Treball de Fi de Grau

Sistema per la gestió i filtratge d'intel·ligència sobre amenaces cibernètiques

Informe final

Biel Camprubí Trujillo, NIU: 1526442

Resum — Aquest treball pretén millorar la gestió d'indicadors de ciberseguretat (IOCs) en empreses mitjançant la creació d'un sistema modular basat en MISP, que filtri indicadors potencialment erronis per evitar falsos positius.

El sistema es desenvoluparà seguint una metodologia de disseny, implementació i proves, i estarà enllaçat amb un repositori a Github. El sistema inclourà un mòdul de filtratge d'IOCs i un mòdul de control del sistema a través d'un portal web. Es podrà desplegar al núvol i connectar-se amb sistemes de seguretat com SIEMs, EDRs o firewalls, optimitzant així la gestió de la ciberseguretat en l'empresa.

Paraules clau — Indicadors de ciberseguretat (IOCs), Empreses, Sistema modular, MISP, Núvol, Sistemes de seguretat, SIEMs, EDRs, Firewalls, Gestió de la ciberseguretat, ReactJS, Python, APIs, Github, Web.

Abstract — This project aims to improve the management of cybersecurity indicators (IOCs) in companies through the creation of a modular system based on MISP, which filters potentially erroneous indicators to prevent false positives.

The system will be developed following a methodology of design, implementation and testing, and will be linked to a Github repository. The system will include an IOC filtering module and a system control module through a web portal. It will be able to be deployed in the cloud and connect with security systems such as SIEMs, EDRs, or firewalls, thus optimizing the company's cybersecurity management.

Index Terms — Cyber Security Indicators (IOCs), Businesses, Modular System, MISP, Cloud, Security Systems, SIEMs, EDRs, Firewalls, Cyber Security Management, ReactJS, Python, APIs, Github, Web.

1 Introducció

Alhora d'evitar possibles atacs informàtics les empreses solen recol·lectar indicadors (per exemple hashes, ips, dominis…) anomenats IOC [1] que representen possibles amenaces per l'empresa. Aquest procediment s'anomena cyber threat intelligence (CTI) [2].

Aquestes dades s'obtenen a partir de projectes open source, intel·ligència de xarxes socials, anàlisis forenses... Normalment tots aquests indicadors es bloquegen en els diferents sistemes de l'empresa (com ara tallafocs o antivirus).

El problema és que moltes vegades aquests indicadors poden ser erronis i generar problemes al ser bloquejats. Un exemple seria si es bloqueja una ip de google i els seus serveis deixen de funcionar dins l'empresa.

Un altre problema és que, tot i que existeixen solucions per emmagatzemar aquests indicadors, aquestes solucions no estan preparades per connectar-se amb els sistemes de seguretat que solen tenir les empreses, com ara SIEMs [3], EDRs [4] o firewalls [5].

2 Objectiu

L'objectiu d'aquest treball és solucionar els problemes mencionats anteriorment. Per aconseguir-ho es crearà un sistema que s'encarregarà de la ingesta i emmagatzematge d'indicadors de ciberseguretat (principalment hashes, ips i dominis) de diverses fonts, utilitzant una sèrie de mètriques, filtrarà els indicadors que no suposin un risc per l'empresa i enviarà els indicadors restants als diferents sistemes de seguretat de l'empresa (principalment tallafocs i SIEMS).

Aquest sistema s'ha de poder instal·lar al cloud [6] i ha de ser capaç d'acceptar un gran flux de dades sense pèrdues ni errors. També ha de ser modular, la solució final s'oferirà com un servei i haurà de ser personalitzable segons els requisits del client.

El sistema tindrà com a element central un programa existent de gestió de IOCs que es farà servir com a base de dades.

3 METODOLOGIA

La metodologia que s'utilitzarà per complir tots els objectius del projecte dins del termini acordat tindrà les següents fases:

1. Definició del problema: En aquesta fase s'ha d'identificar el problema que es vol resoldre, això inclou els requisits principals i una petita explicació de com es volen aconseguir els resultats esperats. Aquest document correspon a aquesta fase.
2. Pla: Desenvolupar un pla que inclogui l'abast del projecte, els terminis, les fites i els lliuraments. També està contingut en aquest document en el següent apartat.
3. Disseny: En aquesta fase es definirà un disseny per al programari, inclosa la seva arquitectura, model de dades, interfície d'usuari i funcionalitat.
4. Implementació: En aquesta fase és on es desenvoluparà el codi i s'implementaran les diferents funcionalitats.
5. Test: En aquesta fase es provarà el programari per assegurar que compleix els requisits i les funcions previstes.

4 PLANIFICACIÓ

Seguint la metodologia definida anteriorment s'ha preparat el següent pla de projecte que tindrà tres fases, també s'han afegit unes dates d'entrega previstes per cada tasca (vegeu la figura **A.1** - diagrama de Gantt per entrar en mes detall):

- Disseny: S'estudiaran les diferents possibles solucions i es dissenyarà la solució final.
- Implementació: Es desenvoluparà el codi. Les parts que s'hauran de desenvolupar principalment seran el mòdul de filtratge i el programa IOC Manager, però és possible que també s'hagin de desenvolupar altres scripts.
- Test: Es crearan testos unitaris per comprovar que el sistema funciona correctament. Consistirà de dos parts, el test general que verificarà el mòdul score i un altre test que validarà el IOC Manager.

Per dur a terme aquest projecte s'ha creat un petit equip que està format per tres persones, un cloud architect (que s'ha encarregat de fer el desplegament del sistema a Azure), un project manager (que s'ha encarregat de vendre el projecte a diferents clients i les comunicacions entre les dues parts) i un senior security engineer el qual soc jo mateix i m'he encarregat de dissenyar i programar tota l'arquitectura (la totalitat del que s'explica en aquest document ha estat feta únicament per mi).

5 ANÀLISI

5.1 Estat de l'art

La majoria de plataformes de CTI son privades, com per exemple Threat connect [7] o IBM X-force [8]. En el nostre cas volem crear una solució que sigui adaptable a qualsevol entorn, per tant no podem utilitzar aquests programes ja que el codi font no està obert.

Per la part de plataformes open source existeixen dos solucions que destaquen:

- MISP [9]: Solució de programari de codi obert per recopilar, emmagatzemar, distribuir i compartir indicadors de cyberseguretat i amenaces sobre l'anàlisi d'incidents de cyberseguretat i l'anàlisi de programari maliciós. MISP està dissenyat per i per a analistes d'incidents, professionals de seguretat i TIC per donar suport a les seves operacions diàries per compartir informació estructurada de manera eficient.
- OpenCTI [10]: OpenCTI és una plataforma de codi obert que permet a les organitzacions gestionar els seus coneixements i observables d'intel·ligència sobre amenaces cibernètiques. Ha estat creat per estructurar, emmagatzemar, organitzar i visualitzar informació tècnica i no tècnica sobre les cyberamenaces.

En la següent taula es comparen les funcionalitats mes rellevants per complir els nostres requisits que tenen cada una de les solucions:

Funcionalitat	MISP	OpenCTI
Filtratge IOCs	X	✓
Connectivitat amb sistemes de seguretat (SIEM, Firewall...)	X	✓
API	✓	✓
Fonts de dades	✓	X

Com podem veure en la taula anterior OpenCTI és una opció molt mes completa i per aconseguir totes les funcionalitats necessàries s'haurien de fer pocs canvis.

Tot i això el programa que s'ha escollit amb el qual basar la solució final és MISP, això és degut a que fa molt mes temps que existeix i la majoria d'empreses ja l'utilitzen, també hi ha una gran quantitat de fonts d'alimentació de dades open source compatibles amb aquest programa.

MISP no té capacitat per filtrar IOCs, com s'ha comentat anteriorment això pot introduir falsos positius en el sistema. Per resoldre aquest problema s'estendran les capacitats de MISP per aconseguir els resultats esperats.

5.2 Entorn

Per desenvolupar aquest projecte es crearà un únic repositori a github [11], ja que tot i tenir varis mòduls jo

seré l'únic desenvolupador i això simplificarà el seu desenvolupament, ja que si s'escollís utilitzar una estratègia multi-repo hi haurien mes arxius i no seria tan còmode [12]. En aquest repositori es guardarà tot el codi font del sistema.

Per altre banda i connectat a aquest repositori hi haurà un projecte de Jira [13] on s'aniran creant totes les característiques a desenvolupar i bugs que es vagin trobant. Cada vegada que es vulgui arreglar un bug o crear una nova característica es crearà una branca al repositori i allà es farà la implementació. Una vegada acabat el desenvolupament es farà un pull request i si no dona cap problema un merge.

Alhora de debugar i programar s'utilitzarà l'editor de codi VSCode [14], per obtenir una bona experiència de desenvolupament s'instal·larà el plugin de python (que ens permet debugar [15]), i el formatejador Autopep8 (formatarà el codi perquè segueixi l'estil de programació PEP8 [16]).

Per últim també es tindrà d'instal·lar Docker Desktop [17] per poder desplegar els containers creats per el sistema i assegurar-nos que funciona correctament.

6 DISSENY

Una vegada escollida la plataforma amb la que es basarà la solució final ja podem formular la proposta.

A sobre de la plataforma escollida (MISP) s'aniran afegint mòduls que interactuaran amb ell a través de la seva API.

Inicialment el sistema tindrà dos mòduls desenvolupats per nosaltres:

- Mòdul de filtratge: S'encarregarà d'assegurar-se que els IOCs del sistema suposen un perill real i que no hi ha falsos positius.
- Mòdul de control del sistema: Serà un portal web que permetrà consultar l'estat del sistema així com carregar nous indicadors a aquest.

A part d'aquests dos mòduls generals també es podran afegir altres per importar o exportar els IOCs segons conveniència (per exemple enviar les dades a un SIEM, EDR o tallafoc).

Per tant, el sistema tindrà tres programes principals:

- MISP: S'utilitzarà com a base de dades i es connectarà a diverses fonts d'IOCs, aquest programa està mantingut per CIRCL, tot i que contindrà un addon creat i mantingut per Cyberproof.
- IOC Manager: Portal web per mirar el estat actual del sistema, permetrà afegir, esborrar i consultar els IOCs que hi ha dins de MISP. També tindrà una ruta que retornarà una llista amb els indicadors que han passat el filtre, aquesta llista s'utilitzarà per injectar els indicadors a plataformes de seguretat com ara un Firewall. Aquest programa serà creat i mantingut per Cyberproof.
- Score: S'encarrega de filtrar els indicadors que hi ha dins de MISP. També estarà creat i mantingut per Cyberproof.

El sistema funcionarà sobre docker, es podrà desplegar en un entorn cloud i estarà constitut per els següents containers:

1. IOC Manager Backend: Servidor backend del IOC manager (backend del mòdul de control del sistema).
2. IOC Manager Frontend: Component de gestió manual de IOCs (frontend del mòdul de control del sistema).
3. MISP Web: Aplicació principal de MISP (codi del programa MISP).
4. MISP Database: Base de dades de MISP (base de dades del programa MISP).
5. Score: Component per enriquir i calcular el Score dels IOCs (Mòdul de filtratge).

En el diagrama **A.2** es pot observar com es comunicaran els diferents containers entre ells, en el punt **7** s'entrarà en mes detall sobre el seu funcionament.

En verd estan pintades les parts desenvolupades completament per Cyberproof, en groc les que s'han fet modificacions (Un mòdul de MISP que s'explicarà en el punt **7.1**) i en blau les parts que no han estat fetes per Cyberproof:

El mòdul score s'executarà cada 6 hores per filtrar els IOCs que hi hagin a MISP, les peticions a l'API es faran de forma concurrent perquè el temps de processament no sigui molt alt.

El portal IOC Manager podrà afegir, esborrar i consultar indicadors, per fer-ho utilitzarà l'api de MISP. També podrà llistar tots els indicador que han passat el filtre per ser exportats a altres sistemes (això correspon al requadre anomenat Feed Server de la figura **A.2**).

7 IMPLEMENTACIÓ

En aquest apartat s'explicarà com s'ha implementat el sistema. Com s'ha esmentat en l'apartat de la metodologia, la implementació és la segona fase del projecte i al començar-la ja es coneix l'arquitectura que el sistema tindrà (està documentada en l'apartat anterior).

Aquest apartat s'ha dividit en els diferents subsistemes que conté l'arquitectura:

7.1 Misp

Ja que la infraestructura s'executarà sobre docker per desplegar aquesta solució s'utilitzarà la seva versió containeritzada (misp-docker [18]). Per desplegar-se s'utilitza un fitxer docker-compose que crea dos containers anomenats MISP Web i MISP Database).

Com que la resta de subsistemes utilitzen una API de MISP, perquè tot funcioni correctament al desplegar la solució, s'haurà de crear una clau API dins de MISP i actualitzar les variables d'entorn amb aquestes dades. Això només s'haurà de configurar una vegada, ja que les dades persisteixen.

Misp té una funcionalitat anomenada decaying models [19], on pots configurar un algorisme que modifica el score segons varis paràmetres. S'ha desenvolupat una funció que té en compte tres variables:

- Sighting info: Aquesta variable s'obté a través d'una funcionalitat de MISP anomenada sightings [20], que permet notificar cada vegada que es troba un IOC a un sistema extern passant una sèrie d'informació sobre el que s'ha trobat (l'idea és connectar solucions com firewalls, SIEMs i EDRs a aquest endpoint per rebre dades reals d'iocs que es troben en els sistemes de l'empresa).
- Sighting date: Aquesta variable s'obté a través de la mateixa funcionalitat comentada anteriorment i és el timestamp de l'últim sighting obtingut per un determinat IOC.
- Score: És la mitjana entre els diferents scores obtinguts de les APIs (Es calcula en el mòdul score).

Podeu trobar mes detalls d'aquests camps en la figura **A.3**. Per aconseguir el score final s'utilitza la següent fórmula:

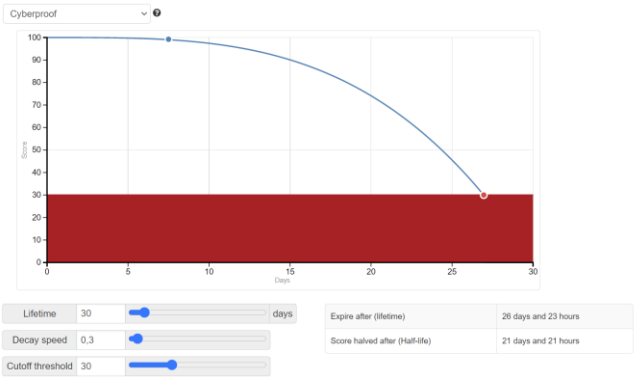
$$score = basescore \cdot \left(1 - \left(\frac{t}{ta}\right)^{\frac{1}{\delta a}}\right)$$

Aquesta és una funció decreixent que modifica el basescore (el resultant de la mitjana de tots els scores mes el modificador calculat amb la variable Sighting info) segons l'última vegada que el IOC s'ha vist (Sighting date).

La variable t és el temps actual menys el temps del darrer sighting (Sighting date), ta és el temps de vida o lifetime de l'IOC (per exemple 3 mesos, aquesta constant és configurable i canvia depenent del tipus d'IOC) i δa és el pendent de la funció, és a dir com de ràpid decau (decay speed), que també és configurable.

Aquest algorisme està programat utilitzant PHP i carrega automàticament a MISP creant un volum en el docker compose amb el codi i s'executa per cada IOC una vegada al dia. Una vegada calculat el score final el programa mira si aquest és superior o inferior a una altre constant configurable anomenada cutoff threshold, si és inferior seteja una flag que proporciona MISP (IDS [21]) a false i si és superior la seteja a true, cosa que farà que s'exporti a sistemes externs o no.

Misp proporciona un frontend on es poden configurar totes les constants mencionades anteriorment, en aquesta figura podeu veure un exemple de configuració:



7.2 Score

Aquest és el mòdul principal on es filtren els IOCs i està desplegat al container Score.

El flux de filtratge (la part dins del requadre que fica Every 6h de la figura **A.2**) consisteix en les següents accions:

- Primer, utilitzant l'API de MISP, s'obtindrà una llista amb tots els indicadors que no tenen la flag IDS mencionada anteriorment setejada a false. Aquests indicadors es reordenaran ficant els que han estat introduïts des del IOC Manager en les primeres posicions.
- A continuació s'obtindrà una puntuació per cada IOC que indicarà la perillositat d'aquest seguint l'ordre de la llista, això s'obtindrà utilitzant dos APIs: Virustotal [22] i AbuseIPDB [23].
- Un cop obtinguda la puntuació s'assignarà al seu IOC corresponent utilitzant una característica de MISP anomenada taxonomia [24], que introdueix una sèrie de tags que s'especifiquen dins d'objectes anomenats taxonomies. En el nostre cas s'ha creat una taxonomia anomenada score amb tags que prenen valors de 0 a 100 (representant així el score calculat).
- Quan s'hagi inserit el tag al seu IOC es mirarà si té una puntuació suficientment alta com per

passar el filtre (aquesta puntuació és la constant cutoff threshold mencionada anteriorment), si és així s'actualitzarà l'indicador a MISP contenint el tag corresponent i activarà el flag IDS, si no passa el filtre, s'eliminarà.

En el cas de que hi hagi algun error al processar un IOC no es farà cap canvi i a la següent execució es tornarà a processar, ja que el flag IDS seguirà sent fals. Aquesta part és crítica perquè les APIs tenen un límit de peticions, i si s'acaben no ha de generar cap problema ni falsos positius.

Ja que el sistema ha de poder suportar filtrar una gran quantitat de dades i diferents APIs de filtratge (s'han implementat Virustotal i AbuseIPDB però en un futur està planejat implementar-ne mes) el programa s'ha creat de forma que accepta mòduls (un per cada API) i utilitzant la llibreria concurrent.futures [25] de forma que es poden fer vàries peticions al mateix temps (el factor de paral·lelisme per defecte és 10, però es pot canviar si s'escau).

Els mòduls per cada api segueixen un format estàndard, cada un ha d'estar implementat dins d'una classe que contingui la variable pública _SOURCE_NAME (que ha de tenir com a valor el nom del mòdul) i el mètode _process_message(self, ioc, callback). Aquest mètode ha d'acceptar l'IOC del qual es vol obtenir el score i un callback callback(ioc, score, nom_modul) que acceptarà el ioc que s'ha processat, el score obtingut i el nom del mòdul que està sent executat (_SOURCE_NAME). Aquest callback es cridarà un cop processat l'IOC i és el que s'encarrega d'enviar les dades a MISP.

Aquests mòduls es carreguen a una carpeta i el programa els detecta automàticament cridant-los quan s'escau.

Com s'ha mencionat anteriorment, el programa s'executa cada 6h, per aconseguir-ho es configura un crontab en el dockerfile.

Per testejar el programa, s'ha creat un petit script que introdueix 1500 IOCs (el màxim d'indicadors que es poden processar en un dia) en el sistema dels quals ja es coneixen el resultat, executa el programa i una vegada acabada l'execució s'assegura que els resultats son els esperats.

7.3 IOC Manager

Aquest mòdul ofereix un frontend des d'on es poden carregar o eliminar IOCs, visualitzar els indicadors que s'han introduït des del portal i qui ho ha fet, ja que té un sistema d'autenticació per usuari i contrasenya i pot diferenciar qui ha fet cada acció. També té una pàgina amb la documentació de com utilitzar cada funcionalitat.

Aquest servei està constituït per un backend i un frontend:

Backend

El backend està fet utilitzant un framework de python anomenat Flask [26], s'ha escollit aquest framework degut a que, per conveni, és el que s'utilitza a l'empresa. Aquest backend es desplega en el container IOC Manager Backend i proporciona una API tipo REST amb les següents funcionalitats que s'han programat des de zero:

- IOCs: Permet afegir o suprimir IOCs del sistema, la llista d'indicadors a afegir o esborrar es pot passar o bé per un fitxer o per el camp iocs d'un formulari.
  En el cas de que s'estiguin pujant nous IOCs també rep un temps d'expiració que pot estar en mesos, dies o ser automàtic. Si és automàtic els IOCs es pujaran al sistema de forma normal i serà el mòdul Score l'encarregat de filtrar-lo i establir el seu temps de vida, però si aquest camp no és automàtic al pujar els indicadors també es crearà un sighting de tipus expiration especificant quan es vol que l'indicador caduqui, això farà que el mòdul score ignori aquests IOCs i que el flag IDS es seteji a true fins que expiri, una vegada expirat el mòdul score tornaria a valorar aquest indicador com un de normal.
  El programa també s'encarrega d'identificar el tipus d'IOC (ip, domini, hash md5, hash sha1...) automàticament utilitzant regex i els carrega a la instància de misp utilitzant la llibreria pymisp [27].
- Resultats: Permet descarregar un fitxer .txt o .xlsx amb els resultats de l'última vegada que es van carregar/esborrar IOCs, en aquests fitxers s'especifiquen tots els indicadors i el resultat de l'acció que s'intentava realitzar per cada un d'ells (per exemple si es va intentar carregar la ip 1.1.1.1 s'especificaria si es va carregar correctament a MISP i en cas que no fos així quin error va haver-hi).
- Autenticació: Permet iniciar sessió en el sistema, registrar un nou usuari, obtenir informació sobre l'usuari que ha iniciat sessió, tancar la sessió i esborrar un usuari. Les dades sobre els usuaris que es creen es guarden a una base de dades interna i les contrasenyes s'encripten automàticament. Al iniciar sessió es retorna un token que és necessari per accedir a tots els altres endpoints, això està gestionat per la llibreria flask_jwt_extended [28].
- Logger: Retorna informació sobre els IOCs que s'han carregat a MISP, per aconseguir-ho utilitza la llibreria pymisp descarregant tots els IOCs inserits al sistema i en el camp Comment dels indicadors hi ha el nom del usuari que el va

carregar (aquesta informació és inserida per l'endpoint IOCs al carregar dades).
- Feed Server: En aquest endpoint és on es connecten els sistemes externs per recollir les dades de MISP. Retorna la llista d'indicadors amb el flag IDS setejat a True que hi ha en el sistema, permet filtrar-los segons el seu tipus (ip, domini, hash MD5...) i establir una limitació d'indicadors retornats, això és important ja que algunes solucions nomes accepten un tipus d'IOC (per exemple els tallafocs nomes accepten IPs) i algunes altres solucions també tenen un límit d'indicadors que poden ingestar.

Frontend

El frontend està programat amb ReactJS [29], un framework de JavaScript, s'ha utilitzat TypeScript [30] (JavaScript amb anotació de tipus) per tenir una millor experiència de desenvolupament. També s'ha utilitzat el framework CSS Bootstrap [31], específicament el seu port a react anomenat react-bootstrap [32] aquests frameworks s'han escollit degut al mateix motiu que en el backend.

El frontend es desplega en el container IOC Manager Frontend i ofereix 5 pàgines diferents completament desenvolupades per nosaltres:

- Documentació: És on s'explica com utilitzar el sistema, d'aquesta forma nous usuaris poden aprendre ràpidament sense haver de buscar documentació en una pàgina externa. Veure figura **A.4**.
- Add IOC: En aquesta pàgina es poden afegir IOCs al sistema tant utilitzant un formulari com adjuntant un fitxer, també permet establir el temps de vida dels IOCs, per aconseguir-ho utilitza l'endpoint IOCs del backend. Veure figura **A.5**.
  Quan els indicadors s'estan carregant surt una barra que indica el progrés i en finalitzar apareixen dos botons per descarregar els fitxers .xlsx i .txt, aquesta part utilitza l'endpoint Resultats del backend. Veure figura **A.6.**
- Delete IOC: En aquesta pàgina es poden esborrar IOCs del sistema tant utilitzant un formulari com adjuntant un fitxer. Aquesta pàgina utilitza l'endpoint IOCs del backend i també renderitza una barra de progrés com en la pàgina Add IOC. Veure figura **A.7**.
- Misp Logger: En aquesta pàgina es pot veure una taula amb l'historial dels indicadors carregats i qui ho ha fet, també es pot filtrar per indicador o per usuari. Aquesta pàgina utilitza l'endpoint Logger del backend. Veure figura **A.8.**
- Usuaris: Quan l'usuari no ha iniciat sessió o l'ha tancat utilitzant el botó logout (apareix quan cliques la icona de perfil) es renderitza aquest menú, que permet tant iniciar sessió com crear un nou usuari. Aquesta pàgina utilitza l'endpoint Autenticació del backend. Veure figura **A.9.**

En el cas de que es trobi algun error al cridar l'API del backend, es renderitza en un quadre de text (veure figura **A.10**), d'aquesta forma el programa és a prova d'errors i dona feedback en tot moment.

Com que al programar-lo s'han utilitzat grids la web és responsive i s'adapta a la dimensió de qualsevol dispositiu canviant el seu aspecte, per exemple si es carrega des d'un mòbil la barra de navegació desapareix i és substituïda per un botó que desplega una barra lateral (veure figura **A.11**).

Per testejar l'IOC Manager s'han creat proves unitàries [33] que s'asseguren que cada una de les funcionalitats mencionades es comporta de forma esperada i no hi ha pèrdues de dades.

7.4 Docker Compose

L'arquitectura es desplega a través d'un fitxer docker-compose.yml, on s'especifiquen els containers que es crearan (IOC Manager Backend, IOC Manager Frontend, MISP Web, MISP Database i Score).

Com hem establert anteriorment la part del desplegament de MISP bé donada per misp-docker i l'única modificació que s'hi ha afegit son dos volums que es munten a les carpetes score-taxonomy i DecayingModelsFormulas del container MISP Web i que contenen la taxonomia score i la implementació del algorisme que s'ha discutit en l'apartat **7.1** i **7.2**.

Els altres containers es despleguen a partir d'un fitxer Dockerfile que conté cada un dels seus respectius mòduls i, com a variables d'entorn, se'ls passa la informació necessària per executar-los (clau de l'API de MISP, hostname on està ubicat MISP...) totes aquestes variables es configuren a un fitxer .env, de forma que es pot personalitzar el desplegament fàcilment.

Com que docker compose és compatible amb Microsoft Azure [34] i Amazon AWS [35] l'arquitectura també es pot desplegar al cloud de forma simple.

8 RESULTATS

Com s'ha comentat abans, les APIs de Virustotal i AbuseIPDB tenen limitacions de consultes diàries. En el cas de Virustotal la limitació és de 500 IOCs al dia i en el cas d'AbuseIPDB la limitació és de 1000 ips al dia. Degut a aquesta limitació no podem ficar un número il·limitat d'IOCs en el sistema perquè tardaria molt de temps a processar totes aquestes dades. Per evitar aquest problema

hem fet un estudi de les principals fonts OSINT [36] d'indicadors de ciberseguretat (Terme que es refereix a recollir informació de fonts de domini públic per a usos com ara recerca, anàlisi de seguretat o investigació privada) buscant aquelles que s'adapten millor al nostre sistema (un número no massa elevat d'IOCs i bona qualitat de forma que la majoria d'indicadors passin el filtre.

D'aquest anàlisi hem obtingut que les fonts que s'adapten mes a les nostres necessitats son el feed Botvrij [37], que prové de l'empresa de seguretat Cudeso, el feed ipsum [38], que és una llista pública d'ips amb mala reputació i el feed de Talos [39], que prové de la plataforma de cyber inteligència Talos de Cisco. Podeu trobar aquest anàlisi complet a la figura **A.12.**

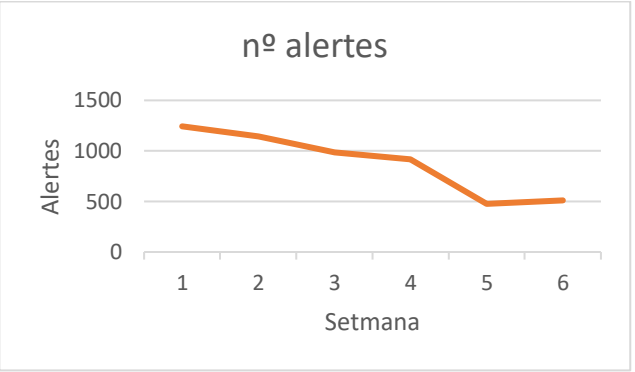
A part d'aquestes fonts OSINT també utilitzarem una de privada propietat de l'empresa que anomenarem CP IOC, que cada mes proporciona uns 200 IOCs i també els que provinguin del IOC Manager, el seu número pot variar segons la quantitat d'esdeveniments que es trobin i no es tindrà en compte en aquest anàlisi perquè el número d'indicadors que proveeix no és molt gran i en alguns casos pot saltar-se el filtre com hem comentat anteriorment.

Una vegada decidides les fonts d'intel·ligència les hem connectat al sistema i s'han començat a processar. Aquest procés ha durat un mes, podeu veure en la figura **A.13** el seu progrés durant aquestes 4 setmanes. Al final s'han ingestat 18965 IOCs dels quals 4673 no han passat el filtre, per tant un 25% d'aquests indicadors s'han eliminat i no es bloquejaran, a mes des del portal del IOC Manager s'han afegit uns 200 indicadors mes.

Durant les últimes 6 setmanes s'ha portat un registre del número d'alertes que es rebia en el SOC [40], que és des d'on es porta la gestió d'incidents de seguretat i el departament que opera l'eina IOC Manager. Durant aquest rang de temps el número d'incidents ha disminuït un 40% passant d'unes 1200 alertes a unes 500 per setmana. Aquest és un número molt significatiu i bàsicament duplica l'efectivitat del SOC al donar mes temps a gestionar les alertes abans que n'apareixen de noves i es millora el servei.

També cal dir que tot i que el sistema ha tingut un paper important en aquesta millora perquè bloquejant els indicadors les alertes que aquests generaven ja no es produïen, també s'han fet altres canvis en el SOC que han pogut influir en aquests números, com per exemple la millora de les regles de correlació que generen les alertes.

En aquesta gràfica es mostra aquesta millora:



Observeu que de la setmana 4 a la 5 hi ha hagut un gran canvi, això és degut a que els operadors del SOC van introduir un gran número d'indicadors relacionats amb incidents bastant recurrents (un total d'uns 200 indicadors com s'ha mencionat anteriorment). Al bloquejar-los s'ha evitat que es produeixi molt de soroll per la seva part. A partir d'ara no es preveu que aquest número canviï molt, ja que el soroll ja s'ha netejat i els pròxims indicadors que es bloquegin no tindran un impacte tant gran.

Durant aquest període de proba del sistema no s'ha bloquejat cap indicador que hagi suposat un problema per l'empresa (per exemple un indicador que podria portar problemes al bloquejar-lo és la ip de google). Això és una molt bona senyal ja que significa que el sistema de filtratge està funcionant correctament evitant aquests problemes.

9 CONCLUSIONS

En aquest projecte, s'ha desenvolupat un sistema per a la ingesta i emmagatzematge d'indicadors de ciberseguretat (IOCs) amb l'objectiu de millorar la seguretat de les empreses i reduir el nombre d'alertes falses. La solució proposada es basa en una plataforma MISP existent, estenent les seves funcionalitats per incloure mòduls addicionals com a filtratge dels IOCs i control del sistema.

Després d'un mes de prova, els resultats han demostrat que el sistema ha tingut un impacte positiu en la reducció del nombre d'alertes falses al SOC, disminuint un 40% de les alertes setmanals. Aquesta millora ha permès als operadors del SOC gestionar més eficientment les alertes i concentrar-se en els problemes reals.

Una de les claus d'aquest èxit ha estat l'ús de fonts OSINT seleccionades curosament per a la ingesta d'IOCs, combinant-les amb el sistema IOC Manager desenvolupat per permetre als operadors introduir indicadors manualment i controlar el temps de vida dels IOCs. El filtratge s'ha realitzat utilitzant APIs com Virustotal i AbuseIPDB, proporcionant una puntuació que indica la perillositat dels indicadors.

El sistema és modular i personalitzable segons les necessitats específiques del client, facilitant-ne la implementació en diferents entorns empresarials. A més,

gràcies a l'ús de Docker Compose, es pot desplegar fàcilment tant en infraestructures locals com en cloud (com Microsoft Azure o Amazon AWS).

Tot i que durant el període de prova no s'ha bloquejat cap indicador crític per error (per exemple: IP's importants), seria interessant continuar monitoritzant el rendiment del sistema a llarg termini per assegurar-se que segueix funcionant correctament sense generar problemes addicionals.

Per tant, aquest projecte demostra que és viable crear un sistema eficient basat en MISP per gestionar indicadors cibernètics i millorar significativament la seguretat informàtica dins una empresa. Les millores en la detecció i bloqueig d'amenaces cibernètiques poden portar a un augment de la confiança dels clients en l'empresa, ja que demostra una actitud proactiva per protegir les seves dades i recursos digitals i és produeix un bon valor afegit molt útil alhora de renegociar contractes amb els clients.

10 BIBLIOGRAFIA

[1] Logsign, «What is IOC in Cyber Security?,» [En línia]. Available: https://www.logsign.com/blog/what-is-ioc-in-cyber-security/.

[2] Cyberproof, «Managed threat intelligence,» [En línia]. Available: https://www.cyberproof.com/cyber-101/managed-threat-intelligence/.

[3] Microsoft, «¿Qué es SIEM?,» [En línia]. Available: https://www.microsoft.com/es-es/security/business/security-101/what-is-siem?SilentAuth=1.

[4] Trend Micro, «¿Qué es EDR?,» [En línia]. Available: https://www.trendmicro.com/es_es/what-is/xdr/edr.html.

[5] Cisco, «What is a Firewall?,» [En línia]. Available: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html.

[6] Wikipedia, «Informàtica en el núvol,» [En línia]. Available: https://ca.wikipedia.org/wiki/Inform%C3%A0tica_en_n%C3%BAvol.

[7] Threat Connect, «ThreatConnect,» [En línia]. Available: https://threatconnect.com/.

[8] Peerspot, «Best Threat Intelligence Platforms,» [En línia]. Available: https://www.peerspot.com/categories/threat-intelligence-platforms.

[9] MISP, «MISP - Threat Intelligence Sharing Platform,» [En línia]. Available: https://github.com/MISP/MISP.

[10] Filigran, «OpenCTI - Introduction,» [En línia]. Available: https://filigran.notion.site/Introduction-f99633ba66ba4ee3af1a4d832208dc99.

[11] Microsoft, «Github,» [En línia]. Available: https://github.com/.

[12] L. Losoviz, «Monorepo VS Multi-Repo,» [En línia]. Available: https://kinsta.com/es/blog/monorepo-vs-multi-repo/.

[13] Atlassian, «Jira,» [En línia]. Available: https://www.atlassian.com/es/software/jira.

[14] Microsoft, «VSCode,» [En línia]. Available: https://code.visualstudio.com/.

[15] Microsoft, «Python - Visual Studio Marketplace,» [En línia]. Available: https://marketplace.visualstudio.com/items?itemName=ms-python.python.

[16] H. Hattori, «Autopep8,» [En línia]. Available: https://github.com/hhatto/autopep8.

[17] Docker, «Docker Desktop,» [En línia]. Available: https://www.docker.com/products/docker-desktop/.

[18] MISP, «misp-docker,» [En línia]. Available: https://github.com/MISP/misp-docker.

[19] MISP, «Decaying of Indicators,» [En línia]. Available: https://www.misp-project.org/2019/09/12/Decaying-Of-Indicators.html/.

[20] CIRCL, «Sightings,» [En línia]. Available: https://www.circl.lu/doc/misp/sightings/.

[21] MISP, «Best practices in threat intelligence,» [En línia]. Available: https://www.misp-project.org/best-practices-in-threat-intelligence.html#_glossary.

[22] Chronicle Security, «VirusTotal - Home,» [En línia]. Available: https://www.virustotal.com/gui/home/search.

[23] AbuseIPDB, «AbuseIPDB,» [En línia]. Available: https://www.abuseipdb.com/.

[24] MISP, «MISP taxonomy format,» [En línia]. Available: https://www.misp-standard.org/rfc/misp-standard-taxonomy-format.html.

[25] Python, «concurrent.futures - Lanzamiento de tareas paralelas,» [En línia]. Available: https://docs.python.org/es/3/library/concurrent.futures.html.

[26] Pallets, «Welcome to Flask,» [En línia]. Available: https://flask.palletsprojects.com/en/2.3.x/.

[27] R. Vinot, «Welcome to PyMISP's documentation!,» [En línia]. Available: https://pymisp.readthedocs.io/en/latest/.

[28] L. A. Gilbert, «Flask-JWT-Extended's Documentation,» [En línia]. Available: https://flask-jwt-extended.readthedocs.io/en/stable/.

[29] Facebook, «React,» [En línia]. Available: https://react.dev/.

[30] Microsoft, «TypeScript: JavaScript with syntax for types,» [En línia]. Available: https://www.typescriptlang.org/.

[31] Twitter, «Bootstrap,» [En línia]. Available: https://getbootstrap.com/.

[32] react-bootstrap, «React Bootstrap,» [En línia]. Available: https://react-bootstrap.github.io/.

[33] Viquipèdia, «Proves unitàries,» [En línia]. Available: https://ca.wikipedia.org/wiki/Proves_unit%C3%A0ries.

[34] Microsoft, «Microsoft Azure,» [En línia]. Available: https://azure.microsoft.com/.

[35] Amazon, «Amazon AWS,» [En línia]. Available: https://aws.amazon.com/.

[36] Incibe, «OSINT - La información es poder,» [En línia]. Available: https://www.incibe.es/incibe-cert/blog/osint-la-informacion-es-poder.

[37] Cudeso, «botvrij.eu - powered by MISP,» [En línia]. Available: https://www.botvrij.eu/.

[38] M. Stampar, «ipsum - daily feed of bad IPs,» [En línia]. Available: https://github.com/stamparm/ipsum.

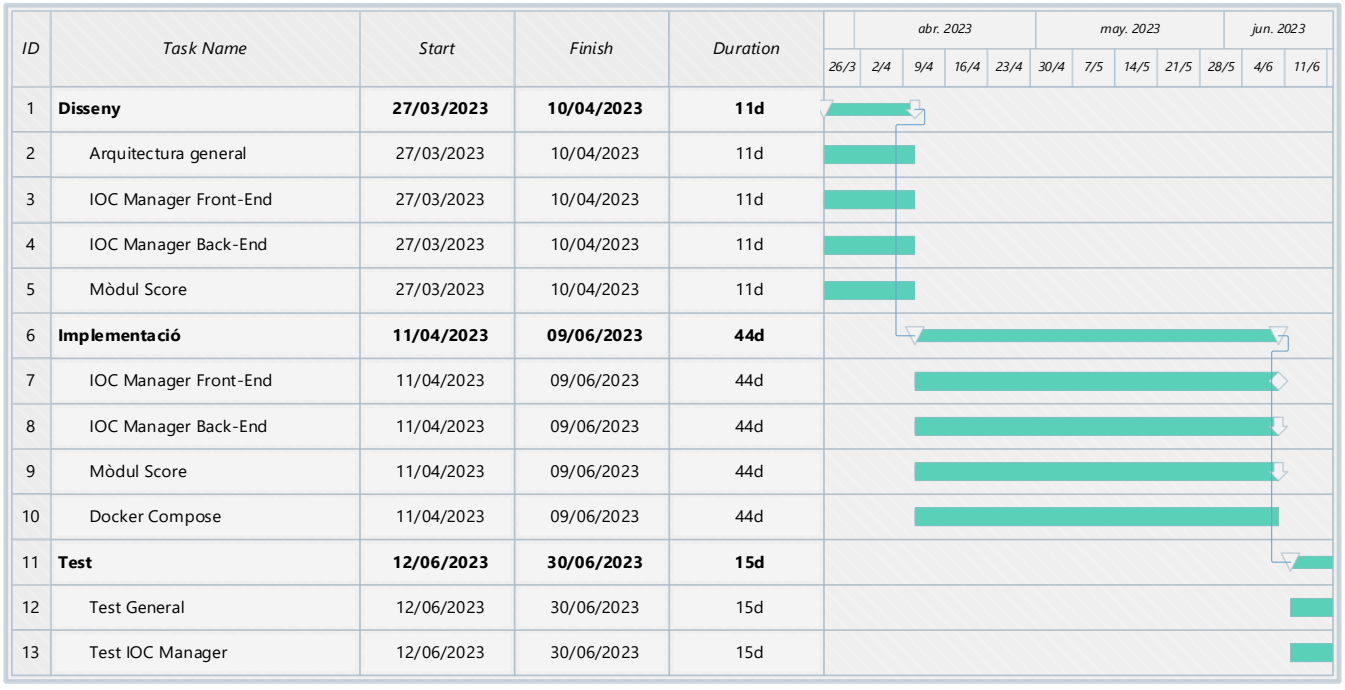
[39] Cisco, «Cisco Talos Intelligence group,» [En línia]. Available: https://www.talosintelligence.com/.

[40] Oracle, «¿Que es un SOC?,» [En línia]. Available: https://www.oracle.com/es/database/security/que-es-un-soc.html.

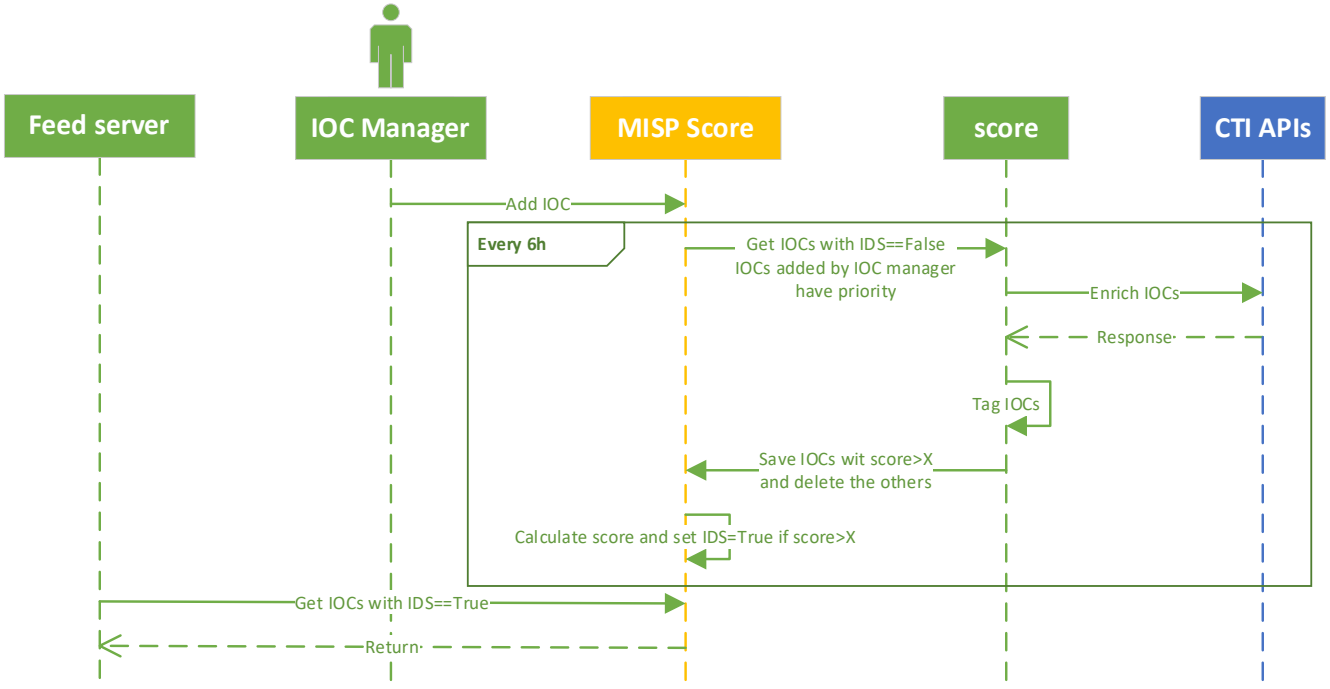
11 APÈNDIX

A.1 Diagrama de Gantt

ID	Task Name	Start	Finish	Duration
1	**Disseny**	**27/03/2023**	**10/04/2023**	**11d**
2	Arquitectura general	27/03/2023	10/04/2023	11d
3	IOC Manager Front-End	27/03/2023	10/04/2023	11d
4	IOC Manager Back-End	27/03/2023	10/04/2023	11d
5	Mòdul Score	27/03/2023	10/04/2023	11d
6	**Implementació**	**11/04/2023**	**09/06/2023**	**44d**
7	IOC Manager Front-End	11/04/2023	09/06/2023	44d
8	IOC Manager Back-End	11/04/2023	09/06/2023	44d
9	Mòdul Score	11/04/2023	09/06/2023	44d
10	Docker Compose	11/04/2023	09/06/2023	44d
11	**Test**	**12/06/2023**	**30/06/2023**	**15d**
12	Test General	12/06/2023	30/06/2023	15d
13	Test IOC Manager	12/06/2023	30/06/2023	15d



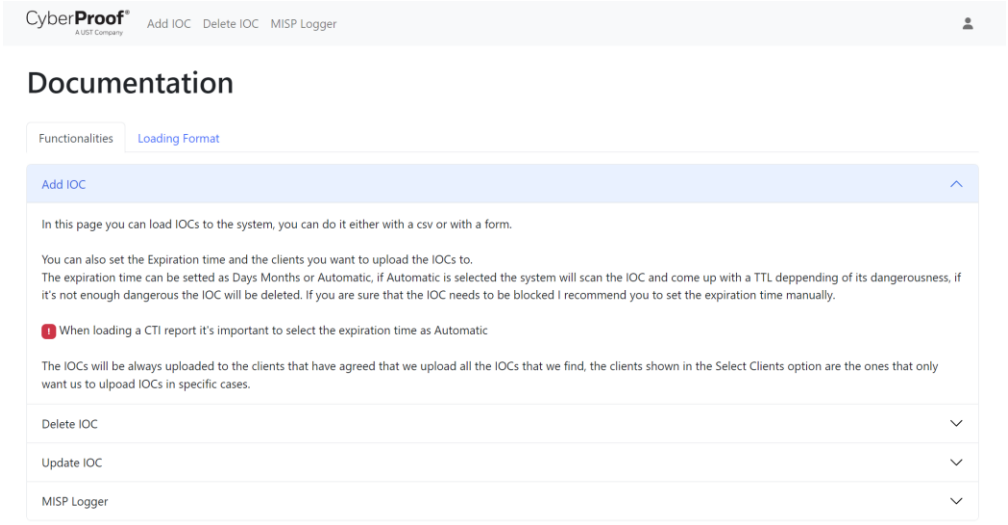
A.2 Diagrama UML Sequence



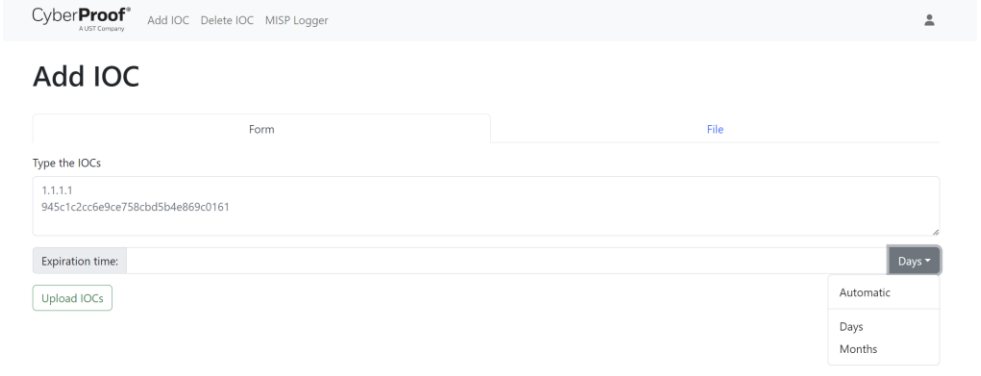
A.3 Algoristme decaying score

Mètrica	Camps	Condicions
Sighting Info	Status (FP - False positive o True positive)	FP resta al score final, TP suma al score final.
	Confidence Level	Multiplica el augment calculat amb el status, per tant com mes confiança hi hagi mes es veurà reflectit en el score final.
	Regió on s'ha observat el IOC	Si és la mateixa on està localitzat el client, augmenta el score final
Sighting date	Last observation (decaying score)	Com mes temps fa que s'ha observat mes baixa el score.
Score	AbuseIPDB score	Es fa la mitjana entre els scores i s'utilitza com a score base, que serà modificat per els sightings
	Virustotal score	
	...	

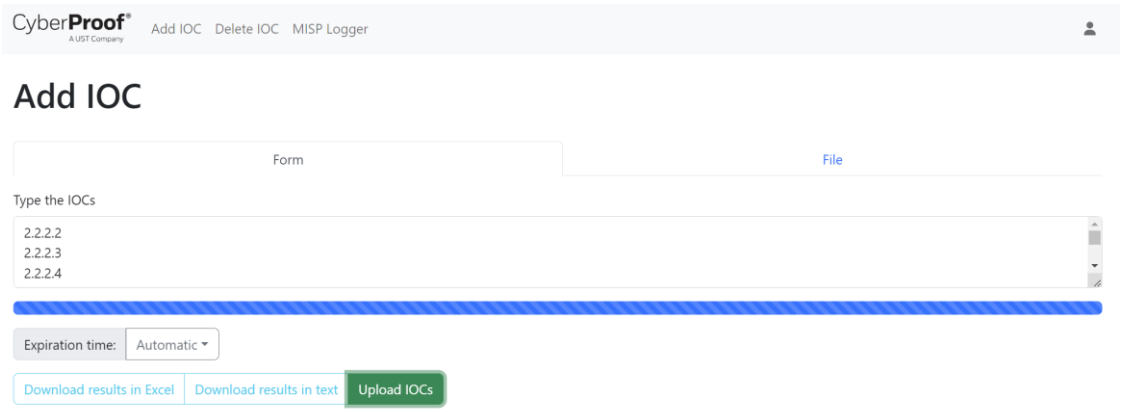
A.4 Frontend - Documentació



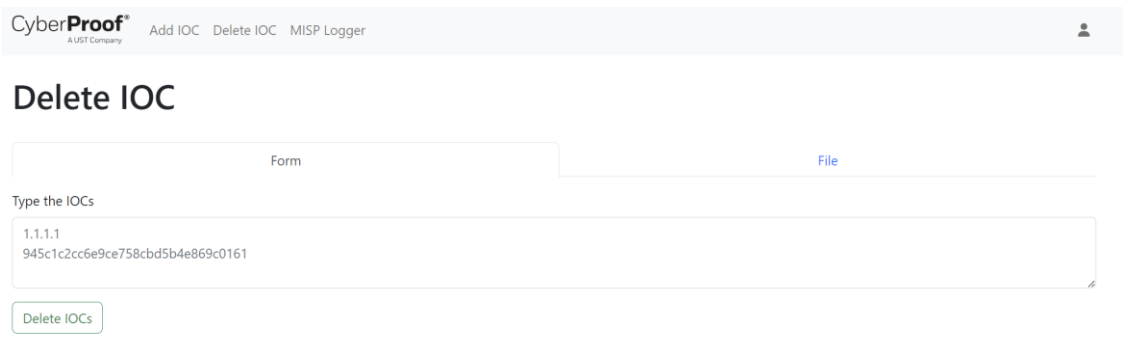
A.5 Frontend - Add IOC



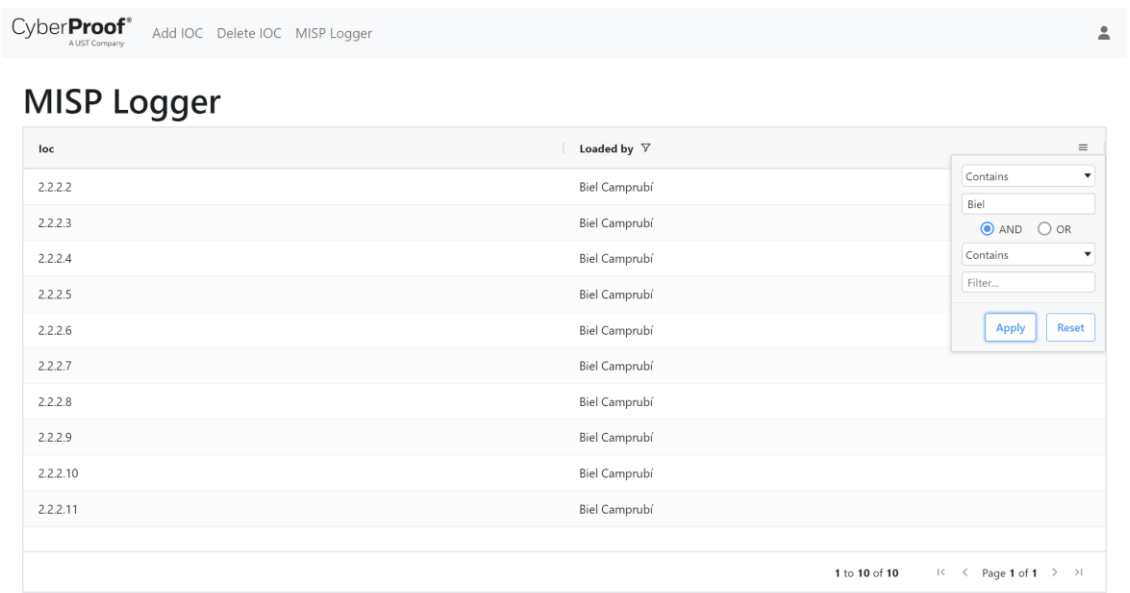
A.6 Frontend - Càrrega completa



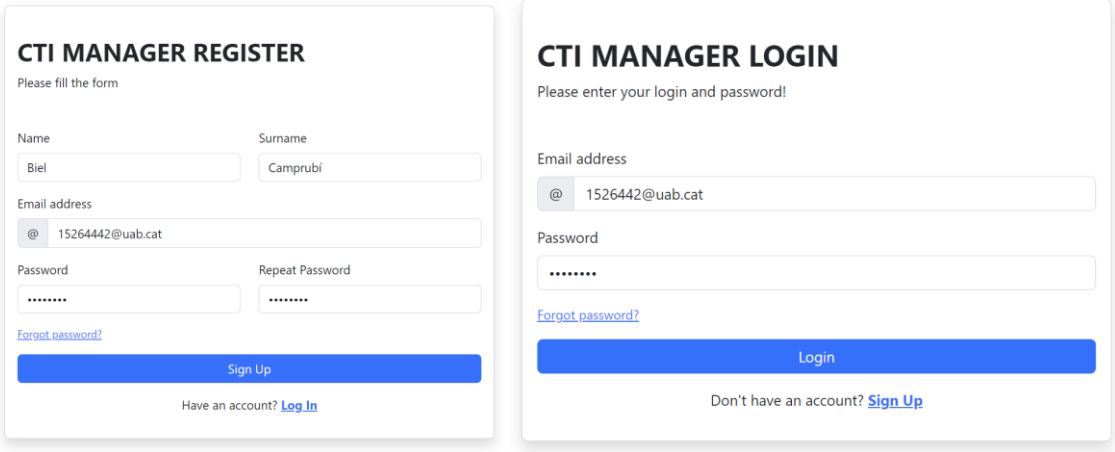
A.7 Frontend - Delete IOC



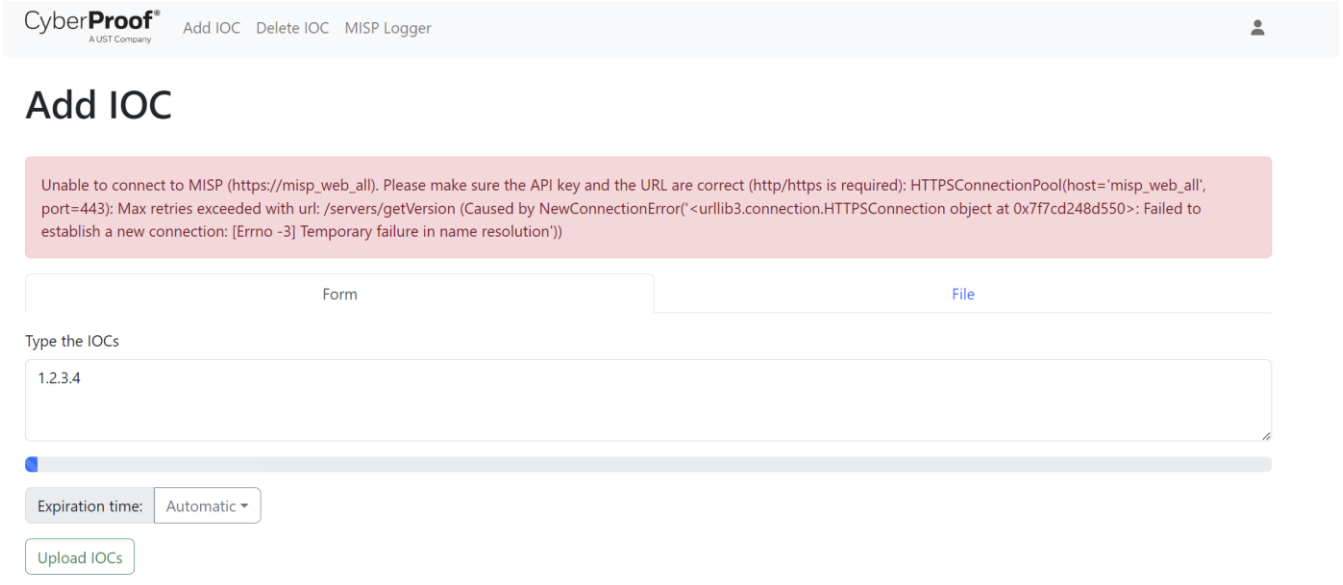
A.8 Frontend - MISP Logger



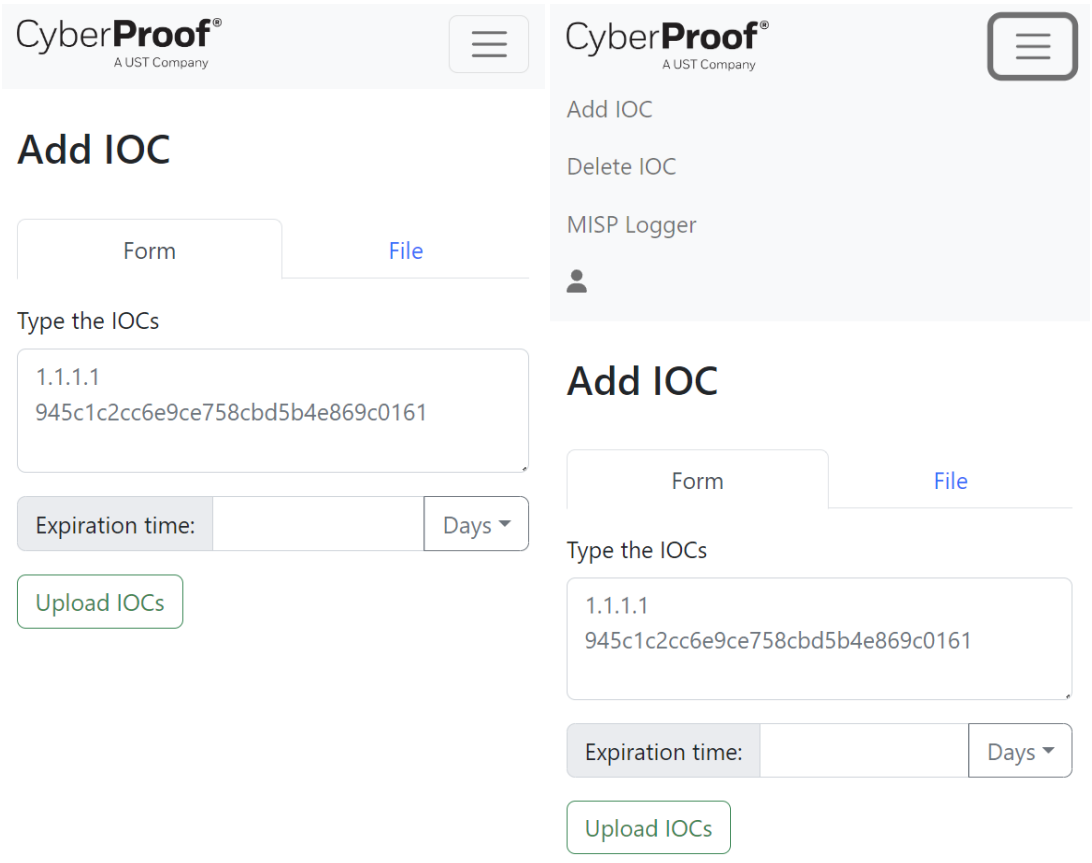
A.9 Frontend - Gestió d'usuaris



A.10 Frontend - Renderitzat d'errors



A.11 Frontend - Disseny responsive



A.12 Anàlisi fonts OSINT

FEED	# IP	#MD5	ANÀLISI
CIRCL OSINT Feed - CIRCL - feed format: misp	31.620		DESCARTAT
The Botvrij.eu Data - Botvrij.eu - feed format: misp	1.311	2.467	SELECCIONAT
DigitalSide Threat-Intel OSINT Feed - osint.digitalside.it - feed format: misp	29.588.181		DESCARTAT
IPsum (aggregation of all feeds) - level 6 - no false positives - IPsum - feed format: freetext	3.584		SELECCIONAT
IPsum (aggregation of all feeds) - level 7 - no false positives - IPsum - feed format: freetext	1.740		SELECCIONAT
Psum (aggregation of all feeds) - level 8 - no false positives - IPsum - feed format: freetext	208		SELECCIONAT
Malware Bazaar - abuse.ch - feed format: csv		617625	DESCARTAT
CISCOtalosIpBlacklist	791		SELECCIONAT

A.13 Resultat ingesta de dades

DATA					01/05/2023	08/05/2023	15/05/2023	22/05/2023	
FEED	**The Botvrij.eu Data**	**TEMPS DE VIDA (MESOS)**	**ENRIQUIMENT**	**# IOC / SETMANA**	**WEEK 1**	**WEEK 2**	**WEEK 3**	**WEEK 4**	**TOTAL**
# IP	1311	2	AbuseIPDB	7000	1311	0			1311
# DOMAIN	2971	2	VirusTotal	3500	2971				2971
# URL	715	2	VirusTotal	3500	529	186			715
# MD5	2467	6	VirusTotal	3500		2467			2467
# SHA1	1304	6	VirusTotal	3500		847	457		1304
# SHA256	3674	6	VirusTotal	3500			3043	631	3674
TOTAL	**12442**				**4811**	**3500**	**3500**	**631**	**12442**

FEED	IPsum	**TEMPS DE VIDA (MESOS)**	**ENRIQUIMENT**	**# IOC / SETMANA**	**WEEK 1**	**WEEK 2**	**WEEK 3**	**WEEK 4**	**TOTAL**
# IP - level 6	3584	2	AbuseIPDB	7000	3584				3584
# IP - level 7	1740	2	AbuseIPDB	7000	1740				1740
# IP - level 8	208	2	AbuseIPDB	7000	208				208
TOTAL	**5532**				208	0	0	0	5532

FEED	CP IOC MAIG 2023	**TEMPS DE VIDA (MESOS)**	**ENRIQUIMENT**	**# IOC / SETMANA**	**WEEK 1**	**WEEK 2**	**WEEK 3**	**WEEK 4**	**TOTAL**
# IP	3	2	AbuseIPDB	7000	3				3
# DOMAIN	1	2	VirusTotal	3500	1				1
# HASH	196	2	VirusTotal	3500	208				208
TOTAL	**200**				208	0	0	0	212

FEED	CISCOtalosIpBlacklist	**TEMPS DE VIDA (MESOS)**	**ENRIQUIMENT**	**# IOC / SETMANA**	**WEEK 1**	**WEEK 2**	**WEEK 3**	**WEEK 4**	**TOTAL**
# IP	791	2	AbuseIPDB	7000	154	637			791
TOTAL	**791**								791

TOTAL #IP	**7637**
TOTAL	**18965**