es realitzarà la instal·lació i configuració de Nginx[18], un servidor web i proxy invers lleuger, i d'alt rendiment. És programari lliure i de codi obert, llicenciat sota la Llicència BSD simplificada.

Per instal·lar Nginx i les utilitats necessàries (utilitats del servidor web Apache) mitjançant els repositoris APT, executar la comanda:

```
sudo apt-get install nginx apache2-utils
```

Per l'accés extern es configurarà un usuari/contrasenya. Per a crear aquestes credencials de Nginx, s'executa la següent la comanda per crear l'usuari "kibanaadmin" i on posteriorment s'escriu la contrasenya (*kibanaadmin*):

```
sudo htpasswd -c /etc/nginx/htpasswd.users kibanaadmin
```

Un cop Nginx instal·lat, es modifica l'arxiu del lloc web per defecte:

```
sudo nano /etc/nginx/sites-available/default
```

Per a habilitar connexions al servidor HTTP al port 80 i fer de pasarel·la cap a la instància web de Kibana al port 5601, configurar el fitxer segons:

```
server {
    listen 80;
    server_name ELKserver;
    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;
    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Per a comprovar que la sintaxi és correcte, executar:

```
sudo nginx -t
```

[18] Lloc web de Nginx: https://www.nginx.com/

La sortida ha de ser:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Nginx queda instal·lat i configurat, i per defecte ja es configura per arrencar a l'inici del sistema. El fitxer de la definició del servei de Nginx és */lib/systemd/system/nginx.service* .

```
sudo systemctl start nginx.service

sudo systemctl restart nginx
```

Des d'un equip que tingui visibilitat al servidor, es realitza la connexió al servei web publicat per HTTP al port 80, per comprovar que Nginx està ben configurat i s'accedeix correctament a Kibana (l'accés està restringit a les credencials de l'usuari "kibanaadmin"):

- http://192.168.1.144/app/kibana#/home

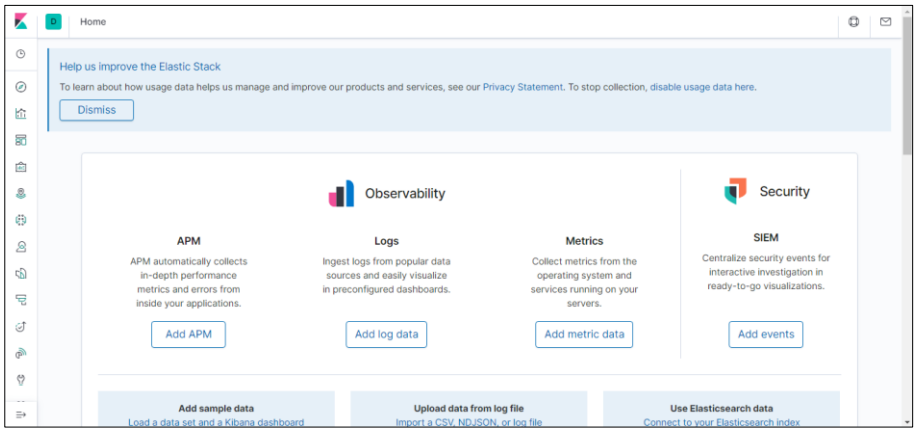


FIGURA 3-5: PANTALLA INICIAL DE KIBANA

3.4.2.2. Inicialització d'índex

Un cop Kibana està en marxa, cal configurar-hi a quins índex d'Elasticsearch buscar la informació per a poder mostrar las dades i construir les diferents visualitzacions i dashboards. Per fer-ho, cal crear almenys un patró d'índex (*index pattern*). Els patrons d'índex diuen poden coincidir amb el nom d'un índex específic i únic d'Elasticsearch, o incloure un asterisc (*) i combinar així diversos índexs definits.

Les dades provinents de Logstash s'emmagatzemen en els següents índex d'Elasticsearch:

- filebeat-*entorn*-connection
- filebeat-*entorn*-dns
- filebeat-*entorn*-files
- filebeat-*entorn*-http

- filebeat-*entorn*-intel
- filebeat-*entorn*-notice
- filebeat-*entorn*-ssh
- filebeat-*entorn*-ssl

Per tant, es crea un patró d'índex per cada un dels dos entorns que hi tenim dades, per fer consultes en tot el conjunt dels logs:

- **filebeat-test-*** → Engloba totes les dades de cada un dels diferents tipus de logs de l'entorn de test.
- **filebeat-real-*** → Engloba totes les dades de cada un dels diferents tipus de logs de l'entorn de real.

També es creen els següents patrons d'índex

- **filebeat-*-files** → Engloba dades corresponents a fitxers.
- **filebeat-*-intel** → Engloba dades corresponents a intel·ligència.

A Kibana, per crear un patró d'índex, cal fer-ho des del menú Management → Kibana → Index Patterns, i seleccionar **Create índex pattern**. A continuació (Figura 3-6), escriure el nom del patró corresponent (veure Figura 3-6) i en el pas següent, seleccionar com a camp que defineix el temps, l'anomenat *@timestamp* (Figura 3-7).

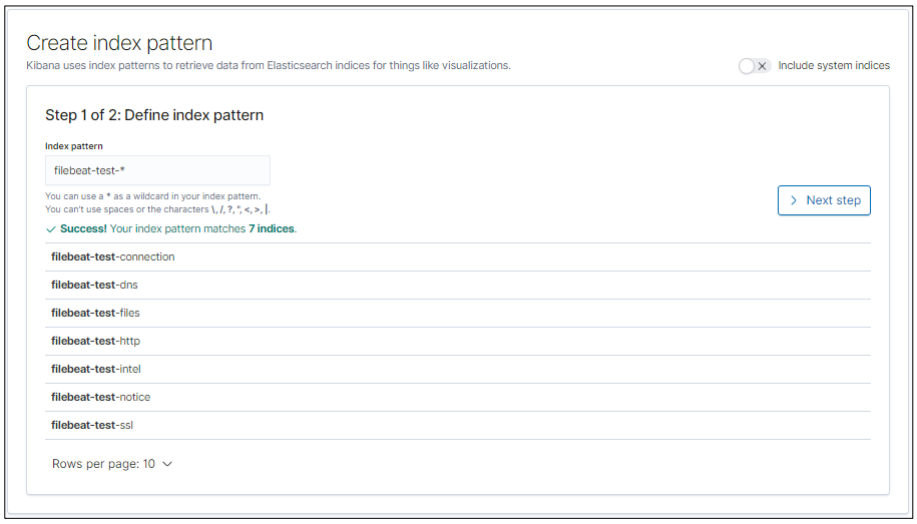


FIGURA 3-6: KIBANA, CREAR INDEX PATTERN (1)

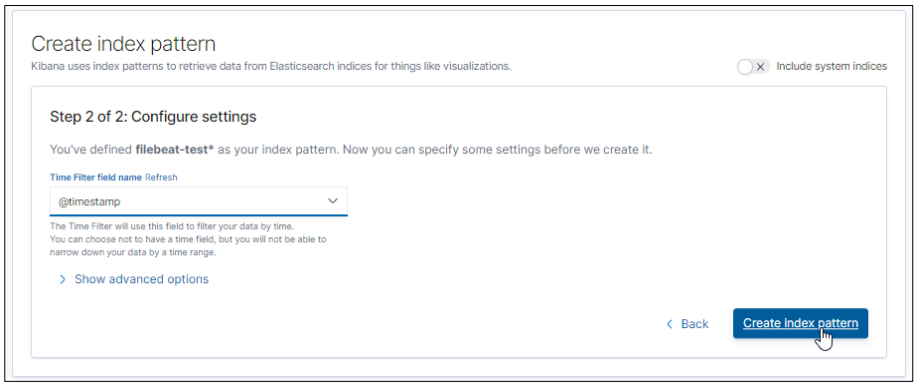


FIGURA 3-7: KIBANA, CREAR INDEX PATTERN (2)

Un cop es disposa dels patrons d'índex, també es defineix el que s'anomenen plantilles d'índex (*index templates*) en la configuració d'Elasticsearch (Figura 3-8). Aquestes, permeten definir unes plantilles que s'aplicaran automàticament quan es creïn nous índexs, les quals inclouen configuració i una definició del mapeig de camps (Figura 3-9).

Per crear una plantilla d'índex d'Elasticsearch, cal fer-ho des del menú de Kibana: Management → Elasticsearch → Index Management → Index Templates, i seleccionar **Create a template**. A continuació, escriure el nom de la plantilla i l'índex a aplicar, que serà *filebeat-** (veure Figura 3-8Figura 3-6). Això significa que tots els índexs que comencin per "filebeat-", s'aplicarà la configuració definida. En el pas següent (veure Figura 3-9), es defineix el mapeig dels diferents camps de la plantilla "*filebeat-**". Els més rellevants són:

- **av_score**: Camp del tipus número enter (*integer*) i que tindrà un valor enter referent al nombre de deteccions d'antivirus d'un fitxer.
- **geoip**: Camp del tipus *geoip*, necessari per a poder mostrar dades mapes a Kibana. Aquestes dades proven del plugin geoip de Logstash.

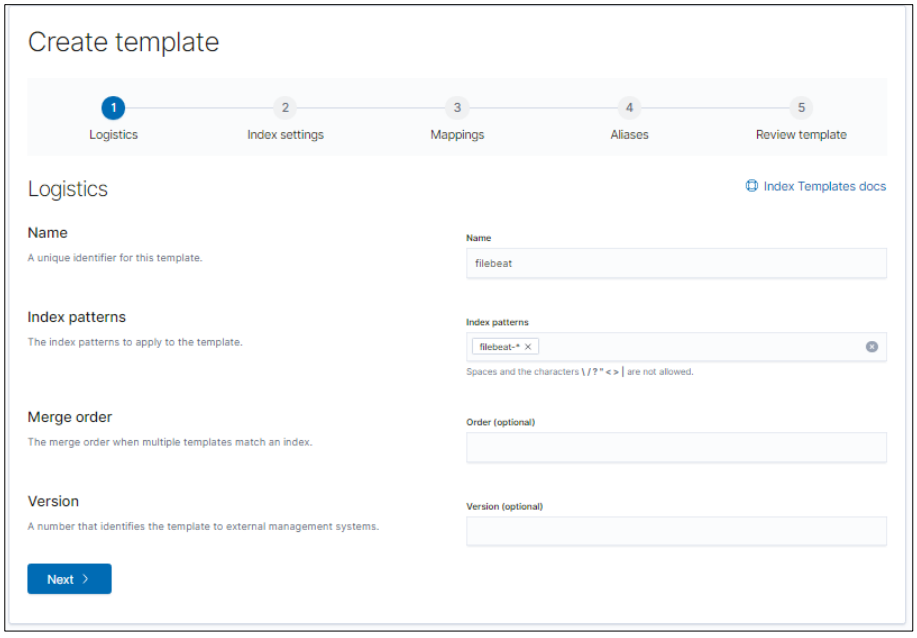


FIGURA 3-8: ELASTICSEARCH, CREAR INDEX TEMPLATE (1)

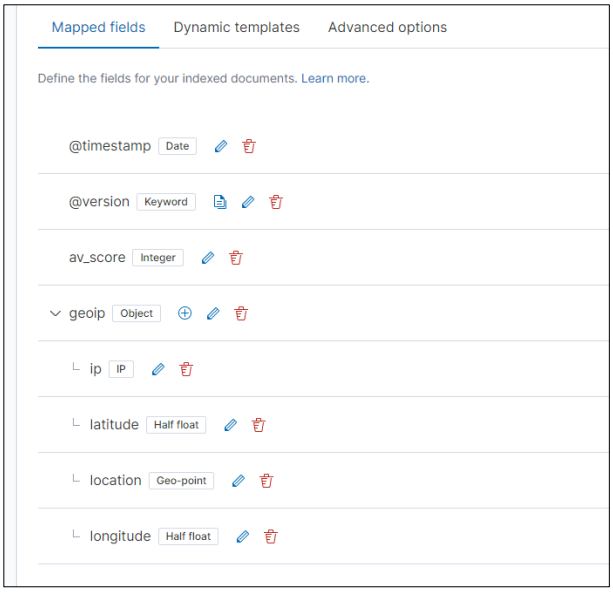


FIGURA 3-9: ELASTICSEARCH, CONFIGURACIÓ DE CAMPS

3.4.3. Logstash

Logstash s'instal·la al servidor virtual "ELKserver", i no a la Raspberry Pi, perquè consumeix uns recursos considerables a l'utilitzar una màquina virtual de Java. Per tant, s'ha d'instal·lar la versió de Java 8 o 11 al servidor (la descàrrega aproximadament són 110MB):

```
sudo apt install openjdk-8-jre-headless
```

Per comprovar que s'ha instal·lat correctament, executar:

```
java -version
```

La versió de Java instal·lada és:

```
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-8u242-b08-
0ubuntu3~18.04-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

El paquet Debian per Logstash es pot descarregar des del lloc web d'Elastic[19] o a través dels repositoris APT. Com s'instal·la en el servidor Ubuntu, es pot utilitzar el repositori APT per a instal·lar Logstash.

S'instal·la la versió més recent estable de Logstash, corresponent a la versió 7.6.2.

[19] Instal·lador Logstash per a Debian: https://www.elastic.co/es/downloads/logstash

Cal descarregar la clau del repositori i afegir-la el repositori de claus, així com afegir la definició d'Elastic 7.6 dels repositoris; però com prèviament ja s'ha instal·lat Elasticsearch i Kibana en aquest servidor, no cal tornar a fer-ho.

Així que s'actualitzen els repositoris i s'instal·la Kibana (la descàrrega són aproximadament uns 175MB):

```
sudo apt-get update && sudo apt-get install logstash
```

Logstash queda instal·lat, però aquest no s'inicia automàticament. L'arrencada o parada de Logstash és diferent segons el sistema operatiu, ja que es pot realitzar de dues maneres, depenent si aquest utilitza SysV init o systemd. En el cas de Ubuntu Server és systemd (es pot saber executant la comanda "*ps -p 1*").

Per tant, es configura Logstash perquè s'iniciï automàticament quan s'inicia el sistema, executant:

```
sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable logstash.service
```

Per a iniciar manualment el servei de Kibana en aquest moment, executar la següent comanda:

```
sudo systemctl start logstash.service
```

I en cas que sigui necessari, es pot aturar manualment mitjançant la següent comanda:

```
sudo systemctl stop logstash.service
```

El fitxer de la definició del servei de Kibana és */etc/systemd/system/logstash.service* , i el de la configuració del programa */etc/logstash/logstash.yml* .

Per comprovar un funcionament bàsic, s'executa Logstash en mode interactiu per línia de comandes, fent que demani informació per teclat (*input*):

```
sudo /usr/share/logstash/bin/logstash -e 'input { stdin { } }
output { stdout {} }'
```

I la retorni per la sortida estàndard (output), és a dir, hauria de sortir per pantalla el missatge introduït, junt amb un segell de temps:

```
{
      "@version" => "1",
    "@timestamp" => 2020-05-20T15:32:33.393Z,
       "message" => "missatge de prova",
          "host" => "elkserver"
}
```

3.4.3.1. Pipeline de processament

En la carpeta /etc/logstash/conf.d/ es troben els fitxers de configuració o pipelines que es vol que s'executin. En aquest treball es defineix el fitxer **zeek-pipeline.conf** per a que Logstash posi en marxa el procés de recepció dels logs des de Filebeat (input), el tractament (filter) i l'enviament (output) a Elasticsearch.

- **Input**
  - La ingesta de dades és a través de Beats, des del port 5044.
  - La comunicació és SSL (*veure apartat següent*).
- **Filter**
  - S'estableixen unes variables (*[@metadata][entorn]* i *[@metadata][elastic_output]*) per configurar fàcilment la sortida tipus debug i en diferents índex de Elasticsearch.
  - Segons el tipus de log provinent (paràmetre *[fileset][name]*), com per exemple del tipus DNS, HTTP, INTEL, etc, es poden duu a terme diferents tipus de filtres.
  - Els filtres destacats d'aquest apartat són:
    - **geoip**: Per tal d'afegir les dades relacionades en la geolocalització de les diferents comunicacions.
    - **http**: Per tal de realitzar consultes REST API a serveis web, en aquest cas a VirtusTotal. En aquest cas, és quan prové un log del tipus "files" i conté el camp *[zeek][files][md5]*.
    - **mutate**: Per tal de realitzar modificacions en els camps, crear-ne nous o eliminar-los. El camp més important que es crea és el *[av_score]*, quan els logs són del tipus "files", on se li assginen els número de deteccions per part dels diferents motors d'antivirus de VirusTotal.
- **Output**
  - La sortida és a Elasticsearch, que es troba en el propi servidor (localhost) i en el port 9200.
  - L'índex d'Elasticsearch és anomenat automàticament segons l'entorn d'execució i el tipus de log (exemple: "filebeat-real-notice"):

```
"%{[agent][type]}-%{[@metadata][entorn]}-%{[fileset][name]}"
```

El contingut íntegre del fitxer *zeek-pipeline.conf* es troba en l'Annex 4: Pipeline de processament Logstash.

3.4.3.2. Comunicació xifrada entre Filebeat i Logstash

A l'utilitzar el client Filebeat i el servei Logstash en equips separats, és important configurar el xifrat SSL en la comunicació entre ells. Primer, cal generar un certificat (autofirmat) al servidor, i a continuació es configurar la connexió SSL, tant per Filebeat (emissor), com per Logstash (receptor).

Per generar un certificat SSL mitjançant OpenSSL al servidor ELK, es realitza una copia del fitxer de configuració d'OpenSSL i a continuació, cal editar-lo.

```
sudo mkdir /etc/pki
```

```
sudo cp /etc/ssl/openssl.cnf /etc/pki/openssl_elk.cnf

sudo nano /etc/pki/openssl_elk.cnf
```

En l'apartat *[v3_ca]*, afegir la següent línia a continuacço, i guardar el fitxer.

```
subjectAltName = IP: 192.168.1.144
```

Per generar el certificat SSL i la clau privada, executar:

```
sudo openssl req -x509 -batch -nodes -days 3650 -newkey
rsa:4096 -keyout /etc/pki/logstash.key -out
/etc/pki/logstash.crt -config /etc/pki/openssl_elk.cnf
```

Es configura Logstash per utilitzar el certificat i clau creades. Editar el fitxer pipeline de configuració corresponent i modificar l'input.

```
sudo nano /etc/logstash/conf.d/zeek-pipeline.conf
```

En l'input beats, a part de configurar el port d'entrada, cal afegir la configuració SSL següent:

```
input {
        beats {
                port => "5044"
                ssl => true
                ssl_certificate => "/etc/pki/logstash.crt"
                ssl_key => "/etc/pki/logstash.key"
        }
}
```

Guardar el fitxer i reiniciar el servei Logstash:

```
sudo systemctl restart logstash.service
```

Per configurar el client Filebeat, es necessita copiar el fitxer "logstash.crt" al dispositiu ZeekPi. S'ubica a la ruta "/etc/pki/logstash.crt". Es configura Filebeat per utilitzar el certificat SSL, per això, cal editar el fitxer de configuració de Filebeat.

```
sudo nano /etc/filebeat/filebeat.yml
```

En l'apartat "output.logstash", afegir la següent configuració referent al certificat:

```
output.logstash:
        hosts: ["192.168.1.144:5044"]
        ssl.certificate_authorities: ["/etc/pki/logstash.crt"]
```

Guardar el fitxer i reiniciar el servei Filebeat:

```
sudo systemctl restart filebeat.service
```

3.5. Detectar activitats sospitoses

3.5.1. Connexions malicioses: Intel Critical Stack

Zeek disposa d'un Framework d'intel·ligència capaç de detectar llocs web de programari maliciós o altres tipus per mitjà de signatures. Per aconseguir-ho necessita alimentar-se dels fitxers que contenen un llistat de direccions web, adreces IP i dominis fraudulents.

Per alimentar la base de dades d'intel·ligència de Zeek, s'obta per la instal·lació de **Intel Critical Stack**[20], un complement desenvolupat específicament per a Zeek i capaç d'alimentar-lo amb una gran varietat de Feeds externs configurables. Per a saber quins llocs web, IPs o dominis maliciosos es visiten, caldrà tenir en compte el fitxer de log "intel.log", el fitxer de log corresponen al framework d'intel·ligència de Zeek. Els tipus d'alertes que podrem tenir són:

- **Intel::DOMAIN** → Indica que l domini on s'accedeix podria ser maliciós.
- **Intel::ADDR** → Indica que l'adreça IP on s'accedeix podria ser maliciosa.
- **Intel::URL** → Indica que la direcció URL on s'accedeix podria ser maliciosa.

3.5.1.1. Preparar Feeds d'intel·ligència

Per integrar Zeek amb Intel Critical Stack i poder utilitzar-lo, abans cal crear un compte d'usuari al seu lloc web i realitzar aquestes accions, per tal de preparar un conjunt de dades que formaran part del fitxer d'intel·ligència:

1. Crear una Col·lecció (*Collection*).
2. Seleccionar les Fonts d'intel·ligència d'amenaces (*threat intelligence Feeds*) per emplenar la col·lecció.
3. Crear un Sensor. Un Sensor és el sistema on es desplegarà la Col·lecció creada anteriorment. En aquest cas, correspondrà a la Raspberry Pi.

Una col·lecció és un contenidor per a la intel·ligència d'amenaces (*threat intelligence*) requerida. Es crea la col·leció "Threat Intel Collection", mitjançant el botó "Create New Collection" (Figura 3-10).

[20] Lloc web Intel Critical Stack: https://intel.criticalstack.com

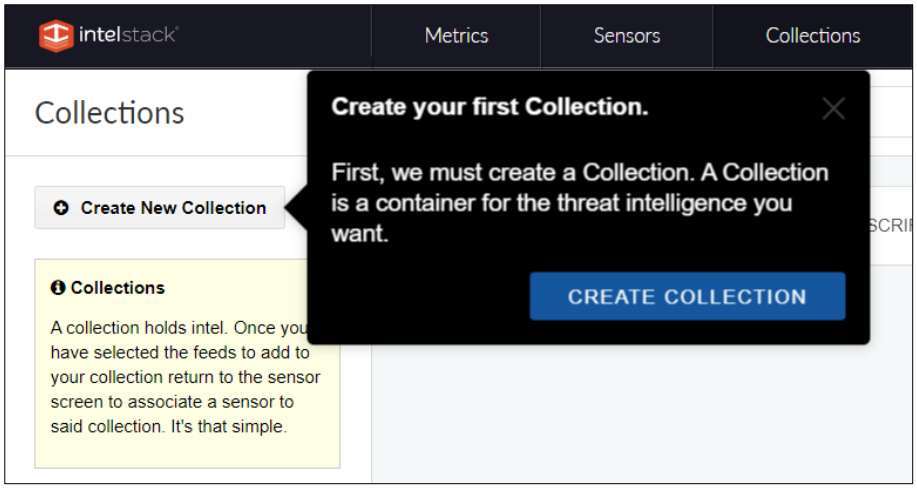


FIGURA 3-10: INTEL CRITICAL STACK, CREAR NOVA COL·LECIÓ

La col·lecció “Threat Intel Collection” s’ha creat (Figura 3-11).

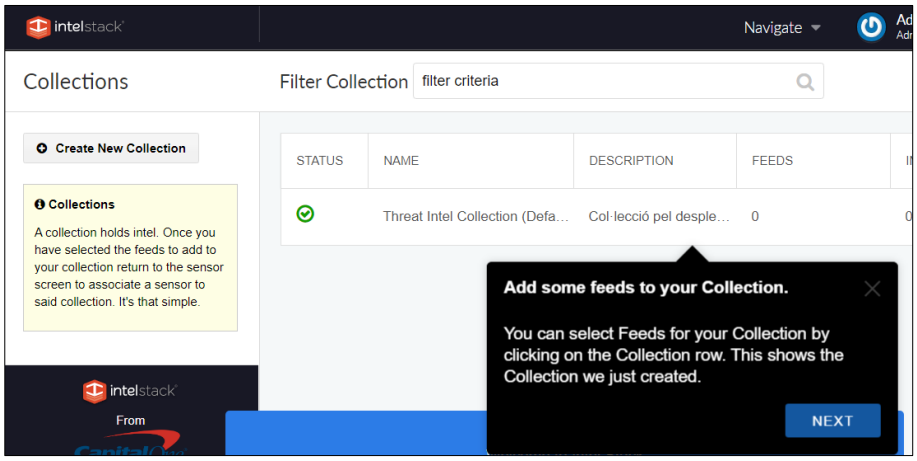


FIGURA 3-11: INTEL CRITICAL STACK, COL·LECCIÓ "THREAD INTEL COLLECTION"

Afegir les fonts (Feeds) a la col·lecció creada (Figura 3-12). Cal seleccionar les fonts de les quals volem activar la subscripció. Un cop es vagin seleccionant, apareixen a la secció “My Feeds”.

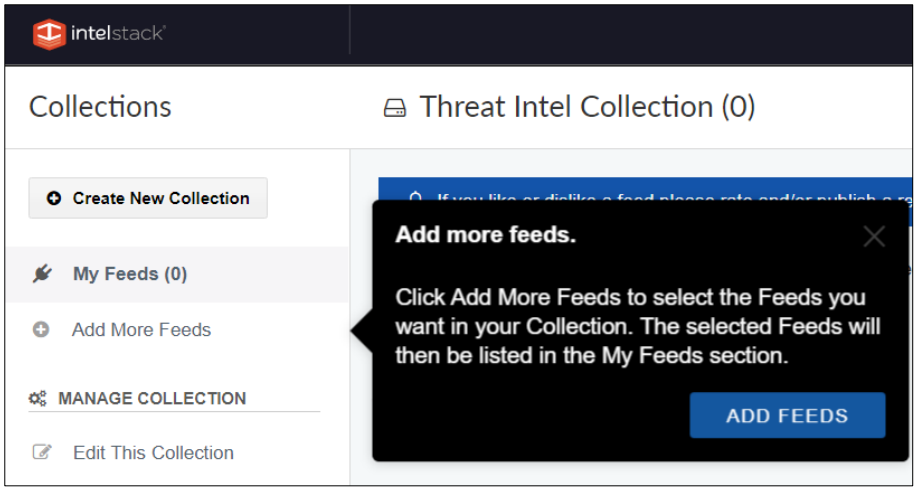


FIGURA 3-12: INTEL CRITICAL STACK, AFEGIR FEEDS A LA COL·LECCIÓ

Es poden seleccionar tants Feeds com es vulgui, però es considera que per realitzar la implantació d'aquest Projecte, amb una selecció de sis Feeds serà correcte. En total, les fonts seleccionades contindran un total d'uns 235.000 hosts i/o IP's malicioses, phising, etc.

Els Feeds seleccionats (Figura 3-13) són:

- **malware URLs from the URLhaus data base**
  - abuse.ch URLhaus Database Dump
  - URL: https://urlhaus.abuse.ch
- **blocklist.de IP Blocklist**
  - Free and voluntary service provided by a Fraud/Abuse-specialist.
  - URL: http://www.blocklist.de
- **PhishTank Intel Feed (Verified)**
  - PhishTank is a collaborative clearing house for data and information about phishing on the Internet.
  - URL: https://www.phishtank.com
- **hosts-file.net-Phishing-Domains**
  - This file contains phishing sites listed in the hpHosts database.
  - URL: https://hosts-file.net
- **snort.org IP Blacklist**
  - IP blacklist provided by the makers of Snort
  - URL: https://snort.org
- **torproject.org Official Exit Node List**
  - The official exit node list published by the Torproject.
  - URL: https://torproject.org

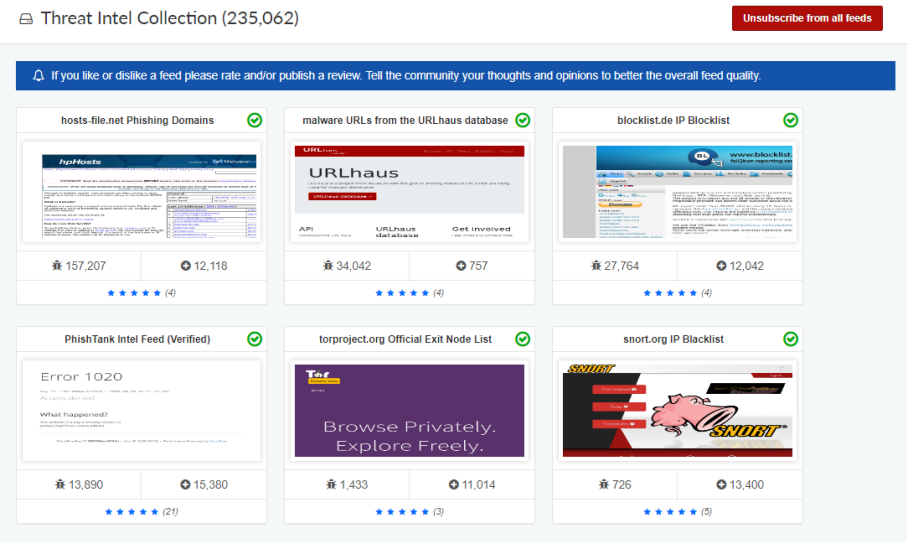


FIGURA 3-13: INTEL CRITICAL STACK, FEEDS

En l'apartat "Sensors" (Figura 3-14), crear el Sensor "ZeekPi", lligat a la col·lecció "Threat Intel Collection".

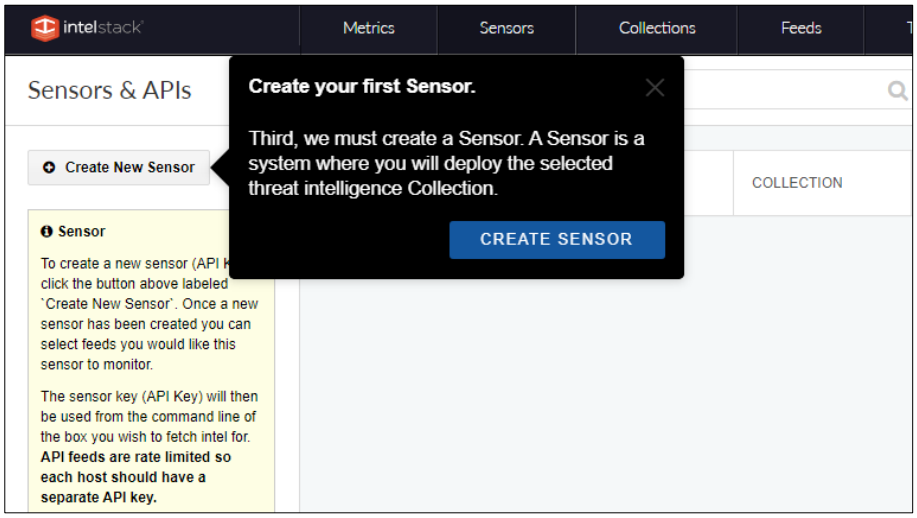


FIGURA 3-14: INTEL CRITICAL STACK, CREAR SENSOR (1)

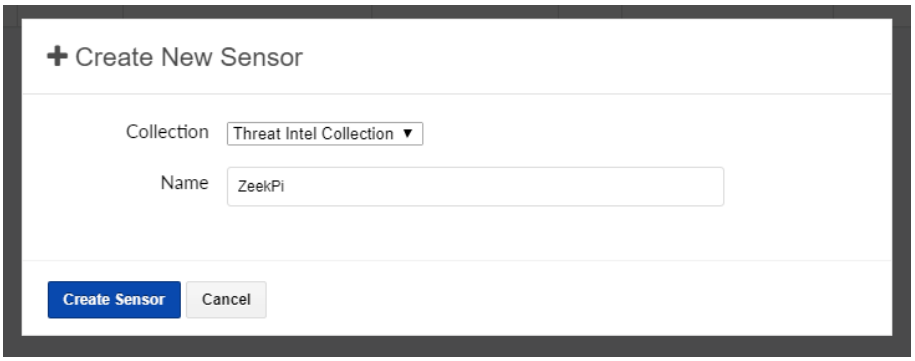


FIGURA 3-15: INTEL CRITICAL STACK, CREAR SENSOR (2)

Un cop es disposa de una col·lecció (*Threat Intel Collection*) emplenada amb els Feeds, i el Sensor *ZeekPi* (Figura 3-16) per buscar-hi amenaces d'intel·ligència, cal accedir a la clau API del Sensor i posteriorment instal·lar el software a la Raspberry Pi.

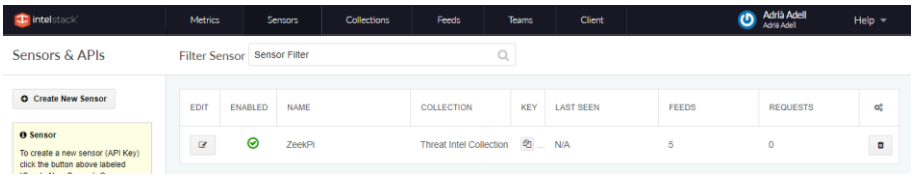


FIGURA 3-16: INTEL CRITICAL STACK, SENSOR ZEEKPI

La clau API (Figura 3-17) necessària per realitzar les descàrregues dels Feeds associats al Sensor "ZeekPi" és la següent:

```
7b058f6f-1869-4070-4714-fe460c66274e
```

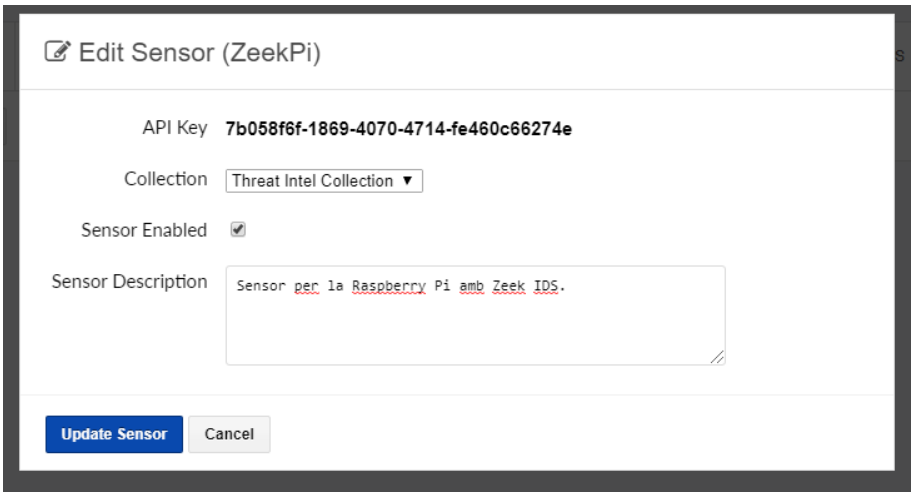


FIGURA 3-17: INTEL CRITICAL STACK, CLAU API

3.5.1.2. Instalar client Intel Stack

Un cop disposem de la Col·lecció amb els Feeds configurats, i el Sensor creat pel client "ZeekPi", cal instal·lar el client d'Intel Stack a la Raspberry Pi, on hi ha Zeek IDS instal·lat.

Per la instal·lació del client Intel Stack, cal afegir el repositori corresponent i la clau del repositori per a poder confiar-hi. La finalitat és poder distribuir paquets i actualitzacions. La descàrrega de la clau i la incorporació del nou repositori és fa automàticament amb l'execució del següent script. L'script també instal·la el paquet "apt-transport-https" i actualitza els repositoris corresponents.

```
curl
https://packagecloud.io/install/repositories/intelstack/client/
script.deb.sh | sudo bash
```

S'instal·la el client Intel Stack (versió 0.6.2), mitjançant la comanda:

```
sudo apt-get install intel-stack-client
```

S'ha d'especificar que Zeek és el nostre motor de detecció (Network Security Monitoring) per a que el client d'Intel Stack generi les dades amb el format correcte:

```
sudo -u intel-stack-client -g intel-stack-client intel-stack-
client nsm zeek
```

Es configura el Sensor, mitjançant la clau API corresponent:

```
sudo -u intel-stack-client -g intel-stack-client intel-stack-
client api 7b058f6f-1869-4070-4714-fe460c66274e
```

Es desactiva la inclusió automàtica de la configuració al fitxer "local.zeek".

```
sudo -u intel-stack-client -g intel-stack-client intel-stack-
client config --set zeek.include=false
```

Per tant, es modificar el fitxer de Zeek manualment:

```
sudo nano /opt/zeek/share/zeek/site/local.zeek
```

Afegir al final, la càrrega dels scripts i frameworks d'intel·ligència:

```
# Framework Intel (Critical Stack)
@load base/frameworks/intel
@load frameworks/intel/seen
@load frameworks/intel/do_notice

redef Intel::read_files += {
        "/opt/intel-stack-client/frameworks/intel/master-
public.dat"
};
```

Realitzar un *Pull Feeds* del client Intel Stack:

```
 sudo -u intel-stack-client -g intel-stack-client intel-stack-
client pull
```

Per veure els Feeds configurats (Figura 3-18), i veure les dades que contenen o la seva última data d'actulització, es pot executar:

```
sudo -u intel-stack-client -g intel-stack-client intel-stack-
client list
```

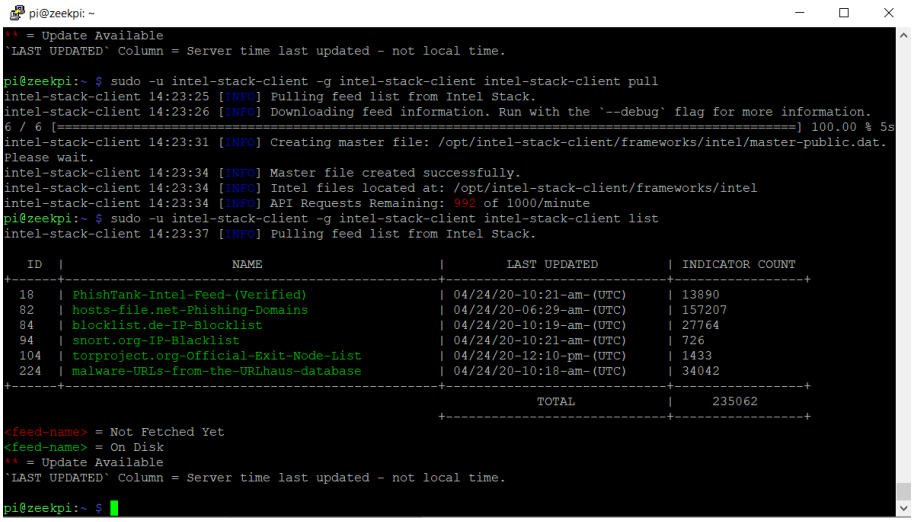


FIGURA 3-18: CLIENT INTEL STACK, FEEDS DISPONIBLES

Ara, mitjançant Zeekctl, comprovar la nova configuració i instal·lar les actualitzacions de recepció de feeds a Zeek:

```
sudo -u zeek -g zeek /opt/zeek/bin/zeekctl check

sudo -u zeek -g zeek /opt/zeek/bin/zeekctl install

sudo -u zeek -g zeek /opt/zeek/bin/zeekctl restart
```

3.5.1.3. Actualització automàtica de Feeds

Per a configurar la descàrrega dels Feeds i el reinici de Zeekctl automàticament, es configuren tres tasques al “CRON” (una per l’usuari “intel-stack-client”, i dues per l’usuari “zeek”) per a que executi tasques corresponents.

Editar el crontab de l'usuari intel-stack-client:.

```
sudo crontab -u intel-stack-client -e
```

Afegir la següent tasca cron (s’executa diàriament a la 1:00):

```
0 1 * * * intel-stack-client pull
```

Editar el crontab de l'usuari zeek:.

```
sudo crontab -u zeek -e
```

Afegir les següents tasques cron per realitzar l’actualització de les fonts d’intel·ligència a Zeek (s’executen diàriament a les 1:05 i 1:06, uns minuts després de l’actualització del Feeds):

```
5 1 * * * /opt/zeek/bin/zeekctl install
6 1 * * * /opt/zeek/bin/zeekctl restart
```

3.5.2. Fitxers maliciosos: Integració amb VirusTotal

VirusTotal[21] és un servei cloud ampliament reconegut i amb gran reputació per analitzar arxius, URLs, dominis i direccions IP en busca de malware, ja que conta amb una gran base de dades. En l'actualitat, és capaç d'inspeccionar els elements amb més de 70 escàners d'antivirus i disposa d'un servei de llistes negres de URLs i dominis.

Mitjançant el navegador web, un software pels clients o extensions d'alguns navegadors, qualsevol usuari és capaç d'analitzar un fitxer del seu equip enviant-lo a VirusTotal. A més però, VirusTotal ofereix realitzar peticions mitjançant programació a través de la seva API pública, basada en HTTP.

Així que, gràcies al Framework "Files" de Zeek, que ens proporciona el valor hash MD5 i SHA-1 de cada un dels fitxers de la xarxa, es realitzaran peticions HTTP a VirusTotal per tal d'esbrinar les deteccions d'antivirus que genera aquell fitxer.

Per poder consultar a VirtusTotal, abans cal crear un compte d'usuari al seu lloc web[22] i obtenir la clau API corresponent.

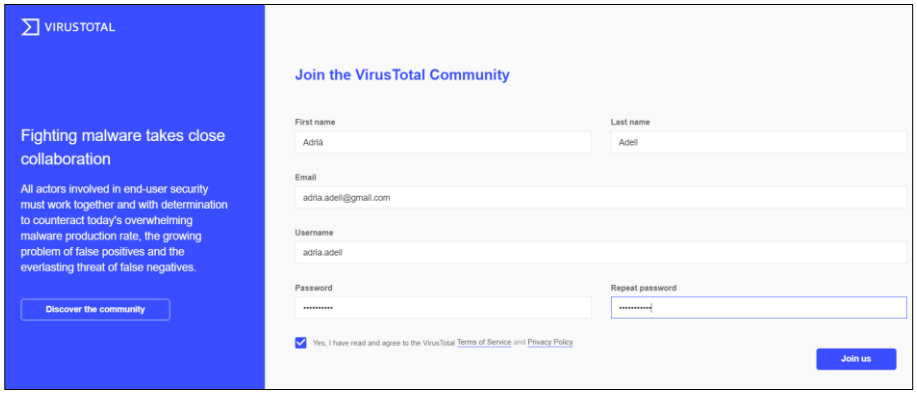


FIGURA 3-19: COMUNITAT VIRUSTOTAL

La clau API assignada és:

```
fc59c20bf3002b44262f058efc647a55fc1a68391f5c2d69b0bd5704d6e8305a
```

La documentació API (versió 3, la darrera versió i més actual) de referència es troba a:

- https://developers.virustotal.com/v3.0/reference

La clau API utilitzada pel projecte és la Public. A la Figura 3-20 es mostra una taula comparativa amb les diferències existents entre la Public API i la Private API.

[21] Web VirusTotal: https://www.virustotal.com
[22] Comunitat VirusTotal: https://www.virustotal.com/gui/join-us

	Public API	Private API
Cas d'us	No-Comercial Acadèmic	Comercial Governamental
Nº peticions	4 per minut	Configurable
Informació addicional	Bàsica	Informació de fiters i URL mitjançant eines integrades a VirusTotal
Sandbox	No	Segons el tipus de fitxer
API búsqueda	Per hash	Avançada
Preu	Gratuït	Pagament, segons ús.
SLA	Cap	99% de disponibilitat

FIGURA 3-20: VIRUSTOTAL, COMPARATIVA PUBLIC API VS PRIVATE API

Entre totes les funcions que ofereix, s'utilitzarà només la "files"[23]. Aquesta funció permet que solament disposant del hash del fitxer (SHA-256, SHA-1 o MD5), ens retorna el resultat complet de l'anàlisi de la seva base de dades.

Les consultes sobre la reputació de cada un dels fitxers es llençarà en el pipeline de Logstash, en l'apartat dels filtres. El filtre HTTP[24] de Logstash proporciona integració amb serveis web externs (REST APIs).

La consulta s'ha de realitzar mitjançant una petició HTTP del tipus GET, a la URL "*https://www.virustotal.com/api/v3/files/Hash_Fitxer*". A més, cal afegir a la capçalera HTTP de la petició, la clau API assignada. Així doncs, el filtre Logstash corresponent a la consulta té el següent format:

```
http {
        url =>
"https://www.virustotal.com/api/v3/files/Hash_Fitxer"
        headers => { "x-apikey" => "
fc59c20bf3002b44262f058efc647a55fc1a68391f5c2d69b0bd5704d6e8305a" }
}
```

Només es realitzen les consultes del fitxers capturats per Zeek que no són descarregats via SSL, ja que en aquest cas el contingut és xifrat i el hash no seria vàlid realment.

La consulta REST API mitjançant retorna un fitxer JSON (veure Figura 3-21: VirusTotal, exemple resposta en format JSONFigura 3-21), el qual per defecte s'emmagatzema a un camp nou anomenat "body".

[23] Funció File de l'API VirtusTotal https://developers.virustotal.com/v3.0/reference#file
[24] Filtre HTTP de Logstash: https://www.elastic.co/guide/en/logstash/current/plugins-filters-http.html

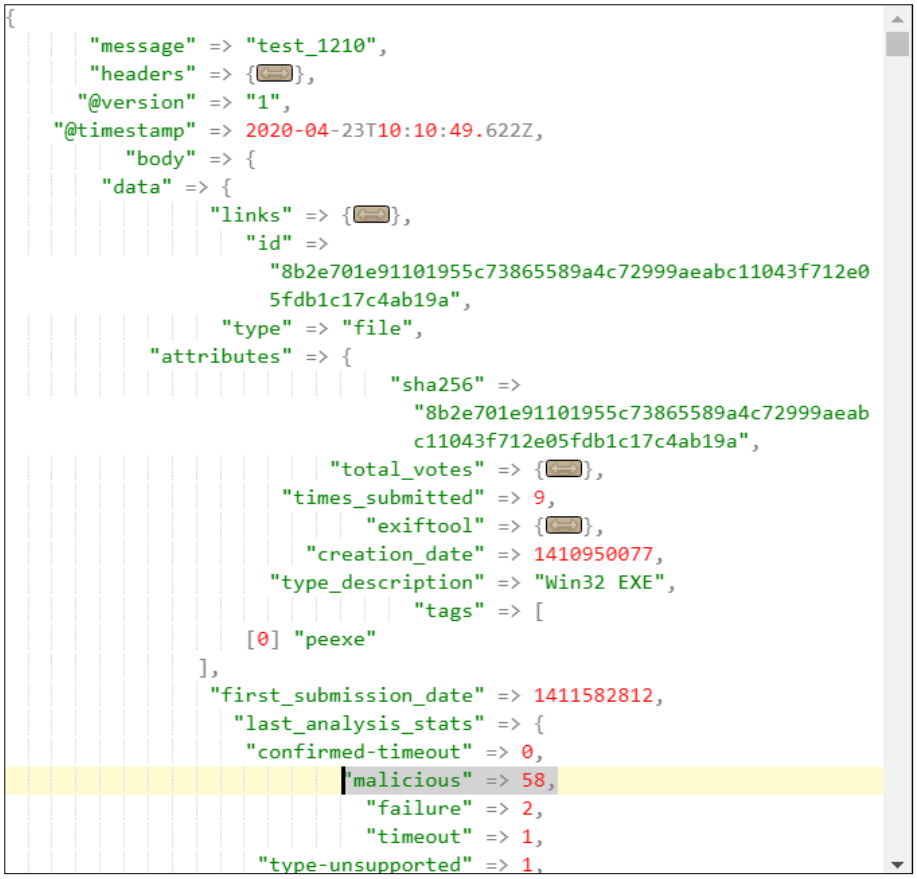


FIGURA 3-21: VIRUSTOTAL, EXEMPLE RESPOSTA EN FORMAT JSON

De tota la informació obtinguda (molt extensa i detallada, ja que conté més de 1.000 camps), només es guarda el nombre de positius (contingut maliciós) dels motors d'antivirus. Aquest número es guarda en un camp nou anomenat "av_score" (de tipus integer). Per crear el camp, s'utilitza el plugin "mutate" de Logstash.

```
av_score =
[body][data][attributes][last_analysis_stats][malicious]
```

La resta de informació no es guarda a Elasticsearch (s'elimina el camp "body" amb el plugin "mutate" de Logstash); es considera que no és útil guardar uns resultats estàtics de l'anàlisi d'un fitxer en el dia que es detecta, ja que tenint el hash del fitxer, es poden iniciar les investigacions pertinents i tornar a realitzar el anàlisis.

3.5.3. Altres activitats sospitoses

Es detectaran mitjançant el framework Notice de Zeek altres activitats sospitoses que es puguin produir a la xarxa. Aquest framework permet a Zeek detectar activitat estranya o comportament potencialment dolents a la xarxa mitjançant un anàlisi en profunditat del tràfic de xarxa.

Segons les diferents necessitats dels usuaris de l'aplicació, Zeek permet personalitzar quins mòduls es volen activar o no, mitjançant la configuració local.

Els elements que Zeek pot arribar a detectar mitjançant aquest framework i els diferents tipus d'alertes que podrem tenir són:

- **Software::Vulnerable_Version** → Indica que s'ha detectat una versió vulnerable d'un software. Tenir versions vulnerables pot conduir a un incident de seguretat, com execució arbitrària de codi o escalada de privilegis.
- **CaptureLoss::Too_Much_Loss** → Informa si hi ha un rati alt de pèrdua de paquets capturats detectada. Això en si no és incident de seguretat, però si es produeix, és probable que degut a la pèrdua de paquets, no es puguin detectar la resta d'atacs.
- **Scan::Port_Scan** → Detecta que un host està escanejant un equip en diversos ports. L'equip en qüestió pot ser víctima d'un escàner de ports, amb la finalitat de trobar alguna vulnerabilitat per on poder atacar.
- **FTP::Bruteforcing** → Detecta que un host ha realitzat un gran nombre d'errors en l'inici de sessió FTP. És important prendre les mesures adients o investigar-ne el motiu. En un cas greu podria servir per inclús tancar el servei FTP per evitar mals majors.
- **HTTP::SQL_Injection_Attacker** → Detecta que un host està realitzant un atac d'injecció SQL. Si el host és de la xarxa interna és un problema, ja que o un usuari intenta vulnerar alguna aplicació, o pot ser un equip infectat per algun tipus de malware que ho realitzi.
- **HTTP::SQL_Injection_Victim** → Detecta que un host està sent víctima d'un atac d'injecció SQL. En aquest cas, cal protegir el servidor, ja sigui realitzant una desconnexió de la xarxa de l'atacant, i si es pot, apagar el servei.
- **SSH::Password_Guessing** → Indica que un host ha superat un llindar d'inicis de sessió fallits en SSH. Si es permeten les connexions SSL a la xarxa interna, podria indicar algun tipus de suplantació o un intent per robar credencials.
- **SSL::Certificate_Expired** → Indica que un certificat SSL està caducat. Cal tenir cura que el lloc web continuï sent verídic.
- **SSL::Invalid_Server_Cert** → El certificat d'un lloc web no és vàlid. Cal tenir compte, ja que es podria tractar de servidors fraudulents amb certificats que no són de confiança, o un fals positiu i que realment no estigués ben configurat.

3.6. Dashboards Kibana

Per la creació dels taulers de control, o Dashboards, cal accedir a l'apartat Dashboards de Kibana. Un Dashboard conté una recull de visualitzacions com gràfiques, cerques, mapes[25] o mètriques en temps real o en dates concretes, per tal d'oferir-nos una informació general de la xarxa. L'objectiu és que de forma

[25] Els mapes de Kibana tenen com a base el següent tipus de mapa: https://basemap.nationalmap.gov/arcgis/services/USGSImageryTopo/MapServer/WMSServer

molt intuïtiva veure alertes, i poder observar certs comportaments anòmals i, que a partir d'aquí, ens permeti aprofundir en els detalls si cal.

Es creen els següents Dashboards, tots ells amb una visualització predeterminada dels últims 7 dies:

- **Dashboard general de seguretat**
- **Dashboard de fitxers i deteccions de virus**
- **Dashboard de connexions malicioses**
- **Dashboard d'altres activitats sospitoses**

3.6.1. Dashboard general de seguretat

El Dashboard general de seguretat és el que conté una visió generals de tots els aspectes a tenir en compte de forma resumida i concisa. En cas de voler aprofundir en algun dels apartats, es pot anar a qualsevol dels altres Dashboards per veure més detalls, ja sigui el que conté informació de fitxers, de connexions o d'altres activitats anòmales.

FIGURA 3-22: DASHBOARD GENERAL DE SEGURETAT

El panell general de seguretat (Figura 3-22) conté les següents visualitzacions de Kibana (ordenades de esquerra a dreta, i de dalt a baix):

- **Comptador de deteccions d'antivirus**: Mostra el nombre de fitxers que han donat almenys 1 positiu en algun motor d'antivirus.
  - Type: *Gauge - Arc*
  - Metrics: *Count*
  - Buckets
    - Aggregation: *Filters*
    - Filter: *av_score >= 1 and zeek.files.source.keyword : "HTTP"*
- **Comptador de connexions malicioses**: Mostra el nombre de connexions malicioses detectades.
  - Type: *Gauge - Arc*

    - Metrics: *Count*
    - Buckets
        - Aggregation: *Filters*
        - Filter: *zeek.intel.matched : **
- **Comptador d'activitats sospitoses a la xarxa**: Mostra el nombre d'events detectats que podrien ser rellevants per la seguretat.
    - Type: *Gauge - Arc*
    - Metrics: *Count*
    - Buckets
        - Aggregation: *Filters*
        - Filter: *zeek.notice.note.keyword : **
- **Timeline de l'activitat a la xarxa (per tipus de log)**: Mostra un historial en el temps de cada un dels tipus de log. Pot servir per observar activitats en hores estranyes, per exemple.
    - Type: *TSVB – Time Series*
    - Metrics: *Count*
    - Group by: *Terms*
        - Terms: *fileset.name.keyword*
- **Mapa d'informació geogràfica de connexions (general)**.
    - Type: *Maps*
    - Data source: *Grid aggregation*
    - Index pattern: *filebeat-entorn-**
    - Geospatial field: *geoip.location*
    - Show as: *heatmap*
    - Metrics: *Count*
    - Grid resolution: *fine*
- **Mètriques sobre volum de dades descarregades**: Descàrrega total en bytes.
    - Type: *Metric*
    - Metric:
        - Aggregation: *Sum*
        - Field: *destination.bytes*
- **Mètriques sobre volum de dades carregades**: Càrrega total en bytes.
    - Type: *Metric*
    - Metric:
        - Aggregation: *Sum*
        - Field: *source.bytes*
- **Quantitat de logs (per tipus)**. Gràfic amb la quantitat absoluta de cada dels diferents logs.
    - Type: *Area*
    - Metrics: *Y-axis*
        - Aggregation: *Count*
    - Buckets: *X-axis*
        - Aggregation: *Terms*
            - Terms: *fileset.name.keyword*
- **Informació geogràfica de connexions per país i ciutat (general)**: Gràfic circular amb el detall de les connexions de cada país i ciutat.
    - Type: *Pie – Donut*
    - Metrics: *Count*
    - Buckets:
        - Split slices
            - Aggregation: *Terms*
            - Field: *geoip.country_name.keyword*
        - Split slices

- Sub aggregation: *Terms*
- Field: *geoip.city_name.keyword*

3.6.2. Dashboard de connexions malicioses

El Dashboard de connexions malicioses és el que conté una visió de les connexions realitzades a dominis, URLs o IPs amb mala reputació, segons els *Feeds* configurats a Intel Critical Stack (framework d'intel·ligència de Zeek). En cas de voler aprofundir en algun incident d'aquest tipus, també s'ha afegit una cerca establerta a la part inferior amb la informació dels incidents trobats.

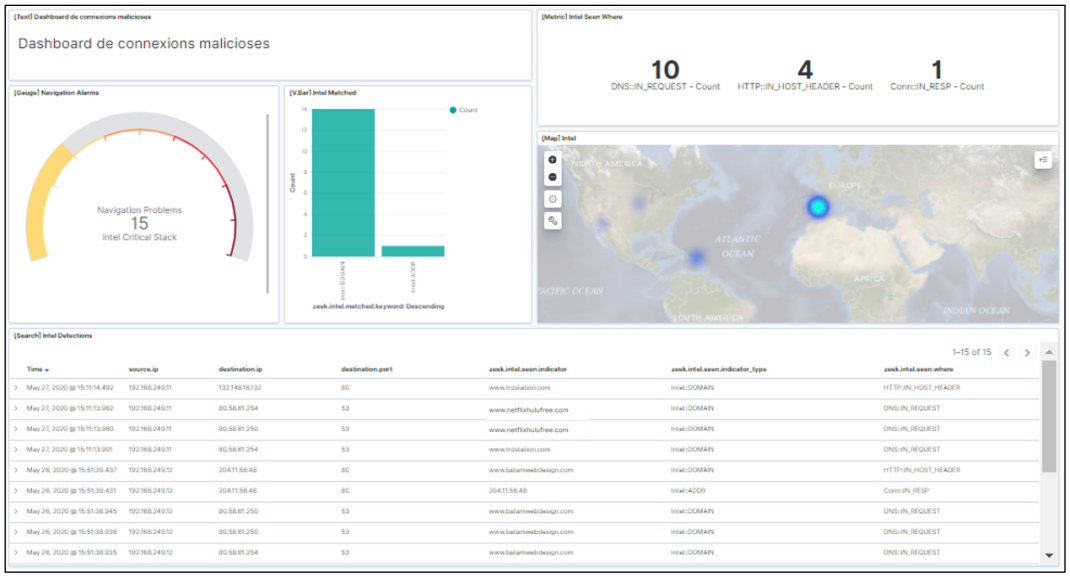


FIGURA 3-23: DASHBOARD DE CONNEXIONS MALICIOSES

El panell de connexions malicioses (Figura 3-23) conté les següents visualitzacions de Kibana (ordenades de esquerra a dreta, i de dalt a baix):

- **Comptador de connexions malicioses**: Mostra el nombre de connexions malicioses detectades (és el mateix que el del Dashboard general).
  - Type: *Gauge - Arc*
  - Metrics: *Count*
  - Buckets
    - Aggregation: *Filters*
    - Filter: *zeek.intel.matched : **
- **Tipus d'alerta d'intel·ligència**: Mostra el nombre de alertes per cada una de les troballes realitzades, ja sigui domini, URL o IP.
  - Type: *Vertical Bar*
  - Metrics: *Y-axis*
    - Aggregation: *Count*
  - Buckets: *X-axis*
    - Aggregation: *Terms*
    - Field: *zeek.intel.matched.keyword*
- **Diferents mètriques sobre les connexions malicioses**. Mostra la quantitat d'alertes trobades i en quin lloc s'ha detectat (consulta DNS, capçalera HTTP, resposta entrant, etc).

    - Type: *Metric*
    - Metrics: *Count*
    - Buckets: *Split group*
        - Aggregation: *Terms*
        - Field: *zeek.intel.seen.where.keyword*
- **Mapa d'informació geogràfica de les connexions malicioses**.
    - Type: *Maps*
    - Data source: *Grid aggregation*
    - Index pattern: *filebeat-entorn-intel*
    - Geospatial field: *geoip.location*
    - Show as: *heatmap*
    - Metrics: *Count*
    - Grid resolution: *fine*
- **Cerca amb informació bàsica de les connexions malicioses**: Cerca en forma de taula per mostrar les connexions potencialment malicioses, ordenades de més recent a més antiga. Conté informació de l'equip origen, la IP i port de destí, i els diferents indicadors que provenen de Zeek, com el tipus (domini, URL o IP), el tipus i on s'ha detectat.
    - Type: *Search*
    - Cerca realitzada: *fileset.name : intel*
    - Camps mostrats: *Time, source.ip, destination.ip, destination.port, zeek.intel.seen.indicator, zeek.intel.seen.indicator_type, zeek.intel.seen.where*.

3.6.3. Dashboard de fitxers i deteccions de virus

El Dashboard de fitxers i deteccions de virus és el que conté una visió dels fitxers que han sigut detectats a la xarxa i si hi ha alguna alerta. En cas de voler aprofundir en algun incident d'aquest tipus, també s'ha afegit una cerca establerta a la part inferior amb els fitxers que poden ser problemàtics (almenys ha sigut detectat per un motor d'antivirus, segons VirusTotal).

El panell de fitxers i deteccions de virus (Figura 3-24) conté les següents visualitzacions de Kibana (ordenades de esquerra a dreta, i de dalt a baix):

- **Comptador de deteccions d'antivirus**: Mostra el nombre de fitxers que han donat almenys 1 positiu en algun motor d'antivirus (és el mateix que el del Dashboard general).
    - Type: *Gauge - Arc*
    - Metrics: *Count*
    - Buckets
        - Aggregation: *Filters*
        - Filter: *av_score >= 1 and zeek.files.source.keyword : "HTTP"*
- **Tipus de connexió dels fitxers**: Mostra el nombre de fitxers que han passat per la xarxa segons si ho han fet des d'una connexió SSL o una HTTP.
    - Type: *Vertical Bar*
    - Metrics: *Y-axis*
        - Aggregation: *Count*
    - Buckets: *X-axis*
        - Aggregation: *Terms*
        - Field: *zeek.files.source.keyword*

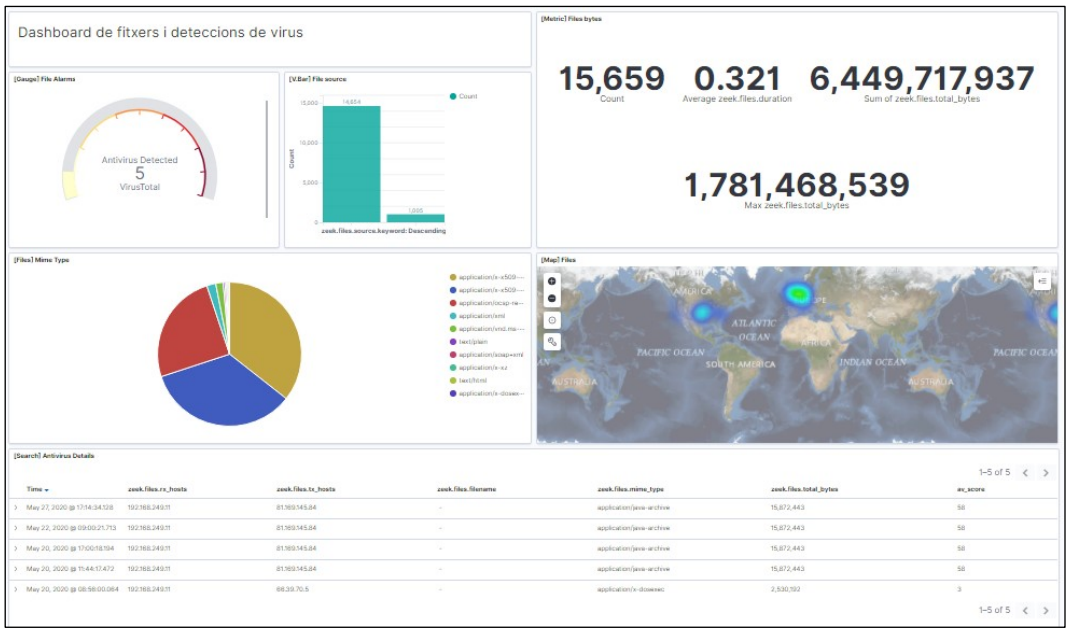


FIGURA 3-24: DASHBOARD DE FITXERS I DETECCIONS DE VIRUS

- **Diferents mètriques sobre fitxers**.
  - Type: *Metric*
  - Metrics:
    1. Nombre total de fitxers vistos
       - Aggregation: *Count*
    2. Duració mitjana del fitxers en tràfic (en segons).
       - Aggregation: *Average*
       - Field: *zeek.files.duration*
    3. Mida total dels fitxers vistos (en bytes).
       - Aggregation: *Sum*
       - Field: *zeek.files.total_bytes*
    4. Mida del fitxer més gran vist (en bytes).
       - Aggregation: *Max*
       - Field: *zeek.files.total_bytes*
- **Mètriques sobre volum de dades carregades**: Càrrega total en bytes.
  - Type: *Metric*
  - Metric:
    - Aggregation: *Sum*
    - Field: *source.bytes*
- **Tipus MIME dels fitxers observats**: Gràfic circular amb el detall del tipus MIME dels fitxers observats a la xarxa.
  - Type: *Pie*
  - Metrics:
    - *Slice size*
    - Aggregation: *Count*
  - Buckets: *Split slices*
    - Aggregation: *Terms*
    - Field: *zeek.files.mime_type.keyword*
- **Mapa d'informació geogràfica de fitxers potencialment maliciosos**.
  - Type: *Maps*
  - Data source: *Grid aggregation*
  - Index pattern: *filebeat-entorn-files*

    - Geospatial field: *geoip.location*
    - Show as: *heatmap*
    - Metrics: *Count*
    - Grid resolution: *fine*
    - Filtering: *av_score >= 1*
- **Cerca amb informació bàsica de fitxers potencialment maliciosos**: Cerca en forma de taula per mostrar els fitxers potencialment maliciosos, ordenats per major número de deteccions dels motors d'antivirus. Conté informació de l'equip origen, la IP de descàrrega, i els diferents indicadors que provenen de Zeek, com el tipus de fitxer, la mida d'aquest, i el nombre de positius detectats.
    - Type: *Search*
    - Cerca realitzada: *fileset.name : files and av_score >= 1 and zeek.files.source.keyword :"HTTP"*
    - Camps mostrats: *Time, zeek.files.rx_hosts, zeek.files.tx_hosts, zeek.files.filename, zeek.files.mime_type, zeek.files.total_bytes, av_score*.

3.6.4. Dashboard d'altres activitats sospitoses

El Dashboard d'altres activitats sospitoses és el que conté una visió d'altres problemes detectats a la xarxa mitjançant l'anàlisis de Zeek (framework Notice). En cas de voler aprofundir en algun incident d'aquest tipus, també s'ha afegit una cerca establerta a la part inferior amb la informació dels incidents trobats.

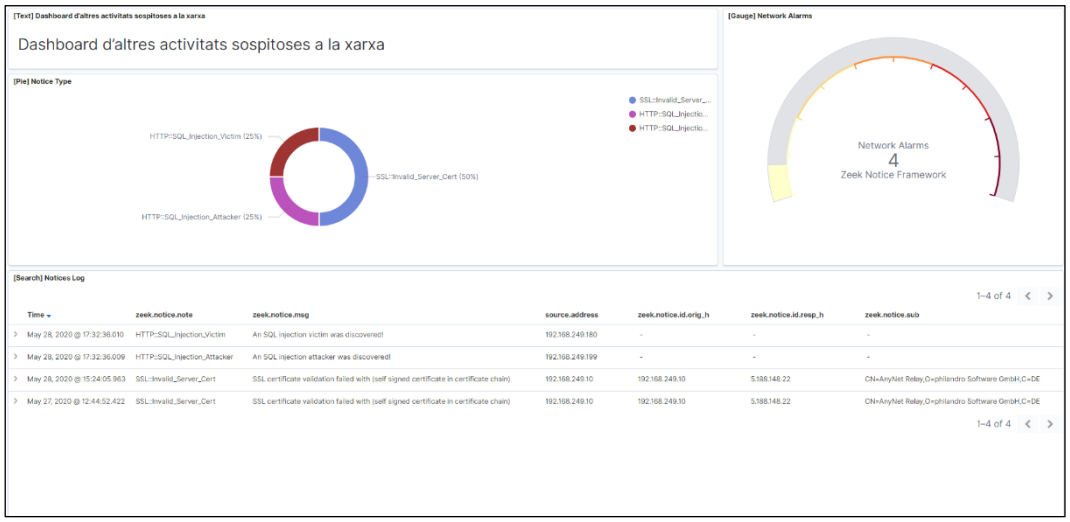


FIGURA 3-25: DASHBOARD D'ALTRES ACTIVITATS SOSPITOSES

El panell d'altres activitats sospitoses (Figura 3-23) conté les següents visualitzacions de Kibana (ordenades de esquerra a dreta, i de dalt a baix):

- **Gràfic circular sobre les diferents alertes de Zeek**.
    - Type: *Pie – Donut*
    - Metrics: *Count*
    - Buckets:
        - Split slices
            - Aggregation: *Terms*
            - Field: *zeek.notice.note.keyword*

- **Comptador d'activitats sospitoses a la xarxa**: Mostra el nombre d'events detectats que podrien ser rellevants per la seguretat (és el mateix que el del Dashboard general).
    - Type: *Gauge - Arc*
    - Metrics: *Count*
    - Buckets
        - Aggregation: *Filters*
        - Filter: *zeek.notice.note.keyword : **
- **Cerca amb informació bàsica de activitats sospitoses**: Cerca en forma de taula per mostrar els diferents tipus d'alertes que pot detectar Zeek, ordenades de més recent a més antiga. Conté informació de l'equip origen/destí o d'on s'ha vist el tipus d'alerta, el tipus de problema i la descripció del problema.
    - Type: *Search*
    - Cerca realitzada: *fileset.name : notice*
    - Camps mostrats: *Time, zeek.notice.note, zeek.notice.msg, source.address, zeek.notice.id.orig_h, zeek.notice.id.resp_h, zeek.notice.sub*.

3.7. Proves de funcionament

Es realitza un joc de proves que consisteix en la generació de tràfic a la xarxa mitjançant dues màquines virtuals, de forma automàtica mitjançant l'execució d'scripts, que simulen navegació per diferents pàgines web. Així com la creació d'un portal web vulnerable a varis tipus d'atacs.

Es creen dues màquines virtual amb el sistema operatiu "Linux Lite"[26] (distribució que necessita poca quantitat de recursos). Aquests clients se'ls configura l'execució d'uns scripts programats mitjançant el Cron:

```
crontab -e
```

Al fitxer Cron, es programa l'execució de tres scripts diferents, un que genera tràfic legítim (*Trafic.sh*), un altre amb tràfic maliciós (*Bad_Trafic.sh*) i per últim un que descarrega fitxers (*Down_Files.sh*).

```
#Visitar webs cada 15 minuts els dies de feina
*/15 9-17 * * 1-5 /home/user1/Desktop/Trafic.sh &>/dev/null
#Visitar webs dolentes per la nit
30 0-8/3 * * * /home/user1/Desktop/Bad_Trafic.sh &>/dev/null
#Descarregar fitxers durant el dia en els dies de feina
0 9-17/3 * * 1-5 /home/user1/Desktop/Down_Files.sh &>/dev/null
```

Cada un dels scripts, fa una petició o descarrega un fitxer de forma aleatòria a partir de tres fitxers de text amb un llistat de URL legítimes, URLs malicioses i de fitxers, respectivament. En l'Annex 6: Detalls en la instal·lació del joc de proves es detallen els scripts i aquests fitxers de text.

[26] https://www.linuxliteos.com

3.7.1. Detectar tràfic i integració de Zeek amb ELK Stack

El primer que es vol comprovar és que Zeek és capaç de veure el tràfic generat i emmagatzemar-lo (als seus propis logs), per tant, primer cal comprovar que Zeek està en marxa i que es van generant els fitxers de log corresponents.

```
sudo -u zeek -g zeek /opt/zeek/bin/zeekctl status
```

L'estat de Zeek és "running", és correcte:

```
Name         Type       Host          Status    Pid    Started
zeek         standalone localhost     running   8087   29 May 01:06:05
```

Es comprova que els logs apareixen a la carpeta */opt/zeek/logs/current/* (Figura 3-26). Si Zeek no està en marxa, la carpeta "current" no apareix.

```
pi@zeekpi:~ $ sudo -u zeek -g zeek ls -l /opt/zeek/logs/current/
total 300
-rw-r--r-- 1 zeek zeek   216 May 29 16:21 capture_loss.log
-rw-r--r-- 1 zeek zeek 89032 May 29 16:25 conn.log
-rw-r--r-- 1 zeek zeek 70145 May 29 16:24 dns.log
-rw-r--r-- 1 zeek zeek 34721 May 29 16:24 files.log
-rw-r--r-- 1 zeek zeek  4769 May 29 16:23 http.log
-rw-r--r-- 1 zeek zeek   791 May 29 16:07 ntp.log
-rw-r--r-- 1 zeek zeek 16699 May 29 16:24 ssl.log
-rw-r--r-- 1 zeek zeek  2468 May 29 16:21 stats.log
-rw-r--r-- 1 zeek zeek    19 May 29 01:06 stderr.log
-rw-r--r-- 1 zeek zeek   188 May 29 01:06 stdout.log
-rw-r--r-- 1 zeek zeek   182 May 29 16:17 weird.log
-rw-r--r-- 1 zeek zeek 33474 May 29 16:24 x509.log
```

FIGURA 3-26: ZEEK EN MARXA I GUARDANT ELS LOGS

Amb l'objectiu de realitzar una demostració del funcionament de solució proposada i de cada una de les casuístiques, es realitzaran les següents proves amb peticions web de forma manual per a comprovar que Zeek detecta el tràfic en viu. Es realitza una petició a la web "**www.google.cat**" des del "**Client 2**". Per comprovar els logs de Zeek (per exemple el DNS):

```
cat /opt/zeek/logs/current/dns.log
```

L'última entrada del log *dns.log*, conté la resposta amb la IP corresponent de domini sol·licitat:


```
{"ts":1590764928.320227,"uid":"Co6XIx3VKdQJ8Mrkfd","id.orig_h":
"192.168.249.12","id.orig_p":49108,"id.resp_h":"80.58.61.250","
id.resp_p":53,"proto":"udp","trans_id":9068,"rtt":0.00646114349
3652344,"query":"www.google.cat","qclass":1,"qclass_name":"C_IN
TERNET","qtype":28,"qtype_name":"AAAA","rcode":0,"rcode_name":"
NOERROR","AA":false,"TC":false,"RD":true,"RA":true,"Z":0,"answe
rs":["2a00:1450:4003:801::2003"],"TTLs":[110.0],"rejected":fals
e}
```


Per comprovar que aquest mateix log és extret per Filebeat, enviat a Logstash, i emmagatzemat a Elasticsearch, es realitza una consulta a Kibana en l'apartat

"Discover" (per veure tot el que està disponible a l'índex seleccionat). En la Figura 3-27 es pot comprovar que l'esdeveniment es mostra correctament a Kibana.

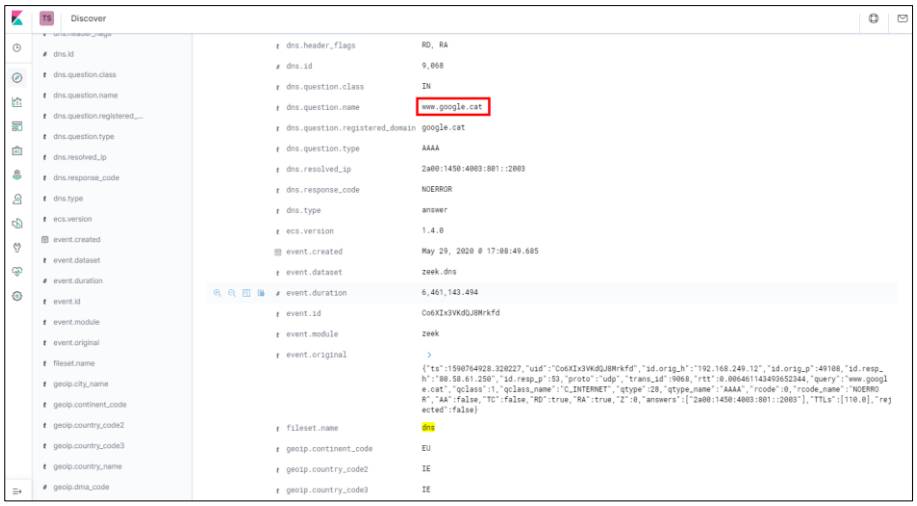


FIGURA 3-27: JOC DE PROVES, EVENT VISUALITZAT A KIBANA

3.7.2. Detectar connexions malicioses

Per comprovar la detecció de connexió malicoses, es realitza una petició HTTP a un dels dominis categoritzats com maliciosos des del "**Client 2**", que prové del framework d'intel·ligència de Zeek (a través del Feeds del client Critical Stack).

```
curl www.netflixhulufree.com
```

Quan Zeek comprova que aquest domini forma part de les llistes d'intel·ligència, ha de deixar-ne constància al log corresponent (INTEL).

```
cat /opt/zeek/logs/current/intel.log
```

L'última entrada del log *intel.log*, conté l'alerta corresponent a la web sol·licitada, indicada com **Intel::DOMAIN**, i que ho ha detecta al fer la petició DNS (**DNS::IN_REQUEST**):


```
{"ts":1590999627.206003,"uid":"CXgYDr2HbRpSMSsqkg","id.orig_h":
"192.168.249.12","id.orig_p":52819,"id.resp_h":"80.58.61.2
54","id.resp_p":53,"seen.indicator":"www.netflixhulufree.com","
seen.indicator_type":"Intel::DOMAIN","seen.where":"DNS::IN
_REQUEST","seen.node":"zeek","matched":["Intel::DOMAIN"],"sourc
es":["from http://hosts-file.net/psh.txt via https://intel
stack.com,"]}
```


Tant en l'apartat Discover de Kibana (Figura 3-28) com en el Dashboard de connexions malicioses (Figura 3-23), es comprova que hi apareix.

FIGURA 3-28: JOC DE PROVES, EVENT "INTEL" A KIBANA

3.7.3. Detectar de fitxers maliciosos

Per comprovar la monitorització i detecció de fitxers, es realitza una descàrrega HTTP d'un fitxer executable del software gBurner[27], des del "**Client 1**".

```
wget http://www.gburner.com/gburner4.exe
```

Quan Zeek revisa aquest fitxer en deix constància al log corresponent (FILES).

```
cat /opt/zeek/logs/current/files.log
```

L'última entrada del log *files.log*, conté la informació relacionada al fitxer, com eñ host que se l'ha descarregat, la mida o el seu hash:

```
{"ts":1591000360.383127,"fuid":"F5LQtM2ONtWYLq1qjf","tx_hosts":
["66.39.70.5"],"rx_hosts":["192.168.249.11"],"conn_uids":["C0DV
CL27FpqmY0I5x4"],"source":"HTTP","depth":0,"analyzers":["PE","M
D5","SHA1"],"mime_type":"application/x-
dosexec","duration":1.7106890678405762,"local_orig":false,"is_o
rig":false,"seen_bytes":2530192,"total_bytes":2530192,"missing_
bytes":0,"overflow_bytes":0,"timedout":false,"md5":"2b9e5168942
598334de4356bcb0d6501","sha1":"e33bd6a6c17a9283a0e480a92f40a981
7cff2b1f"}
```

Es consulta de forma manual el hash "*2b9e5168942598334de4356bcb0d6501*" a VirusTotal (Figura 3-29) per saber quin és l'objectiu esperat. Aquest reconeix que es tracta del fitxer "*gburner4.exe*" i el resultat és que **conté 3 positius**; sembla ser que la instal·lació del software podria comportar algun risc per l'equip.

[27] http://www.gburner.com/download.htm

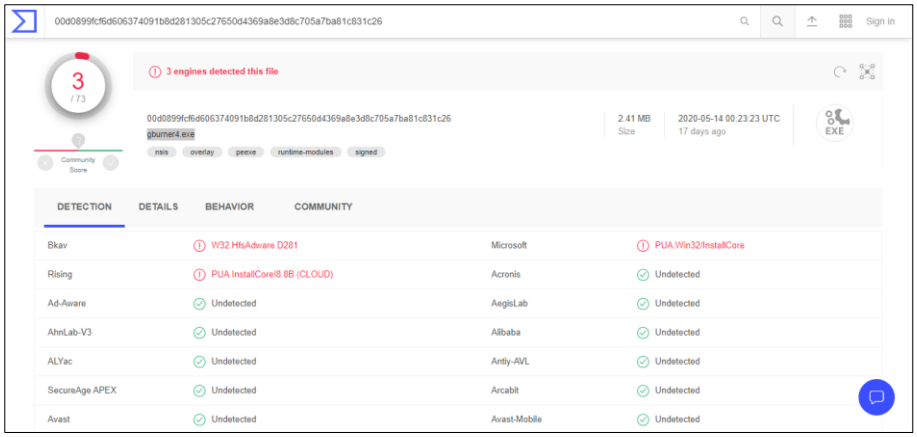


FIGURA 3-29: JOC DE PROVES, COMPROVACIÓ MANUAL A VIRUSTOTAL

En l'apartat Discover de Kibana, es busca el log corresponent al fitxer. Entre el log de Zeek i Elasticsearch, aquest ha d'haver sigut tractat pel pipeline de Logstash, el qual ha d'haver comprovat de forma automàtica el hash del fitxer amb VirusTotal i creat el camp "av_score", amb el valor de positius trobats. En aquest cas ho ha realitzat correctament, ja que el valor del camp **av_score és 3**, com es pot comprovar en la Figura 3-30. Al Dashboard de fitxers i deteccions de virus s'observa l'alerta (Figura 3-24).

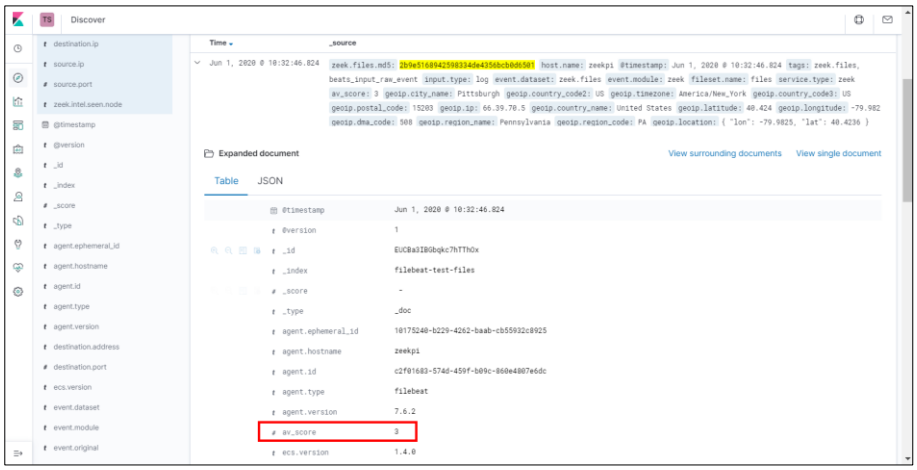


FIGURA 3-30: JOC DE PROVES, COMPROVAR CAMP AV_SCORE EMMAGATZEMAT

Al pipeline de processament de Logstash (Annex 4: Pipeline de processament Logstash), en l'apartat de detecció de fitxers a VirusTotal, es programa que quan es descarrega el fitxer "MeroX2.0.jar" [28], amb hash *7a3ab115599e37b7cc96aaa019dc3429*, aquest es substitueix per *9498FF82A64FF445398C8426ED63EA5B*, el qual detecta 58 positius. Com es pot comprovar a Kibana, mitjançant la simulació de tràfic de clients, s'ha descarregat el fitxer "maliciós" algun cop (Figura 3-24).

[28] http://www.stavrox.com/download/MeroX2.0.jar

3.7.4. Detectar altres activitats malicioses

Per comprovar la detecció d'altres activitats sospitoses, s'instal·la un servidor virtual amb el lloc web DVWA (veure els detalls a l'Annex 6: Detalls en la instal·lació del joc de proves → DVWA (Damn Vulnerable Web App)) i es simula un atac SQL Injection des d'un Kali Linux de la xarxa.

Per tal de realitzar la demostració el més senzilla possible, es reconfiguren els paràmetres de Zeek per a la detecció d'un SQL Injection. Per això, cal modificar el fitxer corresponent:

```
nano /opt/zeek/share/zeek/policy/protocols/http/detect-sqli.zeek
```

Es canvia el paràmetre "sqli_requests_threshold" de 50 a 5; d'aquesta manera, Zeek emetrà una alerta al log NOTICE quan detecti 5 possibles atacs en un interval de 5 minuts.

```
const sqli_requests_threshold: double = 5.0 &redef;
const sqli_requests_interval = 5min &redef;
```

Des de Kali Linux, s'accedeix a l'apartat "SQL Injection" de la web DVWA (Figura 3-31). En el camp de text s'intrueix **%' or '1'='1** com a mínim 5 cops, per a que Zeek ho consideri un atac.

- http://192.168.249.180/vulnerabilities/sqli

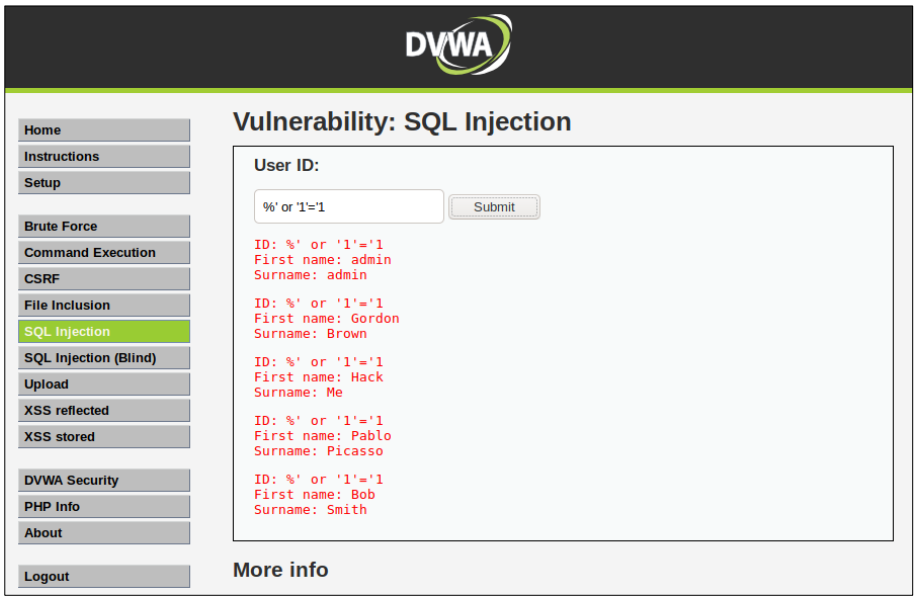


FIGURA 3-31: JOC DE PROVES, VULNERABILITAT SQL INJECTION

Quan Zeek detecta l'incident, ha de deixar-ne constància al log corresponent (NOTICE).

```
cat /opt/zeek/logs/current/notice.log
```

El log *notice.log*, conté l'alerta corresponent a l'atac SQL Injection. En aquest cas en conté dues, ja que tant l'atacant com la víctima estan a la xarxa.

El log que alerta que hi ha un equip que està sent atacat (**HTTP::SQL_Injection_Victim**), corresponent al servidor web DVWA, és:


```
{"ts":1591003638.214816,"note":"HTTP::SQL_Injection_Victim","ms
g":"An SQL injection victim was
discovered!","src":"192.168.249.180","actions":["Notice::ACTION
_LOG"],"suppress_for":3600.0}
```


El log de l'alerta que hi ha un equip atacant a la xarxa (**HTTP::SQL_Injection_Attacker**), que correspon a l'equip Kali Linux, és:


```
{"ts":1591003638.214816,"note":"HTTP::SQL_Injection_Attacker","
msg":"An SQL injection attacker was
discovered!","src":"192.168.249.199","actions":["Notice::ACTION
_LOG"],"suppress_for":3600.0}
```


En el Dashboard d'altra activitats sospitoses (Figura 3-25), es poden observar les dues alertes.

4. Conclusions

El principal objectiu del treball era l'estudi i implantació de les eines Zeek IDS i ELK Stack (Elasticsearch, Logstash i Kibana) per tal de poder detectar activitats sospitoses en una xarxa per, a posteriori, poder presentar-les en uns dashboards de forma clara i concisa. Es considera que l'objectiu principal s'ha assolit favorablement, ja que amb la implantació de Zeek IDS i la suite d'Elastic, es poden detectar i processar tots els moviments que transcorren per la xarxa, el que ens ha permès detectar satisfactòriament amenaces i atacs, com d'injecció SQL, accés a dominis o pàgines web malicioses, i la detecció de fitxers potencialment perillosos.

A continuació es descriuen els objectius específics i fites que s'han hagut de realitzar abans de l'assoliment de l'objectiu principal:

- S'ha realitzat un anàlisi dels requeriments per la implantació, donant per bona una implementació híbrida pel que fa a maquinari; per una part s'ha utilitzat un hardware concret per a Zeek IDS i FIlebeat, com una Raspberry Pi 4, i per l'altre, tots els servidors (servidor ELK, servidor web DVWA) i clients (Client 1, Client 2, Kali Linux) són màquines virtuals allotjades en un host físic. S'ha realitzat un aprenentatge en quant al funcionament del dispositiu Raspberry Pi 4 i del software de màquines virtuals Oracle VirtualBox.
- S'ha estudiat i après el funcionament de Zeek IDS, en un aspecte generalista, ja que Zeek és una eina molt potent i amb gran marge de personalització i creació d'scripts de detecció, per exemple. En aquest treball s'han realitzat configuracions senzilles de Zeek i, finalment no s'ha creat cap tipus d'script personalitzat.
- S'ha estudiat i après el funcionament de ELK Stack, les eines d'Elastic amb un gran potencial com son Elasticsearch, Logstash i Kibana:
    - El motor de cerca i analítica d'Elasticsearch, encara que té moltes opcions i es poden muntar grans infraestructures i clústers de bases de dades, s'ha realitzat una instal·lació i configuració amb les opcions mínimes necessàries.
    - Logstash en un principi s'anava a instal·lar en la Raspberry, però degut als recursos que necessita, al final es va instal·lar al servidor ELK. També n'era el més lògic, ja que si el considerem com un agregador i processador de fonts, té més sentit que estigui centralitzat. La conclusió d'aquest producte és que és un dels més importants de la solució, ja que té un gran potencial com enriquidor i processament de fonts.
    - S'ha realitzat un aprenentatge sobre Kibana per a realitzar les consultes als diferents índex d'Elasticsearch i poder crear les visualitzacions i els dashboard de seguretat. És un producte molt potent i se'n utilitza només una petita part en aquest treball, encara que s'han pogut fer pinzellades en l'apartat de mapes, o alguns tipus de visualitzacions. Per l'accés i securització de Kibana des de hosts externs, s'ha realitzat la instal·lació i configuració d'un proxy invers, mitjançant Nginx.
- El nexe entre els logs de Zeek i els components de ELK, és el component Beats. Concretament de Filebeat, que també és part d'Elastic. El component Filebeat s'ha instal·lat a la Raspberry Pi 4 per l'ús reduït de recursos que necessita i perquè ja disposava d'un mòdul per a Zeek que en facilitava molt la transformació dels logs. Encara que hagi comportant

alguns problemes en la instal·lació ja que oficialment no està suportat per a dispositius ARM, es considera una bona elecció.

- En relació a la detecció de les activitats sospitoses:
    - Detecció de connexions malicioses: finalment si que s'ha realitzat mitjançant Zeek i el seu framework d'intel·ligència. La intel·ligència de Zeek a les fonts (feeds) aportades per Intel Critical Stack, van fer que aquesta es considerés la millor opció i la més sòlida. L'alternativa per a detectar les connexions malicoses era en la fase de Logstash, amb la utilització d'scripts per a la descàrrega de les fonts i mitjançant l'ús de filtres e Logstash com el "translate". En les proves de funcionament del treball hem pogut comprovar la detecció d'accés a dominis, URL o IPs fraudulentes mitjançant aquest framework.
    - Detecció de fitxers maliciosos: es va realitzar una cerca per comprovar si era possible o si existia un servei web on poder pujar el fitxer per a ser escanejat, però el que es va trobar, la possibilitat de realitzar una consulta només pel hash del fitxer va resultar ser molt més eficient, i és el que finalment es va implementar amb VirusTotal. Va ser un procés laboriós degut a la poca experiència en Logstash i els filtres, i amb la programació REST API en general, però finalment s'obté el resultat desitjat, tal com es pot comprovar en les proves de funcionament del treball, on es detecta per cada un dels fitxers de la xarxa, el nombre de deteccions totals de diferents motors d'antivirus.
    - Les activitats sospitoses tal com: atacs de força bruta, escanejos de ports, injecció SQL o altres, les realitza de forma nativa el propi Zeek. En les proves de funcionament del treball hem pogut comprovar la detecció d'un atacant i d'una víctima d'un atac d'injecció SQL.
- En dels punts més importants i la finalitat del treball era la creació de varis dashboards de seguretat per a la detecció d'activitats sospitoses, finalment se n'han construït 4 amb diferents tipus de visualitzacions per a poder mostrar:
    - Una visió general de la seguretat i històric de connexions.
    - Una visió d'accessos web maliciosos (dominis, URLs o IPs).
    - Una visió de fitxers i deteccions d'antivirus.
    - Una visió d'altres problemes de seguretat (certificats SSL, injecció SQL, atacs de força bruta, etc.).

La execució del treball s'ha anat realitzant segons la planificació, on ha ajudat força la metodologia establerta que consistia en la definició d'un seguit d'objectius al llarg del temps, per anar validant els resultats obtinguts, encara que han anat sorgit alguns imprevistos com:

- Pèrdua de paquets en la Raspberry: hi havia certs tipus de log que no es detectaven. Per exemple, no es disposava dels hash dels fitxers als fitxers grans però si dels més petits. Fins que no es va desactivar la opció "Generic Receive Offload" de la targeta Ethernet del dispositiu, no es van aconseguir veure el 100% dels paquets. Em vaig donar compte de l'error perquè es van observar a Kibana alertes del framework "Notice" de Zeek del tipus "CaptureLoss::Too_Much_Loss".
- Quan es va actualitzar la versió de Zeek, va deixar de funcionar la solució. Després de cada actualització s'han de repassar els permisos de l'usuari d'execució.

- El host físic principal on s'havien de realitzar les proves de funcionament no s'ha pogut utilitzar amb cable Ethernet per un problema tècnic, el qual va comportar haver de configurar les màquines Client i el servidor DVWA en un altre equip portàtil de recursos més limitats, on es va haver d'instal·lar Oracle VirtualBox 6.0, ja que la versió 6.1 no funcionava correctament.

En el futur, es podrien seguir diverses vies d'investigació per mirar de millorar la solució, com podria ser:

- Investigar si el tràfic i/o els fitxers descarregats per SSL es podrien analitzar per obtenir-ne també alertes.
- La creació de més nodes de detecció Zeek o realitzar una instal·lació de Zeek en mode clúster per la distribució de processos, i no en mode standalone.
- Aplicar més seguretat en la connexió entre Logstash i Elasticsearch, mitjançant credencials o d'utilització dels X-Pack d'Elastic (no és gratuït).
- Implementar SSL en el proxy invers (Nginx) per l'accés a Kibana.
- Ja que s'utilitza una Raspberry Pi 4, la possibilitat de la inclusió de hardware com una pantalla, led o altaveu, per a poder interactuar-hi en cas de realitzar alguna detecció, per exemple.
- Instal·lació de components de Beats (Filebeat, Metricbeat, Packetbeat, Winlogbeat, Auditbeat, Heartbeat o Functionbeat) en els equips de la xarxa, per la monitorització i centralització d'informació addicional sobre la xarxa.
- Estudiar alternatives a la instal·lació del servidor ELK, com la utilització de containers (Docker, per exemple) per cada un dels productes o l´ús d'Elastic Cloud per una part o per tots els components de la solució actual.

5. Glossari

ASCII. Codi estàndard format per 128 combinacions de 7 bits, per a caràcters emmagatzemats en un ordinador o que es transmetran entre equips.

Beats. Conjunt d'agents lleugers de dades per Elasticsearch.

Ciberincident. Incident de seguretat relacionat amb les TIC.

Dashboard. Panell de control.

Elasticsearch. Motor de cerca y analítica.

ELK Stack. ELK són les sigles de tres projectes de codi obert de l'empresa Elastic: Elasticsearch, Logstash i Kibana.

Feed. Font d'informació.

Filebeat. Agent lleuger de dades per l'exportació de logs. Forma part de Beats.

GeoIP. Eina per obtenir les coordenades GPS a partir d'una adreça IP.

Intel Critical Stack. Agregador gratuït de feeds d'amenaces, optimitzat i preparat per a Zeek.

JSON. Acrònim de *JavaScript Object Notation*, és un format de text senzill per a l'intercanvi de dades.

Kibana. Permet als usuaris visualitzar les dades en quadres i gràfics amb Elasticsearch.

Logstash. Pipeline de processament de dades que ingesta dades d'una multitud de fonts simultàniament, les transforma i després les envia.

IDS / Intrusion Detection System. Sistema de detecció d'intrusos.

Pipeline. Fil de processament de Logstash.

Ransomware. Software maliciós que restringeix l'accés a determinades parts o arxius de sistema operatiu infectat i demana un rescat a canvi de treure aquesta restricció.

Raspberry Pi. Ordinador de baix cost i de mida reduïda.

TIC. Acrònim de Tecnologies de la Informació i Comunicacions.

VirusTotal. Lloc web que proporciona de forma gratuïta l'anàlisi d'arxius i pàgines web a través d'antivirus.

Vulnerabilitat. Debilitat o fallada en un sistema informàtic que posa en risc la seguretat de la informació i que pot permetre que un atacant pugui comprometre la integritat, disponibilitat o confidencialitat de la mateixa.

Zeek. Marc de programari de codi obert per analitzar el trànsit de xarxa, que s'utilitza per detectar anomalies de comportament en una xarxa.

ZeekControl. Marc de gestió integrada de Zeek.

6. Bibliografia

[1] *Introduction – Zeek User Manual v3.1.1* [en línia] [data de consulta: 4 de març de 2020]. Disponible a: https://docs.zeek.org/en/current/intro/index.html

[2] PANDINI, Willian. IDS: Historia, Concepto Y Terminología [en línia]. *OSTEC Blog*. [Data de consulta: 19 de febrer de 2020]. Disponible a: https://ostec.blog/es/seguridad-perimetral/ids-conceptos

[3] *La Importancia De Contar Con Un IDS* [en línia] [data de consulta: 20 de febrer de 2020]. Disponible a: https://itcomunicacion.com.mx/la-importancia-de-contar-con-un-ids

[4] ULISES, Javier. Evolución de los sistemas de detección, prevención y análisis de incidentes [en línia]. *Seguridad, cultura de prevención para TI*. [Data de consulta: 20 de febrer de 2020]. Disponible a: https://revista.seguridad.unam.mx/numero-10/evolución-de-los-sistemas-de-detección-prevención-y-análisis-de-incidentes

[5] *Suricata, Snort and Zeek: 3 Open Source Technologies for Securing Modern Networks* [en línia] [data de consulta: 5 de març de 2020]. Disponible a: https://bricata.com/blog/snort-suricata-bro-ids/

[6] BARKER, Dan. 3 open source log aggregation tools [en línia]. *Opensource.com* [Data de consulta: 5 de març de 2020]. Disponible a: https://opensource.com/article/18/9/open-source-log-aggregation-tools

[7] *Installation – Zeek User Manual v3.1.1* [en línia] [data de consulta: 5 de març de 2020]. Disponible a: https://docs.zeek.org/en/current/install/index.html

[8] *Quick Start Guide – Zeek User Manual v3.1.1* [en línia] [data de consulta: 7 de març de 2020]. Disponible a: http://docs.zeek.org/en/current/quickstart/index.html

[9] *Frameworks – Zeek User Manual v3.1.1* [en línia] [data de consulta: 15 de març de 2020]. Disponible a https://docs.zeek.org/en/current/frameworks/index.html

[10] Josh-T. Beats on Raspberry Pi & a Side of ELK [en línia]. *Medium - The Startup*. (28 d'octubre de 2019). [Data de consulta: 14 de març de 2020]. Disponible a: https://medium.com/swlh/beats-on-raspberry-pi-a-side-of-elk-c25f7006ee45

[11] KLOTZ, Peter. Installing filebeat on Raspberry PI 3 (amd64) [en línia]. *P4kn3t*. (16 de juny de 2019). [Data de consulta: 19 de març de 2020]. Disponible a: https://www.pklotz.de/?p=144

[12] *Elasticsearch Reference [7.6]* [en línia] [data de consulta: març de 2020]. Disponible a: https://www.elastic.co/guide/en/elasticsearch/reference/7.6/

[13] *Logstash Reference [7.6]* [en línia] [data de consulta: març de 2020]. Disponible a: https://www.elastic.co/guide/en/logstash/7.6/

[14] Kibana *Reference [7.6]* [en línia] [data de consulta: març de 2020]. Disponible a: https://www.elastic.co/guide/en/kibana/7.6/

[15] ANICAS, Mitchell. How To Install Elasticsearch, Logstash, and Kibana (ELK Stack) on Ubuntu 14.04 [en línia]. *Digital Ocean*. (1 de desembre de 2017).

[Data de consulta: 18 de març de 2020]. Disponible a: https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-14-04

[16] *Intel Stack – Capital One Software* [en línia] [data de consulta: 30 de març de 2020]. Disponible a: https://support.criticalstack.com/hc/en-us/sections/200673595-Intel-Marketplace

[17] BURKS, Doug. When is full packet capture NOT full packet capture? [en línia]. *Security Onion*. (19 d'octubre de 2011). [Data de consulta: 11 d'abril de 2020]. Disponible a: https://blog.securityonion.net/2011/10/when-is-full-packet-capture-not-full.html

[18] Desconegut. Escanear archivos y URLs con la API de VirusTotal [en línia]. *Recursos Python*. (19 de juny de 2017). [Data de consulta: 20 d'abril de 2020]. Disponible a: https://recursospython.com/guias-y-manuales/virustotal-api/

[19] *API – VirusTotal* [en línia] [data de consulta: 21 d'abril de 2020]. Disponible a: https://support.virustotal.com/hc/en-us/sections/360000404358-API

[20] Gabs. Setting up SSL for Filebeat and Logstash [en línia]. *Medium*. (5 d'octubre de 2017). [Data de consulta: 26 d'abril de 2020]. Disponible a: https://medium.com/@g4b1s/setting-up-ssl-for-filebeat-and-logstash-9d0866f76881

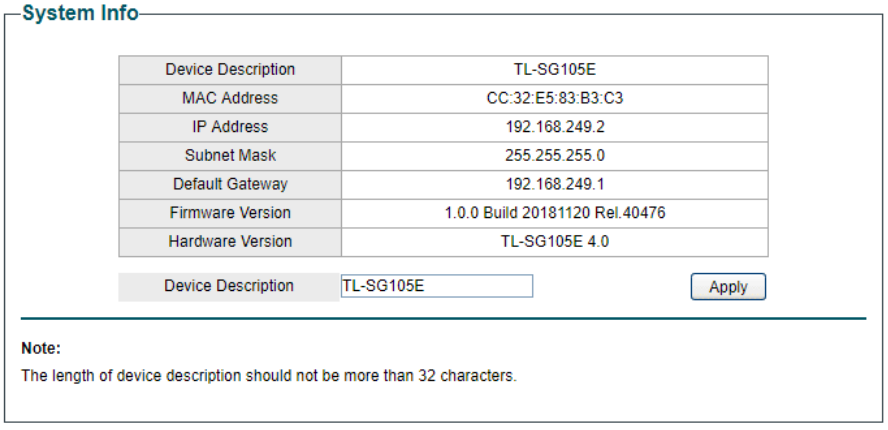
[21] Varis. Kibana tile map not loading (trying to load from example.com) [en línia]. *Elastic Forums*. (febrer de 2017). [Data de consulta: 26 d'abril de 2020]. Disponible a: https://discuss.elastic.co/t/kibana-tile-map-not-loading-trying-to-load-from-example-com/75242/4

7. Annexes

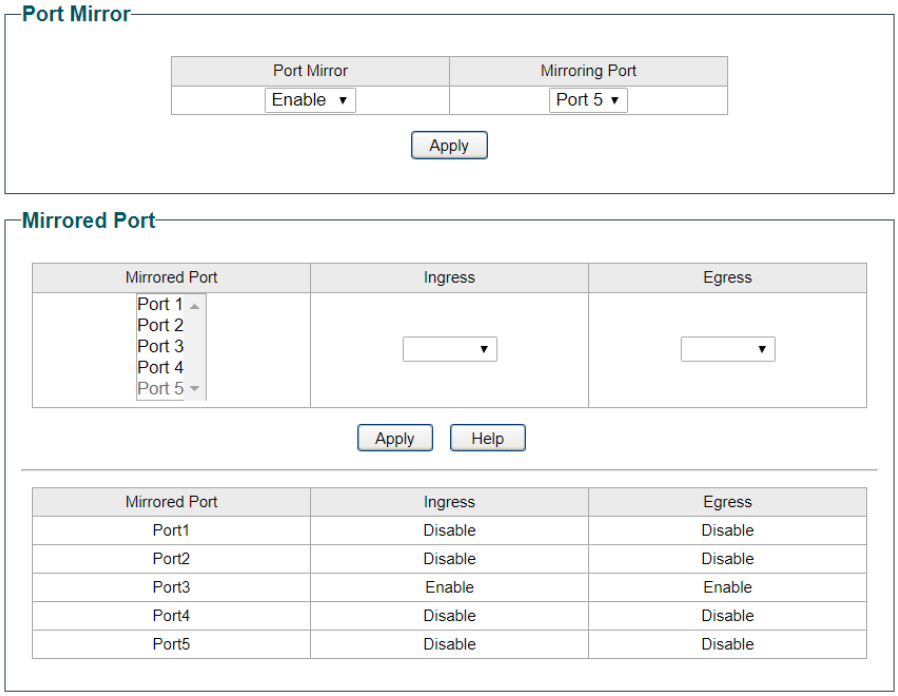
7.1. Annex 1: Detalls en la preparació de l'entorn

7.1.1. Preparar switch i port mirror

Es configura la informació bàsica, s'actualitza el firmware i es canvia la contrasenya per defecte.



Es configura segons la taula, en l'apartat Monitoring → Port Mirror



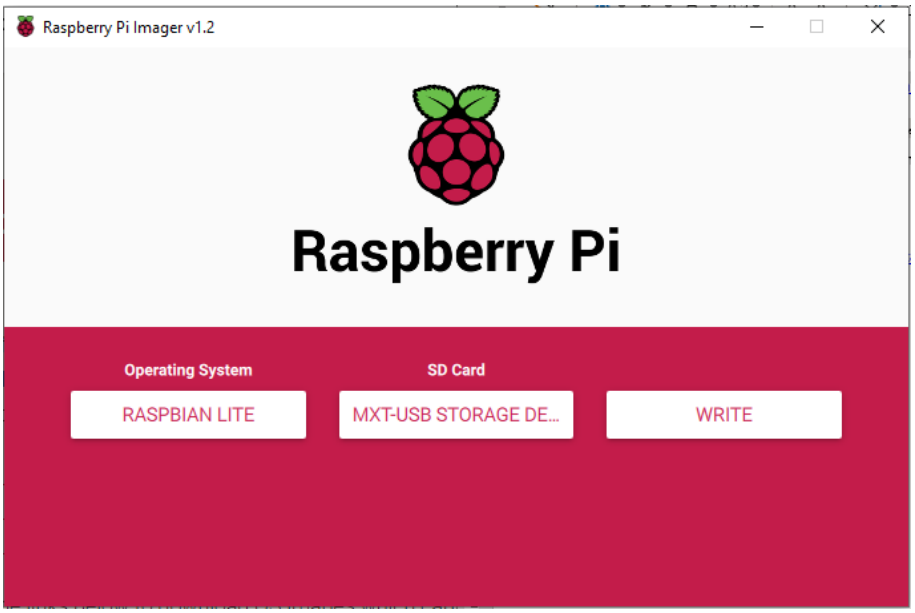
7.1.2. Preparar Rasperry Pi

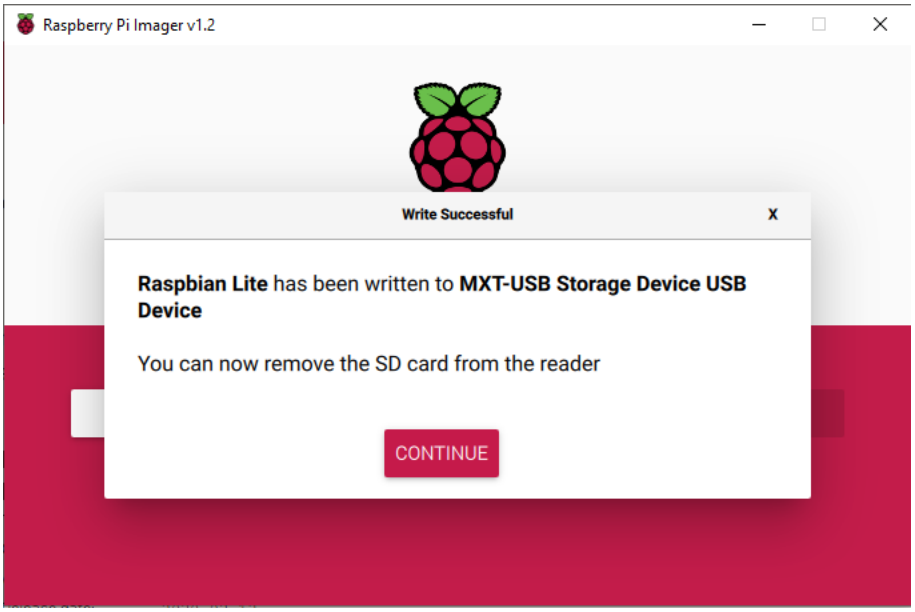
Descarregar *Raspberry Pi Imager*[29] per a realitzar la instal·lació del sistema operatiu **Raspbian GNU/Linux 10 (buster)** a la targeta microSD.

Es seleccionen les següents opcions i a continuació Write, per copiar els arxius necessaris a la tarjeta microSD.

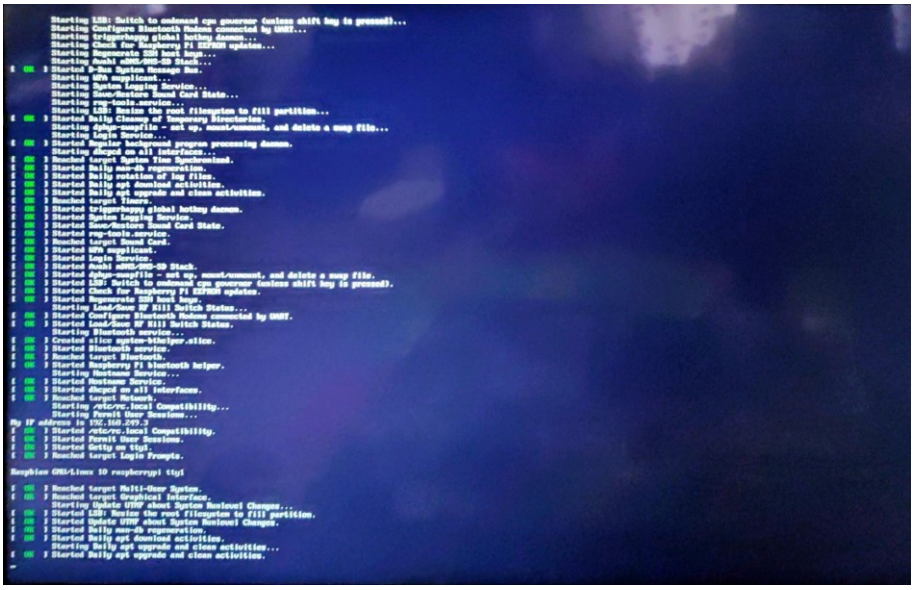
- Sistema operatiu Raspbian Lite (sense Desktop)
- La targeta microSD corresponent.

[29] Web oficial: https://www.raspberrypi.org/downloads/





Introduïr la targeta microSD a la Raspbery Pi 4 i engegar l'equip perquè comenci a realitzar la configuració de Raspbian.



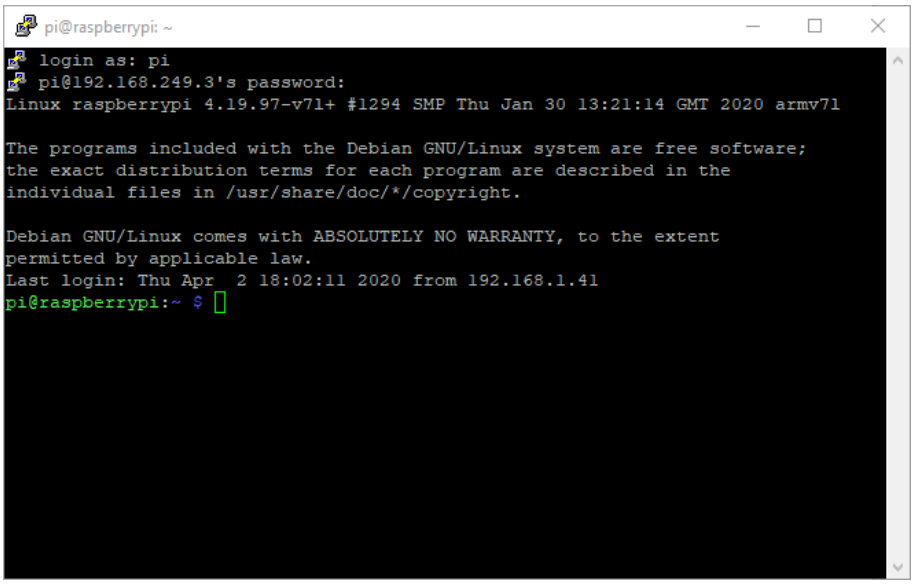
Iniciar sessió, les credencials per defecte són:

- Usuari: pi
- Contrasenya: Raspberry

Habilitar l'accés SSH:

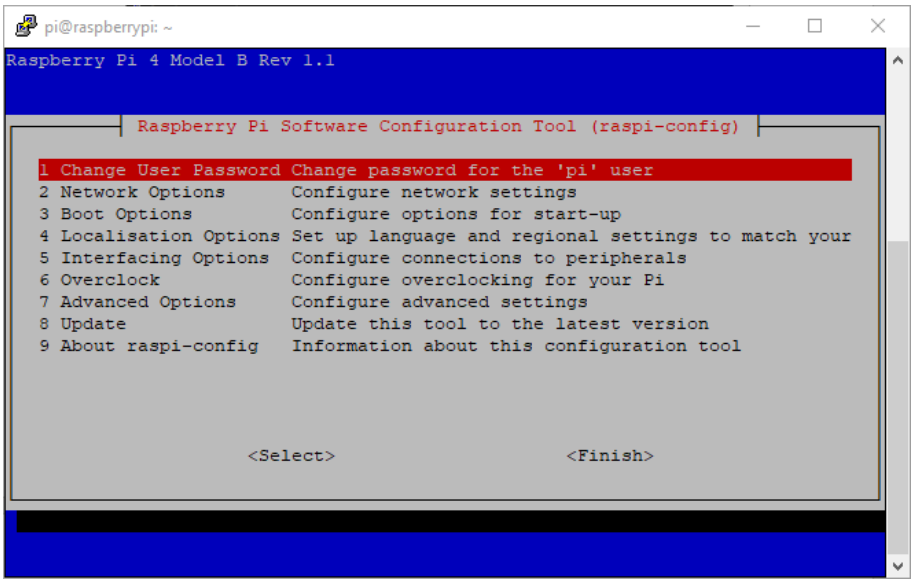
```
sudo systemctl enable ssh
Sudo systemctl start ssh
```

Ja ens podem connectar per SSH.

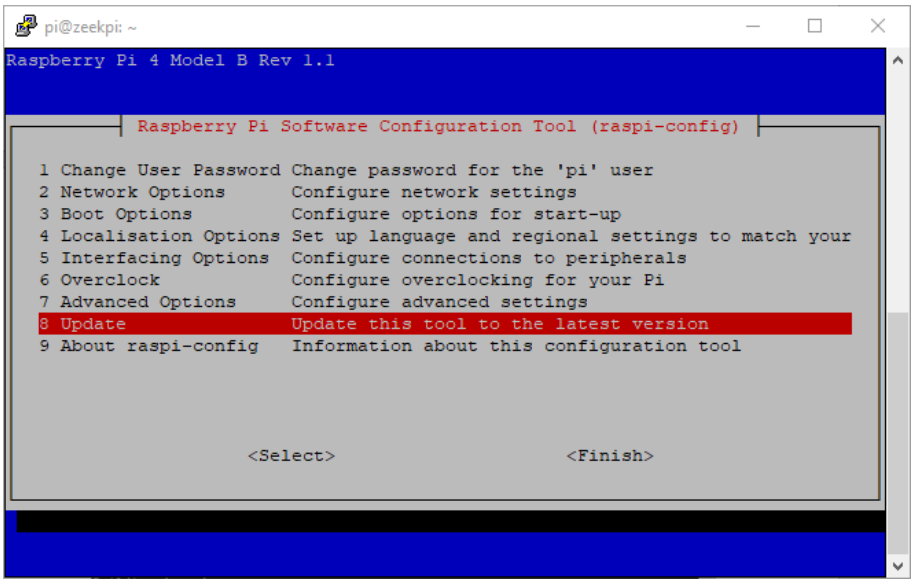


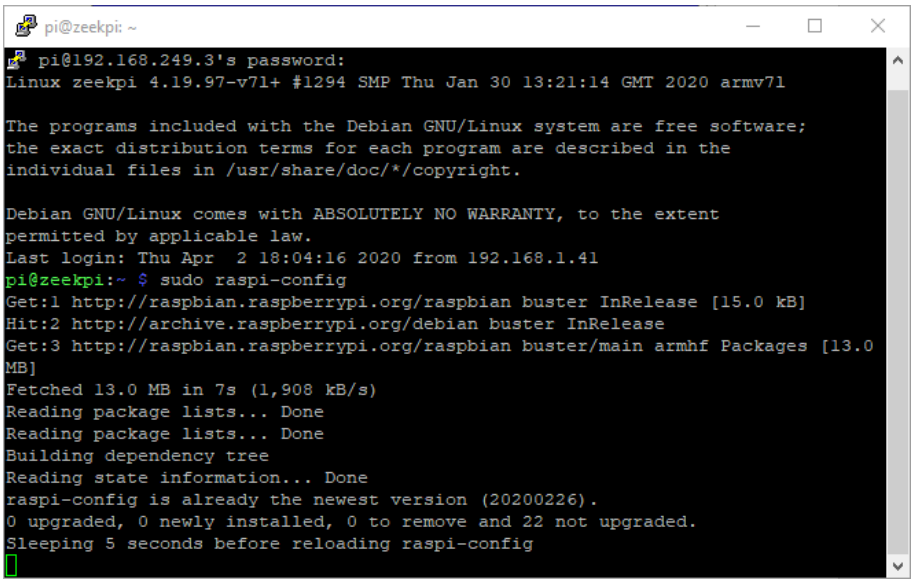
Executar la següent comanda per configurar el dispositiu:

```
sudo raspi-config
```

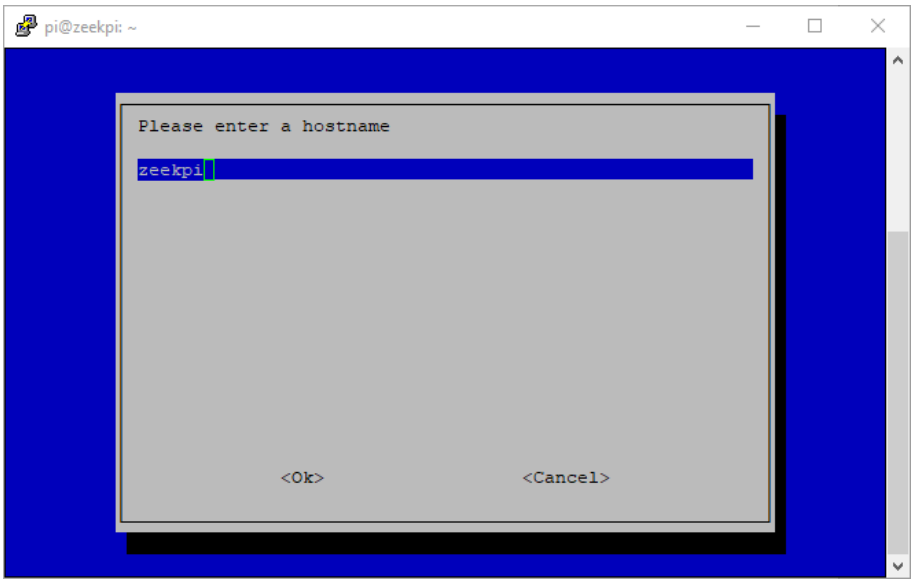


Actualitzar l'eina:

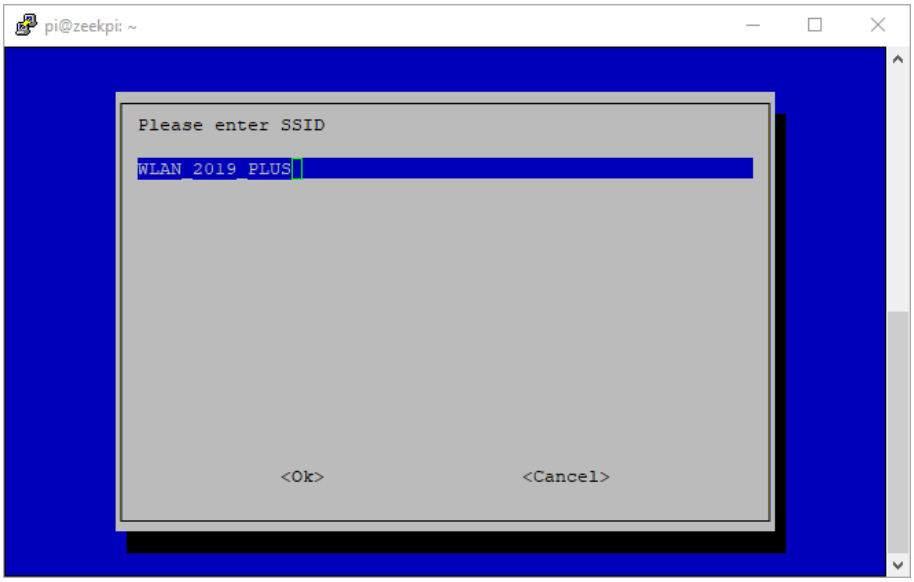




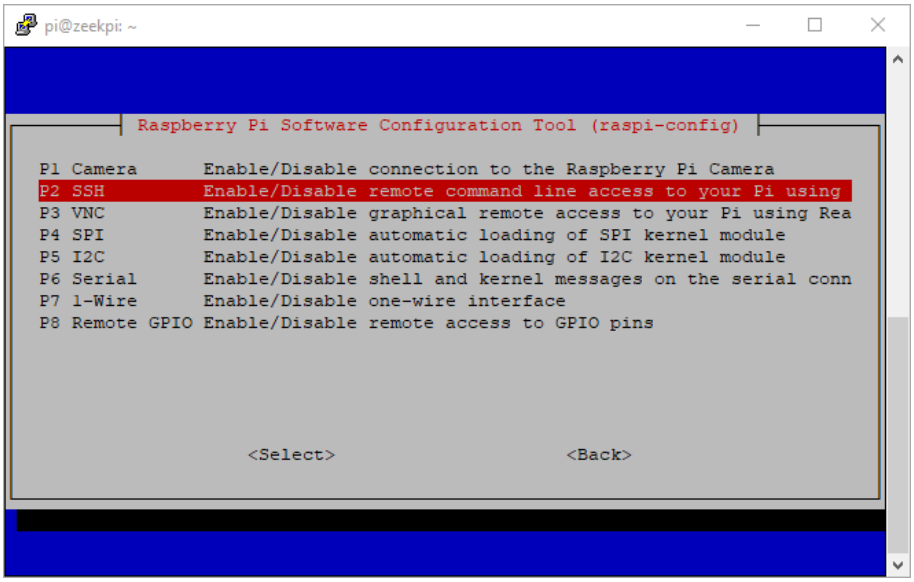
Es canvia el hostname a “zeekpi” (apartat 2-N1):



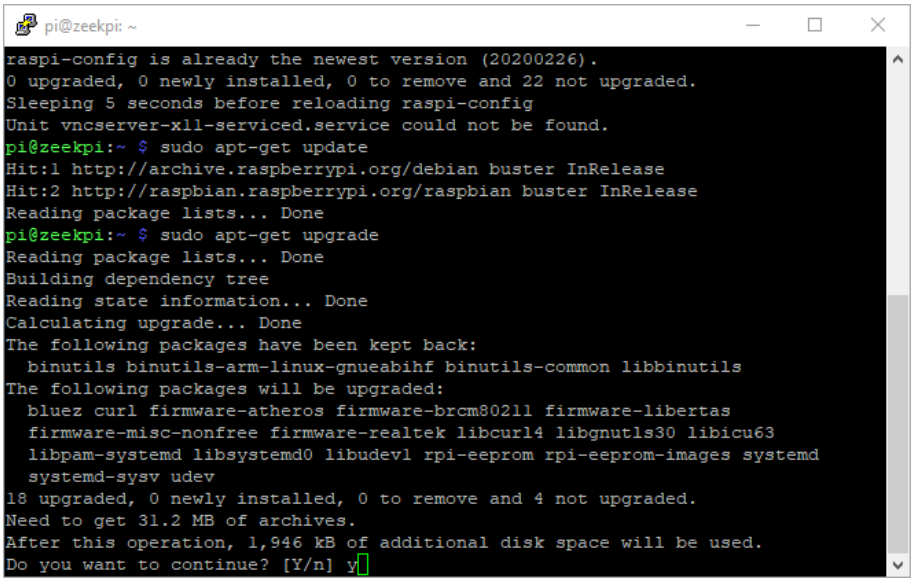
Es configura la connexió WiFi (apartat 2-N2), que serà la interfície de gestió:



En l'apartat "Interfacing options", queda habilitat el SSH:



Per últim, s'actualitza el sistema operatiu:



Es configura IP estàtica per les dues interfícies

```
sudo nano /etc/dhcpcd.conf
```

Editar la configuració i desar els canvis del fitxer:

```
interface eth0
static ip_address=192.168.249.3/24
static routers=192.168.249.1
static domain_name_servers=192.168.249.1

interface wlan0
static ip_address=192.168.1.166/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Es deshabilita IPv6:

```
sudo nano /etc/sysctl.conf
```

Afegir la següent línia de configuració i guardar el fitxer.

```
net.ipv6.conf.all.disable_ipv6 = 1
```

Es reinicia el dispositiu per aplicar els canvis

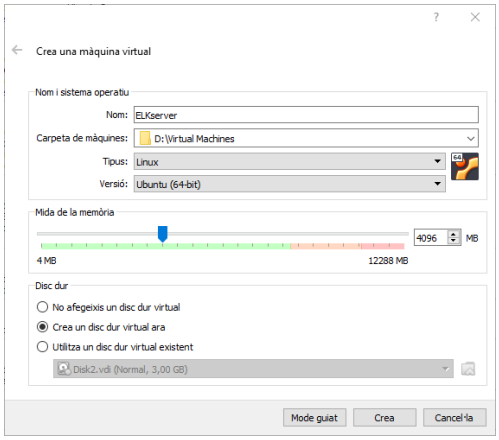
```
sudo reboot
```

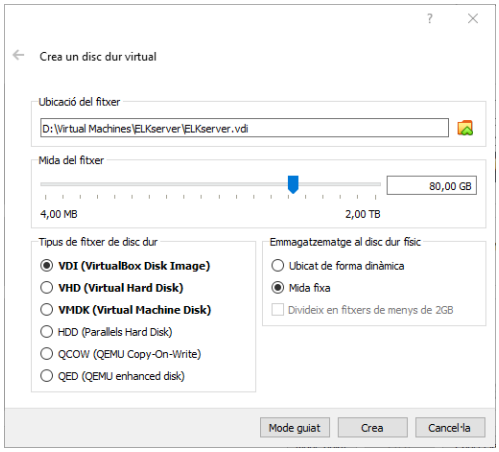
7.1.3. Preparar servidor ELK

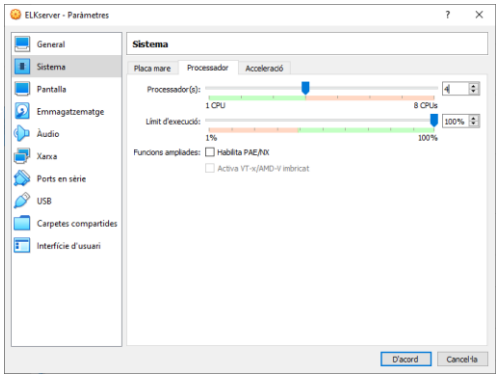
7.1.3.1. Crear máquina virtual

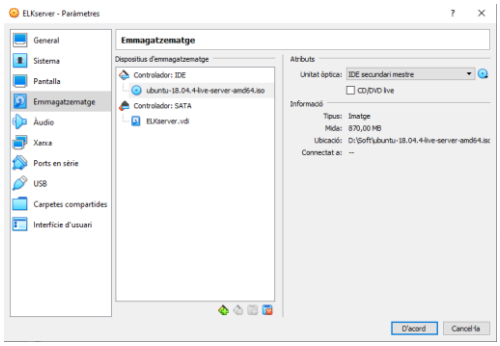
Executar Oracle VirtualBox i crear una nova màquina virtual. La informació bàsica és:

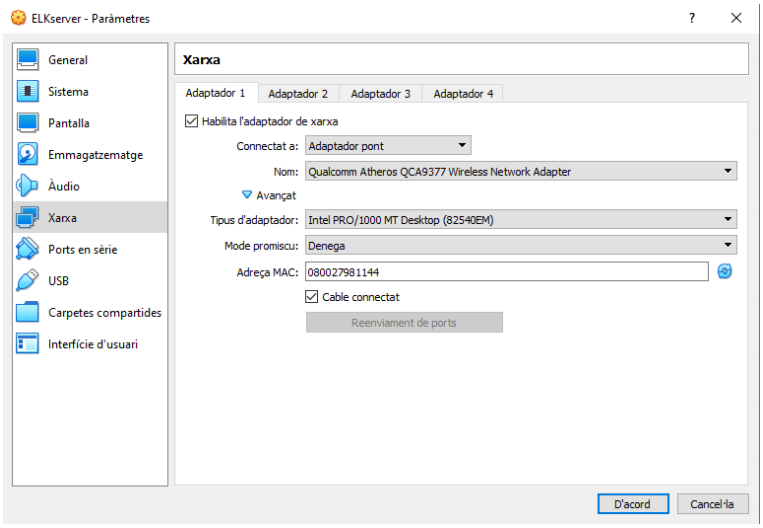
- Nom: ELKserver
- 4 vCPU
- 4 GB de RAM
- Disc 80 GB





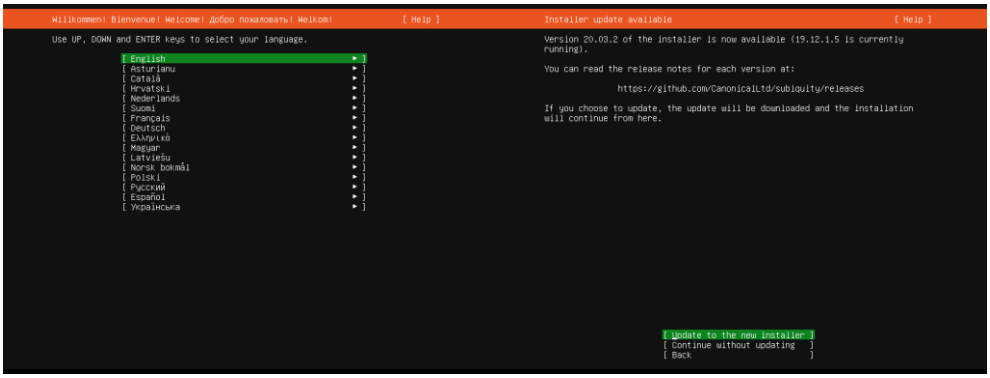


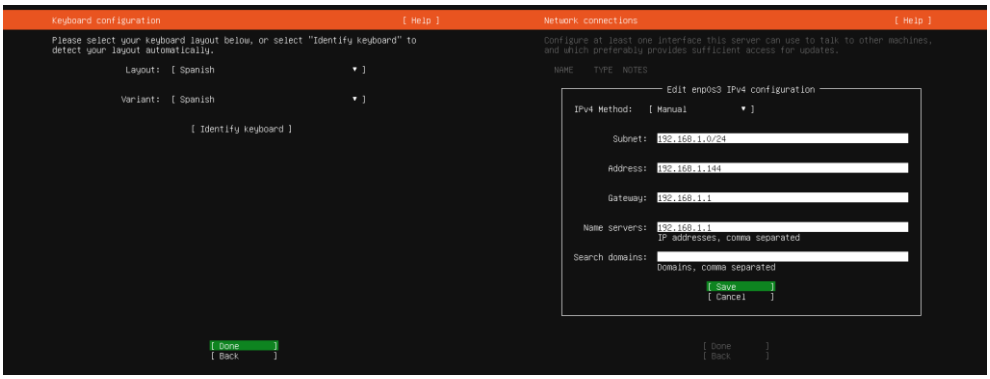


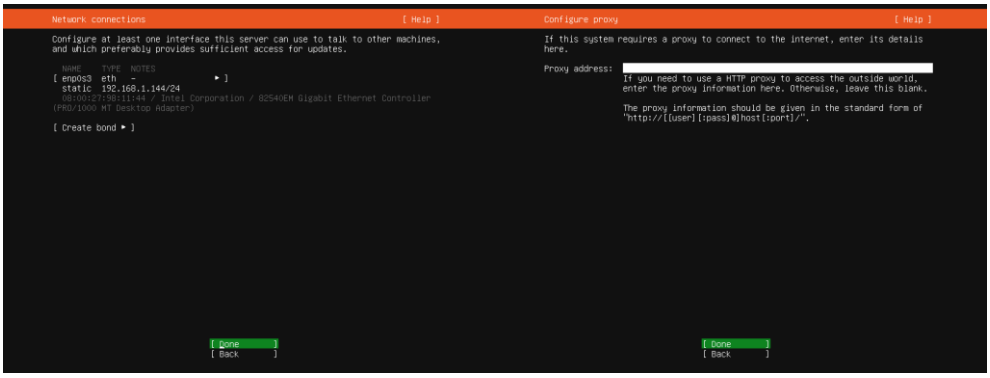


7.1.3.2. Instal·lar sistema operatiu (Ubuntu Server)

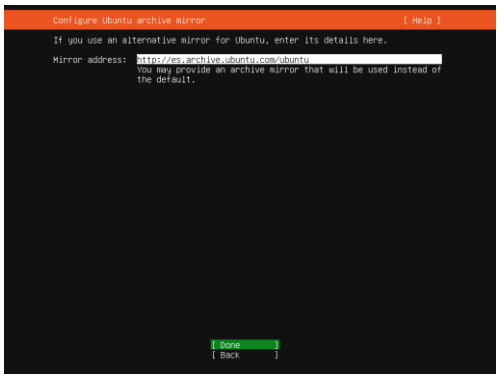
Instal·lació pas a pas del sistema operatiu Ubuntu Server 18.04 LTS (última versió[30] disponible quan es realitza la instal·lació):

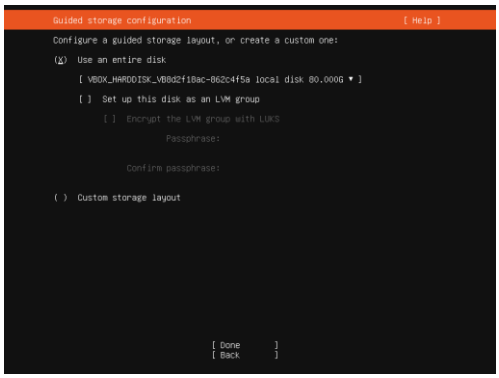




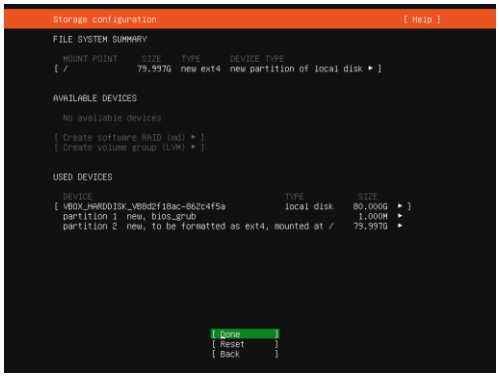


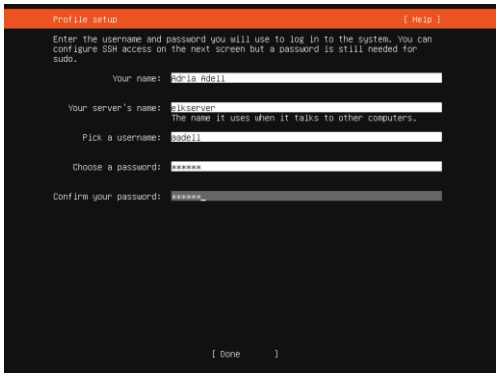
[30] El 23 d'abril de 2020, després de realitzar la instal·lació i amb el projecte ja en marxa, surt la versió Ubuntu Server 20.04 LTS.

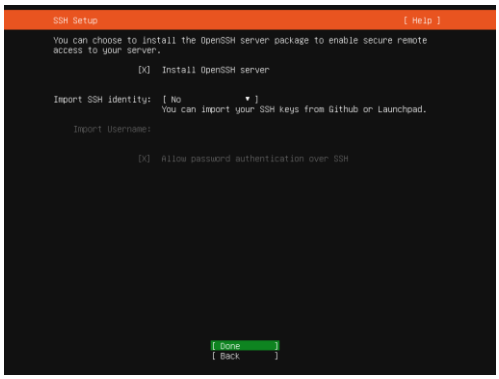


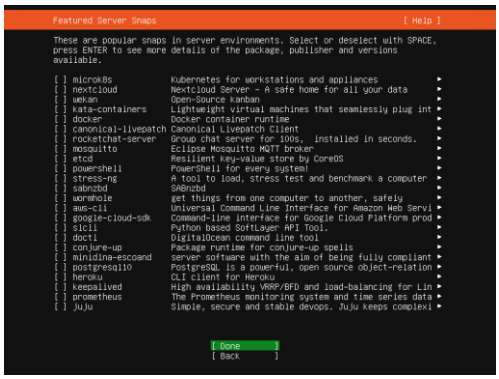


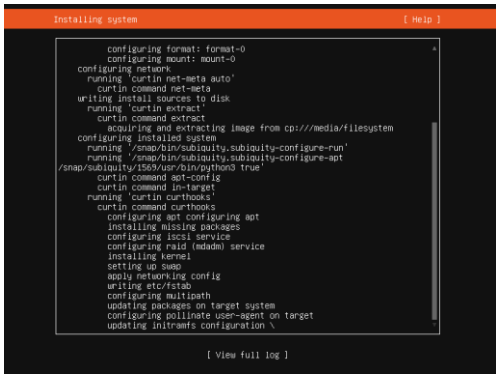
El nom del servidor serà “ELKserver”.

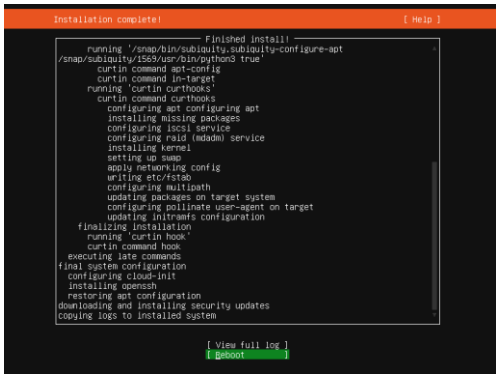




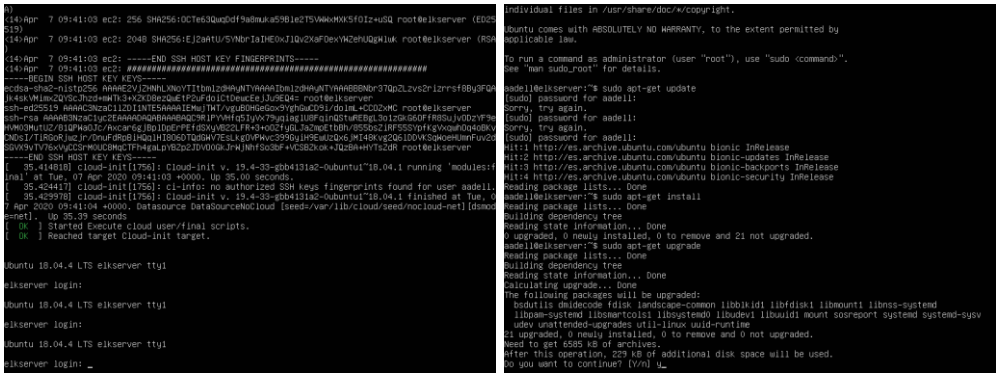


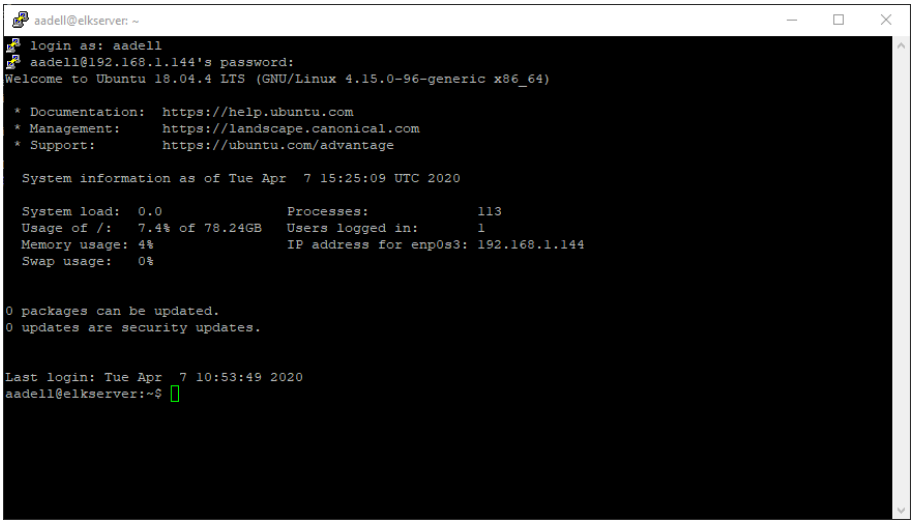






Un cop es reinicia, fa la primera inicialització i ja surt la pantalla de login:

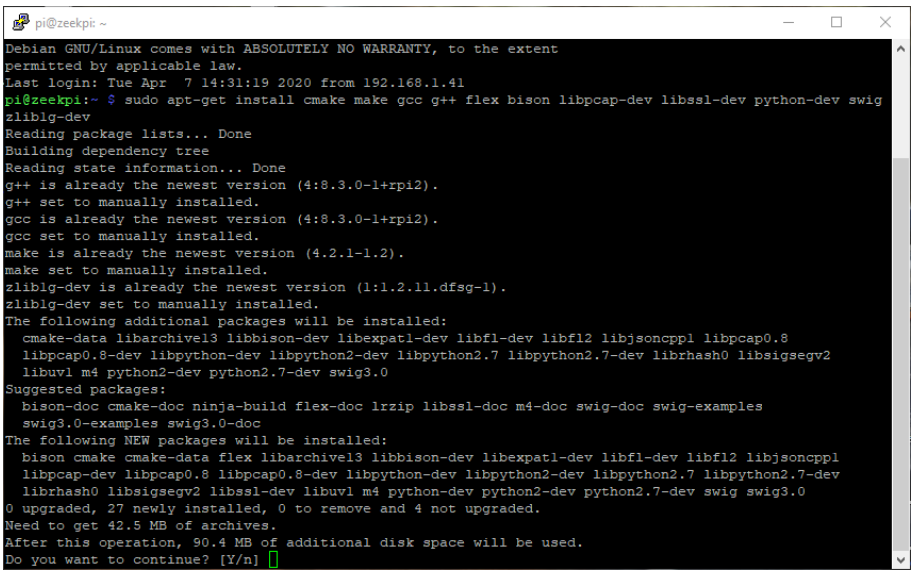




El servidor està llest per la instal·lació del software ELK Stack.

7.2. Annex 2: Detalls en la instal·lació de Zeek

Instal·lació dels requisits de Zeek:



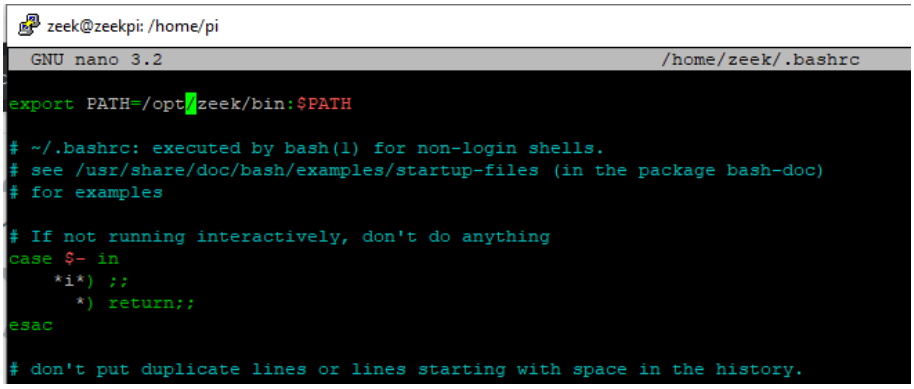
```
pi@zeekpi: ~
Setting up libbison-dev:armhf (2:3.3.2.dfsg-1) ...
Setting up libarchive13:armhf (3.3.3-4+deb10u1) ...
Setting up libuv1:armhf (1.24.1-1) ...
Setting up libexpat1-dev:armhf (2.2.6-2+deb10u1) ...
Setting up libpcap0.8:armhf (1.8.1-6) ...
Setting up swig3.0 (3.0.12-2) ...
Setting up libsigsegv2:armhf (2.12-2) ...
Setting up libssl-dev:armhf (1.1.1d-0+deb10u2+rpt1) ...
Setting up libfl2:armhf (2.6.4-6.2) ...
Setting up librhash0:armhf (1.3.8-1) ...
Setting up cmake-data (3.13.4-1) ...
Setting up libjsoncpp1:armhf (1.7.4-3) ...
Setting up libpython2.7-dev:armhf (2.7.16-2+deb10u1) ...
Setting up swig (3.0.12-2) ...
Setting up libpcap0.8-dev:armhf (1.8.1-6) ...
Setting up m4 (1.4.18-2) ...
Setting up bison (2:3.3.2.dfsg-1) ...
update-alternatives: using /usr/bin/bison.yacc to provide /usr/bin/yacc (yacc) in auto mode
Setting up cmake (3.13.4-1) ...
Setting up libpython2-dev:armhf (2.7.16-1) ...
Setting up python2.7-dev (2.7.16-2+deb10u1) ...
Setting up flex (2.6.4-6.2) ...
Setting up libpcap-dev:armhf (1.8.1-6) ...
Setting up python2-dev (2.7.16-1) ...
Setting up libfl-dev:armhf (2.6.4-6.2) ...
Setting up libpython-dev:armhf (2.7.16-1) ...
Setting up python-dev (2.7.16-1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for install-info (6.5.0.dfsg.1-4+b1) ...
Processing triggers for libc-bin (2.28-10+rpi1) ...
pi@zeekpi:~ $
```

Instal·lació de Zeek:

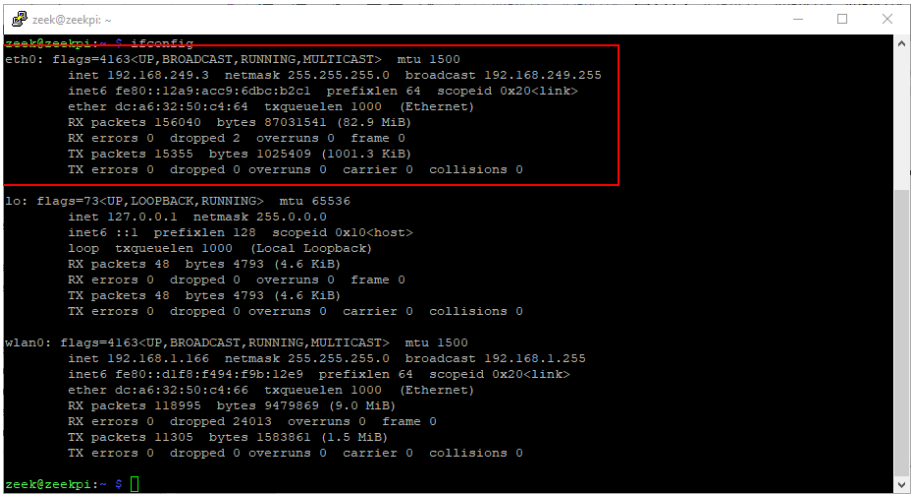
```
pi@zeekpi: ~/zeek
pi@zeekpi:~/zeek $ sudo apt-key add - < Release.key
OK
pi@zeekpi:~/zeek $ apt-get update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)
pi@zeekpi:~/zeek $ sudo apt-get update
Get:1 http://download.opensuse.org/repositories/security:/zeek/Raspbian_10  InRelease [1,540 B]
Hit:2 http://archive.raspberrypi.org/debian buster InRelease
Get:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Get:4 http://download.opensuse.org/repositories/security:/zeek/Raspbian_10  Packages [4,369 B]
Get:5 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0 MB]
Fetched 13.0 MB in 6s (2,034 kB/s)
Reading package lists... Done
pi@zeekpi:~/zeek $ sudo apt-get install zeek
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  liblockfile-bin liblockfile1 lockfile-progs sendmail-base sendmail-bin sendmail-cf zeek-core zeekctl
Suggested packages:
  sendmail-doc logcheck sasl2-bin libsasl2-modules
The following NEW packages will be installed:
  liblockfile-bin liblockfile1 lockfile-progs sendmail-base sendmail-bin sendmail-cf zeek zeek-core
  zeekctl
0 upgraded, 9 newly installed, 0 to remove and 4 not upgraded.
Need to get 7,204 kB of archives.
After this operation, 26.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

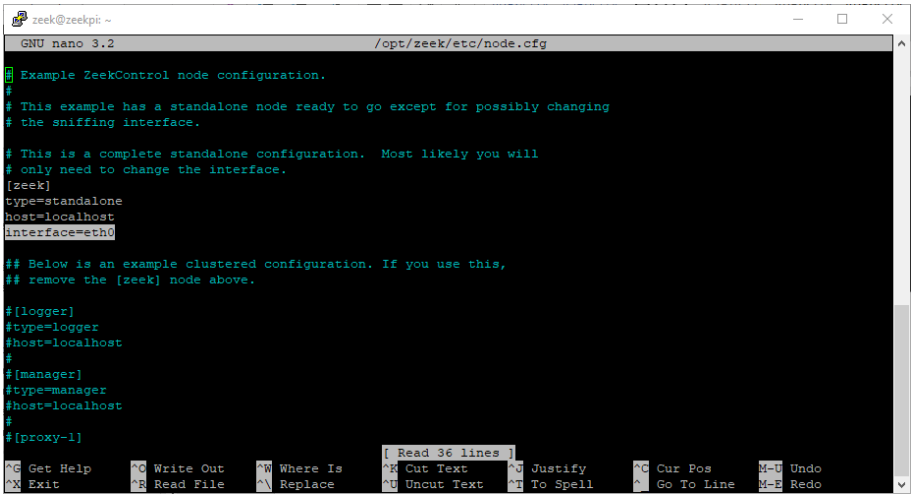
```
pi@zeekpi: ~/zeek
Validating configuration.
Creating /etc/mail/Makefile...
Reading configuration from /etc/mail/sendmail.conf.
Validating configuration.
Writing configuration to /etc/mail/sendmail.conf.
Writing /etc/cron.d/sendmail.
Disabling HOST statistics file(/var/lib/sendmail/host_status).
Creating /etc/mail/sendmail.cf...
Creating /etc/mail/submit.cf...
Informational: confCR_FILE file empty: /etc/mail/relay-domains
Warning: confCT_FILE source file not found: /etc/mail/trusted-users
 it was created
Informational: confCT_FILE file empty: /etc/mail/trusted-users
Warning: confCW_FILE source file not found: /etc/mail/local-host-names
 it was created
Warning: access_db source file not found: /etc/mail/access
 it was created
Updating /etc/mail/access...
Linking /etc/aliases to /etc/mail/aliases
Informational: ALIAS_FILE file empty: /etc/mail/aliases
Updating /etc/mail/aliases...
WARNING: local host name (zeekpi) is not qualified; see cf/README: WHO AM I?
/etc/mail/aliases: 0 aliases, longest 0 bytes, 0 bytes total

Warning: 3 database(s) sources
        were not found, (but were created)
        please investigate.
Processing triggers for systemd (241-7~deb10u3+rpi1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rpi1) ...
pi@zeekpi:~/zeek $
```

Configuració de les variables d'entorn:

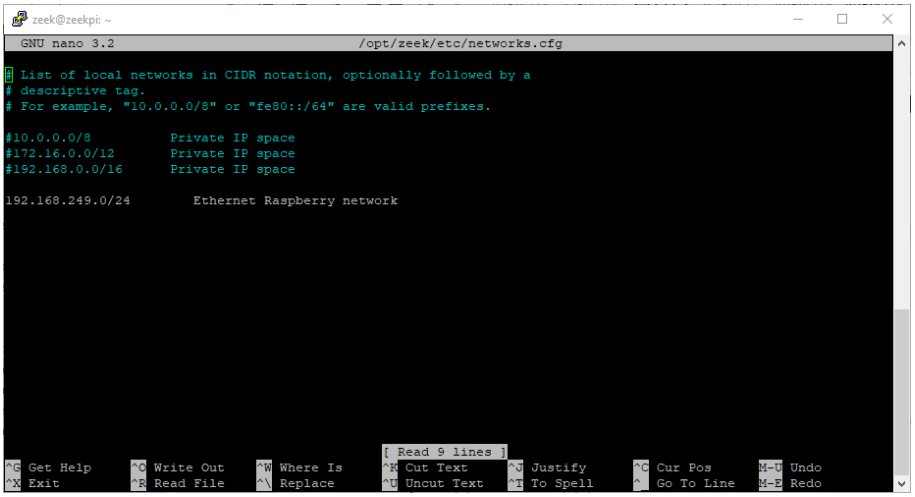


La interfície a monitoritzar és la "eth0":

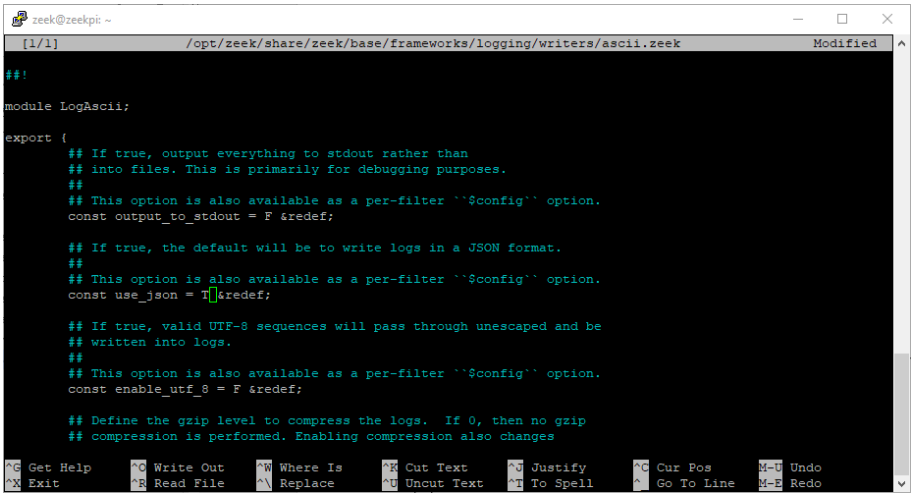




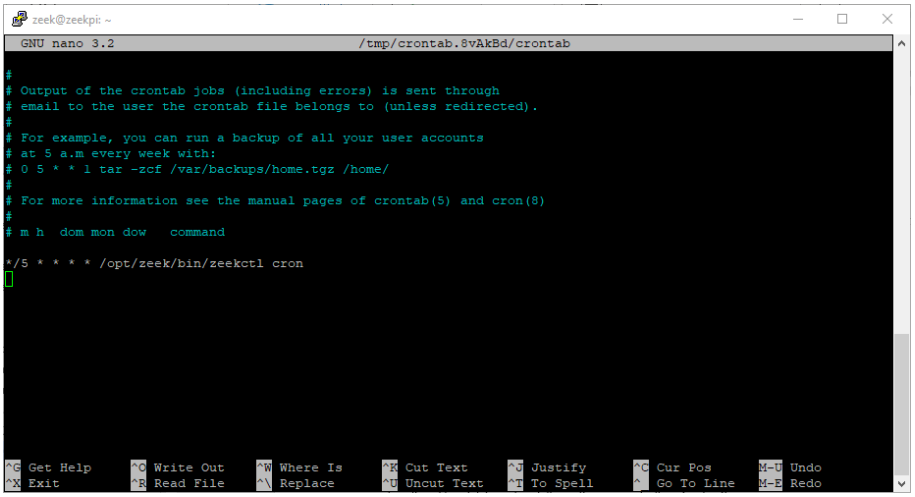
Definir direccionament de xarxa a capturar per a Zeek:



Redefinir la sortida dels logs a format JSON:

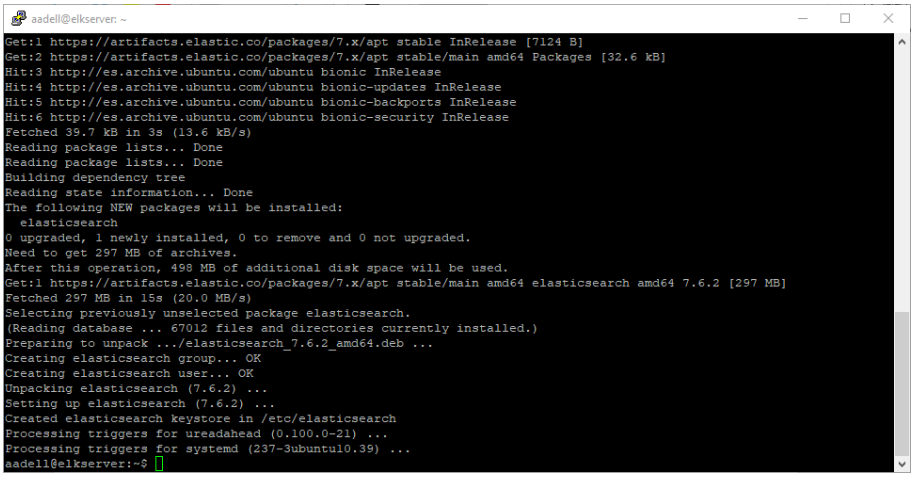


Tasques de manteniment mitjançant el Crontab de l'usuari Zeek:

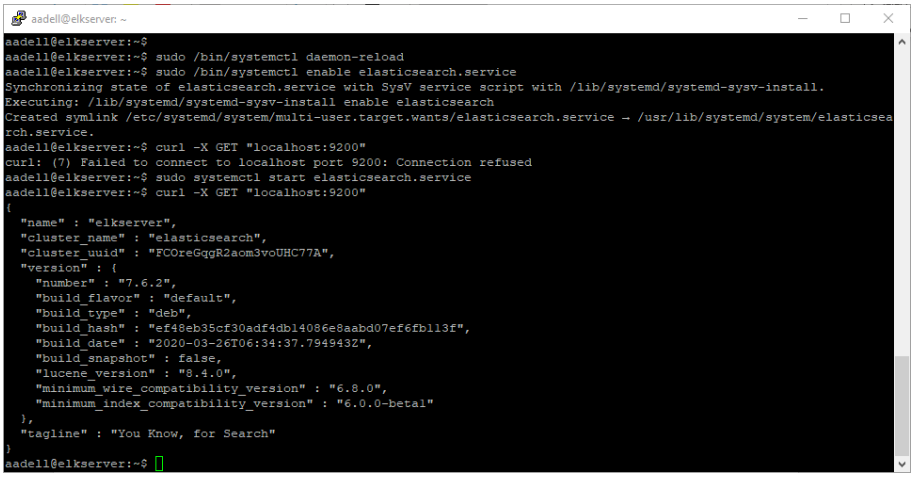


7.3. Annex 3: Detalls en la instal·lació de ELK Stack

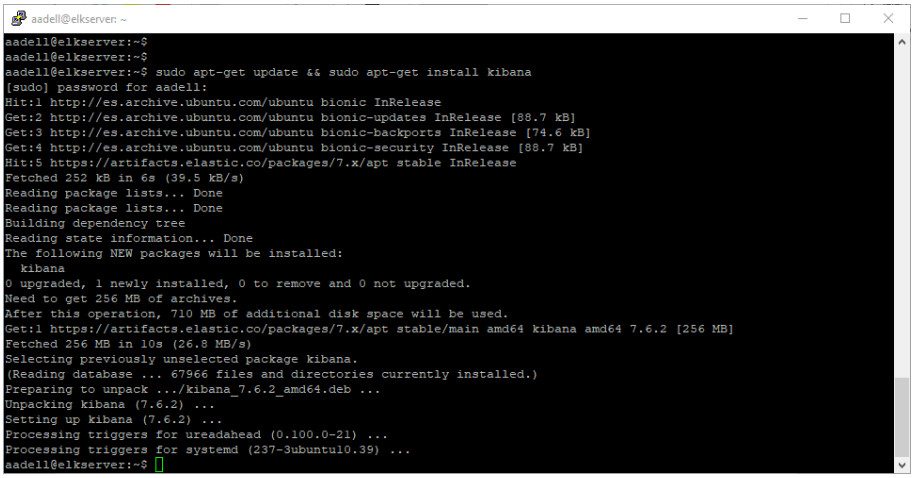
Instal·lació d'Elasticsearch:



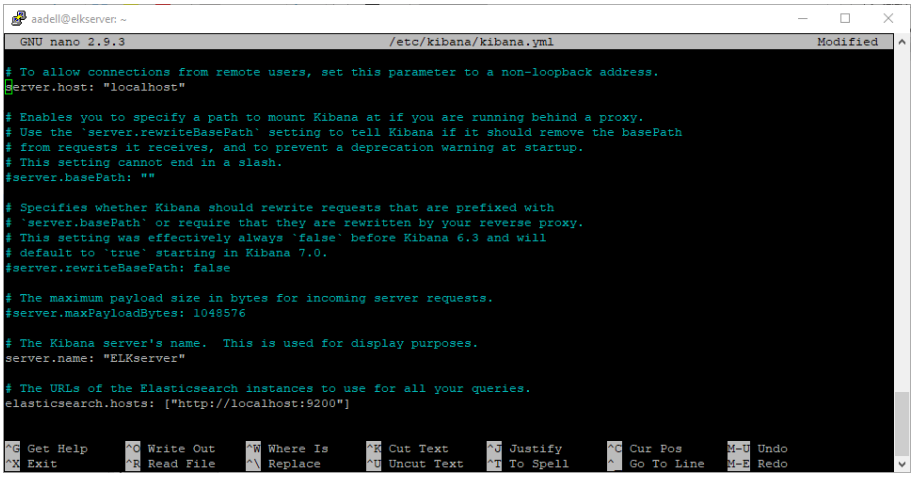
Servei Elasticsearch contesta correctament a una petició HTTP:



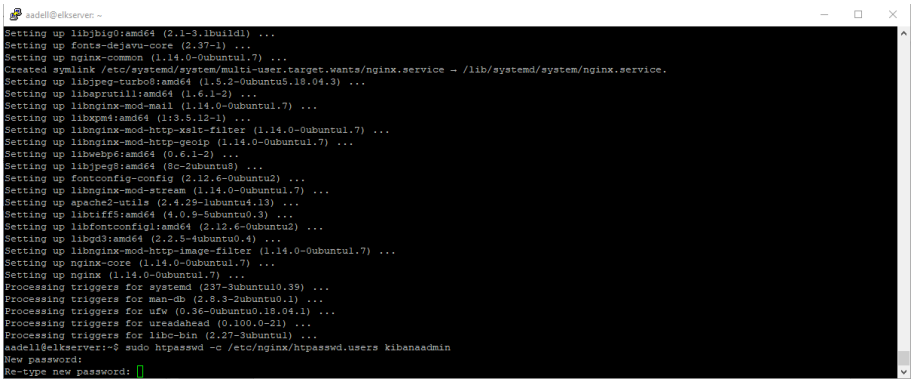
Instal·lació de Kibana:

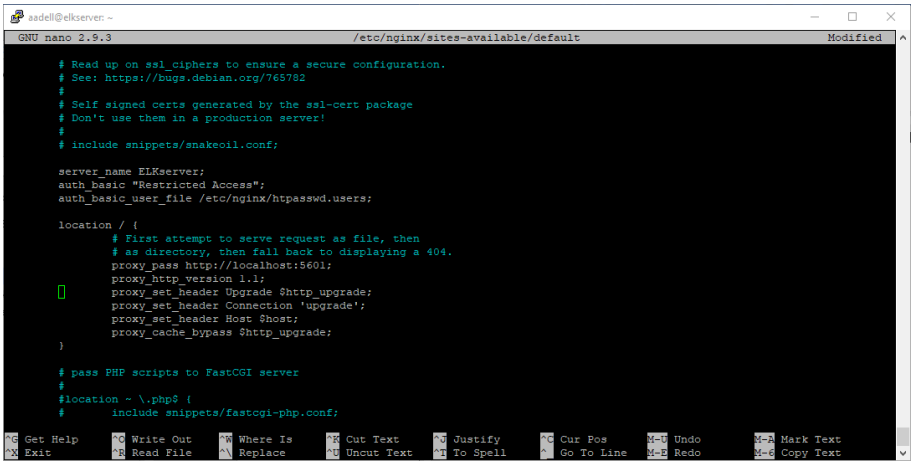


Arxiu de configuració de Kibana:

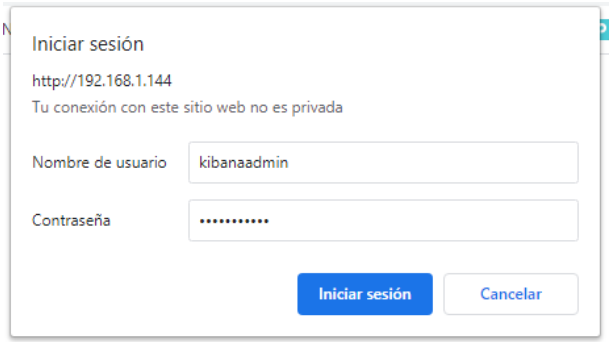


Instal·lació i configuració de Nginx, per l'accés a Kibana:

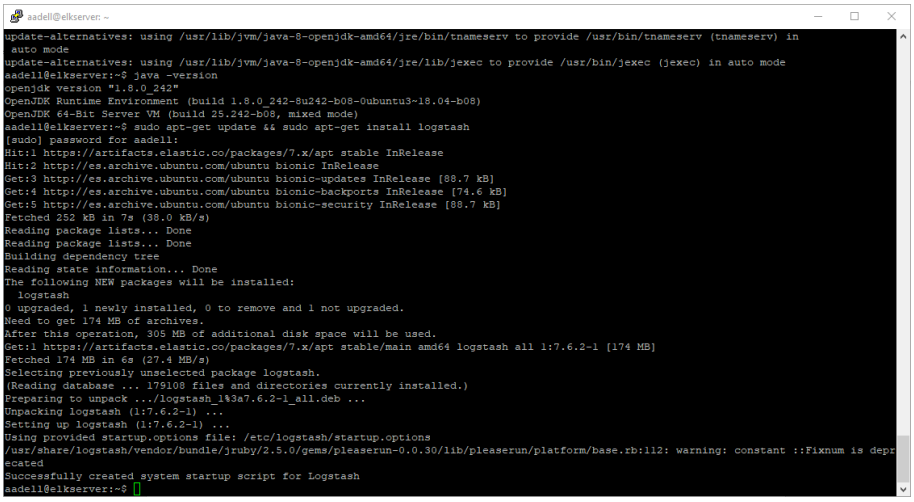




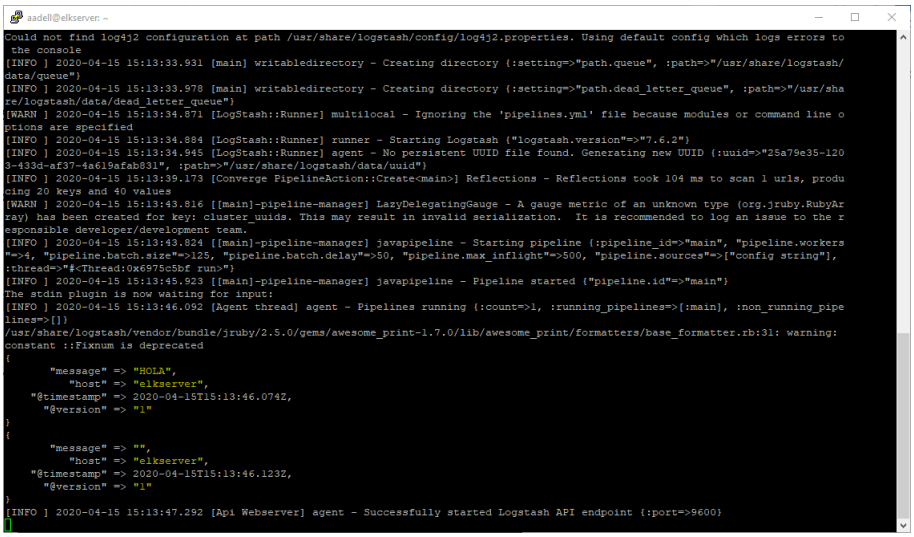
L’accés a Kibana està restringit a les credencials de l’usuari “kibanaadmin”:



Instal·lació de Logstash:



Comprovació senzilla i interactiva del funcionament de Logstash:



7.4. Annex 4: Pipeline de processament Logstash

A continuació, el contingut íntegre del fitxer “zeek-pipeline.conf”:

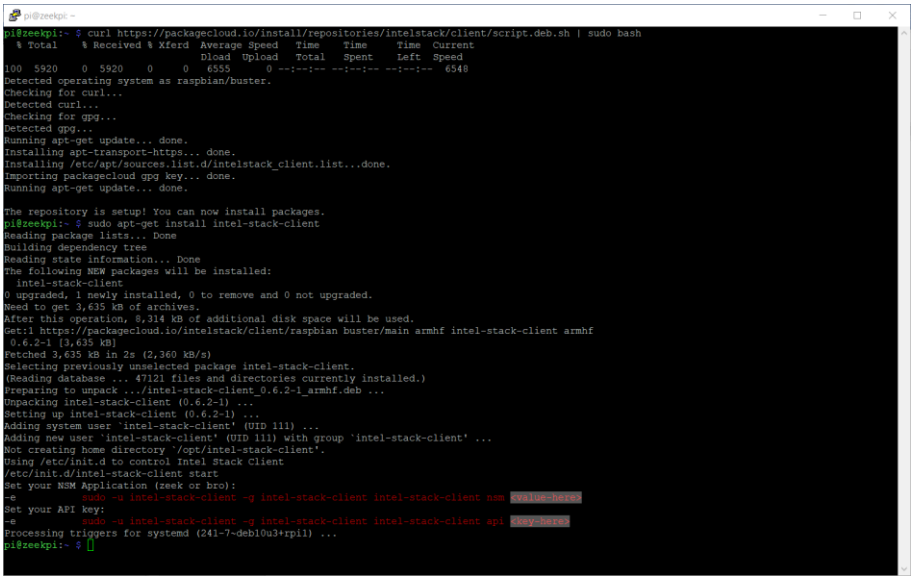
```
## INPUT - Entrada de Filebeat
input {
 beats {
          port => "5044"
          ssl => true
          ssl_certificate => "/etc/pki/logstash.crt"
          ssl_key => "/etc/pki/logstash.key"
 }
}

## FILTER - Processament
filter {

 ## Filtar sorolls no desitjats de la xarxa
 if [source][address] in
["192.168.1.16","192.168.1.41","192.168.1.36","192.168.1.39","192.168.
1.56","192.168.1.68"] { drop { } }
 if [destination][address] in
["192.168.1.41","192.168.1.255","224.0.0.251","224.0.0.252","ff02::16"
,"ff02::fb","ff02::1:3"] { drop { } }
        if [dns][question][name] in
["_googlecast._tcp.local","_spotify-
connect._tcp.local","_smb._tcp.local","RED","wpad.local","wpad”]
{ drop { } }

 ## Establir entorn (test/real) i sortida a Elasticsearch
 mutate {
          #add_field => { "[@metadata][entorn]" => "test" }
          add_field => { "[@metadata][entorn]" => "real" }
          add_field => { "[@metadata][elastic_output]" => "true" }
          #add_field => { "[@metadata][elastic_output]" => "false" }
 }

 ## Aqui comença el tractament, diferenciat per cada tipus de LOG

 ## CONNECTION actions
        if [fileset][name] == "connection" {
          # Cap acció especial
 }

        ## DNS actions
        if [fileset][name] == "dns" {
          ## GeoIP
          if [dns][resolved_ip] {
                   geoip {
                            source => "[dns][resolved_ip]"
                   }
          }
 }

 ## FILES actions
        if [fileset][name] == "files" {
                ## GeoIP
                if [zeek][files][rx_hosts] {
                  split {
                            field => "[zeek][files][tx_hosts]"
                  }
                  geoip {
                                    source => "[zeek][files][tx_hosts]"
                  }
          }
```

```
                ## Check VirusTotal (només contingut no ssl)
                if [zeek][files][md5] and [zeek][files][mime_type] !=
"application/x-x509-ca-cert" {
                 ## MD5 d'un fitxer bo conegut, es simula que és un
malicios substituint el hash
                 if [zeek][files][md5] ==
"7a3ab115599e37b7cc96aaa019dc3429" {
                          mutate {
                                   update => { "[zeek][files][md5]" =>
"9498FF82A64FF445398C8426ED63EA5B" }
                          }
                 }
                 http {
                          url =>
"https://www.virustotal.com/api/v3/files/%{[zeek][files][md5]}"
                          headers => { "x-apikey" =>
"fc59c20bf3002b44262f058efc647a55fc1a68391f5c2d69b0bd5704d6e8305a" }
                 }
                 mutate {
                          add_field => { "av_score" => "0" }
                 }
                 if
[body][data][attributes][last_analysis_stats][malicious] {
                          mutate {
                                   update => { "av_score" =>
"%{[body][data][attributes][last_analysis_stats][malicious]}" }
                          }
                          mutate {
                                   remove_field => [ "headers" ]
                                   remove_field => [ "body" ]
                          }
                 }
         }
        }

        ## FTP actions
        if [fileset][name] == "ftp" {
         # Cap acció especial
}

        ## HTTP actions
        if [fileset][name] == "http" {
         # Cap acció especial
}

         ## INTEL actions
        if [fileset][name] == "intel" {
                ## GeoIP
         if [destination][ip] {
                  geoip {
                          source => "[destination][ip]"
                  }
         }
}

        ## NOTICE actions
        if [fileset][name] == "notice" {
         # Cap acció especial
}

        ## SSH actions
        if [fileset][name] == "ssh" {
         # Cap acció especial
}

## SSL actions
if [fileset][ssl] == "ssl" {
         ## GeoIP
```

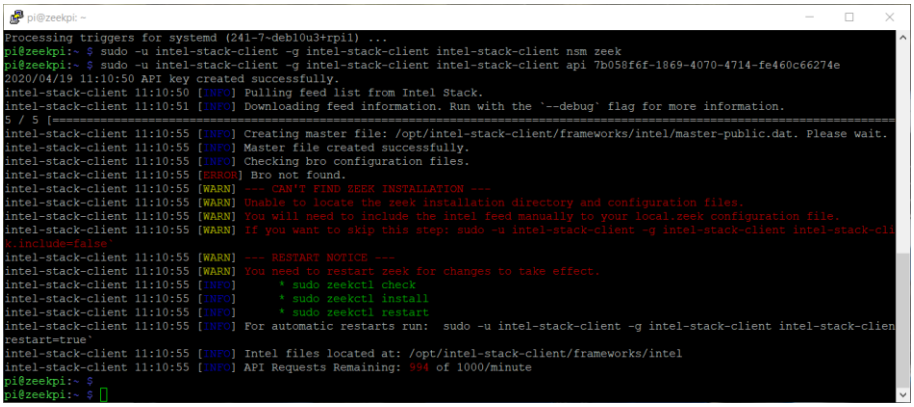
```
                if [destination][address] {
                        geoip {
                                source => "[destination][address]"
                        }
                }
 }

}

## OUTPUT - Enviar a Elasticsearch
output {

 # En proves / test, es mostra el debug
 if [@metadata][entorn] == "test" {
          stdout { codec => rubydebug }
 }

 # Sortida Elasticsearch si està especificat en true
 if [@metadata][elastic_output] == "true" {
          elasticsearch {
                   hosts => ["localhost:9200"]
                   index =>
"%{[agent][type]}-%{[@metadata][entorn]}-%{[fileset][name]}"
          }
 }

}
```

7.5. Annex 5: Detalls en la instal·lació d'Intel Stack

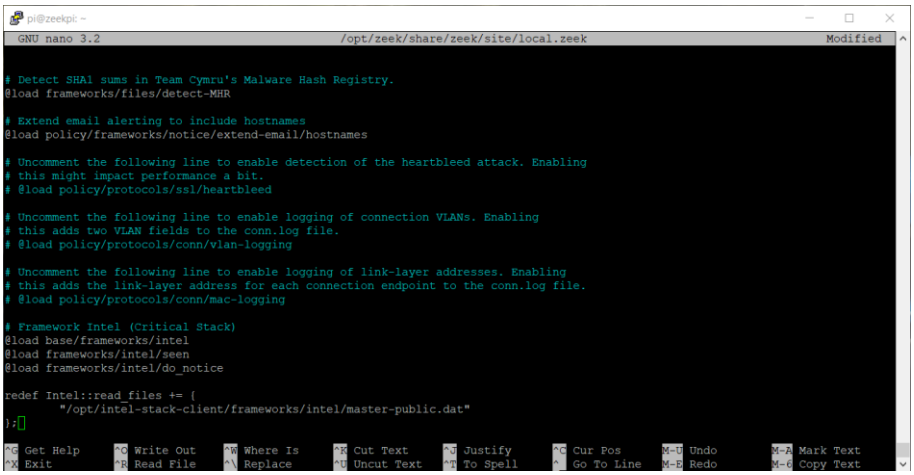
Instal·lació del client Intel Stack:



Configurar el Sensor amb la clau API corresponent:



Modificar arxiu de configuració Zeek per a carregar el framework d'intel·ligància:



7.6. Annex 6: Detalls en la instal·lació del joc de proves

7.6.1. Scripts de generació de tràfic

Els contingut dels scripts és:

- Trafic.sh

```
#! /bin/sh
curl $(shuf -n 1 /home/user1/Desktop/Websites.txt)
```

- Bad_Trafic.sh

```
#! /bin/sh
curl $(shuf -n 1 /home/user1/Desktop/Bad_Websites.txt)
```

- Down_Files.sh

```
#! /bin/sh
wget $(shuf -n 1 /home/user1/Desktop/Files.txt) -P temp/
rm -rf temp
```

I un exemple del contingut dels respectius fitxers, font de URLs, són:

- Websites.txt

```
https://www.uoc.edu
https://www.urv.cat
https://www.uab.cat
https://www.youtube.com
https://www.facebook.com
https://www.baidu.com
https://www.yahoo.com
https://www.amazon.com
https://www.wikipedia.org
https://www.qq.com
...
```

- Bad_Websites.txt

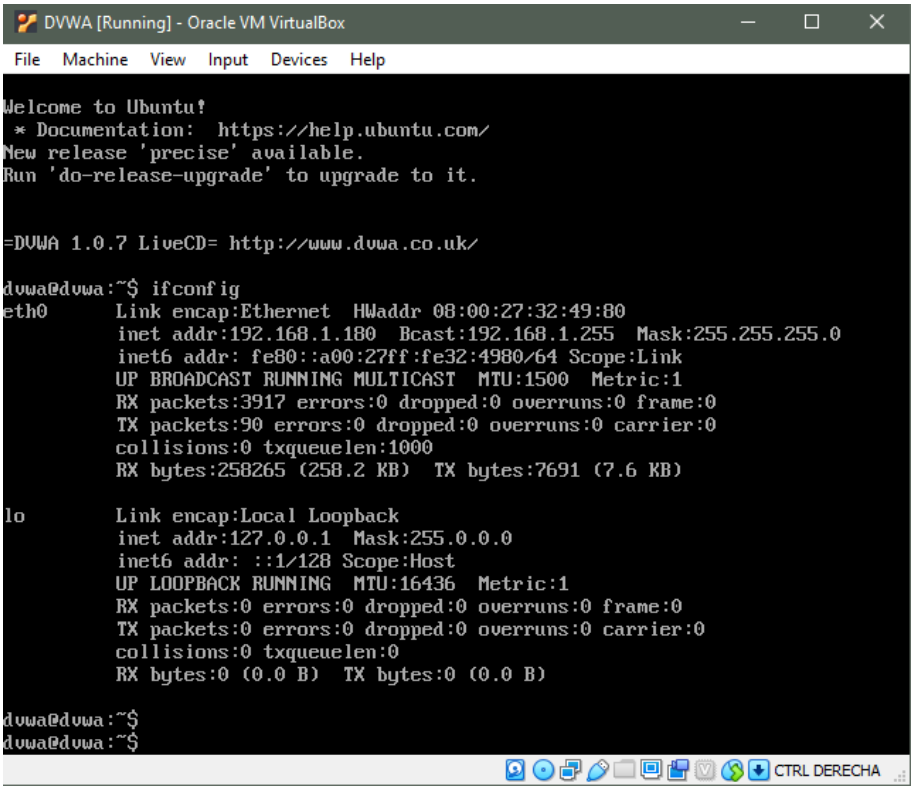
```
bakron.co.za
www.baseballsuccesssecrets.com
www.bttsystems.pl
128.199.155.218
47.176.39.218
173.234.227.184
104.244.73.126
www.blog-e-pedia.com
www.netflixhulufree.com
duffelbagadventures.com/mls/bbva-compass-stadium-houston-
dynamo/
...
```

- Files.txt

```
http://download.winimage.com/wimaia81.zip
http://downloads.sourceforge.net/gnuwin32/zip-3.0-setup.exe
http://www.stavrox.com/download/MeroX2.0.jar
http://www.gburner.com/gburner4.exe
http://www.usblyzer.com/files/USBlyzer.zip
http://bostjan-grandovec.si/Files/Pantheon.zip
https://codeload.github.com/elastic/beats/zip/v7.6.1
https://www.zeek.org/downloads/zeek-3.1.0.tar.gz
https://netcologne.dl.sourceforge.net/project/win32diskimager/A
rchive/win32diskimager-1.0.0-install.exe
https://download.skype.com/s4l/download/win/Skype-
8.56.0.103.exe
...
```

7.6.2. DVWA (Damn Vulnerable Web App)

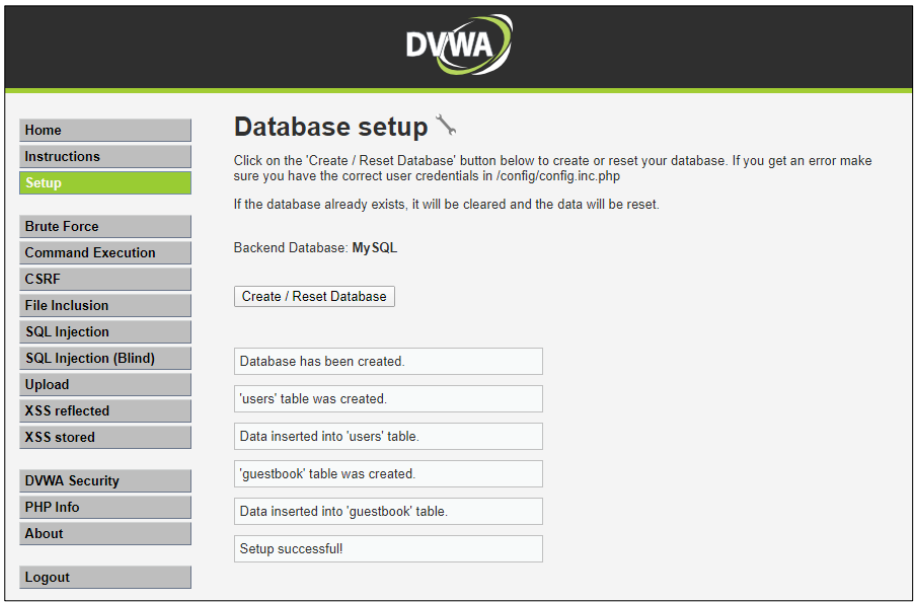
DVWA versió 1.0.7, és un arxiu ISO que es configura en una màquina virtual i arrenca el sistema operatiu en viu (Live).



Obrir navegador web a la IP corresponent:

- http://192.168.249.180

Anar a l'apartat Setup i seleccionar "Create Database".



Anar a l'apartat "DVWA Security" i seleccionar seguretat "Low".