Universitat de Girona
Escola Politècnica Superior

Projecte - Treball final de carrera

Estudi: EINF

Títol: Sistema de captura i anàlisi de trànsit Wi-Fi orientat a la detecció i a la resposta d'incidents de seguretat

Document: Memòria

Alumne: Pau Ochoa Cañigueral

Tutor: Lluís Fàbrega Soler / Antonio Bueno Delgado
Departament: Arquitectura i Tecnologia de Computadors
Àrea: Arquitectura i Tecnologia de Computadors

Convocatòria: Setembre / 2016

Agraïments

A l'Antonio Bueno i Lluís Fàbrega i Soler per la seva orientació, ajuda, recolzament i tracte durant tot el desenvolupament del projecte.

A totes aquelles persones que m'he trobat al llarg dels meus estudis i carrera professional per enriquir-me i fer-me millorar, d'una manera o d'una altra.

A la Cristina i en Roc, per la seva paciència, recolzament incondicional i per tot el que signifiquen per a mi.

Moltes gràcies.

Índex

Índex de figures

Índex de taules

1 Introducció

Aquest primer capítol serveix per situar el marc del projecte, s'estableixen les raons per les quals s'ha decidit desenvolupar un sistema de monitoratge de xarxes Wi-Fi orientat a incidents de seguretat, i què s'espera del mateix.

També s'ha inclòs en aquest capítol una petita guia amb l'estructura d'aquesta memòria, per facilitar-ne la lectura i comprensió.

1.1 Motivació

La popularitat de les xarxes Wi-Fi és innegable: és estrany el dia en el que no les fem servir o veiem a altres interactuar amb les mateixes tant a l'àmbit professional com al personal. De fet no és estrany veure a persones que el primer que fan és demanar la contrasenya de la xarxa Wi-Fi al accedir a un establiment o lloc públic. Aquesta tecnologia de connectivitat de xarxa està tan estesa que fins i tot és la única opció de connectivitat per defecte de molts dispositius.

En conseqüència les xarxes Wi-Fi formen part de la majoria d'infraestructures de xarxes i sistemes actuals. Fins i tot en aquells casos en els quals no estan oficialment implantades, donat que molts dispositius permeten la creació de xarxes Wi-Fi de forma fàcil i senzilla.

És habitual que els administradors de sistemes i responsables d'infraestructures disposin d'eines i mecanismes per a monitorar-ne l'estat, tant pel que respecta al seu funcionament i rendiment, com a la seguretat. Però no és tant habitual el monitoratge específic de les xarxes Wi-Fi sinó que sovint s'integren dins els elements monitorats a altres nivells, com per exemple a nivell de comunicacions IP i capes superiors. Existeixen amenaces que treballen específicament als nivells definits pels protocols IEEE 802.11, que regeixen l'estàndard Wi-Fi, i que de no realitzar-ne un monitoratge específic, poden passar desapercebudes a les capes superiors. Com per exemple atacs de suplantació de punts d'accés que poden posar en risc als usuaris de la xarxa legítima.

La majoria de sistemes de monitoratge Wi-Fi es centren en la disponibilitat i el rendiment, i sovint es tracta de solucions tancades i propietàries que van íntimament lligades amb les solucions de maquinari ofertes per diferents grans fabricants. A banda de tenir costos prohibitius o requerir la utilització d'equipament de xarxa d'un fabricant concret. És per aquest motiu que s'ha decidit afrontar aquest projecte, per intentar aportar solucions genèriques i obertes al monitoratge de xarxes Wi-Fi, fent èmfasi en la detecció i resposta d'incidents de seguretat.

1.2 Objectius i abast

L'objectiu d'aquest projecte és dissenyar i implementar un sistema de captura, emmagatzematge i anàlisi de trànsit Wi-Fi. Aquest sistema té com a principal propòsit donar suport i facilitar la detecció i la resposta d'incidents de seguretat. Però tot i estar orientat a aquest àmbit, es vol desenvolupar un sistema prou flexible com per a permetre cobrir altres necessitats que requereixen el monitoratge de trànsit Wi-Fi. Per tant més que ser un sistema dissenyat per a un propòsit concret es tracta d'un entorn de treball o *framework* obert del qual se'n podran derivar solucions específiques.

Per assolir aquest objectiu cal:

- Estudiar les particularitats del trànsit Wi-Fi
- Avaluar el format del trànsit i la forma de capturar-lo
- Dissenyar i implementar components i processos, tant a nivell de maquinari com de programari, que permetin realitzar la captura de trànsit.
- Dissenyar i implementar els components i processos necessaris per processar i emmagatzemar de forma adient el trànsit capturat.
- Dissenyar i implementar mecanismes que permetin analitzar el trànsit emmagatzemat i extreure'n informació rellevant per al propòsit escollit.

1.3 Estructura d'aquesta memòria

L'estructura d'aquesta memòria presenta el projecte desenvolupat utilitzant diferents vistes, a on els capítols principals es corresponen amb vistes concurrents, que el descriuen segons diferents punts de vista[*]:

[*] Aquesta aproximació s'inspira en el que es coneix com el model de vistes 4+1 per a l'arquitectura de programari dissenyat per *Philippe Kruchten*: http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf

- Capítol 3 Anàlisi: Aquest apartat recull les necessitats i requisits del sistema de monitoratge de xarxes Wi-Fi desenvolupat en el marc d'aquest projecte. Així com d'altres apartats relacionats amb les fases d'anàlisis, la metodologia emprada i la planificació del projecte. El punt de vista principal és el d'un perfil similar al d'un gestor de projectes.
- Capítol 4 Disseny: Descriu l'estructura i funcionalitat del sistema de monitoratge desenvolupat. S'inclou informació rellevant dels principals elements que composen el sistema i com es comuniquen entre sí. Es tracta per tant d'una vista enfocada al punt de vista d'un arquitecte del software i/o d'aquelles persones que vulguin entendre el funcionament general del sistema.
- Capítol 5 Implementació: S'aprofundeix en els detalls d'implementació, integració i/o configuració dels components del sistema. Es detalla el sistema de monitoratge des de la perspectiva d'un perfil tècnic, per alguns components més pròpiament d'un desenvolupador i per d'altres d'un operador.
- Capítol 6 Implantació: La informació continguda en aquest apartat està orientada al punt de vista d'un enginyer de sistemes o operador encarregat d'implantar o desplegar una solució basada en el sistema desenvolupat.

El Capítol 2 Conceptes previs, inclou una pinzellada dels estàndards, tecnologies i conceptes clau necessaris per comprendre les particularitats i problemàtiques de monitorar el trànsit Wi-Fi i, en conseqüència la solució proposada en aquest projecte. No es pretén explicar aquests conceptes en detall, donat que el propòsit principal d'aquesta memòria és el de documentar la solució implementada. A més a més existeixen gran quantitat d'excel·lents recursos i bibliografia a on es tracten en profunditat. A l'apartat 8 Bibliografia es recullen alguns dels mateixos que s'han consultat durant aquest projecte.

En general s'ha intentat separar en annexes tota aquella informació més detallada, com configuracions, formats de dades i d'altres, que podrien dificultar la lectura de la memòria.

Tot i haver-se intentat fer capítols independents i auto continguts es recomana la lectura total i seqüencial de la memòria per una millor comprensió del projecte i la solució de captura, emmagatzemament i anàlisi de trànsit Wi-Fi desenvolupada.

Finalment cal destacar que aquest projecte no es correspon amb un projecte només de gestió, tal i com es descriu a la *Guia dels projectes/treballs de final de carrera de les enginyeries informàtiques*[1]. I tot i ser més proper a la descripció de projecte de sistemes, tampoc hi encaixa al 100%. És per això que tot hi intentar respectar tots els apartats de la memòria que s'especifiquen a la guia citada anteriorment, no tots s'han adaptat a la naturalesa d'aquest projecte amb el mateix títol o als mateixos aparats que els que s'especifiquen a la guia.

2 Conceptes previs

Aquest capítol descriu aquells conceptes imprescindibles per poder fer un bon seguiment d'aquest projecte i de la seva memòria. S'inclou també l'apartat marc de treball on es menciona l'equip de persones que han tingut relació amb el desenvolupament del projecte.

2.1 Wi-Fi i estàndards IEEE 802.11

Wi-Fi és el nom comercial, i més conegut, de la implementació del conjunt d'estàndards IEEE 802.11[2][2] que descriuen les xarxes d'àrea local sense fils o WLANs. Actualment existeixen més de 15 especificacions diferents, a on es detallen característiques físiques especifiques i/o que suposen millores o correccions a l'estàndard original.

Tot i la magnitud d'aquest conjunt d'estàndards en essència es resumeix a detallar les especificacions necessàries per implementar xarxes sense fils d'àrea local. Per a fer-ho s'especifiquen la subcapa de Control d'Accés al Medi, coneguda com a MAC de l'anglès *Medium Access Control*, i la capa física o PHY, abreviatura de *Physical Layer*.

La subcapa MAC és la més baixa dins la capa d'enllaç o capa 2, en el model OSI[3] de 7 capes. Aquesta subcapa proveeix l'adreçament i els mecanismes de control d'accés al canal que fan possible que diferents elements de xarxa es comuniquin dins una xarxa d'accés múltiple. Actua com a interfície entre la subcapa de control d'enllaç lògic (*Logical Link Control* o LLC) i la capa física (PHY) de la xarxa.

[2] IEEE 802.11: Wireless LANs - http://standards.ieee.org/about/get/802/802.11.html
[3] OSI: *Open Systems Interconnection* - http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_ics_browse.htm?ICS1=35&ICS2=100

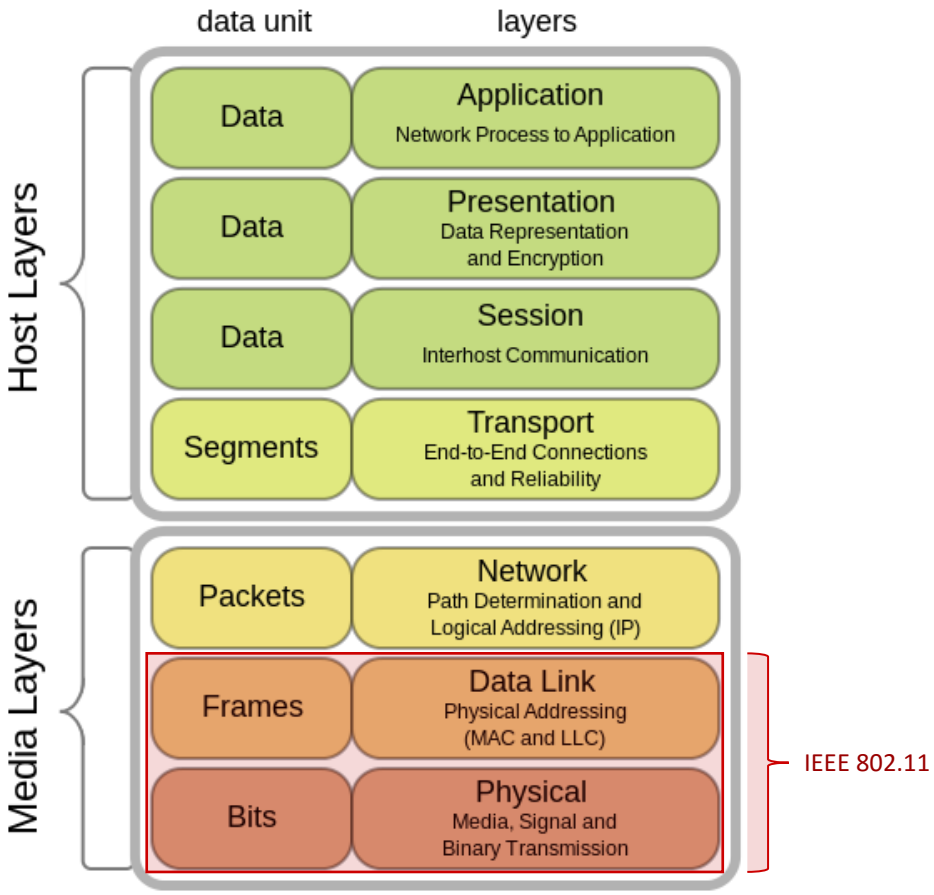


Figura 2-1 IEEE 802.11 en el model OSI

(font de la imatge original: https://commons.wikimedia.org/wiki/File:OSI_Model_v1.svg)

2.1.1 Components bàsics

A molt alt nivell les xarxes basades en els estàndards IEEE 802.11 estan formades principalment per quatre elements:

- Medi sense fils: Per a poder transmetre paquets entre estacions es necessita un medi. IEEE 802.11 defineix diferents capes físiques per suportar el MAC 802.11, utilitzant freqüències radio elèctriques (RF) i infrarojos. A la pràctica només es fan servir les primeres.
- Estacions: Les xarxes són creades amb la finalitat d' intercanviar dades entre estacions. En aquest cas, una estació és un aparell equipat amb una targeta de xarxa sense fils (o més d' una). Actualment hi ha dispositius de tota mena que actuen com estacions, des d'ordinadors portàtils, mòbils i tauletes, fins a impressores, càmeres de seguretat, electrodomèstics, cotxes, etc.

- Punts d'accés: També coneguts com *Acces Points* o APs per abreujar. Les trames 802.11 tenen que ser convertides a altres tipus de trames o paquets si es vol que els elements de la xarxa es puguin comunicar amb altres xarxes, com per exemple una xarxa Ethernet convencional o sistemes que es troben a Internet. Els punts d' accés s' encarreguen, entre d' altres, d'aquesta tasca d'interconnexió. També actuen com a element central per a facilitar la comunicació entre estacions.
- Sistema de distribució: Quan es connecten diversos punts d'accés per a crear àrees de cobertura majors, aquests tenen que comunicar-se amb els demés per a poder "seguir" el moviment de les estacions mòbils (*roaming*). Aquest Sistema de Distribució és el component lògic del 802.11 emprat per a redireccionar les trames a la seva destinació. L' estàndard no defineix cap tecnologia en particular pel sistema de distribució, però a la pràctica aquest és implementat com una combinació d' un sistema de pont (*bridging*) i un medi de sistema de distribució, que és la xarxa principal o *backbone* utilitzada per a transmetre trames entre punts d' accés. Sovint aquesta xarxa principal es correspon amb una xarxa Ethernet.

2.1.2 Topologies de xarxa

Basic Service Set

La topologia de xarxa bàsica de les xarxes sense fils d'àrea local és la que s'anomena *Basic Service Set* o BSS, a on un conjunt d' estacions es comuniquen entre elles. Les comunicacions tenen lloc a una àrea definida per la característica de propagació del medi, anomenada àrea de servei bàsica (*basic service area*).

Aquesta topologia bàsica es pot implementar de dues maneres:

- Xarxa independent: També coneguda com *Ad-Hoc* o *Peer to Peer*.
- Xarxa d'infraestructura: També coneguda com a *Managed*.

Xarxa independent o Ad-Hoc

A les xarxes independents (també conegudes com IBSS de *Independent Basic Service Set*) les estacions es comuniquen les unes amb les altres directament, per tant totes tenen que estar dins un rang que permeti totes les comunicacions possibles entre elles.

Figura 2-2 Xarxa independent o *Ad-Hoc*

La xarxa 802.11 més petita possible és una IBSS formada per dues estacions. Normalment aquest tipus de xarxes són xarxes petites formades per a un objectiu específic i durant breus períodes de temps (compartir algun arxiu en una modesta conferencia per exemple).

Xarxa d'infraestructura o Managed

Les xarxes d'infraestructura es distingeixen per la presència de punts d' accés.

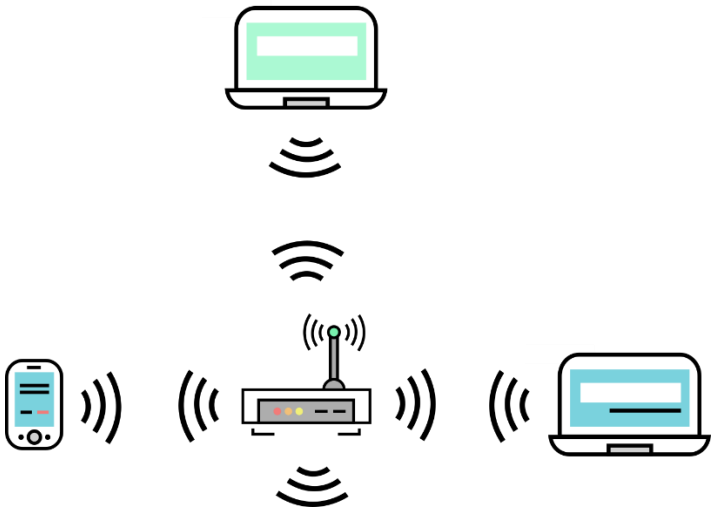


Figura 2-3 Xarxa infraestructura o *Managed*

Els punts d' accés són emprats per a totes les comunicacions en aquest tipus de xarxa. Si una estació que forma part d' una xarxa BSS en mode infraestructura necessita comunicar-se amb una altra estació, la comunicació es fa en dos salts. En primer lloc, l'estació transmet la trama al punt d' accés i tot seguit aquest la retransmet a l'estació de destí. Tot i que les comunicacions passen pel punt d' accés i requereixen per tant més capacitat de transmissió, el mode infraestructura té dos avantatges principals :

- La cobertura de la xarxa esta definida per la distància al punt d' accés. Totes les estacions han d'estar dins el rang de cobertura de l' AP, però no cal que ho estiguin en el de tota la resta. Tot i fer perdre una mica d'eficiència en la velocitat de les transmissió, aquest mode ofereix molta flexibilitat alhora de dissenyar la xarxa.

- Un altre punt a favor és que permet que les estacions es posin en mode d'estalvi amb més facilitat. Els punts d' accés poden emmagatzemar paquets dirigits a una estació que estigui actualment estalviant energia, i un cop el buffer d' emmagatzematge estigui ple transmetre les trames. D'aquesta forma les estacions (que sovint operen amb algun tipus de bateries) poden posar a "dormir" les seves interfícies de xarxa i "despertar-los" només per a transmetre i rebre totes les trames del buffer de cop.

En aquesta tipologia de xarxa les estacions s' han d' associar amb el punt d'accés per a poder fer ús dels serveis de la xarxa. L' associació és el procés en el qual una estació entra a formar part d' una xarxa 802.11. A nivell lògic és l'equivalent a connectar el cable de xarxa a una xarxa Ethernet a nivell físic. És important tenir present que una estació amb una sola interfície sense fils tan sols pot estar associada a un punt d'accés en un moment donat. Per contra, en principi no hi ha límit teòric del nombre d'estacions que poden estar associades a un AP, tot i que a la pràctica, sí que existeix aquest límit degut a les especificacions del maquinari i a altres limitacions físiques.

Extended Service Set

La topologia Basic Service Set pot oferir cobertura a una zona física notable però no pot oferir la possibilitat de crear una xarxa de grans dimensions que cobreixi una zona geogràfica considerable.

Per aquest propòsit l'estàndard IEEE 802.11 defineix els mecanismes per unir diversos BSS formant el que es coneix com a Extended Service Set o ESS.

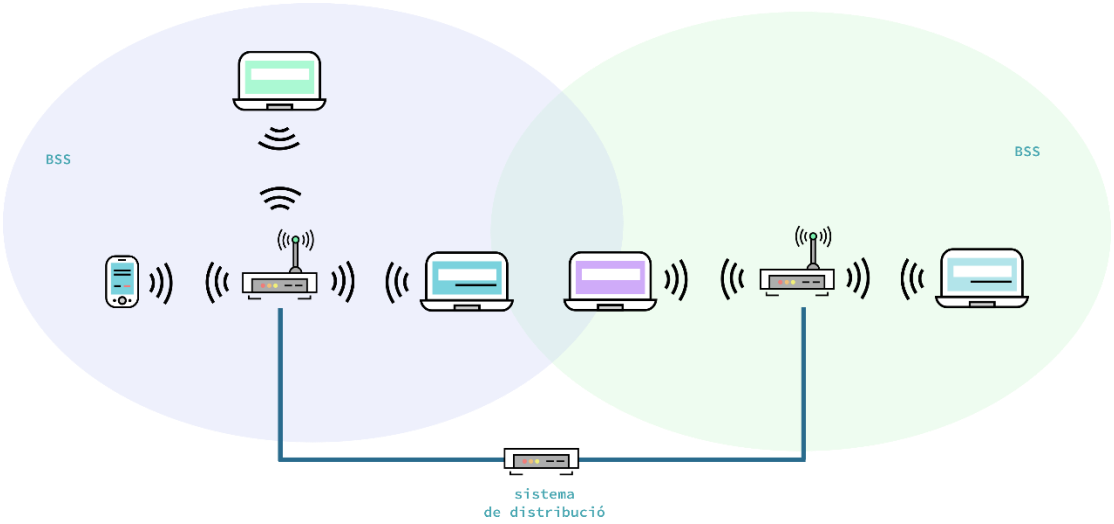


Figura 2-4 Extended Service Set o ESS

En aquest cas cada punt d'accés de cadascun dels BSS haurà de ser configurat per a formar part del mateix ESS i connectar-se al mateix sistema de distribució. Les estacions

que estiguin dins el rang d'algun punt d'accés que formi part d'aquest ESS es podran comunicar amb la resta d'estacions, encara que aquestes no estiguin al mateix BSS.

2.1.3 Capa física i espectre radioelèctric

La capa física de qualsevol xarxa és aquella que defineix la modulació i la senyalització utilitzada per a les transmissions de dades. L'IEEE 802.11 defineix tres possibles opcions per a la capa física, dos basades en freqüències de radio:

- Espectre expandit per seqüència directa o DSSS (*Direct Sequence Spread Spectrum*)
- Espectre expandit per salt de freqüències o FHSS (*Frequence Hopping Spread Spectrum*)

I una darrera basada en llum infraroja, tot i que a la pràctica no existeixen implementacions, i ha passat a ser més aviat una anècdota del estàndard original que no una opció utilitzada habitualment:

- Llum infraroja en banda base, sense modular

Freqüències de radio

Tot i que existeixen principalment dos tipus diferents de tecnologia que fan servir la radiofreqüència, la banda "estreta" i la banda "ampla", és aquesta última, també anomenada d'espectre expandit, la més utilitzada.

La tecnologia d' espectre expandit utilitza tot l' ample de banda disponible, en lloc de fer servir una portadora per concentrar energia al seu voltant. No va ser casualitat que es triés aquesta tecnologia sobre les demés, ja que l'espectre expandit ofereix diverses avantatges importants respecte a les demés, de les que destaquen les seves excel·lents propietats respecte a la tolerància a interferències i les seves possibilitats de xifrat.

Els estàndards 802.11b, 802.11g i 802.11n-2.4 utilitzen l'espectre dels 2.400 - 2.500 GHz, una de les bandes reservades internacionalment per propòsits de l'industria, ciència i/o medicina. Per altra banda els estàndards 802.11a i 802.11n-5 treballen en la banda dels 4.915 – 5.825 GHz, molt més regulades en tot el món. Aquestes bandes són més conegudes com dels 2.4 GHz i dels 5 GHz per abreujar.

Cadascun d'aquests espectres radioelèctrics es subdivideix en canals amb una freqüència central i una amplada.

La banda dels 2.4 GHz es divideix en 14 canals separats cada 5 MHz, començant amb el canal 1 centrat als 2.412 GHz. Cal destacar que els darrers canals tenen restriccions addicionals o no son disponibles segons alguns marcs de regulació.

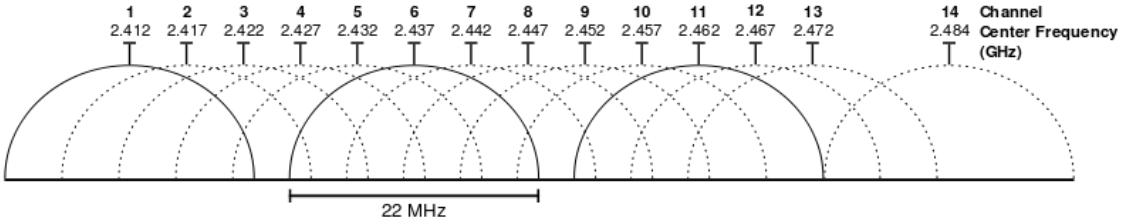


Figura 2-5 Canals Wi-Fi a la banda dels 2.4 GHz

(font de la imatge original: https://commons.wikimedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).png)

Canal	Freqüència en MHz	Nord-Amèrica	Japó	Espanya
1	2412	Sí	Sí	Sí
2	2417	Sí	Sí	Sí
3	2422	Sí	Sí	Sí
4	2427	Sí	Sí	Sí
5	2432	Sí	Sí	Sí
6	2437	Sí	Sí	Sí
7	2442	Sí	Sí	Sí
8	2447	Sí	Sí	Sí
9	2452	Sí	Sí	Sí
10	2457	Sí	Sí	Sí
11	2462	Sí	Sí	Sí
12	2467	No	Sí	Sí
13	2472	No	Sí	Sí
14	2484	No	11b només	No

Taula 1 Freqüències dels canals a 2.4 GHz i regulació per país

La numeració dels canals de la banda dels 5 GHz és molt menys intuïtiva degut a les diferents regulacions a diferents països. La següent taula en resum els canals a nivell europeu, amb les següents anotacions:

- Si s'especifica interiors indica que només es pot utilitzar per interiors i no per crear punts de connexió entre xarxes per l'exterior.
- Quan s'especifica DFS vol dir que cal fer servir el que es coneix com a *Dynamic Frequency Selection*.
- TPC implica l'ús obligatori de mecanismes de control de la potència de transmissió (*Transmit Power Control*).
- Finalment SRD especifica que només es pot utilitzar per a dispositius de rang limitat o *Short Range Devices*.

Canal	Freqüència Central (MHz)	Rang de freqüència (MHz)	Amplada (MHz)	Europe
7	5035	5030-5040	10	No
8	5040	5030-5050	20	No
9	5045	5040-5050	10	No
11	5055	5050-5060	10	No
12	5060	5050-5070	20	No
16	5080	5070-5090	20	No
34	5170	desconegut	desconegut	No
36	5180	5170-5190	20	Interiors
38	5190	5170-5210	40	No
40	5200	5190-5210	20	Interiors
42	5210	5170-5250	80	No
44	5220	5210-5230	20	Interiors
46	5230	5210-5250	40	No
48	5240	5230-5250	20	Interiors
50	5250	5170-5330	160	No

Canal	Freqüència Central (MHz)	Rang de freqüència (MHz)	Amplada (MHz)	Europe
52	5260	5250-5270	20	Interiors/DFS/TPC
54	5270	5250-5290	40	No
56	5280	5270-5290	20	Interiors/DFS/TPC
58	5290	5250-5330	80	No
60	5300	5290-5310	20	Interiors/DFS/TPC
62	5310	5290-5330	40	No
64	5320	5310-5330	20	Interiors/DFS/TPC
100	5500	5490-5510	20	DFS/TPC
102	5510	5490-5530	40	No
104	5520	5510-5530	20	DFS/TPC
106	5530	5490-5570	80	No
108	5540	5530-5550	20	DFS/TPC
110	5550	5530-5570	40	No
112	5560	5550-5570	20	DFS/TPC
114	5570	5490-5650	160	No
116	5580	5570-5590	20	DFS/TPC
118	5590	5570-5610	40	No
120	5600	5590-5610	20	DFS/TPC
122	5610	5570-5650	80	No
124	5620	5610-5630	20	DFS/TPC
126	5630	5610-5650	40	No
128	5640	5630-5650	20	DFS/TPC
132	5660	5650-5670	20	DFS/TPC

Canal	Freqüència Central (MHz)	Rang de freqüència (MHz)	Amplada (MHz)	Europe
134	5670	5650-5690	40	No
136	5680	5670-5690	20	DFS/TPC
138	5690	5650-5730	80	No
140	5700	5690-5710	20	DFS/TPC
142	5710	5690-5730	40	No
144	5720	5710-5730	20	No
149	5745	5735-5755	20	SRD (25 mW)
151	5755	5735-5775	40	SRD (25 mW)
153	5765	5755-5775	20	SRD (25 mW)
155	5775	5735-5815	80	SRD (25 mW)
157	5785	5775-5795	20	SRD (25 mW)
159	5795	5775-5815	40	SRD (25 mW)
161	5805	5795-5815	20	SRD (25 mW)
165	5825	5815-5835	20	SRD (25 mW)
183	4915	4910-4920	10	No
184	4920	4910-4930	20	No
185	4925	4920-4930	10	No
187	4935	4930-4940	10	No
188	4940	4930-4950	20	No
189	4945	4940-4950	10	No
192	4960	4950-4970	20	No
196	4980	4970-4990	20	No

Taula 2 Freqüències dels canals a 5 GHz i regulació Europea

A l'actualitat també hi ha alguns estàndards que especifiquen altres freqüències menys freqüents com per exemple la dels 3.65 GHz a l'estàndard 802.11y, la dels 4.9 GHz per xarxes WLAN de seguretat pública 802.11j, de 5.9 GHz al 802.11p, 60 GHz 802.11ad o 900 MHz al 802.11ah. Però es tracta d'estàndards encara no implementats i/o no tant estesos com els citats anteriorment.

2.1.4 Serveis de xarxa

Una forma senzilla, però pràctica, de definir una tecnologia de xarxa és definir quins serveis defineix. En el cas dels estàndards IEEE 802.11 son 9 els serveis principals que es proveeixen.

Tres d'aquests s'usen per a l' intercanvi de dades i els sis restants es corresponen amb operacions de gestió i control que permeten a la xarxa mantenir un control de les estacions i d' aquesta forma poder enviar les trames adequadament.

A continuació es descriuen breument:

- Distribució : Aquest servei és utilitzat per les estacions que formen part d'una xarxa en mode infraestructura, cada cop que envien dades. Un cop una trama arriba al punt d'accés, aquest utilitza el servei de distribució per a fer arribar les trames al seu destí.
- Integració : El servei d' integració és un subservei que proveeix el servei de distribució. Aquest permet la connexió del servei de distribució a una altra xarxa (per exemple una xarxa no IEEE 802.11).
- Associació : L'enviament de trames a les estacions és possible gràcies a que aquestes s'associen o registren prèviament als punts d' accés. Llavors el sistema de distribució pot fer servir la informació d'associació per a saber quin punt d'accés fer servir per a cada estació (en cas d' una xarxa ESS amb múltiples punts d'accés).
- Re-associació : Quan una estació que es mou dins una àrea de servei estès té cobertura a través de diverses àrees de servei bàsic, pot avaluar la qualitat del senyal de recepció, i en cas que creure-ho oportú, canviar de punt d'accés al que està associat mitjançant trames de re-associació.
- De-associació : Aquest servei, com el seu nom indica, serveix per finalitzar una associació. Per tant és un servei que fan servir les estacions quan abandonar una xarxa.
- Autenticació : Les xarxes sense fils 802.11 no ofereixen les possibilitats d'assegurar la part física de la mateixa, al menys en un grau similar al de les xarxes Ethernet (en el qual es requereix l'accés físic al cablejat i/o dispositius de xarxa). És per això que incorporen unes rutines d' autenticació per a saber que les estacions que accedeixen a la mateixa són legítimes. És un prerequisit opcional

per a la posterior associació (tot i que pot no ser així, depenent de la configuració del AP).

- De-autenticació: Procés invers a l'autenticació, que també forma part dels serveis utilitzats quan s'abandona una xarxa.
- Xifrat: Al ser el medi emprat per les xarxes 802.11 accessible per a qualsevol (sempre i quan es disposi de l'equipament adequat), aquest servei és de vital importància, ja que sense ell, encara que un atacant no es pogués associar a la xarxa, degut al servei d' autenticació, podria veure tot el tràfic de la mateixa amb total impunitat. És per això que es va dissenyar el protocol WEP inicialment, i posteriorment WPA i WPA2, per a oferir cert grau de privacitat.
- Lliurament: El servei de lliurament, conegut com a lliurament d'unitat de dades del servei MAC o MSDU (Mac Service Data Unit), és l' encarregat de fer arribar les dades al seu punt de destí.

2.1.5 MAC 802.11

Aquesta subcapa és l' encarregada de controlar la transmissió de les dades dels elements de la xarxa al medi. Proveeix totes les operacions per a treballar amb les diferents trames i també per a la interacció amb una xarxa de distribució o xarxa de *backbone* (com per exemple una xarxa Ethernet).

L'estàndard IEEE 802.11 no suposa un allunyament radical respecte als altres estàndards IEEE 802. De fet el seu origen era el de portar les xarxes Ethernet a enllaços per radio. Ethernet utilitza un esquema CSMA (*Carrier Sense Multiple Acces*) per a controlar l' accés al medi, intentant evitar les col·lisions amb un mecanisme de detecció anomenat CSMA/CD (*Collision Detect*). Per contra, ja que les col·lisions malgasten molta capacitat de transmissió en un medi com l'aire, l'estàndard IEEE 802.11 es basa en evitar-les utilitzant el que es coneix com CSMA/CA (*Collision Avoidance*).

Format de les trames

Les trames 802.11 no inclouen alguns dels camps clàssics de les xarxes Ethernet, les absències més notables són el camp de tipus/longitud i el de preàmbul. Aquest darrer és part de la capa física. Els detalls d'encapsulació com el tipus i la longitud són presents a les capçaleres de les dades.

Per altra banda incorpora nous camps per poder sobreposar-se a les dificultats que suposa utilitzar radio freqüències. Un clar exemple és la utilització de fins a quatre camps d'adreces que varien en significat depenent del tipus i subtipus de trama.

El format d'una trama MAC és el següent:

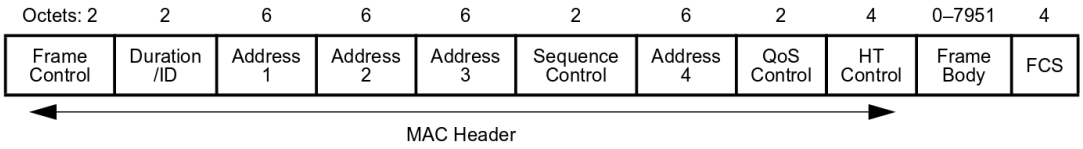


Figure 8-1—MAC frame format

Figura 2-6 Format de trama MAC

(font de la imatge: estàndard IEEE 802.11)

A on els dos primers octets o 16 bits es corresponen amb el camp de control de trama (*Frame Control*), que a l'hora està definit amb els camps:

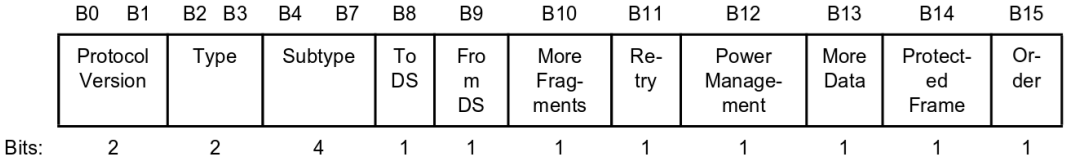


Figure 8-2—Frame Control field

Figura 2-7 Camp de control de trama

(font de la imatge: estàndard IEEE 802.11)

Els components del camp de control de trama són:

- Versió del protocol: 2 bits indiquen quina versió de MAC 802.11 és utilitzada a la resta de la trama.
- Tipus i subtipus: 2 i 4 bits respectivament que especifiquen de quin tipus de trama es tracta. Els seus possibles valors són:

Type value	Type description	Subtype value	Subtype description
00	Management	0000	Association Request
00	Management	0001	Association Response
00	Management	0010	Reassociation Request
00	Management	0011	Reassociation Response
00	Management	0100	Probe Request

Type value	Type description	Subtype value	Subtype description
00	Management	0101	Probe Response
00	Management	0110-0111	Reserved
00	Management	1000	Beacon
00	Management	1001	ATIM
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101	Action
00	Management	1110-1111	Reserved

Taula 3 Trames de gestió o *management*

Type value	Type description	Subtype value	Subtype description
01	Control	0000-0111	Reserved
01	Control	1000	Block Ack Request
01	Control	1001	Block Ack
01	Control	1010	PS-Poll
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	ACK
01	Control	1110	CF-end
01	Control	1111	CF-end + CF-ack

Taula 4 Trames de control

Type value	Type description	Subtype value	Subtype description
10	Data	0000	Data
10	Data	0001	Data + CF-ack
10	Data	0010	Data + CF-poll
10	Data	0011	Data +CF-ack +CF-poll
10	Data	0100	Null
10	Data	0101	CF-ack
10	Data	0110	CF-poll
10	Data	0111	CF-ack +CF-poll
10	Data	1000	QoS data
10	Data	1001	QoS data + CF-ack
10	Data	1010	QoS data + CF-poll
10	Data	1011	QoS data + CF-ack + CF-poll
10	Data	1100	QoS Null
10	Data	1101	Reserved
10	Data	1110	QoS + CF-poll (no data)
10	Data	1111	Qos + CF-ack (no data)
11	Reserved	0000-1111	Reserved

Taula 5 Trames de dades

- To DS i From DS: Aquests dos bits indiquen si la trama és destinada al sistema de distribució o si prové d'aquest. Els seus possibles valors són:

To/From DS values	Meaning
To DS = 0, From DS = 0	*A data frame direct from one STA to another STA within the same IBSS, as well as all management and control type frames.*
To DS = 0, From DS = 1	*Data frame exiting the DS.*
To DS = 1, From DS = 0	*Data frame destined for the DS.*
To DS = 1, From DS = 1	*Wireless distribution system (WDS) frame being distributed from one AP to another AP.*

Taula 6 Valors dels camps To DS i From DS

- Més fragments: Funciona igual que el bit de més fragments del protocol IP. Indica que la trama ha estat fragmentada i que encara manquen per transmetre algunes parts de la mateixa.
- Retransmissió: Indica que es tracta d'una trama retransmesa. Ajuden als elements receptors a detectar i eliminar trames duplicades.
- Estalvi d'energia: Especifica si l'element que ha transmès les dades estarà o no en mode estalvi d'energia.
- Més dades: Serveix per indicar que hi ha més trames amb el mateix destí al buffer d'enviament.
- Protecció: Indica si la trama utilitza algun mecanisme de protecció o no.
- Ordre: Les trames i fragments poden ser enviats en ordre o sense ordre.

A banda d'aquesta capçalera i els seus subcamps, la resta de camps principals de les trames MAC són:

- Identificació de duració: Aquests bits serveixen per indicar quant de temps (en micro segons) es preveu que el medi sigui ocupat per l'estació que esta transmeten actualment.
- Camps d'adreces: Es tracta d'adreces de 48 bits que segueixen les convencions utilitzades per altres xarxes IEEE 802, com per exemple les Ethernet. Segons el tipus de trames cada camp d'adreces té un significat o un altre. Normalment la primera adreça indica el destí, la segona l'origen i la tercera el Basic Service Set ID o BSSID. La quarta no s'utilitza habitualment, però serveix per indicar l'adreça de l'estació de transmissió quan s'utilitza WDS[4].

[4] WDS: *Wireless Distribution System* – http://www.dd-wrt.com/wiki/index.php?title=WDS_Linked_router_network

- Control de seqüencia: Aquests 16 bits s'utilitzen per gestionar la fragmentació i les trames duplicades. Els primers 4 bits indiquen el número de fragment i els 12 restants el de seqüència.
- Cos de la trama: Fins a 2304 bits per encapsular dades de capes superiors. S'especifiquen 2312 bits per poder acomodar l'*overhead* afegit pels mecanismes de xifrat WEP o WPA.
- Seqüència de control de trama o FCS: De la mateixa forma que les trames Ethernet, les trames es finalitzen amb uns bits de control que sovint s'anomenen CRC (*Cyclic Redundancy Code*). Gràcies a aquest camp una estació pot comprovar l' integritat de les dades, ja que tots els camps de la capçalera MAC i el cos de la trama estan inclosos per calcular el valor d'aquesta seqüència.

2.2 Monitoratge de xarxes Wi-Fi

A continuació es detallen alguns aspectes relacionats amb el monitoratge de trànsit Wi-Fi que són utilitzats en aquest projecte.

2.2.1 Mode monitor

Es coneix com a mode monitor o RFMON, de l'anglès Radio Frequency Monitor), a un mode de treball específic que permeten la majoria d'interfícies de xarxa IEEE 802.11. Aquest mode de treball permet l'accés a tot el trànsit IEEE 802.11 a la zona de cobertura i freqüència en la que treballa la interfície.

És similar al que es coneix com a mode promiscu, però en aquest cas no és necessari estar associat amb cap punt d'accés ni xarxa per a monitorar el trànsit.

2.2.2 Channel hopping

El terme conegut com a *channel hopping* o salt de canals es correspon amb un mecanisme de mostreig de les diferents freqüències utilitzades pels estàndards IEEE 802.11.

Si es volen monitorar tots els canals (~freqüències) i només es disposa d'una interfície de xarxa, es pot recórrer a realitzar un mostreig en el temps del trànsit a cada canal.

En el seu mode de funcionament més bàsic s'especifiquen les freqüències a monitorar i el nombre de salts entre elles que es realitzarà en un període de temps (per exemple un segon). Per a implementar-ho es canvia contínuament el canal de treball de la interfície de xarxa, seguint una seqüència definida i seguint el període determinat.

És útil es volen obtenir dades de resum, estadístiques o quan no importa que es perdi trànsit ja que, al realitzar aquests salts de canals, durant tots els períodes en els quals no s'està treballant en un canal determinat no es pot accedir al trànsit d'aquest.

2.3 Sistema de detecció d'intrusions

També coneguts com a IDS, de l'anglès Intrusion Detection System, es tracta de dispositius o aplicacions de programari que permeten monitorar el trànsit de xarxa o els sistemes cercant activitat maliciosa o violacions de les polítiques establertes. A l'apartat de bibliografia s'inclouen recursos[3][4] que permeten aprofundir en el seu funcionament.

2.4 Marc de treball

Tot i tractar-se d'un projecte a iniciativa de l'estudiant, el desenvolupament del mateix s'ha dut a terme sota la supervisió i la tutoria de membres del grup de recerca de Comunicacions i Sistemes Distribuïts[5], del Departament d'Arquitectura i Tecnologia de Computadors. Concretament:

- Antonio Bueno Delgado
- Lluís Fàbrega i Soler

Tots dos han aportat els seu suport en els camps de comunicacions, comunicacions distribuïdes, comunicacions sense fils i seguretat.

[5] BCDS: *Broadband Communications and Distributed Systems* - http://bcds.udg.edu/

3 Anàlisi

Aquest apartat s'ocupa de recollir les necessitats i requisits del sistema de monitoratge de xarxes Wi-Fi desenvolupat en el marc d'aquest projecte de final de carrera. S'inclouen tant els requisits funcionals com els no funcionals.

Així mateix s'ha inclòs en aquest apartat la descripció de la metodologia seguida pel disseny i la implementació del sistema i també l'estratègia adoptada per assolir els objectius plantejats. Es descriu el pla de treball, les tasques, estimació temporal i les principals fites.

3.1 Requisits

3.1.1 Requisits funcionals

Els principals requisits del sistema a desenvolupar són:

- Captura de trànsit
  - Possibilitat d'especificar el canal o canals de captura que es corresponen amb les freqüències utilitzades pels protocols IEEE 802.11
  - Possibilitat de capturar múltiples bandes de freqüència utilitzades pels protocols IEEE 802.11 de forma simultània
  - Possibilitat de realitzar captures simultàniament en diferents zones de cobertura
  - Generació de fitxers amb les dades de captura
- Processament

    - Processar i interpretar els fitxers generats en el procés de captura per tal d'extreure'n:
        - Detall de les xarxes Wi-Fi
        - Detall de les estacions i d'altres elements que realitzin comunicacions IEEE 802.11
        - Generació d'alertes de seguretat en base al trànsit capturat
- Emmagatzemament
    - Permetre la inserció de les dades processades
    - Permetre la modificació i/o eliminació de dades
    - Indexar les dades de forma adient per a la seva consulta
- Anàlisi
    - Oferir una interfície d'usuari mitjançant la qual sigui possible explorar i consultar les dades emmagatzemades
    - Oferir una interfície o API per facilitar la integració amb altres eines
    - Disposar de cerques i visualitzacions prefixades que permetin començar a explorar les dades o veure'n ràpidament un resum de les mateixes

3.1.2 Requisits no funcionals

Els principals requeriments no funcionals del sistema a desenvolupar son:

- Utilitzar programari lliure en la major mesura possible
- Utilitzar estàndards oberts i reconeguts per la representació i emmagatzemament de les dades
- Flexibilitat per adaptar-se a casos d'ús no pensats inicialment
- Contemplar aspectes de seguretat del sistema des de la fase inicial de disseny fins a la d'implementació
- Escalabilitat tant vertical com horitzontal
- Interoperabilitat amb altres eines o solucions
- Dissenyar i implementar els sensors de captura de forma que sigui possible el seu desplegament amb maquinari de baix cost

3.2 Estudi de la viabilitat

3.2.1 Recursos

A nivell de recursos humans, tot i l'abast del projecte i de disposar d'un únic recurs pel desenvolupament (l'autor del mateix), la planificació dilatada en el temps en possibilita l'execució.

En quant a maquinari s'ha decidit reutilitzar-lo del ja disponible per l'autor, pel que no n'ha calgut l'adquisició.

Tot el programari utilitzat per l'anàlisi, desenvolupament i implementació d'aquest projecte és de codi lliure i gratuït.

3.2.2 Viabilitat tecnològica

Abans d'afrontar el projecte s'ha analitzat l'existència d'eines i llibreries que en fessin plausible el desenvolupament.

A nivell tècnic la possibilitat de configurar una interfície de xarxa IEEE 802.11 en el que es coneix com a mode monitor, garanteix la possibilitat de monitorar l'espectre radioelèctric en el qual treballa.

Els dispositius de tipus punt d'accés estan dissenyats per processar grans quantitats de trànsit IEEE 802.11, en moltes ocasions fent-ne operacions com el càlcul d'integritat. És per això que la utilització de maquinari similar únicament per a la captura es creu viable.

Actualment existeixen sistemes d'emmagatzemament i consulta de dades que gestionen volums molt elevats, tant d'inserció de dades com de consultes. Per aquest motiu es veu possible l'emmagatzemament i posterior accés a les dades capturades.

3.2.3 Viabilitat econòmica

Els costos humans són exclusivament en temps personal de l'autor, pel que no suposen en aquest cas un impediment econòmic.

El maquinari utilitzat ja era propietat de l'autor i estava amortitzat prèviament, pel que no ha suposat una despesa. I tot el programari utilitzat és de codi lliure i gratuït, pel que tampoc ha requerit cap inversió.

3.3 Metodologia

Tot i haver-se realitzat un estudi de viabilitat previ, haver-se recollit un llistat de requisits preliminars i, en resum, tenir una idea general del projecte, l'experiència indicava que caldria aprofundir en cadascun dels tres grans blocs del sistema (captura, emmagatzemament i anàlisi) per realment apreciar-ne la magnitud i estimar-ne millor els esforços.

Amb aquesta premissa s'ha considerat que la metodologia més apropiada en aquest cas va alineada al que es coneix actualment com a metodologies àgils o *agile development*.

Agile Development

Es refereix a un grup de metodologies de desenvolupament basades en un procés iteratiu i incremental, a on els requeriments i les seves solucions evolucionen a mesura que s'avança en el desenvolupament.

Aquest comportament és especialment útil quan es vol tenir la possibilitat de revisar qualsevol aspecte del desenvolupament contínuament, fins i tot requeriments i disseny.

Scrum

Scrum[6] és probablement el *framework* agile més conegut. Tot i la seva eficàcia provada, la implementació pura d'Scrum requereix la definició de diferents rols com de:

- *Product owner*: representa l'usuari final del sistema a desenvolupar i s'encarrega que el desenvolupament proporcioni el valor que quest espera.
- *Equip de desenvolupament*: El responsable de lliurar els increments del sistema a desenvolupar al final de cada període de temps fixat (conegut com *Sprint*).
- *Scrum Master*: És l'encarregat de garantir que es segueix el framework Scrum i de gestionar els impediments per garantir que l'equip de desenvolupament compleixi els objectius.

Que no estan pensats per un projecte que serà desenvolupat per una única persona, com és el cas d'aquest projecte. Així doncs el flux de treball que proposa Scrum perd molt del seu sentit si els tres rols són executats per una sola persona.

És per això que s'ha adoptat una metodologia més flexible com Kanban, pel desenvolupament.

[6] Scrum - http://scrummethodology.com/

Kanban

Kanban[7] és una metodologia genèrica per la gestió del coneixement que ha estat molt ben acollida en el context del desenvolupament de programari.

Alguns dels components que utilitza i dels conceptes clau que defineix són:

- *Kanban boards*: Es tracta d'una eina per visualitzar i optimitzar el flux de treball. En la seva vesant més bàsica defineix tres fases, Per Fer, En Progres i Fet.
- *Kanban cards*: Es tracta dels elements que formen part de la Kanban board i representen unitats de treball.
- Limitar la quantitat de treball en progrés: La idea és impedir que s'estigui treballant en paral·lel en moltes tasques sense donar-ne cap per finalitzada. Per aconseguir-ho cal reduir al màxim possible l'abast de les tasques i se'n limita el nombre de tasques concurrents que poden estar en l'estat "Per Fer".

Durant el gruix del desenvolupament del projecte s'han definit tres *Kanban Boards* un par cadascun dels grans blocs del mateix:

- Captura
- Processament i emmagatzemament
- Anàlisi

Un altre amb aspectes relacionats amb la gestió del PFC (full de projecte, comunicacions amb els tutors, ...) i un darrer de documentació.

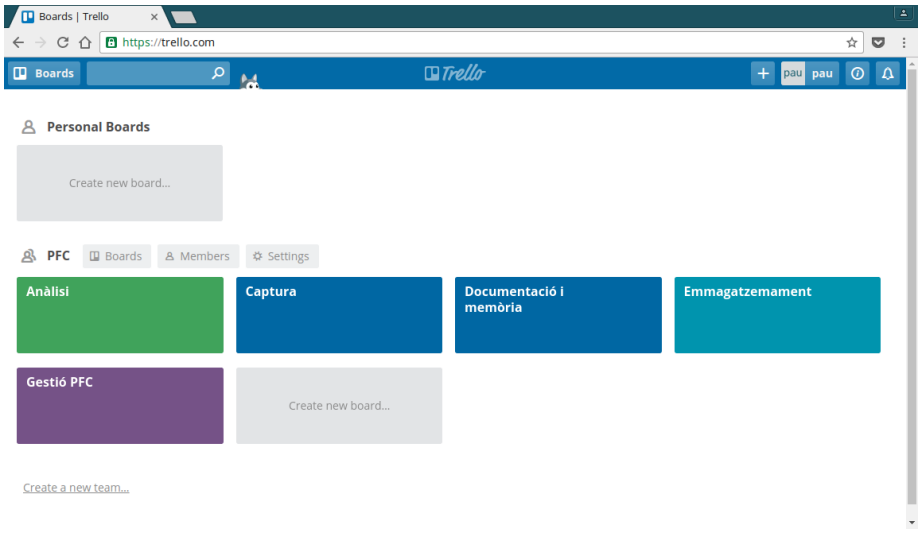


Figura 3-1 Visió dels diferents *Kanban boards* utilitzats

[7] Kanban - https://en.wikipedia.org/wiki/Kanban_(development)

Els principals beneficis que ha aportat aquesta aproximació al desenvolupament del projecte han estat flexibilitat en la planificació, reducció dels cicles de temps de les diferents iteracions del projecte, crear menys colls d'ampolla al limitar el treball en progrés i proporcionar mètriques visuals que faciliten el seguiment i la gestió del projecte.

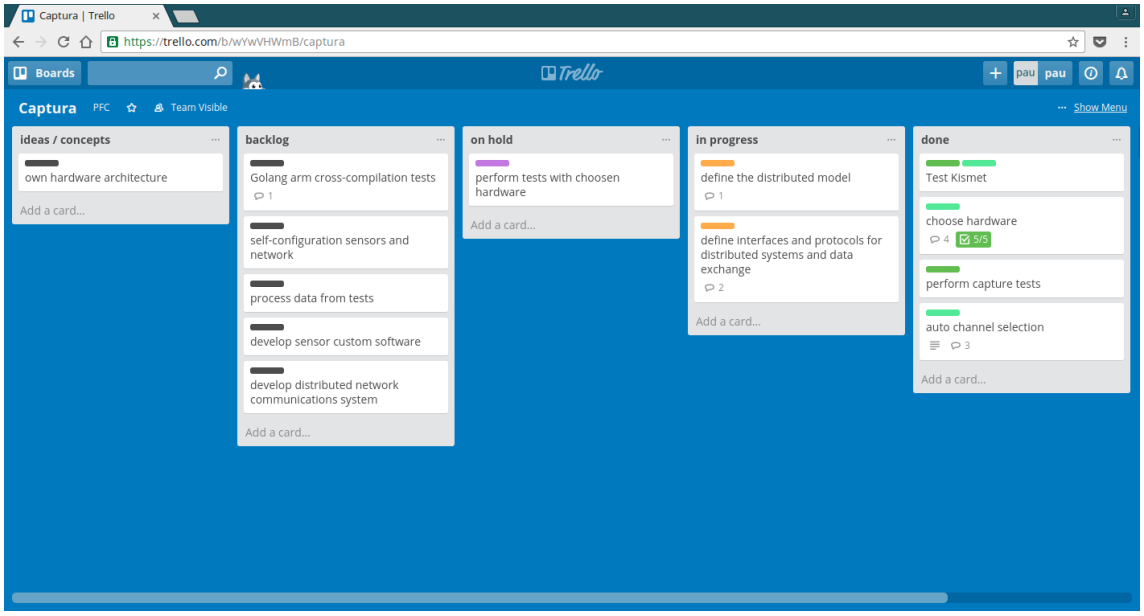


Figura 3-2 Estat d'un *Kanban Board* en un determinat instant del projecte

S'ha utilitzat l'eina Trello[8] per a la gestió dels *Kanban Boards* i les tasques.

3.4 Planificació

La planificació i el treball inicial d'aquest projecte comença molt abans de la seva definició actual, donat que moltes de les problemàtiques tractades han estat viscudes durant la vida laboral de l'autor, al llarg de diversos anys, sense ser possible dedicar-hi el temps i els esforços necessaris per a tractar-les de forma adient.

Al voltant del setembre de 2015 es decideix aquesta temàtica com a projecte de final de carrera i es comença a aprofundir de forma més sistemàtica en els sistemes de monitoratge de xarxes IEEE 802.11. Concretament per veure si és possible donar solució als problemes i requeriments que posteriorment han estat plantejats en aquest projecte de final de carrera.

[8] Trello - https://trello.com/

Després d'una fase preliminar d'investigació i realització de proves, i de corroborar-ne la viabilitat, es defineix una proposta formal que es fa arribar als actuals tutors del projecte, els quals l'accepten sota el seu tutoratge. La previsió era finalitzar-lo en alguna de les convocatòries del curs 2015/2016.

A continuació es detalla el pla de treball i les principals tasques planificades.

3.4.1 Pla de treball i tasques planificades

El pla de treball ve determinat en gran part per les tres tasques principals que donen títol a aquest projecte, relacionades amb el trànsit Wi-Fi:

- Captura
- Processament i emmagatzematge
- Anàlisi

Es tracta de tasques de naturalesa seqüencial donat que cadascuna d'elles depèn de l'anterior. Abans de processar i emmagatzemar el trànsit, cal ser capaç de capturar-lo, conèixer-ne les seves característiques i els seus possibles formats. El mateix succeeix a l'hora d'analitzar-lo, fase que va fortament lligada al processament i al emmagatzematge que se n'hagi fet del mateix.

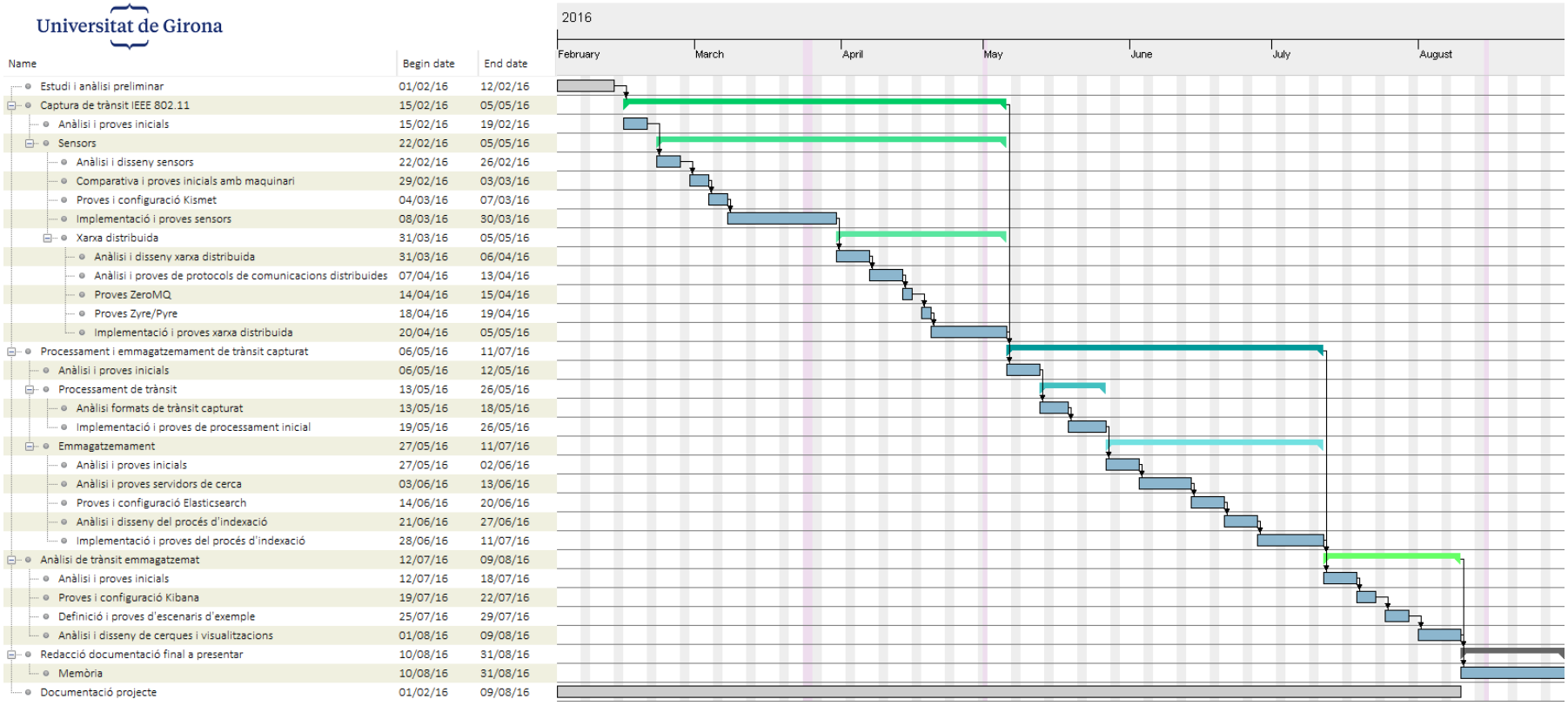
Així doncs, el gruix del projecte es composa de les tasques representades al proper diagrama de Gantt, a on s'inclou una estimació d'esforç i una planificació temporal. El llistat de tasques no s'ha definit en la seva totalitat des del principi, donat que en la majoria de casos calia aprofundir en cada bloc abans de poder desglossar-ne i estimar millor les tasques.

Cal destacar que tot i que el projecte ha estat desenvolupat per una única persona, aquesta planificació està especificada per diferents recursos que es corresponen amb el rol que s'ha exercit en cada moment:

- Arquitecte: Perfil tipus arquitecte del programari experimentat.
- Desenvolupador: Perfil de desenvolupador de programari.
- Operador: Perfil de sistemes o similar.
- Manager: Perfil tipus gestor de projectes.

S'ha utilitzat l'eina de codi obert GanttProject[9] per a la gestió i planificació de tasques i recursos.

[9] GanttProject - http://www.ganttproject.biz/



Name	Begin date	End date
Estudi i anàlisi preliminar	01/02/16	12/02/16
Captura de trànsit IEEE 802.11	15/02/16	05/05/16
Anàlisi i proves inicials	15/02/16	19/02/16
Sensors	22/02/16	05/05/16
Anàlisi i disseny sensors	22/02/16	26/02/16
Comparativa i proves inicials amb maquinari	29/02/16	03/03/16
Proves i configuració Kismet	04/03/16	07/03/16
Implementació i proves sensors	08/03/16	30/03/16
Xarxa distribuida	31/03/16	05/05/16
Anàlisi i disseny xarxa distribuida	31/03/16	06/04/16
Anàlisi i proves de protocols de comunicacions distribuides	07/04/16	13/04/16
Proves ZeroMQ	14/04/16	15/04/16
Proves Zyre/Pyre	18/04/16	19/04/16
Implementació i proves xarxa distribuida	20/04/16	05/05/16
Processament i emmagatzemament de trànsit capturat	06/05/16	11/07/16
Anàlisi i proves inicials	06/05/16	12/05/16
Processament de trànsit	13/05/16	26/05/16
Anàlisi formats de trànsit capturat	13/05/16	18/05/16
Implementació i proves de processament inicial	19/05/16	26/05/16
Emmagatzemament	27/05/16	11/07/16
Anàlisi i proves inicials	27/05/16	02/06/16
Anàlisi i proves servidors de cerca	03/06/16	13/06/16
Proves i configuració Elasticsearch	14/06/16	20/06/16
Anàlisi i disseny del procés d'indexació	21/06/16	27/06/16
Implementació i proves del procés d'indexació	28/06/16	11/07/16
Anàlisi de trànsit emmagatzemat	12/07/16	09/08/16
Anàlisi i proves inicials	12/07/16	18/07/16
Proves i configuració Kibana	19/07/16	22/07/16
Definició i proves d'escenaris d'exemple	25/07/16	29/07/16
Anàlisi i disseny de cerques i visualitzacions	01/08/16	09/08/16
Redacció documentació final a presentar	10/08/16	31/08/16
Memòria	10/08/16	31/08/16
Documentació projecte	01/02/16	09/08/16

Figura 3-3 Diagrama de Gantt simplificat del projecte

4 Disseny

Aquest apartat descriu l'estructura i funcionalitat del sistema de monitoratge desenvolupat. Inclou informació rellevant dels principals elements que composen el sistema i com es comuniquen entre sí.

Es tracta d'una vista enfocada al punt de vista d'un arquitecte del software i d'aquelles persones que vulguin entendre el funcionament general del sistema.

4.1 Visió general

Com qualsevol sistema de monitoratge l'objectiu primordial és poder conèixer i analitzar què està succeint. I per aconseguir-ho cal disposar d'informació rellevant i fiable.

És per això que, a molt alt nivell, el sistema desenvolupat es composa de tres grans blocs:

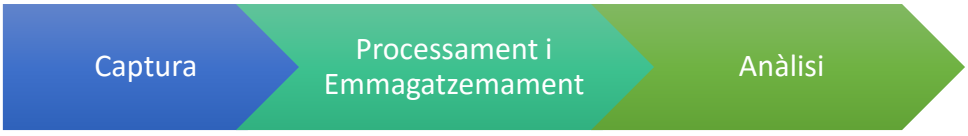


Figura 4-1 Blocs principals del sistema

El bloc de captura es centra en obtenir la informació. En el cas del sistema desenvolupat, al tractar-se d'un sistema de monitoratge de trànsit Wi-Fi, la informació rellevant és justament el trànsit que circula a les bandes de freqüències utilitzades per les xarxes i dispositius Wi-Fi (protocols IEEE 802.11).

El bloc de processament i emmagatzemament s'encarrega de preparar i guardar les dades obtingudes en un format que en permeti i faciliti el seu anàlisi.

Finalment el bloc d'anàlisi proporciona les eines per poder examinar les dades i extreure'n informació rellevant segons el context. Aquest context, des del punt de vista d'aquest projecte, és el de donar suport en la detecció i gestió de possibles incidents de seguretat relacionats amb les xarxes Wi-Fi. S'entenen com a incident de seguretat aquell esdeveniment que posa en risc la confidencialitat, integritat i/o disponibilitat dels nostres actius o sistemes d'informació.

Entrant en un primer nivell de detall, el sistema desenvolupat es composa de diferents elements relacionats amb alguns dels tres grans blocs anteriors, captura, processament i emmagatzemament i/o anàlisi:

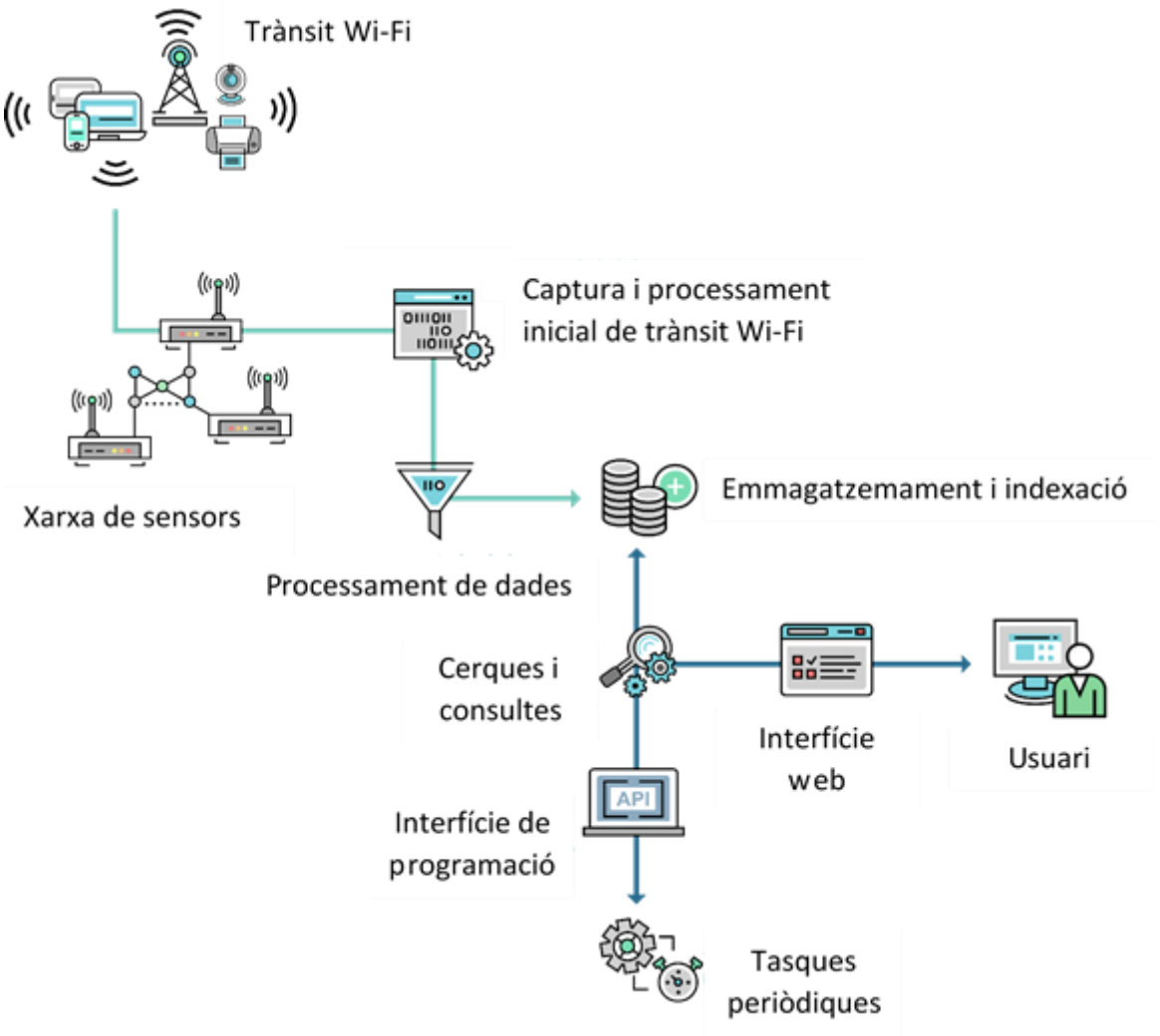


Figura 4-2 Elements principals del sistema

Trànsit Wi-Fi

No és un element del sistema com a tal, sinó que és el medi a monitorar. Per tal de saber que succeeix en les comunicacions Wi-Fi és necessari inspeccionar-ne el trànsit.

Donat que el medi utilitzat per a aquestes comunicacions és l'aire, si no s'afegeixen mesures addicionals, qualsevol dispositiu en el radi de cobertura de les comunicacions les pot escoltar. Per tant és possible per exemple enumerar les xarxes i dispositius que estan transmeten dades i conèixer-ne detalls. Amb aquesta informació es poden detectar potencials incidents de seguretat per exemple si es detecten dispositius no autoritzats que s'estan fent passar per elements d'una xarxa legítim o altres tipus de trànsit anòmal.

Al capítol anterior, 2 Conceptes previs, s'han descrit detalls bàsics del funcionament del protocol IEEE 802.11, conegut amb el nom comercial Wi-Fi, per permetre comprendre en major profunditat el tipus de dades a monitorar.

Xarxa de sensors

Un sensor és un dispositiu capaç de monitorar i capturar trànsit Wi-Fi. En aquest projecte s'ha decidit implementar una xarxa distribuïda de sensors per tal de poder monitorar diverses freqüències del protocol IEEE 802.11 simultàniament i cobrir diferents àrees de cobertura.

Al capítol anterior, 2 Conceptes previs, s'han detallat algunes de les particularitats i problemàtiques que presenta el monitoratge de xarxes Wi-Fi, que justifiquen la idoneïtat de disposar de diversos sensors.

Emmagatzemament i indexació

Cal agrupar, emmagatzemar i indexar tota la informació capturada pels sensors de forma que sigui possible utilitzar-la per detectar possibles incidents de seguretat i/o per investigar-ne de coneguts en profunditat. S'utilitza el que es coneix com a servidor de cerca per a centralitzar les dades capturades per tots els sensors i per permetre la seva consulta.

Anàlisi i explotació de la informació

Finalment cal proporcionar les eines adients per poder realitzar cerques i consultes sobre les dades capturades. Aquestes cerques poden ser automatitzades i periòdiques, per exemple per generar alertes o resums de la informació capturada. O bé ser realitzades a demanda per un usuari o analista mitjançant una interfície d'usuari. Per aquest propòsit s'implementa amb el que es coneix com a servidor d'anàlisi, que és una interfície d'accés a la informació emmagatzemada al servidor de cerca.

4.2 Captura de trànsit

4.2.1 Sensors i xarxes de sensors

Sensor

En el marc d'aquest projecte un sensor és un dispositiu capaç de monitorar l'espectre radioelèctric en les freqüències reservades pels estàndards IEEE 802.11. La seva principal tasca és la de capturar trànsit 802.11 que posteriorment serà processat i analitzat.

Degut a la constant evolució tecnològica i de mercat, que fa que freqüentment aparegui maquinari amb més capacitats i amb un cost i dimensions més contingudes, s'ha decidit utilitzar programari molt genèric i àmpliament difós per implementar la lògica de captura. D'aquesta forma es permet la utilització de maquinari heterogeni i fins i tot, en moltes ocasions, la reutilització d'equipament de xarxa ja desplegat, ja que principalment els únics requeriments d'un sensor són:

- Una o més interfícies de xarxa 802.11
- Una interfície de comunicacions addicional: una altra interfície 802.11, una interfície *ethernet*, etc.
- Executar un sistema operatiu tipus **nix*: GNU/Linux, BSD, etc.

A l'apartat 6.1 Maquinari, a la pàgina núm 78, es poden veure els detalls dels diferents dispositius utilitzats com a sensors durant el desenvolupament d'aquest projecte.

El següent diagrama representa el funcionament simplificat d'un sensor:

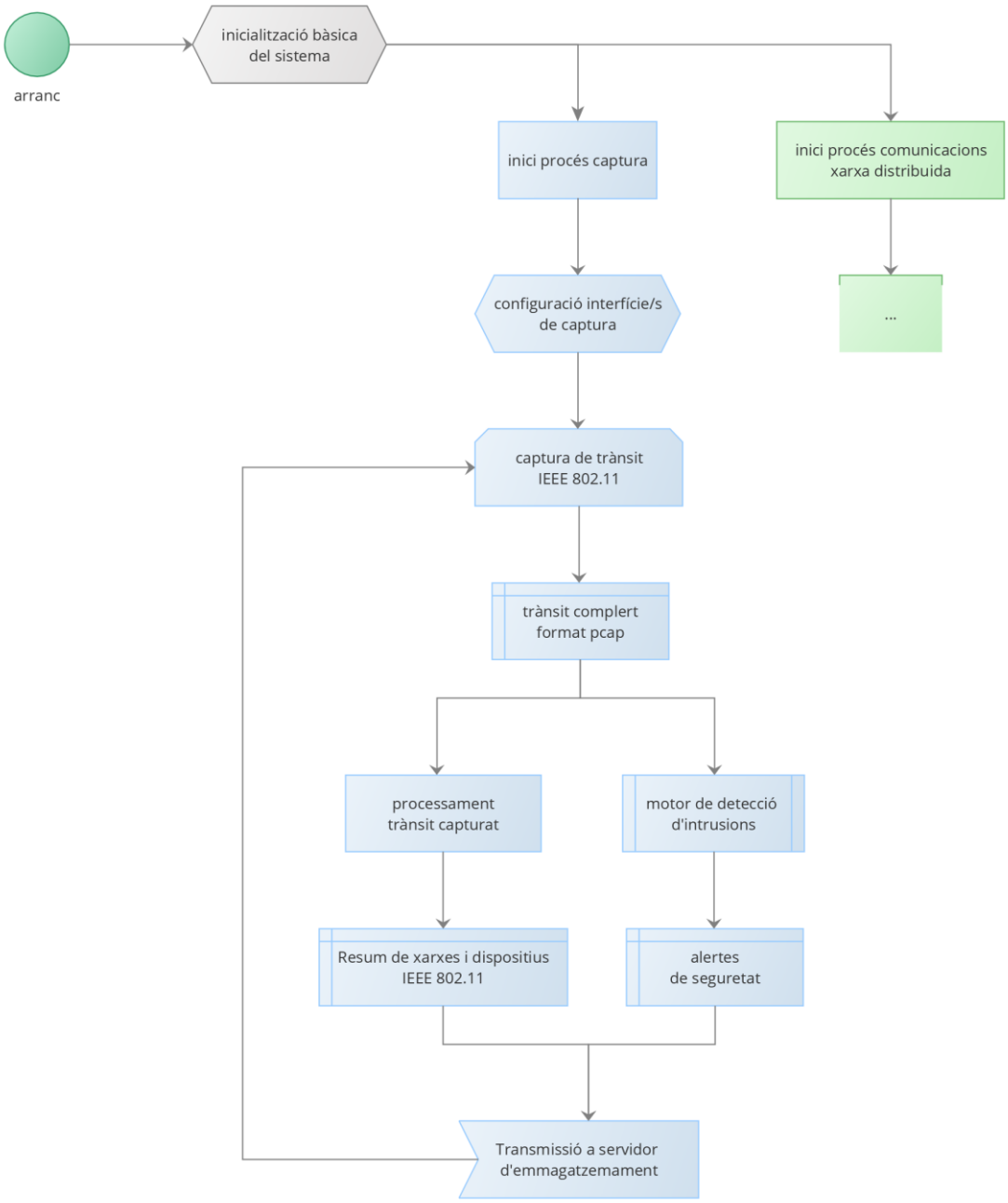


Figura 4-3 Diagrama de flux sensor (captura)

1. Inicialització: Després d'engegar físicament el sensor aquest arranca el firmware i sistema operatiu, inicialitzant les funcionalitats bàsiques del sistema.
2. Configuració d'interfícies en mode *monitor*: S'executen comandes del sistema per configurar la interfície o interfícies de captura en mode *monitor* (a l'apartat 2.2 Monitoratge de xarxes Wi-Fi, es descriu aquest mode de funcionament i perquè s'utilitza per a la captura de trànsit).

3. Captura de trànsit: S'arranca un procés de captura de trànsit que bàsicament enregistra totes les trames i paquets que veuen les interfícies de xarxa.
4. Emmagatzemament del trànsit en format *pcap*[10]: Tot el trànsit capturat s'emmagatzema en temps real.
5. Inspecció del trànsit 802.11: En paral·lel es realitza una inspecció del trànsit per tal d'extreure'n informació rellevant tal com:
    - Extracció d'informació resum de les xarxes i dispositius presents al trànsit: Seguint el format de les trames especificat a l'estàndard IEEE 802.11 es realitza una dissecció i interpretació dels diferents camps. S'ha inclòs informació més detallada del format de les trames a l'apartat, 2.1.5 MAC 802.11. El format en el que s'extreu aquesta informació es pot veure al apartat 4.2.2 Processament de trànsit Wi-Fi.
    - Generació d'alertes de seguretat: En funció a la informació present a les trames i paquets capturats es generen alertes de seguretat. Aquestes alertes poden generar-se per un sol paquet o depenent del context i estan tant basades en signatures com en tendències. A l'apartat 5.1.3 Processament de dades capturades, a la pàgina núm 57 s'inclouen més detalls. El format de les alertes es pot veure a l'apartat 4.2.2 Processament de trànsit Wi-Fi.
6. Transmissió de les dades generades a servidor centralitzat d'emmagatzemament: Periòdicament i en paral·lel els sensors transmeten les dades generades a un servidor a on es centralitzen i processen les dades de tots els sensors.
7. *En paral·lel també s'executa un procés de comunicacions amb la xarxa distribuïda de sensors, que es descriu al proper apartat.*

Xarxa de sensors

Tal i com s'ha vist al capítol 2 Conceptes previs, la naturalesa de les xarxes Wi-Fi fa que sovint sigui necessari utilitzar més d'un sensor per al seu monitoratge. Principalment per:

- Abastar tota la zona de cobertura d'una o més xarxes
- Monitorar simultàniament diferents freqüències radioelèctriques (canals)

Tot i que es podrien desplegar sensors de forma independent per complir amb aquests requeriments, la implementació d'una xarxa de sensors suposa una millora substancial per a la configuració i gestió dels sensors.

[10] pcap: estàndard de-facto per a l'emmagatzemament de trànsit de xarxa - http://www.tcpdump.org/

La solució proposada en aquest projecte es basa en una xarxa de sensors distribuïda on cada sensor és autònom però es pot comunicar amb altres sensors per modificar els paràmetres de monitoratge, captura i centralització d'informació.

Aquesta aproximació té com a principal avantatge permetre dividir les tasques de captura en diferents dispositius, disminuint els requeriments d'aquests i, en conseqüència, el cost. Trobar maquinari amb 12 o més interfícies de xarxa Wi-Fi, amb els busos i la capacitat per a processar tot el trànsit capturat no és fàcil i el seu cost arriba fàcilment a diversos milers d'euros. Per contra utilitzant una dotzena o més de dispositius dedicats a capturar i processar el trànsit de diferents freqüències es redueix el cost a pocs centenars d'euros.

Les comunicacions de la xarxa s'implementen fent servir protocols de missatgeria distribuïda que permeten:

- Que un sensor descobreixi la presència d'altres sensors
- Organitzar grups de sensors
- Enviament i recepció de missatges entre sensors

Dins la xarxa de sensors s'ha definit la figura del *orchestrator* que és l'encarregat de gestionar i decidir la configuració de cada gestor. Aquesta figura pot ser assumida per un mateix sensor o un altre element del sistema, com per exemple el servidor d'emmagatzemament.

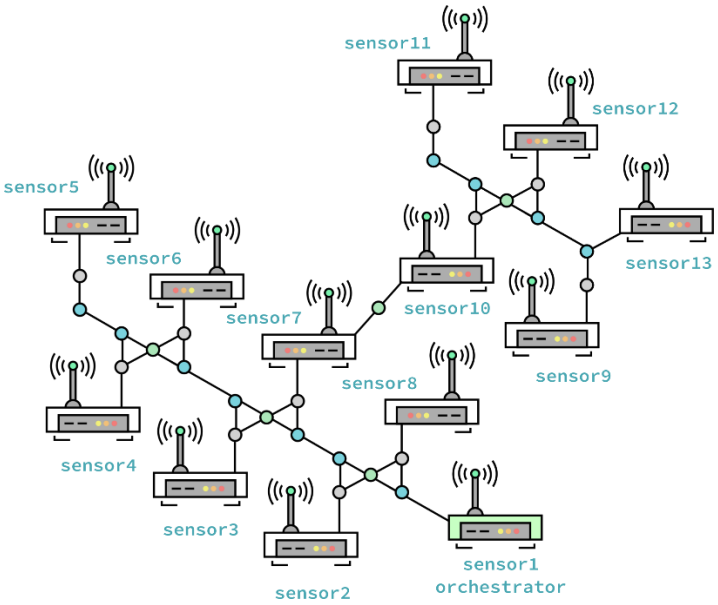


Figura 4-4 Exemple de xarxa distribuïda de sensors

A molt alt nivell el funcionament és el següent:

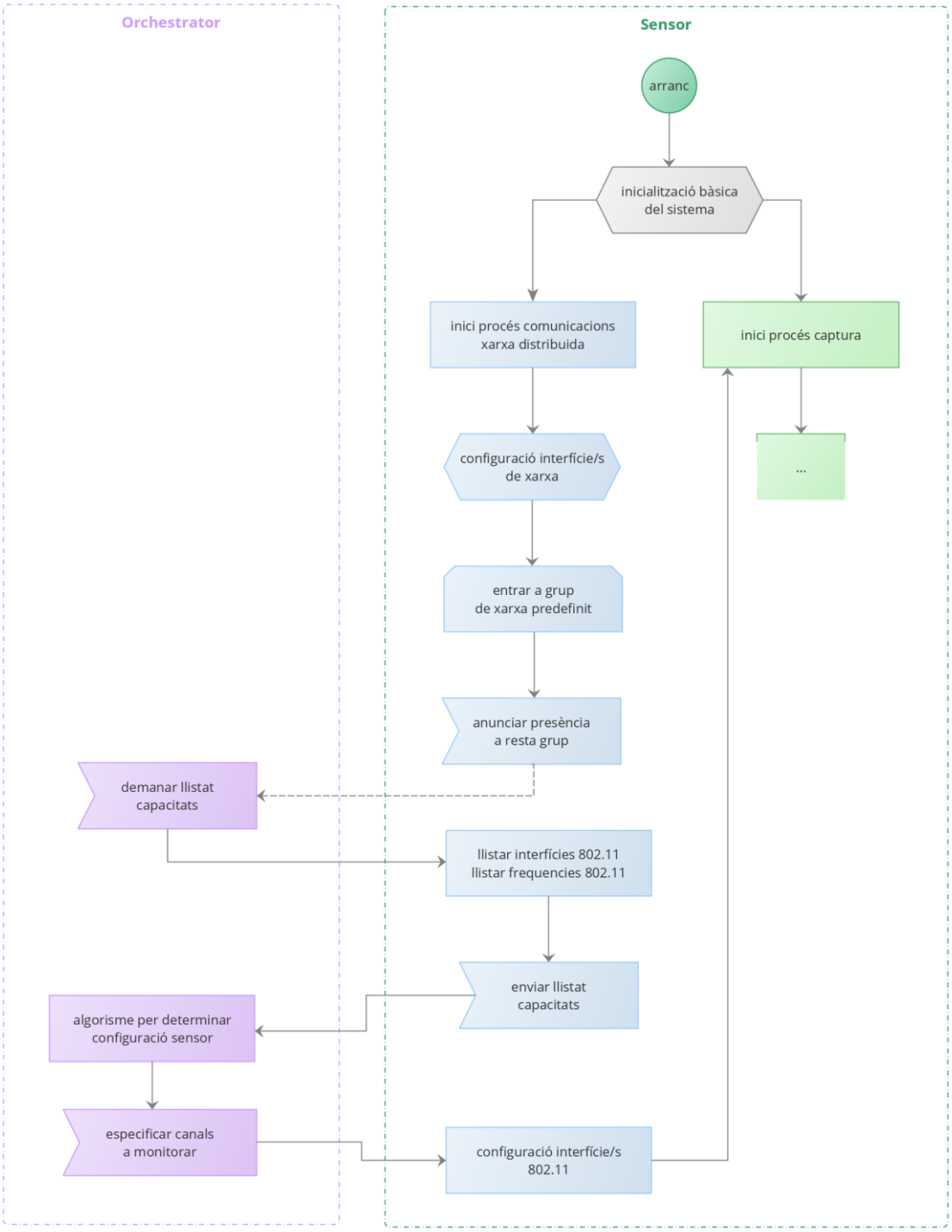


Figura 4-5 Diagrama de flux inicialització sensor (comunicacions)

1. Quan s'inicialitza un sensor, aquest anuncia la seva presència en un grup prefixat.
2. Quan l'*orchestrator* detecta un nou sensor, li demana el seu llistat de capacitats. Actualment el llistat de capacitats està únicament relacionat amb el número

d'interfícies 802.11 de les que disposa el sensor i amb les freqüències en les que poden treballar.

3. El sensor que acaba d'entrar al grup envia les seves capacitats al *orchestrator*.
4. *L'orchestrator* revisa el llistat de capacitats de les que disposa, i si s'escau, envia missatges de re-configuració als sensors. Aquests missatges de configuració especifiquen en quines freqüències tenen que treballar les interfícies del sensor.
5. Quan un sensor deixa d'estar disponible, bé perquè s'ha retirat expressament o per mal funcionament, la resta del grup se n'assabenta i, si cal, l'*orchestrator* envia missatges de re-configuració als sensors per tal de distribuir les tasques que feia el sensor que ha abandonat la xarxa distribuïda.

El disseny i la implementació de la xarxa de sensors s'ha fet sense tenir com a objectiu una única arquitectura de xarxa, sinó de forma que sigui molt flexible i escalable per poder adaptar-se a diferents necessitats. Per exemple una topologia simple a on un únic element realitza totes les tasques:

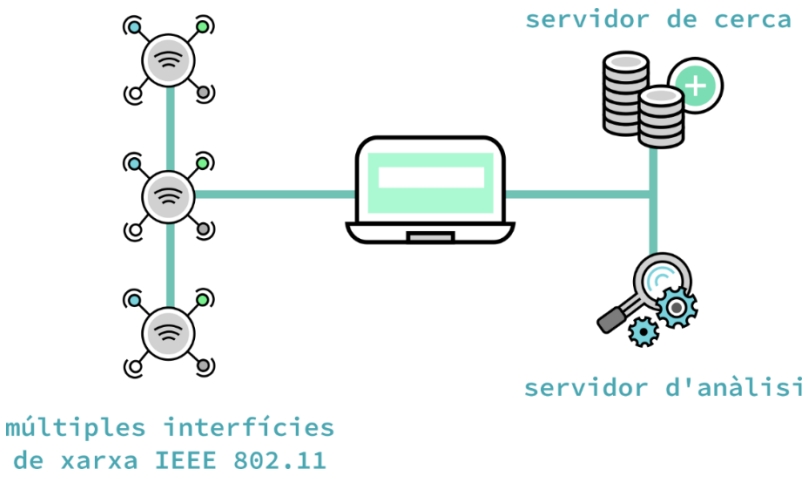


Figura 4-6 Topologia amb un únic element

En aquest cas un únic element, per exemple un ordinador portàtil amb unes capacitats considerables i múltiples interfícies de xarxa IEEE 802.11, executa les tasques de sensor, de servidor de cerca i d'anàlisi.

O bé una de més complexa amb un nombre elevat de sensors i servidors de cerca i anàlisi dedicats:

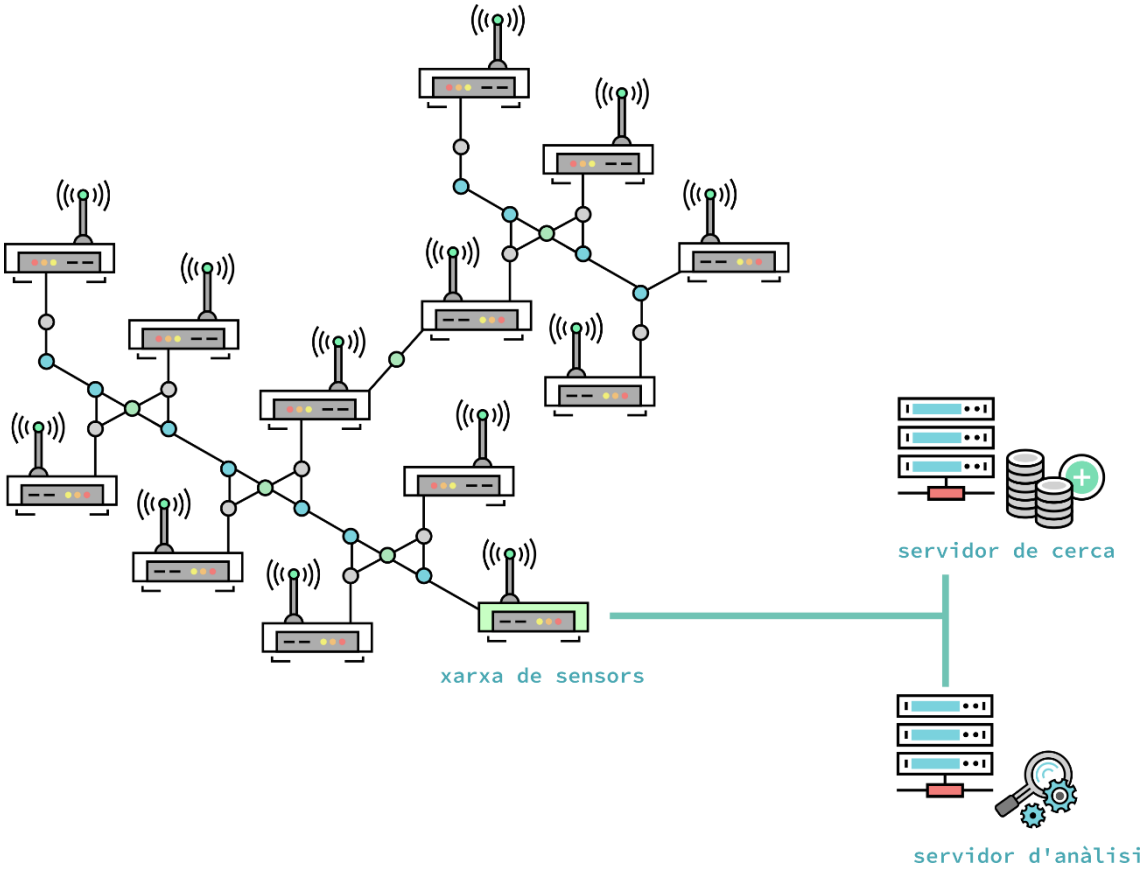


Figura 4-7 Topologia amb 13 sensors

A l'apartat 6.2 Escenaris i casos d'ús, a la pàgina núm 84, es detallen algunes de les topologies implementades durant la realització de proves.

No s'ha implementat en aquesta primera versió, però amb uns canvis mínims fins i tot es podrien definir grups de xarxes de sensors que cooperessin entre sí creant una única xarxa. Per a fer-ho caldria implementar comunicacions entre orchestrators.

4.2.2 Processament de trànsit Wi-Fi

El trànsit capturat es processa utilitzant dissectors del protocol IEEE 802.11. Inicialment es van començar a desenvolupar des de zero aquests dissectors. Però la complexitat del protocol 802.11 i l'existència d'eines madures i provades que ja l'implementen correctament, van ser decisius per no re-inventar la roda i centrar els esforços en altres parts del projecte.

Tot i no realitzar la implementació total, els estudis i les tasques desenvolupades en aquest àmbit han servit per tenir un major coneixement del protocol i aprofundir en el funcionament dels *drivers* i subsistema 802.11 de les distribucions GNU/Linux.

Aquests dissectors s'encarreguen principalment de fer operacions a nivell de bit per interpretar les dades contingudes a les trames Wi-Fi segons ho especificat al estàndard IEEE 802.11[2]. Per exemple el format principal de les trames IEEE 802.11 defineix el que es coneix com a format MAC (de l'anglès *Medium Access Control*):

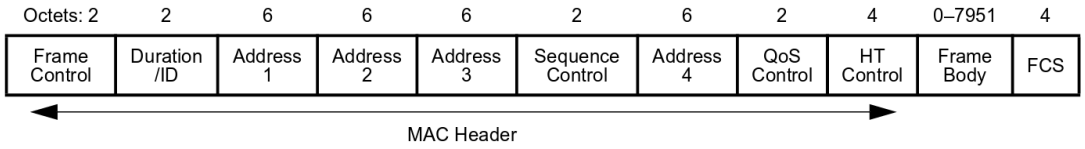


Figure 8-1—MAC frame format

Figura 4-8 Format de trama MAC

(font de la imatge: estàndard IEEE 802.11[2]*)*

Els dos primers octets o 16 bits es corresponen amb el camp de control de trama (*Frame Control*), que a l'hora està definit amb els camps:

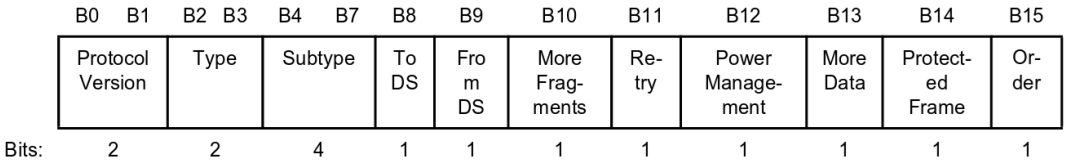


Figure 8-2—Frame Control field

Figura 4-9 Camp de control de trama

(font de la imatge: estàndard IEEE 802.11[2]*)*

A on els possibles valors dels camps tipus i subtipus (*Type* i *Subtype*) defineixen el tipus de trama:

Table 8-1—Valid type and subtype combinations

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110	Timing Advertisement

Figura 4-10 Fragment dels possibles valors de tipus i subtipus de trames

(font de la imatge: estàndard IEEE 802.11 [2]*)*

Per exemple una trama amb els bits de tipus a 00 i els de subtipus 0100 es correspondria amb una trama de gestió, específicament amb un *Probe request*.

Seguint aquesta aproximació per la resta de l'estàndard IEEE 802.11 els dissectors son capaços d'interpretar els diferents camps i informació continguda a les trames capturades.

Durant el processament inicial s'inspeccionen totes les trames 802.11 capturades i se n'extreu informació rellevant que posteriorment serà indexada per poder ser analitzada. Aquesta informació s'associa a la xarxa i/o als dispositius en concret que la generen, per obtenir una visió complerta del flux de comunicacions i poder estudiar-ne el comportament per separat (per exemple per només analitzar la informació relacionada amb una xarxa en concret).

El resum de la informació principal que generen els sensors un cop processat el trànsit Wi-Fi, és el següent:

Camp	Descripció
bssid	Adreça de la xarxa (*Basic Service Set Identifier*)
channel	Canal en el qual s'ha capturat el paquet (es correspon amb una freqüència de radio utilitzada pel protocol 802.11)
cloaked	Booleà que especifica si la xarxa oculta el seu ESSID
crypt_packets	Número de paquets xifrats que s'han vist per a una xarxa o estació en concret

Camp	Descripció
data_packets	Número de paquets de dades que s'han vist per una xarxa o estació en concret
datasize	Dimensions totals de tots els paquets de dades que s'han vist per una xarxa o estació en concret
encryption	Mecanisme de xifrat que s'utilitza (WEP, WPA+AES, ...)
essid	Nom de la xarxa (*Extended Service Set Identifier*)
first_seen	Data i hora a la qual s'ha vist per primer cop algun paquet d'una determinada xarxa o estació
fragmented_packets	Número de paquets fragmentats que s'han vist per una xarxa o estació en concret
last_seen	Data i hora a la qual s'ha vist per darrera vegada algun paquet d'una determinada xarxa o estació
last_signal	Qualitat de senyal amb la qual s'ha rebut el darrer paquet d'una xarxa o estació determinada. Aquesta informació s'obté de les capçaleres *radiotap headers*[11]
llc_packets	Número de paquets de la capa LLC *(Logical Link Control)* que s'han vist per una xarxa o estació en concret
mac	Adreça MAC d'una estació
manufacturer	Fabricant associat a l'adreça de la xarxa o dispositiu. Aquesta informació s'obté correlant les adreces MAC amb el llistat d'adreces MAC reservades a fabricants.
max_signal	Millor qualitat de senyal amb la qual s'ha rebut algun paquet d'una xarxa o estació determinada. Aquesta informació s'obté de les capçaleres *radio headers*
min_signal	Menor qualitat de senyal amb la qual s'ha rebut algun paquet d'una xarxa o estació determinada. Aquesta informació s'obté de les capçaleres *radio headers*
probed_essids	Nom de les xarxes per les quals un determinat client ha preguntat mitjançant l'enviament de trames de tipus *probe request*

[11] Radiotap header: *"The radiotap header format is a mechanism to supply additional information about frames, from the driver to userspace applications"* - http://www.radiotap.org/

Camp	Descripció
retried_packets	Número de paquets retransmesos que s'han vist per una xarxa o estació en concret
total_packets	Número total de paquets que s'han vist per una xarxa o estació en concret

Taula 7 Llistat de camps extrets del trànsit 802.11

Al mateix temps es compara la informació obtinguda amb un motor de detecció d'intrusions que genera alertes de seguretat. Aquest motor es capaç de generar alertes basades en signatures o en tendències, sense estat o basades en el context.

Un exemple d'alerta basada en una signatura és per exemple aquella que busca una cadena concreta en una trama 802.11. Per altra banda una alerta basada en una tendència és aquella que es genera quan es detecta un elevat nombre de trames de des-autenticació, fet altament inusual en el funcionament normal d'una xarxa Wi-Fi.

Les alertes generades s'escriuen en un fitxer de *log* a on s'especifica la data i hora a la qual s'ha generat l'alerta i els detalls de la mateixa.

A propers apartats d'aquest document s'aprofundeix en els formats i informació generada pels sensors.

4.3 Emmagatzemament i indexació

4.3.1 Processament de dades

Tal i com s'ha descrit a l'apartat anterior el processament inicial del trànsit Wi-Fi per extreure'n els camps més rellevants es realitza a cada sensor durant la mateixa captura de trànsit. S'ha decidit fer-ho així principalment per segmentar les tasques donat que és menys costós a nivell computacional fer aquest processament durant la mateixa captura que posteriorment un cop centralitzat el trànsit. A més s'optimitza la transmissió i emmagatzemament de dades ja que només s'inclou un resum de les dades capturades i no la seva totalitat.

Les dades generades pels sensors són centralitzades a un únic punt[12] a on s'agrupen i es tornen a processar per preparar-les pel seu emmagatzemament i indexació. Aquest punt pot coincidir amb el servidor de cerca, però no és obligatori.

Els sensors emmagatzemen les captures complertes en un buffer circular (que depèn de la capacitat d'emmagatzemament dels sensors). En cas de ser necessari durant un anàlisi, és possible accedir a totes les dades capturades en cru per analitzar-les en profunditat.

4.3.2 Indexació a servidor de cerca

Una vegada centralitzades les dades aquestes són indexades al servidor de cerca.

El concepte d'índex, segons el servidor de cerca, fa referència principalment al mecanisme d'organització d'informació utilitzat. En la seva basant més simple i familiar un índex funciona de forma molt similar al d'una base de dades relacional:

- Gestor relacional → Base de dades → Taules → Columnes i entrades
- Servidor de cerca → Índexs → Tipus → Propietats i entrades

Però permet una flexibilitat major a la de les abstraccions basades en bases de dades i taules. Per exemple un servidor de cerca és sovint utilitzat per enregistrar entrades de bitàcola o log. Per aquest propòsit es pot utilitzar un nou índex per cada dia facilitant la cerca per dies en concret o per períodes. Per a poder tractar les dades generades pels sensors s'ha decidit crear diferents índexs que en facilitin la seva posterior consulta:

- Índexs per sensor: D'aquesta forma es pot consultar de forma independent la informació d'un únic sensor. O la de tots els sensors.
- Índexs temporals: Cada cert període de temps es crea un nou índex per limitar les dades a analitzar. Per exemple si el que s'està intentant és detectar intents d'intrusió que s'estiguin dur a terme ara mateix, per prevenir-los, interessa fer cerques sobre les dades dels darrers instants i no sobre dades de l'any passat.
- Índexs per tipus de dades: S'han separat els índexs segons la tipologia de dades que inclouen, per una banda la relacionada amb dades de les estacions i clients. I per altra les alertes de seguretat generades pel motor de detecció d'intrusions.

El procés que es segueix per a la seva creació és el descrit a continuació:

[12] S'han explorat alternatives per fer també aquestes tasques de forma distribuïda, donat que el programari utilitzat ho permet, però finalment s'ha deixat com a treball de futur per no dedicar esforços necessaris en cobrir els requeriments i objectius inicials del projecte en els terminis previstos.

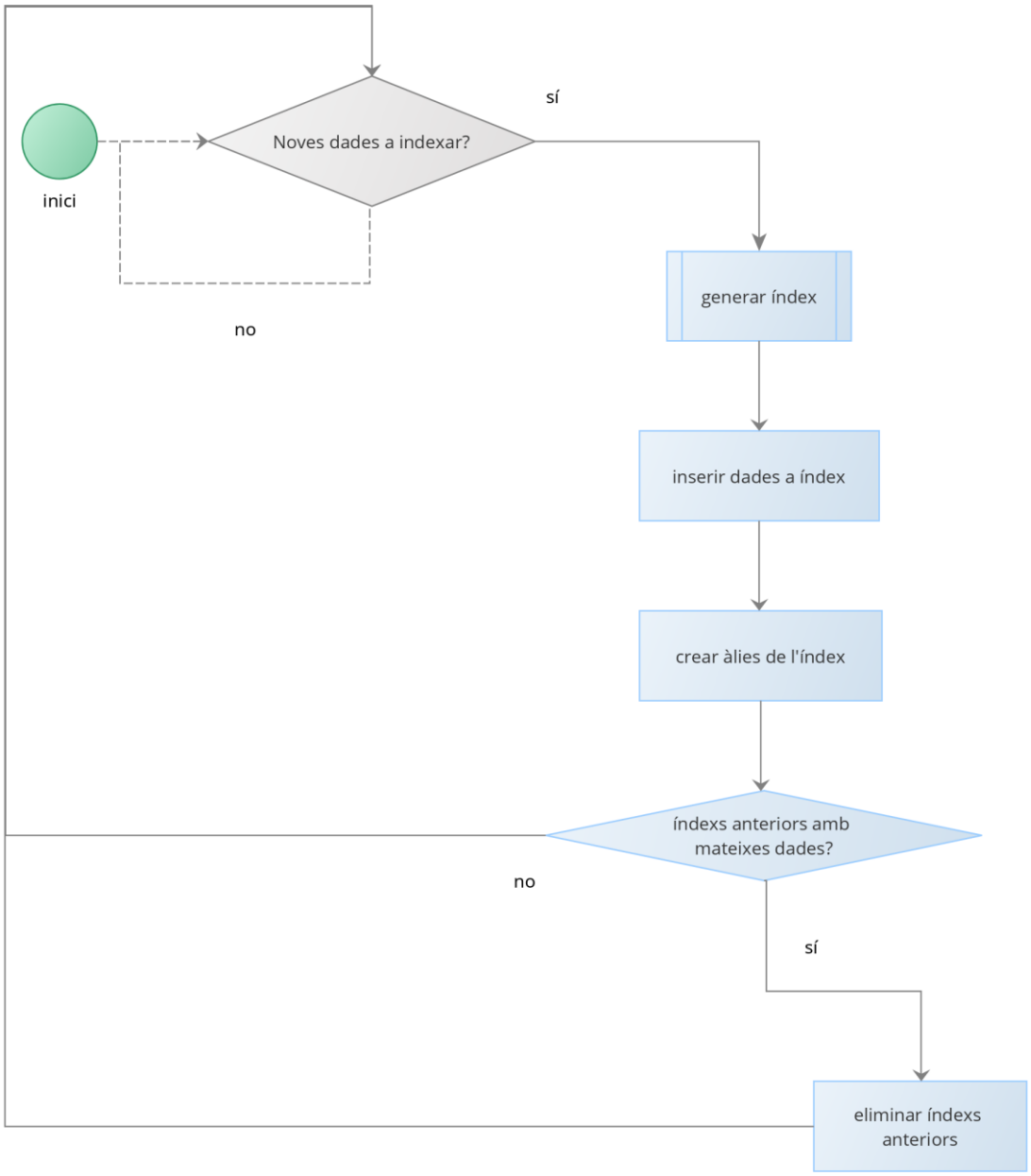


Figura 4-11 Diagrama d'indexació de noves dades

1. S'executa un procés de tipus *watchdog* que detecta quan hi ha dades noves d'un sensor.
2. Quan hi ha noves dades es genera dinàmicament un nou índex que inclou el tipus de dades a carregar, el nom del sensor que les ha transmès i una marca de data i hora.
3. Es carreguen les noves dades al nou índex, utilitzant les eines que proporciona la implementació escollida del servidor de cerca.
4. Es crea un nou àlies del índex creat, per tal de que sempre es pugui fer servir el mateix nom d'índex per a realitzar cerques.

5. Si les noves dades carregades engloben dades carregades anteriorment, s'eliminen del servidor de cerca els índexs que les contenen per evitar duplicats. S'ha decidit fer-ho d'aquesta forma al corroborat que eliminar índexs és molt més eficient que eliminar o modificar dades d'un índex a la solució de servidor de cerca escollida.

4.4 Anàlisi i explotació de la informació

4.4.1 Cerques i vistes

Un cop es disposa de tota la informació centralitzada i indexada al servidor de cerca és possible analitzar-la per detectar-ne patrons de comportament, detectar anomalies o extreure'n informació rellevant segons diferents casos d'ús.

Per a fer-ho s'utilitza el que s'ha anomenat servidor d'anàlisi, que no deixa de ser una interfície per accedir a les dades contingudes al servidor de cerca, i que de fet es pot executar a un mateix sistema.

Per defecte s'han definit una sèrie de cerques i vistes genèriques i versàtils que serveixen per fer-se ràpidament una idea de què està succeint en l'espectre radioelèctric IEEE 802.11 monitorat. Aquestes cerques són accessibles des d'una interfície d'usuari basada en una aplicació Web.

Exemples d'aquestes cerques i visualitzacions són:

- Llistat i característiques principals de les xarxes Wi-Fi presents
- Llistat i característiques dels dispositius i estacions Wi-Fi que intervenen a les comunicacions
- Llistat de les darreres alertes de seguretat generades en funció del trànsit analitzat
- Enumeració de les xarxes i clients que generen més trànsit

S'han agrupat en taulers de visualització (*dashboards*) per tal de que un usuari del sistema pugui veure aquesta informació agrupada i de forma fàcil.

D'altra banda la interfície de cerca implementa un potent motor de consulta que atorga a un usuari experimentat molta flexibilitat a l'hora de buscar, ordenar i visualitzar informació. Entre d'altres permet:

- Definir filtres temporals
- Especificar l'índex o conjunt d'índexs sobre els quals cercar dades

- Cercar cadenes de text, valors o rangs de valors en qualsevol camp de les dades
- Cercar valors o rangs específics per qualsevol camp o conjunt de camps
- Utilitzar comodins de cerca
- Utilitzar cerques difuses (*fuzzy*)
- Utilitzar operadors booleans a les cerques
- Agrupar clàusules de cerca per formar cerques sobre cerques

5 Implementació

Aquest apartat il·lustra el sistema de monitoratge des de la perspectiva d'un perfil tècnic, per alguns components més pròpiament d'un desenvolupador i per d'altres d'un operador. S'aprofundeix en detalls d'implementació, integració i/o configuració de components del sistema que s'han vist a l'apartat anterior.

5.1 Captura de trànsit

5.1.1 Implementació sensors

Requeriments

Tal i com s'ha descrit al apartat de disseny els requeriments de maquinari i programari dels sensors és molt genèric, de forma que es poden implementar sobre gran quantitat de dispositius i sistemes.

Els principals requeriments a nivell de programari són els següents:

- Intèrpret de codi Python[13] (≥ 2.7 o ≥ 3.4)
  - Per a l'execució d'utilitats de tractament i conversió de dades desenvolupades en aquest llenguatge
- Intèrpret de comandes conforme amb l'estàndard POSIX[14]
  - Per a l'execució de programari desenvolupat en llenguatge *shell script*.

[13] Python - https://www.python.org/
[14] POSIX - http://pubs.opengroup.org/onlinepubs/9699919799/

- Suport pel subsistema Inotify[15] del kernel de Linux
  - Utilitzat al programari desenvolupat per monitorar canvis en el sistema de fitxers sense tenir que realitzar tècniques de *polling,* de forma més eficient.
- rsync[16] amb suport SSH amb autenticació per certificat
  - Utilitzat per a la transmissió segura de dades entre els sensors i altres elements del sistema
- Kismet Wireless[17] (≥ 2010-07-R1-1)

La majoria de distribucions GNU/Linux incorporen per defecte o als repositoris oficials aquest programari, pel que són requeriments fàcils de satisfer.

A nivell de requeriments de maquinari, durant la realització d'aquest projecte s'ha utilitzat maquinari divers, en el pitjor dels casos amb les següents característiques:

processador	RAM	Flash	Emmagatzemament	Ethernet	802.11
400MHz	32MiB	4MiB	8GB	1x10/100	1xb/g/n suport mode monitor

Taula 8 Característiques mínimes del maquinari dels sensors

Pel que es pot veure es tracta d'uns requeriments modests i fàcilment assumibles amb dispositius de baix cost. A l'apartat 6 Implantació es detallen els models de maquinari utilitzats durant les proves, les seves característiques principals i com adaptar-los per ser utilitzats com a sensors.

Detalls d'implementació

La tasca principal dels sensors és la de capturar trànsit 802.11. Tot i existir múltiples alternatives per fer la captura s'ha decidit utilitzar el programari Kismet Wireless pel conjunt de funcionalitats que proporciona, la seva estabilitat i maduresa.

S'ha definit a nivell de sistema un servei per engegar automàticament el component servidor de Kismet en segon pla (*background*). D'aquesta forma s'inicia el procés de captura de forma automàtica un cop s'ha inicialitzat el sistema del sensor.

S'ha programat una peça de codi per tal de poder configurar de forma dinàmica i automàtica Kismet. D'aquesta forma s'utilitza una plantilla de configuració que s'adapta al nombre d'interfícies que disposa el sensor i als canals IEEE 802.11 que ha de monitorar

[15] Inotify - http://man7.org/linux/man-pages/man7/inotify.7.html
[16] Rsync - https://rsync.samba.org/
[17] Kismet Wireless - https://www.kismetwireless.net/

en un moment donat. S'ha adjuntat aquesta plantilla de configuració i la seva documentació a l'annex A Sensors: plantilla de configuració servidor Kismet.

Aquest procés de captura genera de forma periòdica diferents sortides:

- Captura complerta del trànsit en format pcap
- Resum de les xarxes i clients Wi-Fi vistos al trànsit capturat, en format netxml
- Log d'alertes de seguretat generades pel motor de detecció d'intrusions de Kismet

Al moment d'inicialitzar el sistema també s'inicia un procés de monitoratge o *watchdog*. Aquest procés s'encarrega de processar les dades generades pel procés de captura, cada vegada que aquest les escriu, i de transmetre'n la informació resultant al servidor d'emmagatzemament. Per defecte aquest procés es realitza cada 60 segons, per tenir dades en quasi temps real, però es podria modificar per fer-ho amb més o menys freqüència.

Per a la generació d'alertes de seguretat s'han utilitzat les següents regles i paràmetres suportats pel motor de detecció d'intrusions que incorpora Kismet:

- APSPOOF
- BSSTIMESTAMP
- CHANCHANGE
- CRYPTODROP
- DEAUTHFLOOD
- BCASTDISCON
- DHCPCLIENTID
- DHCPCONFLICT
- DISASSOCTRAFFIC
- DISCONCODEINVALID
- DEAUTHCODEINVALID
- DHCPNAMECHANGE
- DHCPOSCHANGE
- LONGSSID
- LUCENTTEST
- MSFBCOMSSID
- MSFDLINKRATE

- MSFNETGEARBEACON
- NETSTUMBLER
- NULLPROBERESP
- PROBENOJOIN

A la documentació oficial del Kismet[5] es poden consultar els detalls de cadascuna d'aquestes regles:

APSPOOF Fingerprint

A list of valid MAC addresses for a SSID may be given via the 'apspoof=' configuration file option. If a beacon or probe response for that SSID is seen from a MAC address not in that list, this alert will be raised. This can be used to detect conflicting access points, spoofed access points, or attacks such as Karma/Airbase which respond to all probe requests.

The 'apspoof=' configuration option can specific exact SSID matches, regular expressions (if Kismet is compiled with PCRE support), and single, multiple, or masked MAC addresses:

apspoof=Foo1:ssidregex="(?i:foobar)",validmacs=00:11:22:33:44:55

apspoof=Foo2:ssid="Foobar",
validmacs="00:11:22:33:44:55,AA:BB:CC:DD:EE:FF"

When multiple MAC addresses are specified, they should be enclosed in quotes (as above).

For more information about forming PCRE-compatible regular expressions, see the PCRE docs (man pcrepattern).

BSSTIMESTAMP Trend/Stateful

Invalid/Out-of-sequence BSS Timestamps can indicate AP spoofing. APs with fluctuating BSS timestamps could be suffering an "evil twin" spoofing attack, as many tools do not attempt to sync the BSS timestamp at all, and the fine-grained nature of the BSS timestamp field makes it difficult to spoof accurately. Some APs may reset the BSS timestamp regularly, leading to a false-positive.

References:

WVE-2005-0019

CHANCHANGE Trend/Stateful

A previously detected access point changing channels may indicate a spoofing attack. By spoofing a legitimate AP on a different channel, an attacker can lure clients to the spoofed access point. An AP changing channel during normal operation may indicate such an attack is in process, however centrally managed networks may automatically change AP channels to less-used areas of the spectrum.

References:

WVE-2005-0019

CRYPTODROP Trend/Stateful

Spoofing an AP with less-secure encryption options may fool clients into connecting with compromised credentials. The only situation in which an access point should reduce encryption security is when the AP is reconfigured.

DEAUTHFLOOD Trend/Stateful
BCASTDISCON Trend/Stateful

By spoofing disassociate and deauthenticate packets an attacker may disconnect clients from a network, causing a denial-of-service which lasts only as long as the attacker is able to send the packets.

References:

WVE-2005-0019, WVE-2005-0045, WVE-2005-0046, WVE-2005-0061
http://802.11ninja.net
http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf

DHCPCLIENTID Fingerprint

A client which sends a DHCP DISCOVER packet containing a Client-ID tag (Tag 61) which doesn't match the source MAC of the packet may be doing a DHCP denial-of-service to exhaust the DHCP pool.

DHCPCONFLICT Trend/Stateful

Clients which receive a DHCP address and continue to use a different IP address may indicate a misconfigured or spoofed client.

DISASSOCTRAFFIC Trend/Stateful

A client which is disassociated from a network should not immediately continue exchanging data. This can indicate a spoofed client attempting to incorrectly inject data into a network, or can indicate a client being the victim of a denial-of-service attack.

DISCONCODEINVALID Fingerprint
DEAUTHCODEINVALID Fingerprint

The 802.11 specification defines valid reason codes for disconnect and deauthenticate events. Various client and access point drivers have been reported to improperly handle invalid/undefined reason codes.

DHCPNAMECHANGE Trend/Stateful
DHCPOSCHANGE Trend/Stateful

The DHCP configuration protocol allows clients to optionally put the hostname and DHCP client vendor/operating system in the DHCP Discover packet. These values should only change if the client has changed drastically (such as a dual-boot system). Changing values can often indicate a client spoofing/MAC cloning attack.

LONGSSID Fingerprint

The 802.11 specification allows a maximum of 32 bytes for the

SSID. Over-sized SSIDs are indicative of an attack attempting to exploit vulnerabilities in several drivers.

LUCENTTEST Fingerprint Deprecated

Old Lucent Orinoco cards in certain scanning test modes generate identifiable packets.

MSFBCOMSSID Fingerprint

Some versions of the Windows Broadcom wireless drivers do not properly handle SSID fields longer than the 802.11 specification, leading to system compromise and code execution. This vulnerability is exploited by the Metasploit framework.

References:

WVE-2006-0071

MSFDLINKRATE Fingerprint

Some versions of the Windows D-Link wireless drivers do not properly handle extremely long 802.11 valid rate fields, leading to system compromise and code execution. This vulnerability is exploited by the Metasploit framework.

References:

WVE-2006-0072

MSFNETGEARBEACON Fingerprint

Some versions of the Windows netgear wireless drivers do not properly handle over-sized beacon frames, leading to system compromise and code execution. This vulnerability is exploited by the Metasploit framework.

NETSTUMBLER Fingerprint Deprecated

Older versions of Netstumbler (3.22, 3.23, 3.30) generate, in certain conditions, specific packets.

NULLPROBERESP Fingerprint

Probe-response packets with a SSID IE tag component of length 0 can cause older cards (prism2, orinoco, airport-classic) to fail.

References:

WVE-2005-0019

PROBENOJOIN Trend/Stateful

Active scanning tools such as Netstumbler constantly send network discovery probes but never join any of the networks which respond. This alert can cause excessive false positives while channel hopping, and is disabled by default.

5.1.2 Implementació xarxa de sensors

Requeriments

Per tal d'implantar la xarxa distribuïda de sensors s'han utilitzat estàndards i llibreries de codi obert madures i testejades.

El gruix de la lògica de la xarxa de sensors està implementat utilitzant les llibreries de missatgeria asíncrona ZeroMQ[18], també coneguda com a ØMQ, 0MQ o ZMQ i el *framework* Zyre[19] (i el seu port a Python Pyre[20]) per aplicacions *peer-to-peer* basades en proximitat.

Concretament s'han utilitzat les versions:

- ZeroMQ Stable Release 4.1.5
- Zyre - https://github.com/zeromq/zyre i Pyre - https://github.com/zeromq/pyre

Tot i que qualsevol implementació que compleixi amb l'estàndard ZRE[6][21] hauria de ser compatible.

Detalls d'implementació

Un dels problemes als quals s'arriba quan es dissenyen arquitectures distribuïdes és el del descobriment. De quina forma un component coneix l'existència d'altres components? Això és especialment complex quan algunes d'aquestes peces poden estar o no estar disponibles en diferents instants.

L'aproximació més senzilla és evitar l'autodescobriment i utilitzar una configuració prefixada de forma que la configuració de la xarxa es fa a mà. En aquest cas quan s'afegeix un nou component, o un deixa d'estar disponible, s'ha de dur a terme una re-configuració perquè la resta de components ho sàpiguen.

Un dels requeriments d'aquest projecte és que el desplegament inicial i posterior de sensors pugui ser simple i dinàmic. És per això que l'aproximació simple de tenir que configurar-ho tot a mà no és la més òptima. Per aquest motiu s'han utilitzat mecanismes d'autodescobriment.

---

[18] ZeroMQ: *A a high-performance asynchronous messaging library, aimed at use in distributed or concurrent applications* - http://zeromq.org/

[19] Zyre: *Zyre provides reliable group messaging over local area networks* - https://github.com/zeromq/zyre

[20] Pyre: *Python port of Zyre 1.0, implementing the same ZRE protocol* - https://github.com/zeromq/pyre

[21] ZRE: *ZeroMQ Realtime Exchange Protocol* - http://rfc.zeromq.org/spec:36/ZRE/

Aquest autodescobriment, i en general la implementació de la xarxa distribuïda de sensors, es basa en:

- L'enviament de paquets de tipus UDP *broadcast*
- L'enviament de paquets de tipus *heartbeat* per saber que els elements de la xarxa segueixen actius

Els paquets de tipus UDP broadcast permeten fer arribar missatges a tots els elements que formin part d'una mateixa xarxa. D'aquesta forma quan un element està disponible anuncia la seva presència a la resta. Per la seva banda els paquets de tipus *heartbeat* son paquets que es transmeten de forma periòdica per saber que un element segueix estant a la xarxa.

Les llibreries utilitzades permeten un nivell d'abstracció elevat i s'encarreguen de la transmissió d'aquests paquets de forma transparent. De forma molt simplificada la seva API[7] permet les principals operacions:

- Constructor: crea un nou node de tipus Zyre. Un node es correspon amb un sensor o un altre element de la xarxa.

```
1. import pyre
2. #  Constructor, creates a new Zyre node. Note that until you start the
3. #  node it is silent and invisible to other nodes on the network.
4. node = pyre.Pyre()
```

- Configuració de les capçaleres del node: aquestes capçaleres poden incloure informació del sensor que la resta d'elements de la xarxa poden veure.

```
1. #  Set node header; these are provided to other nodes during discovery
2. #  and come in each ENTER message.
3. node.set_header(name, value)
```

- Iniciar: s'inicia la fase de descobriment i connexió.

```
1. #  (TODO: Currently a Pyre node starts immediately) Start node, after setting h
   eader values. When you start a node it
2. #  begins discovery and connection.
3. node.start()
```

- Aturar: s'anuncia a la resta de nodes que es deixa d'estar disponible. És un missatge de cortesia, si s'atura de forma inesperada la resta de nodes se n'assabenten igualment pel mecanisme de *heartbeat*.

```
1. #  Stop node, this signals to other peers that this node will go away.
2. #  This is polite; however you can also just destroy the node without
3. #  stopping it.
4. node.stop()
```

- Entrar i abandonar grup: el protocol permet crear grups de nodes per si es vol acotar l'abast dels missatges. En aquest projecte s'utilitza un únic grup per a tots els sensors.

```
1. #  Join a named group; after joining a group you can send messages to
2. #  the group and all Zyre nodes in that group will receive them.
```

```
node.join(group)

#  Leave a group
node.leave(group)
```

- Rebre missatge: rebre el següent missatge de la xarxa. Pot ser un missatge enviat a un grup, només a aquest node en concret, o tractar-se d'un missatge de control com que ha entrat o abandonat la xarxa un node.

```
#  Receive next message from network; the message may be a control
#  message (ENTER, EXIT, JOIN, LEAVE) or data (WHISPER, SHOUT).
#  Returns a list of message frames
msgs = node.recv();
```

- Enviar missatge a un únic node: s'envia un missatge a un node en concret.

```
# Send message to single peer, specified as a UUID object (import uuid)
# Destroys message after sending
node.whisper(peer, msg)
```

- Enviar missatge a tot un grup: s'envia un missatge a tots els nodes que pertanyen a un grup.

```
# Send message to a named group
# Destroys message after sending
node.shout(group, msg);
```

Mitjançant aquestes primitives els sensors anuncien la seva presència a la resta de sensors i elements de la xarxa.

Quan l'element de tipus *orchestrator* detecta l'entrada d'un nou sensor en demana les capacitats, principalment el número d'interfícies i a quines freqüències pot treballar. Amb aquesta informació planifica quines freqüències ha de monitorar aquest sensor en concret i li especifica. Si s'escau envia missatges de re-configuració a la resta de sensors per poder cobrir l'espectre radioelèctric a monitorar de la forma més eficient possible.

```
// GetChannels returns a string with all the supported frequencies
func GetChannels() string {
    cmd := exec.Command("sh", "-c", "iw list | grep --color=never dBm | grep --color=never -oP '\\[\\K[^\\]]+'")

    out, err := cmd.Output()
    if err != nil {
        log.Fatal(err)
    }
    channels := strings.Replace(string(out[:len(out)-1]), "\n", ",", -1)
    return channels
}
```

Figura 5-1 Codi per llistar les freqüències permeses per les interfícies del sensor

A mode d'exemple, es suposa el següent cas d'ús del sistema:

1. Es requereix el monitoratge dels canals de radio 1 i 6 de l'estàndard IEEE 802.11. Donat que es volen protegir les xarxes corporatives que treballen en aquests canals.
2. Inicialment hi ha un únic sensor amb una única interfície monitorant els dos canals. Per poder cobrir els dos canals amb una sola interfície s'ha de fer el que es coneix com a *channel hopping*, que es tracta d'anar canviant de canal cada cert interval de temps de forma equitativa. Aquesta aproximació te la limitació que la meitat del temps es captura trànsit d'un canal i l'altra de l'altre, fent que es perdi trànsit que podria ser rellevant.
3. S'engega un nou sensor, que automàticament entra a formar part de la xarxa distribuïda.
4. *L'orchestrator* ho detecta i li demana el llistat de capacitats.
5. El nou sensor li respon al orchestrator indicant que disposa d'una interfície de xarxa que pot treballar en qualsevol dels canals de l'estàndard IEEE 802.11 a les freqüències de 2.4Ghz o 5Ghz.
6. *L'orchestrator*, mitjançant un algorisme, decideix que en aquest cas el millor és que el nou sensor capturi el trànsit del canal 6 i el que ja formava part de la xarxa el canal 1.
7. *L'orchestrator* envia un missatge de configuració al nou sensor dient-li que capturi el trànsit del canal 6, i un missatge de re-configuració al sensor inicial perquè capturi el trànsit del canal 1.
8. Ambdós sensors configuren automàticament les seves interfícies de xarxa per treballar en el canal especificat i inicien la captura del trànsit.

La lògica de l'algorisme utilitzat per *l'orquestator* es pot resumir en:

- Si hi ha el mateix nombre de sensors i interfícies de xarxa que els canals a monitorar, s'adjudica un canal a cada sensor.
- Si hi ha menys sensors i interfícies de xarxa que els canals a monitorar, s'adjudica un canal a cada sensor i els que manquen es distribueixen entre els sensors per ser capturats mitjançant *channel hopping*.
- Si hi ha més sensors i interfícies de xarxa que els canals a monitorar, s'adjudica un canal diferent a cada sensor i la resta de sensors es distribueixen la resta de canals possibles de l'estàndard per ser capturats mitjançant *channel hopping*.

5.1.3 Processament de dades capturades

Els sensors generen dos tipus de dades, un resum de les xarxes i estacions 802.11 vistes en el trànsit capturat i un fitxer de *log* amb alertes generades pel motor de detecció d'intrusions (abreviat com IDS de l'anglès *Intrusion Detection System*).

En aquest apartat se'n detalla el format.

netxml

El format netxml es un format considerat estàndard *de-facto* donat que diverses de les principals eines de captura de trànsit Wi-Fi l'implementen. Va ser definit i implementat inicialment en el marc del projecte Kismet Wireless[22][5] i es tracta d'un format XML estàndard a on s'agrupa informació de xarxes i estacions 802.11.

La generació d'aquest tipus de fitxers és duta a terme per l'eina Kismet. Tot i que la configuració del servidor de Kismet s'ha vist a l'apartat anterior, a continuació es mostren els paràmetres referents a la creació de fitxers netxml:

```
# Prefix on es guardaran els logs, inclosos els fitxers netxml
logprefix=/data/kismet/

# Cada quan s'escriuen els fitxers de log en segons
# Per defecte els sensors es configuren per transmetre les dades cada 60 segons
# és un bon compromís per tenir dades en quasi temps-real sense saturar les
# comunicacions a la xarxa de sensors ni realitzar moltes escriptures a disc
writeinterval=60

# Llista de formats pels quals s'escriu un fitxer de log
# XML es correspon amb el format netxml
logtypes=pcap,xml,gps,text,alert
```

Figura 5-2 Configuració referent a *logs* netxml a *kismet_server.conf*

A continuació es mostra un exemple d'element present en un fitxer netxml generat durant la realització de proves:

```
<wireless-network number="28" type="infrastructure" first-time="Wed Aug 24 10:57:59 2016" last-time="Wed Aug 24 12:18:15 2016">
  <SSID first-time="Wed Aug 24 10:58:22 2016" last-time="Wed Aug 24 12:18:15 2016">
      <type>Beacon</type>
      <max-rate>54.000000</max-rate>
      <packets>82</packets>
      <beaconrate>4</beaconrate>
      <wps>No</wps>
      <encryption>None</encryption>
      <essid cloaked="false">GironaFreeWIFI</essid>
  </SSID>
  <BSSID>00:0D:97:18:4A:40</BSSID>
  <manuf>Abb/Trop</manuf>
  <channel>11</channel>
  <freqmhz>2457 2</freqmhz>
  <freqmhz>2462 157</freqmhz>
  <maxseenrate>18000</maxseenrate>
  <carrier>IEEE 802.11b+</carrier>
  <encoding>CCK</encoding>
   <packets>
    <LLC>82</LLC>
    <data>77</data>
    <crypt>0</crypt>
    <total>159</total>
```

[22] Kismet Wireless - https://www.kismetwireless.net/

```
      <fragments>0</fragments>
      <retries>0</retries>
    </packets>
    <datasize>18826</datasize>
   <snr-info>
     <last_signal_dbm>-91</last_signal_dbm>
     <last_noise_dbm>0</last_noise_dbm>
     <last_signal_rssi>0</last_signal_rssi>
     <last_noise_rssi>0</last_noise_rssi>
     <min_signal_dbm>-96</min_signal_dbm>
     <min_noise_dbm>0</min_noise_dbm>
     <min_signal_rssi>1024</min_signal_rssi>
     <min_noise_rssi>1024</min_noise_rssi>
     <max_signal_dbm>-81</max_signal_dbm>
     <max_noise_dbm>-256</max_noise_dbm>
     <max_signal_rssi>0</max_signal_rssi>
     <max_noise_rssi>0</max_noise_rssi>
   </snr-info>
   <bsstimestamp>Aug 24 12:15:22</bsstimestamp>
   <cdp-device></cdp-device>
   <cdp-portid></cdp-portid>
   <seen-card>
    <seen-uuid>9bdc5b02-69d8-11e6-bd30-5604c5202902</seen-uuid>
    <seen-time>Wed Aug 24 12:18:15 2016</seen-time>
    <seen-packets>159</seen-packets>
   </seen-card>
   <wireless-client number="1" type="fromds" first-
  time="Wed Aug 24 10:57:59 2016" last-time="Wed Aug 24 12:18:15 2016">
     <client-mac>00:0D:97:18:4A:40</client-mac>
     <client-manuf>Abb/Trop</client-manuf>
     <channel>11</channel>
     <freqmhz>2462 145</freqmhz>
     <maxseenrate>18000</maxseenrate>
     <carrier>IEEE 802.11b+</carrier>
     <encoding>CCK</encoding>
      <packets>
        <LLC>82</LLC>
        <data>63</data>
        <crypt>0</crypt>
        <total>145</total>
        <fragments>0</fragments>
        <retries>0</retries>
      </packets>
      <datasize>17934</datasize>
     <snr-info>
       <last_signal_dbm>-91</last_signal_dbm>
       <last_noise_dbm>0</last_noise_dbm>
       <last_signal_rssi>0</last_signal_rssi>
       <last_noise_rssi>0</last_noise_rssi>
       <min_signal_dbm>-93</min_signal_dbm>
       <min_noise_dbm>0</min_noise_dbm>
       <min_signal_rssi>1024</min_signal_rssi>
       <min_noise_rssi>1024</min_noise_rssi>
       <max_signal_dbm>-81</max_signal_dbm>
       <max_noise_dbm>-256</max_noise_dbm>
       <max_signal_rssi>0</max_signal_rssi>
       <max_noise_rssi>0</max_noise_rssi>
     </snr-info>
     <seen-card>
      <seen-uuid>9bdc5b02-69d8-11e6-bd30-5604c5202902</seen-uuid>
      <seen-time>Wed Aug 24 12:18:15 2016</seen-time>
      <seen-packets>145</seen-packets>
     </seen-card>
```

```
  </wireless-client>
  <wireless-client number="2" type="tods" first-time="Wed Aug 24 12:06:40 2016" last-time="Wed Aug 24 12:06:40 2016">
    <client-mac>18:9E:FC:B4:94:16</client-mac>
    <client-manuf>Apple</client-manuf>
    <channel>0</channel>
    <freqmhz>2462 3</freqmhz>
    <maxseenrate>2000</maxseenrate>
     <packets>
       <LLC>0</LLC>
       <data>3</data>
       <crypt>0</crypt>
       <total>3</total>
       <fragments>0</fragments>
       <retries>0</retries>
     </packets>
     <datasize>72</datasize>
    <snr-info>
      <last_signal_dbm>-84</last_signal_dbm>
      <last_noise_dbm>0</last_noise_dbm>
      <last_signal_rssi>0</last_signal_rssi>
      <last_noise_rssi>0</last_noise_rssi>
      <min_signal_dbm>-84</min_signal_dbm>
      <min_noise_dbm>0</min_noise_dbm>
      <min_signal_rssi>1024</min_signal_rssi>
      <min_noise_rssi>1024</min_noise_rssi>
      <max_signal_dbm>-82</max_signal_dbm>
      <max_noise_dbm>-256</max_noise_dbm>
      <max_signal_rssi>0</max_signal_rssi>
      <max_noise_rssi>0</max_noise_rssi>
    </snr-info>
    <seen-card>
     <seen-uuid>9bdc5b02-69d8-11e6-bd30-5604c5202902</seen-uuid>
     <seen-time>Wed Aug 24 12:06:40 2016</seen-time>
     <seen-packets>3</seen-packets>
    </seen-card>
  </wireless-client>
  <wireless-client number="3" type="tods" first-time="Wed Aug 24 12:15:54 2016" last-time="Wed Aug 24 12:15:54 2016">
    <client-mac>28:A0:2B:D2:E5:DF</client-mac>
    <client-manuf>Apple</client-manuf>
    <channel>0</channel>
    <freqmhz>2462 1</freqmhz>
    <maxseenrate>1000</maxseenrate>
     <packets>
       <LLC>0</LLC>
       <data>1</data>
       <crypt>0</crypt>
       <total>1</total>
       <fragments>0</fragments>
       <retries>0</retries>
     </packets>
     <datasize>24</datasize>
    <snr-info>
      <last_signal_dbm>-88</last_signal_dbm>
      <last_noise_dbm>0</last_noise_dbm>
      <last_signal_rssi>0</last_signal_rssi>
      <last_noise_rssi>0</last_noise_rssi>
      <min_signal_dbm>-88</min_signal_dbm>
      <min_noise_dbm>0</min_noise_dbm>
      <min_signal_rssi>1024</min_signal_rssi>
      <min_noise_rssi>1024</min_noise_rssi>
      <max_signal_dbm>-88</max_signal_dbm>
```

```
147.       <max_noise_dbm>-256</max_noise_dbm>
148.       <max_signal_rssi>0</max_signal_rssi>
149.       <max_noise_rssi>0</max_noise_rssi>
150.     </snr-info>
151.     <seen-card>
152.      <seen-uuid>9bdc5b02-69d8-11e6-bd30-5604c5202902</seen-uuid>
153.      <seen-time>Wed Aug 24 12:15:54 2016</seen-time>
154.      <seen-packets>1</seen-packets>
155.     </seen-card>
156.   </wireless-client>
157. </wireless-network>
```

Figura 5-3 Exemple d'element de xarxa en format netxml

Per poder comprendre millor la figura anterior es citen a continuació alguns dels elements i atributs més rellevants:

- Un element de tipus <*wireless-network*> que es correspon amb la xarxa Wi-Fi "GironaFreeWIFI"
  - Els atributs d'aquest element denoten que aquesta xarxa va ser vista per primer cop el 24 d'Agost de 2016 a les 10:57:59 i per darrer el mateix dia a les 12:18:15. Aquest atribut es genera mirant quin va ser el primer i el darrer paquet capturat d'aquesta xarxa.
  - També es veu que és de tipus *infraestructura*, és a dir el model de xarxa més estès a contraposició de xarxes de tipus *ad-hoc* o altres modes permesos per l'estàndard 802.11.
  - Així mateix el sub-element <*encryption*> amb un valor "*None*" indica que es tracta d'una xarxa oberta i sense xifrat.
  - Un altre sub-element de la xarxa recull l'adreça de la mateixa, coneguda com a *BSSID* (*Basic Service Set Identifier*). En aquest cas amb el valor "00:0D:97:18:4A:40".
  - També hi ha presents gran quantitat d'altres elements que recullen informació detallada de la xarxa, tal i com el nombre de paquets que s'han vist per aquesta xarxa o la qualitat de la senyal amb la que s'han rebut.
- Tres sub-elements de tipus <*wireless-client*> que es corresponen amb clients de la xarxa.
  - Aquests a l'hora tenen sub-elements i atributs que en recullen informació tal com la seva adreça MAC, el nombre de paquets transmesos i la qualitat del senyal amb la qual s'han rebut.

alertes IDS

S'ha configurat l'eina Kismet per generar diverses alertes en funció del trànsit analitzat. A l'apartat anterior s'ha descrit la tipologia d'aquestes alertes i s'ha inclòs el detall de la configuració del motor de detecció d'intrusions.

El format de les alertes es pot considerar un format de *log* estàndard a on s'inclou una marca de temps (*timestamp*) i els detalls de l'alerta. A continuació es mostren algunes de les alertes generades durant la realització de proves, per poder veure'n el format:

```
1. Sun Aug 14 08:30:34 2016 CHANCHANGE 11 D4:7B:B0:56:93:75 D4:7B:B0:56:93:75 FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID D4:7B:B0:56:93:75 changed channel
    from 1 to 11
2. Sun Aug 14 08:38:06 2016 CHANCHANGE 1 F8:8E:85:DC:0F:9F F8:8E:85:DC:0F:9F FF:FF
   :FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID F8:8E:85:DC:0F:9F changed channel
   from 11 to 1
3. Sun Aug 14 10:04:45 2016 BCASTDISCON 0 02:1A:11:F9:FA:ED 02:1A:11:F9:FA:ED FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 02:1A:11:F9:FA:ED broadcast deaut
   henticate/disassociation of all clients, possible DoS
4. Sun Aug 14 10:04:45 2016 BCASTDISCON 0 02:1A:11:F9:FA:ED 02:1A:11:F9:FA:ED FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 02:1A:11:F9:FA:ED broadcast deaut
   henticate/disassociation of all clients, possible DoS
5. Sun Aug 14 10:04:48 2016 BCASTDISCON 0 02:1A:11:F9:FA:ED 02:1A:11:F9:FA:ED FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 02:1A:11:F9:FA:ED broadcast deaut
   henticate/disassociation of all clients, possible DoS
6. Sun Aug 14 10:04:48 2016 BCASTDISCON 0 02:1A:11:F9:FA:ED 02:1A:11:F9:FA:ED FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 02:1A:11:F9:FA:ED broadcast deaut
   henticate/disassociation of all clients, possible DoS
7. Sun Aug 14 10:04:52 2016 BCASTDISCON 0 02:1A:11:F9:FA:ED 02:1A:11:F9:FA:ED FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 02:1A:11:F9:FA:ED broadcast deaut
   henticate/disassociation of all clients, possible DoS
8. Sun Aug 14 10:05:54 2016 BCASTDISCON 0 02:1A:11:F9:FA:ED 02:1A:11:F9:FA:ED FF:F
   F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 02:1A:11:F9:FA:ED broadcast deaut
   henticate/disassociation of all clients, possible DoS
9. Sun Aug 14 11:09:01 2016 CHANCHANGE 1 F8:8E:85:DC:0F:9F F8:8E:85:DC:0F:9F FF:FF
   :FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID F8:8E:85:DC:0F:9F changed channel
   from 11 to 1
10. Sun Aug 14 11:24:01 2016 CHANCHANGE 11 F8:8E:85:DC:0F:9F F8:8E:85:DC:0F:9F FF:F
    F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID F8:8E:85:DC:0F:9F changed channel
     from 1 to 11
11. Sun Aug 14 11:38:58 2016 CHANCHANGE 1 F8:8E:85:DC:0F:9F F8:8E:85:DC:0F:9F FF:FF
    :FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID F8:8E:85:DC:0F:9F changed channel
    from 11 to 1
12. Sun Aug 14 13:08:41 2016 BCASTDISCON 0 00:1A:2B:AF:9A:FC 00:1A:2B:AF:9A:FC FF:F
    F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 00:1A:2B:AF:9A:FC broadcast deaut
    henticate/disassociation of all clients, possible DoS
13. Sun Aug 14 13:08:41 2016 BCASTDISCON 0 00:1A:2B:AF:9A:FC 00:1A:2B:AF:9A:FC FF:F
    F:FF:FF:FF:FF 00:00:00:00:00:00 Network BSSID 00:1A:2B:AF:9A:FC broadcast deaut
    henticate/disassociation of all clients, possible DoS
```

Figura 5-4 Exemple d'alertes generades pel motor IDS

Conversió a format JSON

També es realitza un processament de dades que s'encarrega de convertir els fitxers de tipus netxml i d'alertes al format JSON amb capçaleres que espera el servidor de cerca

Elasticsearch. A l'apartat d'annexes s'ha adjuntat el codi que realitza la conversió, donant com a resultat entrades d'aquest tipus:


```
1. {"index": {"_type": "network"}}
2. {"last_signal": "-
   93", "bssid": "00:01:38:FD:76:CE", "datasize": "0", "llc_packets": "33", "min_s
   ignal": "-
   97", "data_packets": "0", "crypt_packets": "0", "first_seen": "Sun Aug 14 08:15
   :57 2016", "max_signal": "-
   91", "manufacturer": "XaviTech", "retried_packets": "33", "fragmented_packets":
    "0", "encryption": "WEP", "cloaked": "false", "essid": "WLAN_88", "last_seen":
    "Sun Aug 14 08:20:32 2016", "type": "infrastructure", "ssid_type": "Beacon", "
   channel": "6", "total_packets": "33"}
3. {"index": {"_type": "network"}}
4. {"last_signal": "-
   97", "bssid": "00:1A:2B:AE:02:33", "datasize": "384", "llc_packets": "29", "min
   _signal": "-
   97", "data_packets": "1", "crypt_packets": "1", "first_seen": "Sun Aug 14 08:15
   :41 2016", "max_signal": "-
   93", "manufacturer": "AyecomTe", "retried_packets": "29", "fragmented_packets":
    "0", "encryption": "WPA+TKIP, WPA+PSK, WPA+AES-
   CCM", "cloaked": "false", "essid": "WLAN_1820", "last_seen": "Sun Aug 14 08:20:
   32 2016", "type": "infrastructure", "ssid_type": "Beacon", "channel": "10", "to
   tal_packets": "30"}
5. {"index": {"_type": "network"}}
6. {"last_signal": "-
   87", "bssid": "00:1A:2B:AE:77:8A", "datasize": "0", "llc_packets": "55", "min_s
   ignal": "-
   89", "data_packets": "0", "crypt_packets": "0", "first_seen": "Sun Aug 14 08:15
   :49 2016", "max_signal": "-
   77", "manufacturer": "AyecomTe", "retried_packets": "55", "fragmented_packets":
    "0", "encryption": "WPA+TKIP, WPA+PSK, WPA+AES-
   CCM", "cloaked": "false", "essid": "WLAN_B94B", "last_seen": "Sun Aug 14 08:20:
   24 2016", "type": "infrastructure", "ssid_type": "Beacon", "channel": "8", "tot
   al_packets": "55"}
7. {"index": {"_type": "network"}}
8. {"last_signal": "-
   89", "bssid": "00:1A:2B:AF:9A:FC", "datasize": "84", "llc_packets": "54", "min_
   signal": "-
   95", "data_packets": "1", "crypt_packets": "1", "first_seen": "Sun Aug 14 08:15
   :41 2016", "max_signal": "-
   87", "manufacturer": "AyecomTe", "retried_packets": "54", "fragmented_packets":
    "0", "encryption": "WPA+TKIP, WPA+PSK, WPA+AES-
   CCM", "cloaked": "false", "essid": "WLAN_03F3", "last_seen": "Sun Aug 14 08:20:
   32 2016", "type": "infrastructure", "ssid_type": "Beacon", "channel": "6", "tot
   al_packets": "55"}
9. {"index": {"_type": "network"}}
10. {"last_signal": "-
   93", "bssid": "00:1A:2B:AF:B8:93", "datasize": "0", "llc_packets": "1", "min_si
   gnal": "-
   93", "data_packets": "0", "crypt_packets": "0", "first_seen": "Sun Aug 14 08:17
   :39 2016", "max_signal": "-
   93", "manufacturer": "AyecomTe", "retried_packets": "1", "fragmented_packets":
   "0", "encryption": "WPA+TKIP, WPA+PSK, WPA+AES-
   CCM", "cloaked": "false", "essid": "WLAN_0D9A", "last_seen": "Sun Aug 14 08:17:
   39 2016", "type": "infrastructure", "ssid_type": "Beacon", "channel": "1", "tot
   al_packets": "1"}
```


Figura 5-5 Exemple de fitxer JSON amb dades de xarxes IEEE 802.11

5.2 Emmagatzemament i indexació

5.2.1 Configuració servidor de cerca

Per emmagatzemar i indexar les dades generades pels sensors cal implementar alguna mena de base de dades que com a mínim permeti inserir noves dades i realitzar consultes sobre les mateixes. Durant l'anàlisi de possibles solucions es varen descartar les bases de dades relacionals per la rigidesa que suposa la necessitat d'associar dades heterogènies a taules, camps i finalment registres. Tot i que inicialment les dades capturades pels sensors i el seu format estan bastant acotats, un dels requeriments desitjables del sistema desenvolupat és el de ser prou flexible com per a permetre ser adaptat de forma àgil per satisfer noves necessitats que no s'havien plantejat inicialment.

És per això que després de veure altres alternatives, s'ha cregut convenient fer servir el que es coneix com a servidor de cerca. Actualment els dos principals servidors de cerca *Open Source* i amb una bona proporció de funcionalitats i suport són:

- Apache Solr - http://lucene.apache.org/solr/
- Elasticsearch - https://www.elastic.co/

Ambdós estan basats en el motor de cerca Apache Lucene, pel que tenen unes funcionalitats similars.

S'ha decidit fer servir Elasticsearch principalment per:

- Proporcionar una sintaxi de cerca més flexible
- Implementar APIs de tipus REST que en faciliten la integració amb altres eines
- Permetre una implantació en mode distribuït de forma quasi transparent

Elasticsearch

Per el desplegament d'Elasticsearch s'ha fet servir la plataforma de creació, distribució i execució d'aplicacions Docker[23]. Queda fóra de l'abast d'aquest projecte descriure en profunditat el funcionament d'aquesta plataforma, però s'ha escollit per les facilitats que suposa a l'hora de crear i desplegar entorns fàcilment reproduïbles.

La versió utilitzada d'Elasticsearch es correspon amb la versió 2.3.5 tot i que el codi desenvolupat hauria de ser compatible amb altres versions, sempre hi quan no hi hagi modificacions a les APIs oficials.

[23] Docker: Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud – https://www.docker.com

La imatge Docker que s'ha fet servir com a base per al desplegament d'Elasticsearch està disponible al Hub oficial de Docker:

- sebp/elk - https://hub.docker.com/r/sebp/elk/

Aquesta imatge inclou suport pel que es coneix com la pila ELK (*Elasticsearch, Logstash* i *Kibana*), pel que també s'utilitza pel desplegament del servidor d'anàlisi, simplificant-ne el desplegament i la gestió.

A l'annex G Dockerfile del servidor de cerca i d'anàlisi s'inclou la definició del fitxer Docker utilitzat a l'entorn de proves.

5.2.2 Detall del procés d'indexació

Plantilles per a l'auto creació d'índexs

Elasticsearch permet definir plantilles d'índexs que són aplicades automàticament quan es creen nous índexs. Aquestes plantilles inclouen tant la configuració del propi índex com el mapeig amb les dades que contindrà.

Per a cobrir els dos tipus de dades principals que s'indexen al servidor de cerca, el resum de les xarxes i clients obtingut de l'anàlisi del trànsit Wi-Fi, i les alertes de seguretat generades pel motor IDS, s'han definit dues plantilles diferents.

La plantilla que defineix els índex de tipus *netxml*, amb les dades de les xarxes i clients 802.11 és el següent:

```
{
  "template": "netxml-*",
  "settings": {
    "number_of_shards": 1
  },
    "mappings": {
        "network": {
            "properties": {
                "bssid": {
                    "type": "string"
                },
                "channel": {
                    "type": "string"
                },
                "cloaked": {
                    "type": "string"
                },
                "crypt_packets": {
                    "type": "long"
                },
                "data_packets": {
                    "type": "long"
                },
                "datasize": {
                    "type": "long"
                },
```

```
                "encryption": {
                    "type": "string"
                },
                "essid": {
                    "type": "string"
                },
                "first_seen": {
                    "type": "date",
                    "format": "EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||epoch_millis"
                },
                "fragmented_packets": {
                    "type": "long"
                },
                "last_seen": {
                    "type": "date",
                    "format": "EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||epoch_millis"

                },
                "last_signal": {
                    "type": "long"
                },
                "llc_packets": {
                    "type": "long"
                },
                "manufacturer": {
                    "type": "string"
                },
                "max_signal": {
                    "type": "long"
                },
                "min_signal": {
                    "type": "long"
                },
                "retried_packets": {
                    "type": "long"
                },
                "ssid_type": {
                    "type": "string"
                },
                "total_packets": {
                    "type": "long"
                },
                "type": {
                    "type": "string"
                }
            }
        },
        "client": {
            "properties": {
                "bssid": {
                    "type": "string"
                },
                "channel": {
                    "type": "string"
                },
                "first_seen": {
                    "type": "date",
                    "format": "EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||epoch_millis"
                },
                "last_seen": {
                    "type": "date",
                    "format": "EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||epoch_millis"

```

```
                },
                "mac": {
                    "type": "string"
                },
                "manufacturer": {
                    "type": "string"
                },
                "probed_essids": {
                    "type": "string"
                },
                 "subtype": {
                     "type": "string"
                 },
                 "type": {
                     "type": "string"
                 }
             }
         }
    }
}
```

Figura 5-6 Plantilla d'índexs de tipus *netxml*

Bàsicament s'especifica el format dels camps, fent especial èmfasi amb els de tipus "data" per tal de que les dates i hores siguin processades correctament.

La plantilla que defineix els índex de tipus *alert*, que inclou la informació de les alertes de seguretat, és la següent:

```
{
  "template": "alert-*",
  "settings": {
    "number_of_shards": 1
  },
    "mappings": {
        "alert": {
            "properties": {
                "type": {
                    "type": "string"
                },
                "channel": {
                    "type": "long"
                },
                "mac1": {
                    "type": "string"
                },
                "mac2": {
                    "type": "string"
                },
                "mac3": {
                    "type": "string"
                },
                "mac4": {
                    "type": "string"
                },
                "bssid": {
                    "type": "string"
                },
                "message": {
                    "type": "string"
```

```
                },
                "date": {
                    "type": "date",
                    "format": "EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||
                                EEE MMM d HH:mm:ss YYYY||epoch_millis"
                }
            }
        }
    }
}
```

Figura 5-7 Plantilla d'índexs de tipus *alert*

Elasticsearch incorpora una API mitjançant la qual és possible definir aquestes plantilles. Concretament s'han executat les següents comandes per definir ambdues plantilles:

```
1. $        curl -XPUT 'https://<ip_servidor_cerca>:9200/_template/netxml' --data-
   binary @index_netxml_template.json

2. $         curl -XPUT 'https://<ip_servidor_cerca>:9200/_template/alert' --data-
   binary @index_alert_template.json
```

Figura 5-8 Comandes per definir plantilles d'índexs

On "*index_netxml_template.json*" i "*index_alert_template.json*" són dos fitxers amb les respectives plantilles que s'han vist anteriorment.

La mateixa API permet consultar quines plantilles estan configurades, d'aquesta forma es pot validar que s'han carregat correctament:

```
1. $ curl -XGET 'https://<ip_servidor_cerca>:localhost:9200/_template/
{"netxml":{"order":0,"template":"netxml-
*","settings":{"index":{"number_of_shards":"1"}},"mappings":{"network":{"propertie
s":{"data_packets":{"type":"long"},"crypt_packets":{"type":"long"},"first_seen":{"
format":"EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||epoch_millis","type":
"date"},"last_seen":{"format":"EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY|
|epoch_millis","type":"date"},"retried_packets":{"type":"long"},"essid":{"type":"s
tring"},"bssid":{"type":"string"},"max_signal":{"type":"long"},"channel":{"type":"
string"},"min_signal":{"type":"long"},"fragmented_packets":{"type":"long"},"last_s
ignal":{"type":"long"},"type":{"type":"string"},"llc_packets":{"type":"long"},"man
ufacturer":{"type":"string"},"cloaked":{"type":"string"},"encryption":{"type":"str
ing"},"ssid_type":{"type":"string"},"datasize":{"type":"long"},"total_packets":{"t
ype":"long"}}},"client":{"properties":{"probed_essids":{"type":"string"},"first_se
en":{"format":"EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss YYYY||epoch_millis","
type":"date"},"last_seen":{"format":"EEE MMM dd HH:mm:ss YYYY||EEE MMM  d HH:mm:ss
 YYYY||epoch_millis","type":"date"},"subtype":{"type":"string"},"bssid":{"type":"s
tring"},"channel":{"type":"string"},"type":{"type":"string"},"mac":{"type":"string
"},"manufacturer":{"type":"string"}}}},"aliases":{}},"alert":{"order":0,"template"
:"alert-
*","settings":{"index":{"number_of_shards":"1"}},"mappings":{"alert":{"properties"
:{"mac2":{"type":"string"},"date":{"format":"EEE MMM dd HH:mm:ss YYYY||EEE MMM  d
HH:mm:ss YYYY||EEE MMM d HH:mm:ss YYYY||epoch_millis","type":"date"},"mac1":{"type
":"string"},"mac4":{"type":"string"},"bssid":{"type":"string"},"mac3":{"type":"str
ing"},"channel":{"type":"long"},"type":{"type":"string"},"message":{"type":"string
"}}}},"aliases":{}}}%
```

Figura 5-9 Comanda per visualitzar plantilles i resultat

Un cop definides aquestes plantilles, s'aplicaran cada vegada que es creï un índex que es correspongui amb una d'elles. Al proper apartat es veurà el detall de la creació d'aquests índexs.

Format i funcionament dels índexs

Tots els sensors transmeten les seves dades a un punt central a on s'emmagatzemen i posteriorment s'indexen. Aquest pot ser un únic element a on es s'executa el programari del servidor de cerca, o per una banda un servidor purament d'emmagatzemament i un altre de cerca. En qualsevol dels casos, cada vegada que es reben noves dades d'un sensor s'executa un procés d'indexació, que s'encarrega de carregar-les i indexar-les. Aquest procés d'indexació funciona de la següent manera:

1. S'executa un procés de tipus *watchdog* que monitoritza canvis en els directoris, per tal de detectar quan hi ha dades noves d'un sensor. *Aquest procés fa servir funcionalitats pròpies del sistema operatiu GNU/Linux*[24] *per evitar tenir que fer polling constantment.*
2. Quan hi ha noves dades es genera dinàmicament un nou nom d'índex amb un format específic que es veurà a continuació. Aquest format inclou el tipus de dades a carregar, el nom del sensor que les ha transmès i una marca de data i hora.
3. Es carreguen les noves dades al nou índex, utilitzant l'API d'Elasticsearch.
4. Es crea un nou àlies del índex creat, per tal de que sempre es pugui fer servir el mateix nom d'índex per a realitzar les cerques.
5. Si les noves dades carregades engloben dades carregades anteriorment, s'eliminen del servidor de cerca els índexs que les contenen per evitar duplicats. S'ha decidit fer-ho d'aquesta forma al corroborat que eliminar índexs és molt més eficient que eliminar o modificar dades d'un índex.

El format del nom dels índexs que es crea de forma dinàmica i automàtica és el següent:

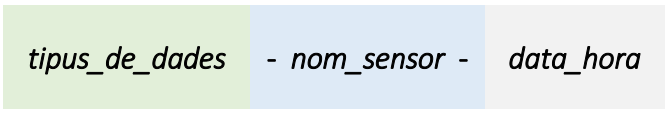


Figura 5-10 Format del nom dels índexs auto generats

A continuació es mostren alguns noms d'índexs per il·lustrar aquest format:

```
1. alert-sensor1-20160801100651
2. netxml-sensor1-20160801100651
3. alert-sensor2-20160801100654
4. netxml-sensor2-20160801100654
5. alert-sensor5-20160801100655
6. netxml-sensor5-20160801100655
7. alert-sensor3-20160801100701
```

[24] Inotify: http://man7.org/linux/man-pages/man7/inotify.7.html

```
8.  netxml-sensor3-20160801100701
```

Figura 5-11 Exemple de noms d'índexs

Donat que els noms dels índexs és dinàmic i que no es pot conèixer d'avant mà, s'utilitza el que Elasticsearch anomena àlies. Aquest àlies permet especificar un altre nom amb el que fer referència a un o més índexs. Si més d'un índex comparteixen un àlies, al moment de fer una cerca sobre aquest àlies la cerca s'expandeix a tots els índexs.

Tal i com s'ha vist anteriorment un dels darrers passos fets després d'indexar noves dades és fer que l'índex amb les darreres dades apunti a un mateix àlies. Pels índexs de tipus alert l'àlies escollit és *alert-current* i pels de tipus netxml és *netxml-current*.

D'aquesta manera en un determinat moment, les darreres dades capturades pels sensors apunten a un mateix àlies, fent possible realitzar cerques sobre les darreres dades capturades per tots els sensors. Per exemple amb el següent estat d'índexs i àlies:

```
1.  alert-sensor1-20160801100451
2.  netxml-sensor1-20160801100451
3.  alert-sensor1-20160801100551
4.  netxml-sensor1-20160801100551
5.  alert-sensor1-20160801100651 ---------> alert-current
6.  netxml-sensor1-20160801100651 --------> netxml-current
7.  alert-sensor2-20160801100454
8.  netxml-sensor2-20160801100454
9.  alert-sensor2-20160801100554
10. netxml-sensor2-20160801100554
11. alert-sensor2-20160801100654 ---------> alert-current
12. netxml-sensor2-20160801100654 --------> netxml-current
13. alert-sensor3-20160801100455
14. netxml-sensor3-20160801100455
15. alert-sensor3-20160801100555
16. netxml-sensor3-20160801100555
17. alert-sensor3-20160801100655 ---------> alert-current
18. netxml-sensor3-20160801100655 --------> netxml-current
```

Figura 5-12 Exemple d'índexs i àlies

Al realitzar una cerca sobre *netxml-current* s'obtindrien els resultats de les dades indexades a:

- *netxml-sensor1-20160801100651*
- *netxml-sensor2-20160801100654*
- *netxml-sensor3-20160801100655*

Si es volen fer cerques sobre tot l'històric de dades i no només les darreres es poden utilitzar comodins de cerca per exemple especificant com a índex: "*netxml-**". Seguint amb l'exemple anterior, en aquest cas s'obtindrien resultats de les dades indexades a:

- *netxml-sensor1-20160801100451*

- *netxml-sensor1-20160801100551*
- *netxml-sensor1-20160801100651*
- *netxml-sensor2-20160801100454*
- *netxml-sensor2-20160801100554*
- *netxml-sensor2-20160801100654*
- *netxml-sensor3-20160801100455*
- *netxml-sensor3-20160801100555*
- *netxml-sensor3-20160801100655*

S'han definit aquests noms d'índexs i els àlies per dotar de molta flexibilitat al sistema a l'hora de realitzar consultes. Per exemple si per qualsevol motiu i per una cerca en concreta tant sols interessen les dades del sensor2, es pot especificar com a índex: "*netxml-2-**".

Per popular les dades als índex s'utilitza la mateixa API dels índex d'Elasticsearch que permet la inserció de dades i auto-creació d'índexs al mateix moment. Bàsicament es fa ús del mètode PUT amb les dades en format JSON ja processades pels sensors que incorporen la capçalera específica vista en altres apartats.

A la documentació oficial d'Elasticsearch[8] i a la bibliografia [9][10] es pot consultar més informació referent a la potència i flexibilitat d'aquesta eina.

5.3 Anàlisi i explotació de la informació

5.3.1 Configuració interfície de cerca i anàlisi

Tot i disposar d'un servidor de cerca si es vol que un usuari pugui realitzar consultes i visualitzar la informació continguda al mateix, de forma amigable, és necessària la utilització d'alguna interfície d'usuari.

Des del primer moment només es van considerar interfícies d'usuari basades en aplicacions Web, donades les avantatges que suposen donat que:

- No requereixen als usuaris realitzar cap mena d'instal·lació
- Es poden utilitzar en dispositius heterogenis com ordinadors de sobretaula, portàtils, tauletes i mòbils.

Un cop escollit Elasticsearch com a servidor de cerca, l'opció més madura i ben acoblada amb el mateix és Kibana[25][11][12].

Kibana

Kibana es defineix com una plataforma per explorar i visualitzar dades que s'integra fàcilment amb Elasticsearch.

El procés per al desplegament de Kibana ha estat idèntic que pel servidor de cerca amb Elasticsearch, fent servir la plataforma Docker.

La versió utilitzada de Kibana es correspon amb la versió 4.5.4 tot i que les cerques i visualitzacions utilitzades en aquest projecte haurien de ser compatible amb altres versions, sempre hi quan no hi hagi canvis majors.

La imatge Docker que s'ha fet servir com a base per al desplegament de Kibana està disponible al Hub oficial de Docker:

- sebp/elk - https://hub.docker.com/r/sebp/elk/

Aquesta imatge inclou suport pel que es coneix com la pila ELK (*Elasticsearch, Logstash* i *Kibana*), pel que també s'utilitza pel desplegament del servidor de cerca, simplificant-ne el desplegament i la gestió.

A l'annex G Dockerfile del servidor de cerca i d'anàlisi, a la pàgina núm 120, s'inclou la definició del fitxer Docker utilitzat a l'entorn de proves per si es vol reproduir.

5.3.2 Detall de pantalles de visualització

En aquest apartat es recull una mostra de captures de pantalla que mostren la interfície d'usuari del sistema desenvolupat amb diferents cerques i visualitzacions que s'incorporen per defecte. Aquestes visualitzacions permeten tenir una idea ràpida i visual del trànsit IEEE 802.11 capturat pel sistema, permetent veure quines xarxes i dispositius estan disponibles o consultar les alertes de seguretat generades.

Al accedir a la interfície Web el primer que es veu és el següent:

[25] Kibana: https://www.elastic.co/products/kibana

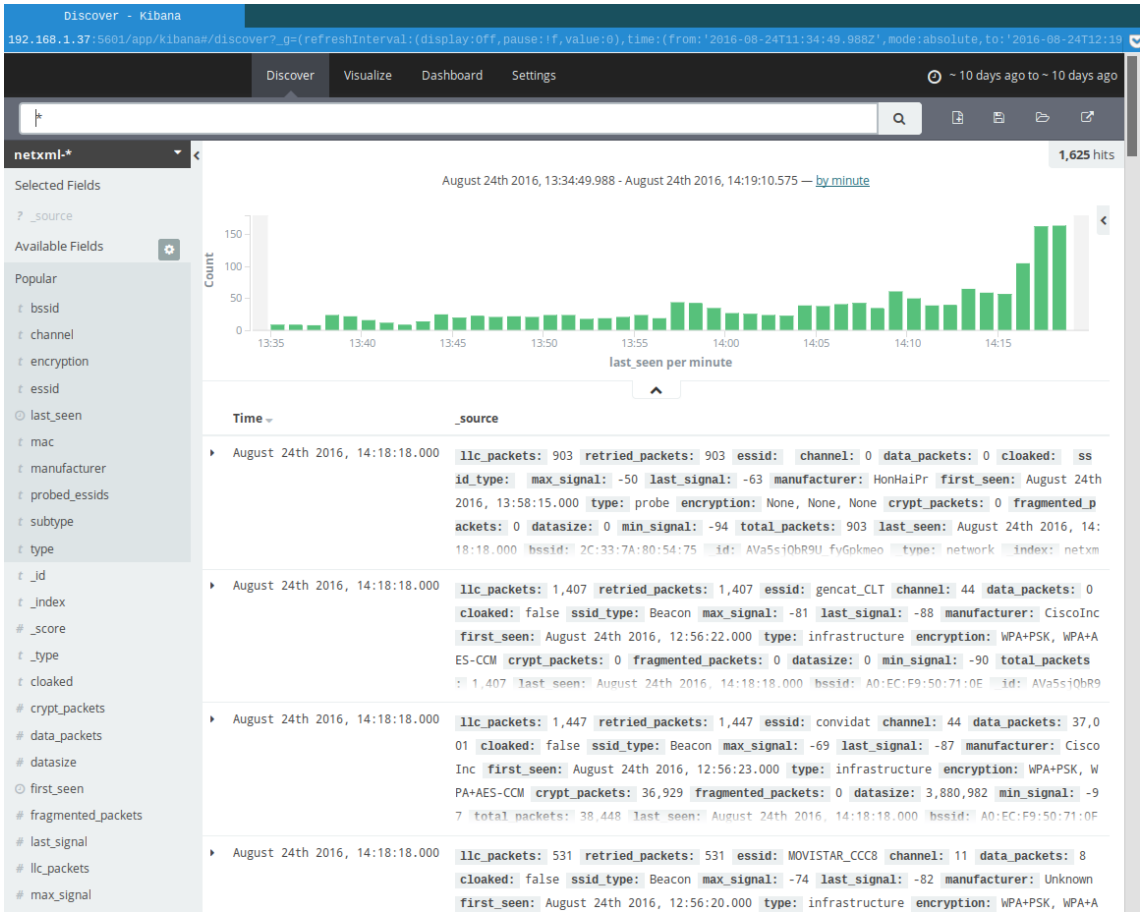


Figura 5-13 Pantalla inicial de la interfície d'usuari del sistema

Per defecte s'està al mode "*Discover*" a on és possible visualitzar les darreres dades capturades pels sensors i realitzar cerques sobre les mateixes.

Seleccionant els camps que es volen visualitzar es poden crear de forma dinàmica taules resum, o utilitzar les creades per defecte al apartat "Visualize". Per exemple la propera captura mostra una taula resum amb els següents camps de les xarxes Wi-Fi vistes en el trànsit capturat:

- Hora de la darrera vegada en la que s'han vist paquets d'aquesta xarxa
- Essid o nom
- Canal
- Tipus de xifrat que fa servir
- Qualitat de la senyal del darrer paquet que s'ha vist
- Número total de paquets

- Dimensions totals dels paquets de dades d'aquesta xarxa (un zero indica que no s'han capturat paquets de dades, possiblement perquè ningú està fent ús de la xarxa).

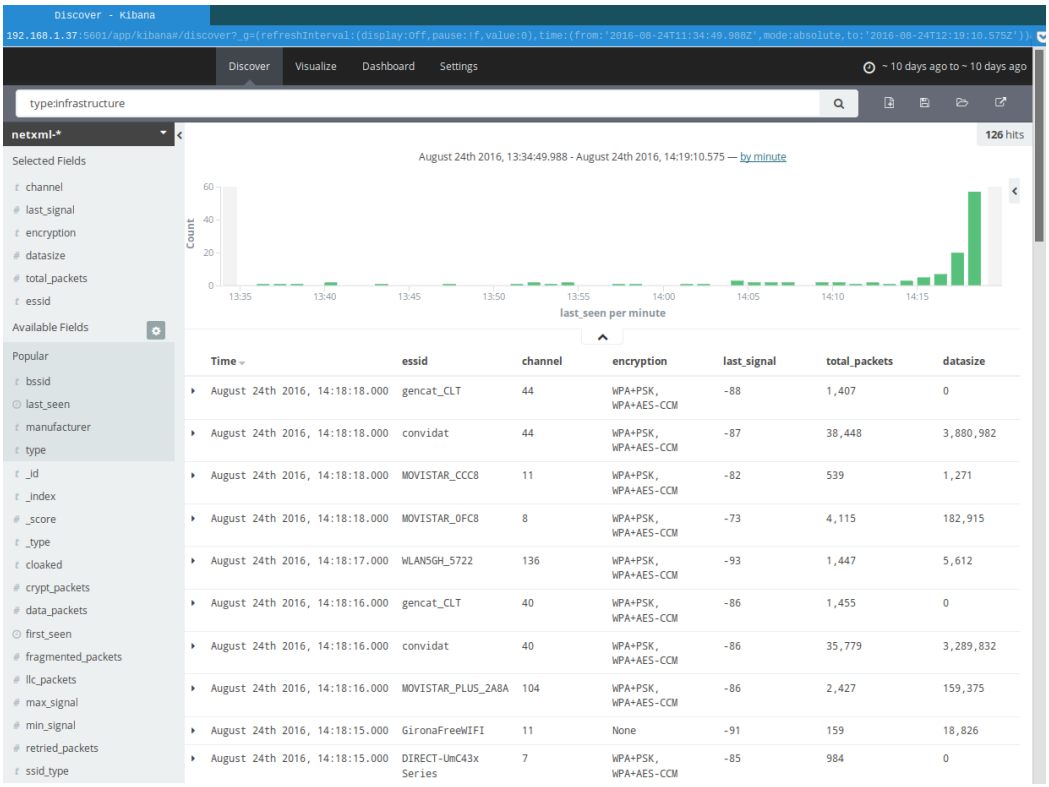


Figura 5-14 Captura amb taula resum amb dades de les xarxes de tipus infraestructura

El camp de cerca permet especificar valors a cercar i permet limitar el camp sobre el qual fer-ho. Per exemple la següent captura mostra una taula resum de les entrades de tipus "client". En aquesta taula resum es poden veure detalls de les estacions clients, entre d'altres el fabricant del dispositiu i el essid de xarxes per les quals han "preguntat", enviant trames de tipus probe request:

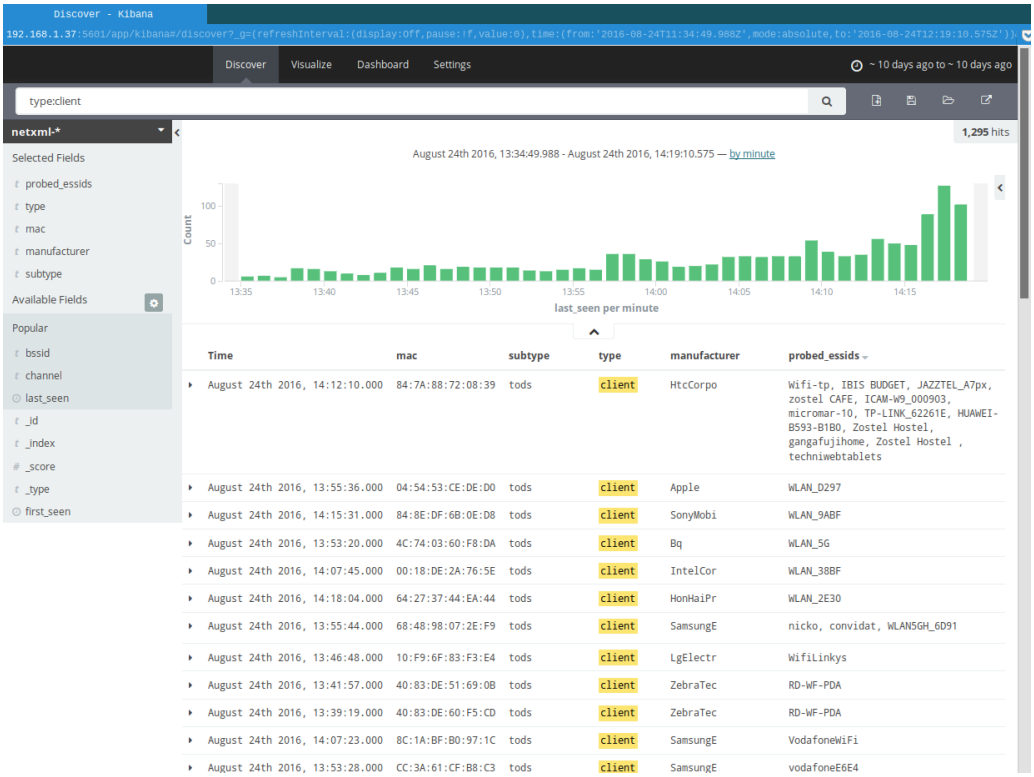


Figura 5-15 Captura amb taula resum de clients

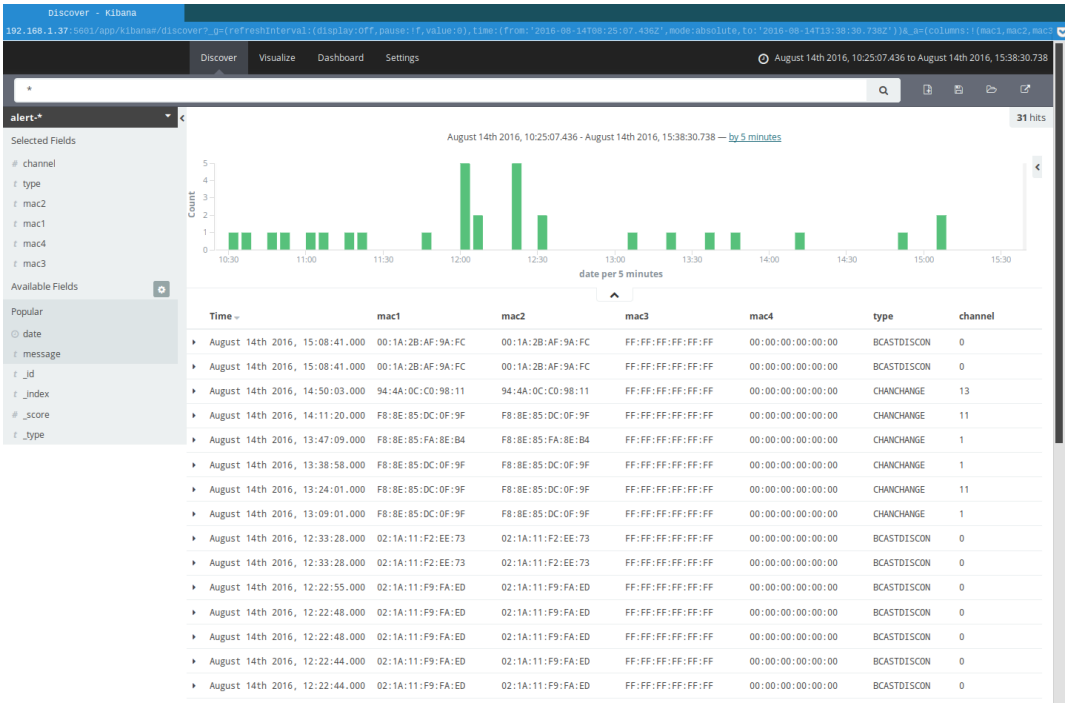


Figura 5-16 Captura amb resum i distribució en el temps de les alertes de seguretat

A l'apartat de visualitzacions, *Visualize*, el sistema permet crear gràfics sobre les dades indexades:

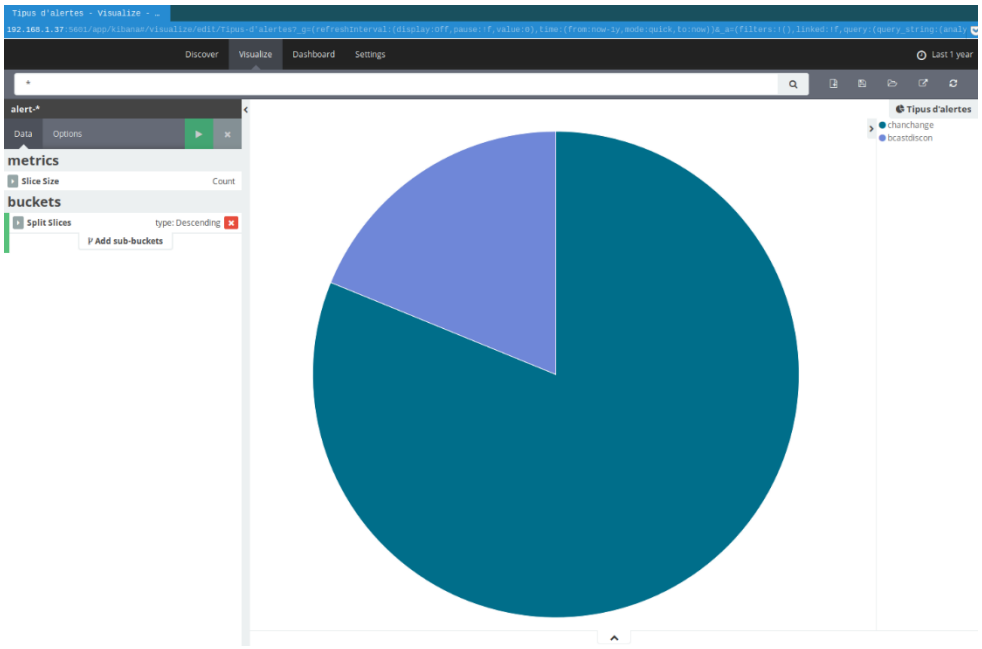


Figura 5-17 Visualització de distribució d'alertes per tipus

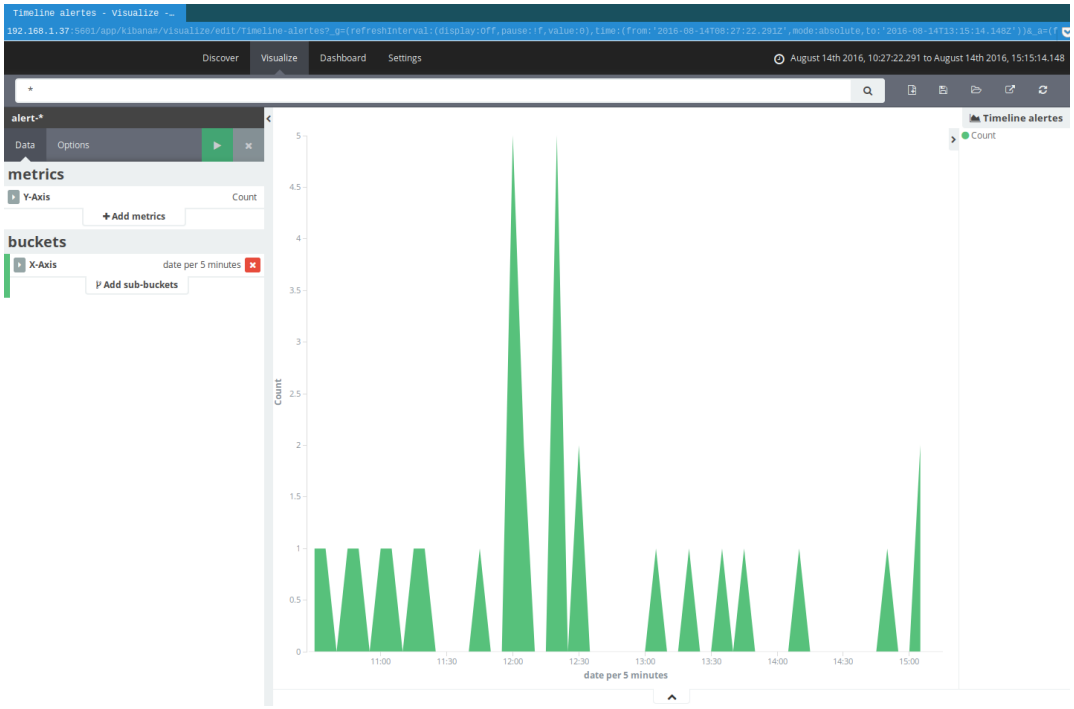


Figura 5-18 Visualització de distribució en el temps de nombre d'alertes de seguretat

Finalment l'apartat *Dashboard* de la interfície permet a l'usuari configurar panells de visualització a on agrupar al seu gust les visualitzacions de les cerques que esculli. Aquesta funcionalitat permet la creació de múltiples *dashboards* que poden facilitar enormement la feina d'un analista recollint a un únic lloc la informació que consideri més rellevant.

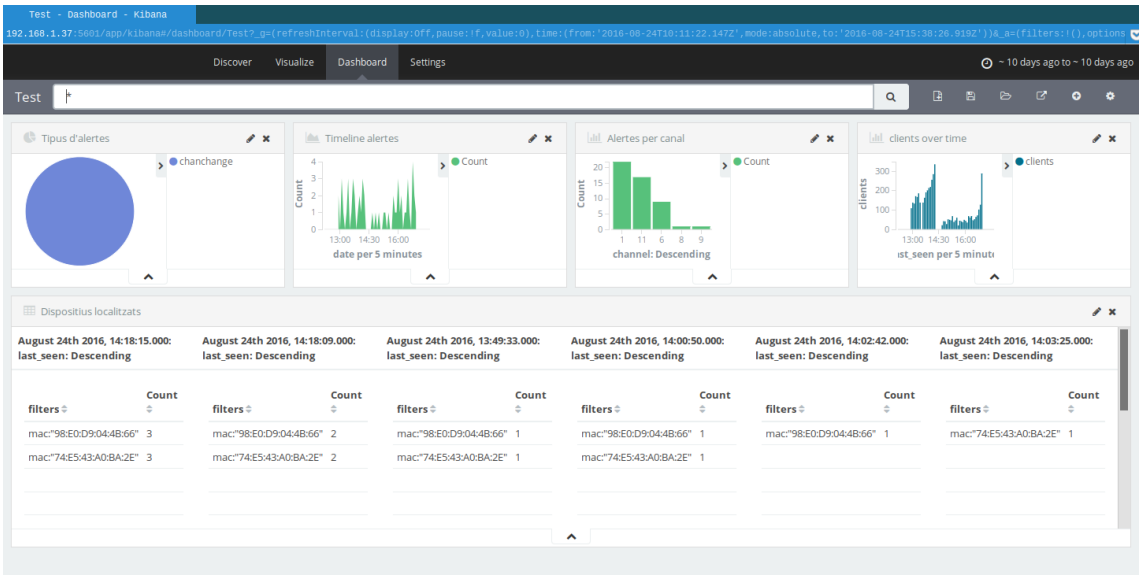


Figura 5-19 Exemple de *dashboard* amb visualitzacions personalitzades

A la documentació oficial de Kibana[11] i a la bibliografia[12] es pot consultar més informació referent a la potència i flexibilitat d'aquesta eina.

6 Implantació

L'objectiu principal del projecte és el de desenvolupar un sistema prou flexible com per a satisfer múltiples necessitats referents al monitoratge de xarxes Wi-Fi, fent especial èmfasi en la detecció i gestió d'incidents de seguretat. És per això que no s'ha dut a terme una implantació del mateix, donat que no hi havia una necessitat específica a satisfer. Un desplegament en un entorn real requeriria d'unes fases d'anàlisi i consultoria prèvies per determinar quin maquinari, quina topologia i configuració del sistema serien més adequats per aquest entorn.

No obstant, per a desenvolupar i provar el sistema, s'ha desplegat en maquinari físic i utilitzat en un entorn de laboratori per a fer proves del mateix. En aquest apartat es recullen els detalls del maquinari i la topologia utilitzada a l'entorn de proves, així com una pinzellada de diferents casos d'ús del sistema.

La informació continguda en aquest apartat està enfocada al punt de vista d'un enginyer de sistemes o al de la persona encarregada d'implantar o desplegar una solució basada en el sistema desenvolupat.

6.1 Maquinari

Un dels objectius del projecte és el de desenvolupar un sistema distribuït i escalable, que es pugui executar sobre maquinari de baix cost i fàcilment disponible. El maquinari que s'ha utilitzat durant les proves de laboratori compleix amb aquests requeriments i ha estat reutilitzat de maquinari del qual ja es disposava.

6.1.1 Sensors

Per al desenvolupament del sistema i durant la realització de proves s'ha utilitzat maquinari heterogeni de baix cost. Per una banda s'han utilitzat el que es coneix comercialment com a punts d'accés Wi-Fi de butxaca i per altra sistemes de tipus SBC

de l'anglès *Single-Board Computer*. En ambdós casos el preu unitari és molt contingut, entre ~20 i 50 euros depenent de la configuració escollida, a data de redacció d'aquest document.

A continuació es detallen les característiques dels diferents elements utilitzats com a sensors, i les modificacions i configuracions realitzades sobre els mateixos.

TP-Link TL-WR703N / TP-Link TL-MR10U / GL.iNet GL-MT300A

Aquests tres models de punts d'accés de butxaca tenen característiques molt similars. A nivell de maquinari les seves principals característiques són:

model	processador	RAM	Flash	Xarxa	802.11	USB
TL-WR703n	AR9331@400MHz	32MiB	4MiB	1x10/100	b/g/n 150Mbps	1x2.0
TL-MR10U	AR9331@400MHz	32MiB	4MiB	1x10/100	b/g/n 150Mbps	1x2.0
GL-MT300A	MT7620A@580MHz	128MiB	16MiB	2x10/100	b/g/n 300Mbps	1x2.0

Taula 9 Característiques del maquinari de punts d'accés de butxaca

Figura 6-1 TP-Link WR703n, TP-Link MR10U i GL.inet MT300A

(font de les imatges: planes web dels fabricants, TP-Link i GL.inet respectivament)

Tot i que originàriament el seu propòsit és el de servir de punts d'accés i/o encaminadors de viatge, per crear i compartir connexions sense fils, és possible modificar-ne el firmware per un altre de propòsit més general. S'ha escollit el firmware OpenWRT[26] degut a la seva compatibilitat no només amb els dispositius de proves, sinó amb gran quantitat d'encaminadors i punts d'accés de múltiples fabricants. A data de redacció d'aquest document, la plana oficial del projecte[27], llista més de 1200 dispositius amb suport.

Encara que les seves característiques puguin semblar molt limitades, cal destacar que el maquinari ha estat dissenyat per realitzar tasques de punt d'accés i encaminador de xarxes 802.11. Per tant està optimitzat just per a la principal tasca que desenvoluparà: tractar trànsit 802.11.

Donada la capacitat limitada a nivell de memòria flash, s'han utilitzat memòries externes per ampliar-la. Pels sensors del fabricant TP-Link s'ha optat per memòries USB de perfil reduït, i pel sensor GL.inet MT300A s'ha soldat un lector de targetes microSD.

Per fer transparent pel sistema l'existència d'aquestes memòries externes i per tal de poder instal·lar utilitzats de la forma convencional, s'ha fet servir el sistema de fitxers *OverlayFS* [28]que implementa Linux. Aquest sistema de fitxers permet presentar un sistema de fitxers que és el resultat de superposar un sistema de fitxers sobre un altre.

Els detalls de com s'ha sobreescrit el firmware original pel firmware OpenWRT s'han separat a l'annex E Sensors: modificació firmware.

Raspberry PI 3

La Raspberry PI és un ordinador de placa reduïda, placa única o de tipus SBC "de l'anglès *Single-Board Computer"* de baix cost. Degut a la seva actual popularitat gaudeix d'un gran nivell de suport per la comunitat i és possible instal·lar-hi distribucions GNU/Linux i la majoria de programari disponible per aquestes.

Les seves principals característiques són:

model	processador	RAM	Flash	Xarxa	802.11	USB
Raspberry PI Model B 3	ARMv8@1.2GHz quad-core	1GB Compartit amb CPU	MicroSD	1x10/100	b/g/n 300Mbps	4x2.0

Taula 10 Característiques Raspberry PI B 3

[26] OpenWRT: A Linux distribution for embedded devices - https://openwrt.org/
[27] OpenWRT *Table of Hardware*: http://wiki.openwrt.org/toh/start
[28] OverlayFS: https://www.kernel.org/doc/Documentation/filesystems/overlayfs.txt

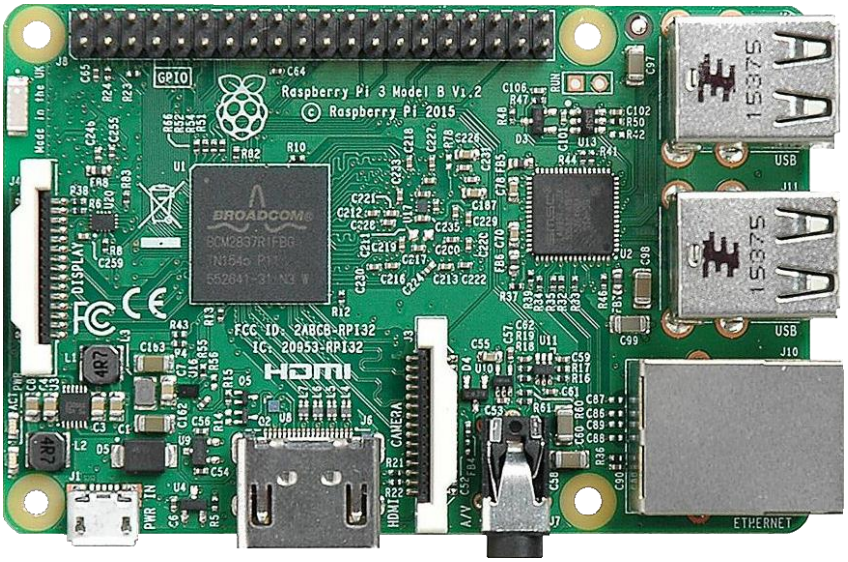


Figura 6-2 Raspberry PI 3

(font de la imatge: https://upload.wikimedia.org/wikipedia/commons/b/b4/Raspberry_Pi_3_Model_B.png)

S'ha decidit realitzar proves sobre la Raspberry PI, tot i de no tractar-se d'un dispositiu orientat a xarxes 802.11, degut a la seva popularitat i facilitat d'adquisició. Cal destacar que les versions anteriors de la Raspberry PI ni tan sols incorporaven connectivitat 802.11 a la mateixa placa, i que la qualitat del firmware del xip 802.11 que incorpora la darrera, a data de redacció d'aquest document, és regular. Per això s'ha decidit utilitzar una targeta de xarxa 802.11 externa, en format USB, pel sensor implementat sobre la Raspberry PI 3, en comptes de fer servir el xip intern que incorpora de sèrie. Tot i aquest inconvenient, només mirant les característiques d'aquest dispositiu envers els anteriors, es pot veure que es tracta d'un sensor amb unes capacitats de computació i memòria molt superiors.

El sistema de sensors s'ha pensat inicialment per poder ser executat en dispositius de capacitat reduïda, però res impedeix afegir més lògica als sensors que tinguin les característiques per executar-la. De fet s'han realitzat algunes proves per executar un model distribuït de servidor de cerca sobre els sensors de Raspberry PI 3, amb totes les avantatges que això comporta. Però s'ha deixat com a treball futur degut a quedar fóra dels objectius inicials del sistema i perquè presenta una sèrie de reptes que caldria afrontar en profunditat.

A l'entorn de proves s'ha utilitzat la distribució oficial Raspbian[29] degut al seu major suport i per tractar-se d'una versió basada en Debian GNU/Linux, amb la qual ja estàvem familiaritzats. La instal·lació i configuració bàsica del sistema s'han realitzat seguint la guia

[29] Raspbian: https://www.raspberrypi.org/downloads/raspbian/

oficial[30] sense cap mena d'incidència, pel que no s'ha cregut necessari tornar a reproduir aquests passos en aquest document.

Altres sensors

A l'entorn de laboratori també s'han utilitzat ordinadors portàtils amb sistema operatiu GNU/Linux, una interfície de xarxa 802.11 i una altra ethernet com a sensors.

6.1.2 Xarxa distribuïda

El sistema dissenyat i implementat no especifica la capa física a utilitzar com a mecanisme de comunicació per a la xarxa distribuïda. Es pot utilitzar qualsevol que permeti definir xarxes d'àrea local (necessari per l'autodescobriment de sensors) i que suporti el protocol IP (per a les comunicacions de comandes i dades dels sensors).

El més habitual i recomanat és fer servir l'estàndard ethernet de xarxes d'àrea local, per les avantatges que comporta:

- Velocitats de gigabits per segon en cas de ser necessari
- No produeix interferències o trànsit addicional a descartar amb el trànsit Wi-Fi que es vol monitoritzar
- Permet utilitzar la tecnologia d'alimentació a través d'ethernet, PoE de l'anglès *Power over Ethernet. Cal que els sensors implantats tinguin suport.*

Durant la realització de proves s'ha dissenyat i implementat una xarxa molt simple, amb els següents elements:

- 4x sensors
- 1x Servidor que realitza les tasques d'emmagatzemament, cerca i anàlisi

Utilitzant el següent maquinari de xarxa:

- Commutador *(Switch) Gigabit Ethernet* de 8 ports
- Cablejat de xarxa Ethernet de categoria 6

La propera figura en mostra la topologia:

[30] https://www.raspberrypi.org/documentation/installation/installing-images/README.md

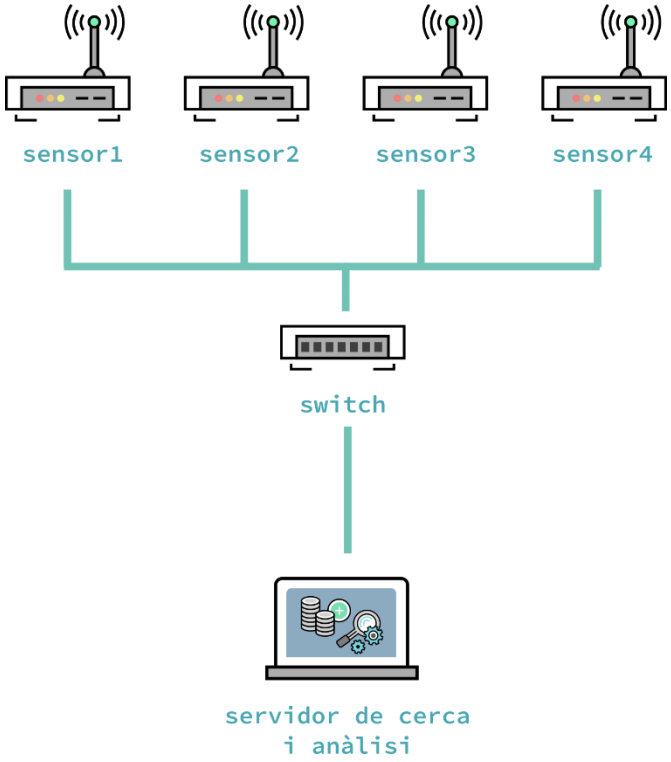


Figura 6-3 Topologia de la xarxa de proves

Servidor d'emmagatzemament, cerca i anàlisi

Depenent del volum de dades a gestionar i emmagatzemar caldrà un sistema amb major o menor capacitat.

Per l'entorn de proves s'ha utilitzat un ordinador portàtil de gama mitja amb les següents característiques principals:

model	processador	RAM	Disc dur	Xarxa
Lenovo Thinkpad T450	Intel Core i5-5200U	16 GB	512 GB SSD	1x10/100/1000

Taula 11 Característiques servidor emmagatzemament, cerca i anàlisi

Cal destacar que depenent de les necessitats seria possible dividir aquest component en tres o més components, per exemple:

- 1x servidor d'emmagatzemament tipus servidor de fitxers
- 1x servidor dedicat de cerca
- 1x servidor dedicat d'anàlisi

L'arquitectura de la solució implementada es prou flexible com per permetre escalar tant de forma vertical com horitzontal. Això és possible perquè els principals elements, tant els sensors, com el servidor de cerca, permeten treballar de forma distribuïda.

A mode orientatiu els requeriments més forts, els dictats pel component servidor de cerca, estan ben detallats i explicats a la plana de requeriments d'Elasticsearch[31]. El volum a gestionar pel servidor de cerca depèn molt del trànsit a monitorar i de l'històric de dades que es vulgui emmagatzemar. No és el mateix guardar un històric de la darrera setmana, per en cas d'incidència poder investigar-ho, que voler-ho fer de tot un any.

6.2 Escenaris i casos d'ús

Tots els casos d'ús descrits a continuació s'han recreat al entorn de laboratori controlat citat anteriorment.

6.2.1 Detecció d'atacs a xarxa Wi-Fi corporativa

La propera figura representa els elements principals simulats en aquest cas d'ús:

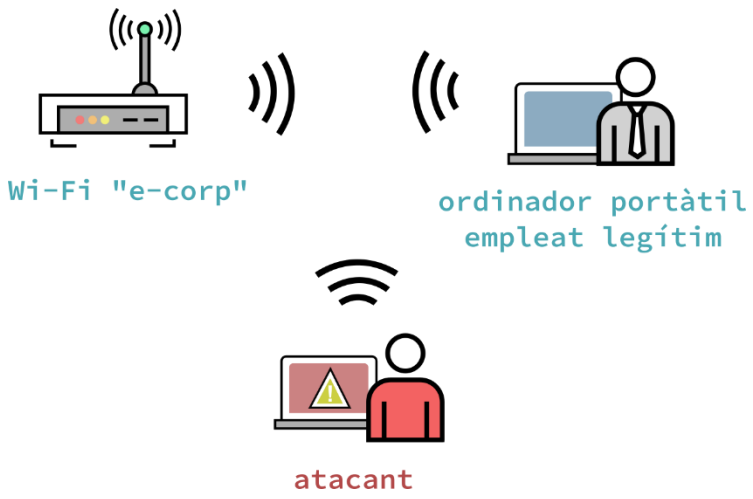


Figura 6-4 Escenari plantejat al cas d'ús detecció d'atacs a xarxa corporativa

A on la xarxa corporativa fictícia té els següents atributs principals:

[31] Elasticsearch requirements - https://www.elastic.co/guide/en/elasticsearch/guide/current/hardware.html

- ESSID: "e-corp"
- BSSID: 02:1A:11:F9:FA:ED

I una estació client legítima associada a aquesta xarxa, per exemple el portàtil d'un empleat, els següents:

- MAC interfície Wi-Fi: 02:DD:51:1A:E6:39

S'ha desplegat la solució de monitoratge desenvolupada en aquest projecte per tal de capturar, emmagatzemar i analitzar el trànsit de la xarxa corporativa, especialment com a sistema de detecció d'intrusions.

Atacs de des-autenticació

Els atacs de des-autenticació es basen en falsejar trames de tipus *deauthentication* suplantant la identitat d'una estació o punt d'accés legítims. D'aquesta forma s'aconsegueix una denegació de servei. Aquest tipus de trames no incorporen cap tipus d'autenticació o xifrat, pel que es tracta d'un error de disseny pel que poc es pot fer, més enllà de detectar els atacs i intentar arribar a l'origen dels mateixos.

Per recrear aquest escenari s'han fet servir els mateixos mètodes i eines que poden utilitzar els atacants per fer-ho, donat que existeixen eines obertes que els implementen. Un atacant que volgués executar un atac de des-autenticació contra la xarxa corporativa podria fer servir l'eina aireplay-ng de la suite aircrack-ng[32] per generar trames d'aquest tipus:

```
1. $ aireplay-ng -0 0 -a 02:1A:11:F9:FA:ED wlan0
```

On els paràmetres signifiquen:

- -0: enviar trames de tipus *deauthentication*
- 0: nombre de trames a enviar (0 significa enviar trames de forma indefinida fins que s'aturi el programa)
- -a: adreça MAC del punt d'accés
  - 02:1A:11:F9:FA:ED – bssid de la xarxa corporativa
  - Al no especificar-se el *flag* -c amb la MAC d'un client en concret, s'envien trames de tipus *deauthentication* massives a l'adreça de *broadcast*.
- wlan0: nom de la interfície de xarxa utilitzada per enviar les trames

Instantàniament tots els usuaris serien des-autenticats de la xarxa i per tant no estarien connectats a la mateixa i no podrien fer-ne ús, donat que l'atacant ha especificat l'enviament de trames de forma ininterrompuda.

[32] aircrack-ng: *"Aircrack-ng is a complete suite of tools to assess WiFi network security."*- https://www.aircrack-ng.org/

Al mateix temps el sistema de monitoratge detectaria aquest atac, generant alertes indicant-ho:

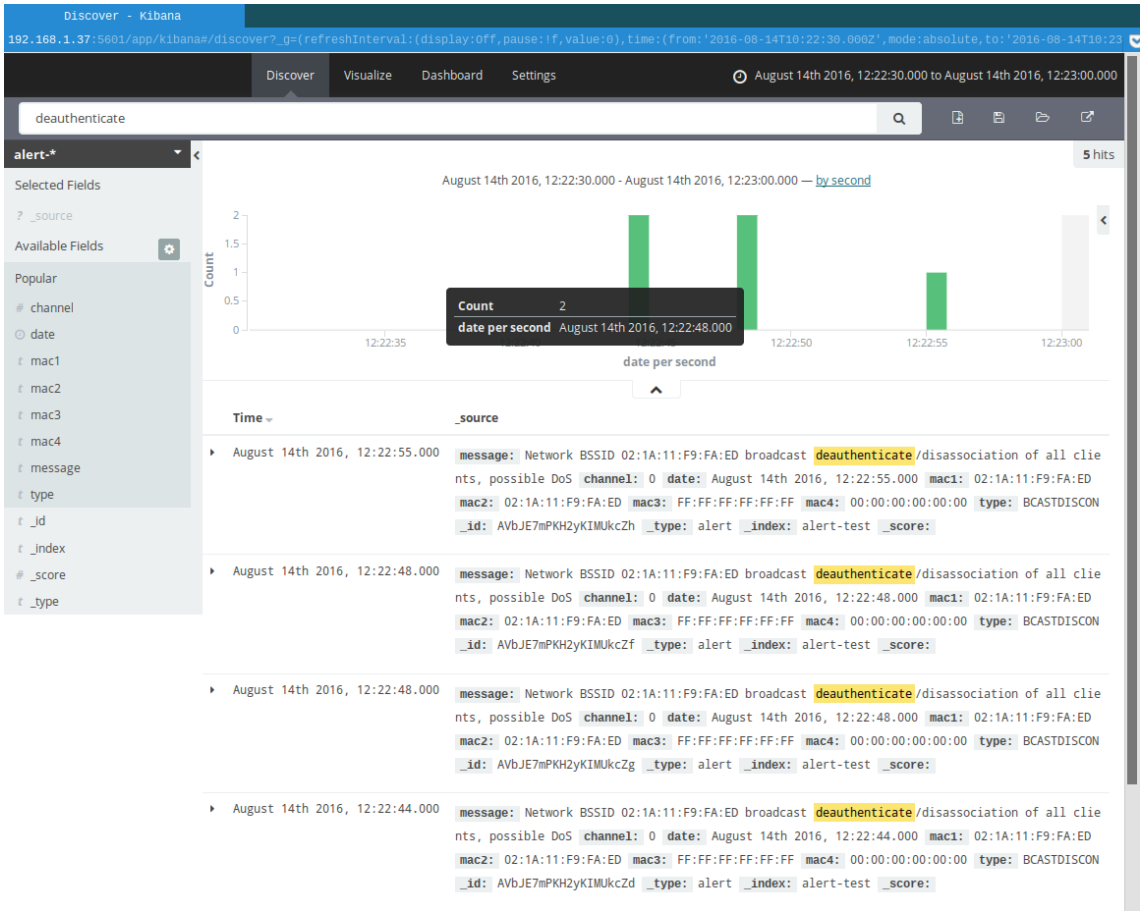


Figura 6-5 Alertes d'atac de des-autenticació a interfície d'anàlisi de la solució

El responsable de seguretat, analista o personal encarregat de la monitorització avisat per aquestes alertes podria iniciar un procés d'investigació analitzant en major profunditat el trànsit capturat per obtenir-ne més informació. Tant mateix aquest tipus d'atacs solen precedir d'altres, com per exemple els intents de suplantació d'estacions legítimes, pel qual es podria incrementar el grau d'alerta i/o posar mesures addicionals per a prevenir-los, contenir-los, minimitzar-ne l'impacte o, en general, gestionar-ne de forma adient el risc.

Suplantació de xarxa corporativa

Un atac molt eficaç és el de suplantar la configuració i característiques d'un punt d'accés legítim per realitzar un atac similar al *phishing*[33] del món Web. La idea principal és la de

[33] *Phishing*: Més informació a http://www.antiphishing.org/resources/overview/avoid-phishing-scams

crear una xarxa similar a la d'una xarxa coneguda per un usuari, per fer que aquest es connecti a aquesta xarxa creient estar connectant-se a la legítima. Per a fer-ho és possible utilitzar el que es coneix com a Karma attacks[34], que s'aprofiten de les implementacions que permeten als usuaris connectar-se de forma automàtica a xarxes ja conegudes.

Un atacant que tingui control a nivell de xarxa pot realitzar atacs molt eficaços a altres nivells: com per exemple modificar les respostes DNS, escoltar les comunicacions i injectar-ne trànsit. És per això que es tracta d'un atac senzill però eficaç, amb un impacte que pot arribar a ser molt elevat per als usuaris que es connectin a la xarxa falsa.

Per recrear aquest atac s'ha utilitzat una implementació moderna de l'atac original anomenat Karma, els detalls d'aquesta implementació i de la eina utilitzada estan a la web del projecte mana[35].

Quan el suposat empleat inicia el seu sistema, al tenir configurada com a xarxa favorita i d'auto-connexió la xarxa corporativa "e-corp", s'envia una trama de tipus *probe request*. El programari executat per l'atacant ho detecta, crea instantàniament una xarxa amb el essid "e-corp" i envia una *probe response* per intentar que l'usuari es connecti a aquesta, en comptes de la legítima.

Al mateix temps el sistema de monitoratge desenvolupat en aquest projecte, i que està monitorant la xarxa corporativa, detecta aquestes trames com a falses (al no correspondre's el BSSID amb l'original) i genera una alerta d'aquest atac:

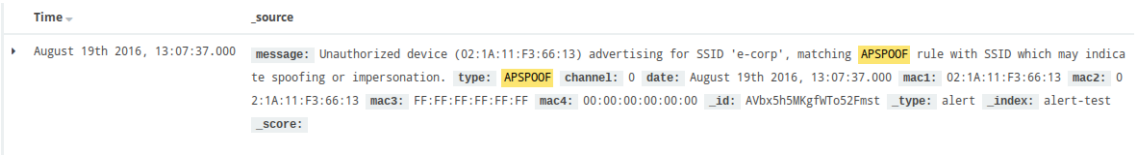


Figura 6-6 Alerta de suplantació vista des de la interfície d'usuari del sistema

El responsable de seguretat, analista o personal encarregat de la monitorització avisat per aquestes alertes podria iniciar un procés d'investigació analitzant en major profunditat el trànsit capturat per obtenir-ne més informació. Correlant la informació de relació senyal-soroll de les trames malicioses, present a les captures de trànsit, seria fins i tot possible aproximar la localització física des de la qual s'està duent a terme l'atac.

[34] Karma attacks - http://theta44.org/karma/
[35] mana - https://github.com/sensepost/mana

6.2.2 Monitoratge de dispositius robats

Aquest escenari utilitza el sistema desenvolupat per intentar detectar dispositius robats. Els dispositius que tenen suport per a connexions Wi-Fi tenen un identificador únic que es correspon amb l'adreça MAC de la interfície IEEE 802.11 i que viatja en clar en el trànsit 802.11 que generen o que va dirigit a aquests dispositius. Aquest identificador fins i tot s'envia en mode *broadcast* encara que el dispositiu no estigui connectat a una xarxa Wi-Fi. Això és degut a la forma de treballar del protocol IEEE 802.11 que permet l'enviament de trames de tipus probe request, per facilitar la connexió amb xarxes conegudes.

Aprofitant aquest comportament és possible monitorar el trànsit IEEE 802.11 cercant trames que incloguin a algun dels seus camps d'adreces alguna de les adreces MAC d'un llistat de dispositius robats. Això és similar al que es fa a les xarxes de telefonia amb el identificador IMSI o *International Mobile Subscriber Identity*, tot i que a la practica les operadores és limiten a afegir els dispositius reportats com a robats a una llista negra, fent que no es puguin connectar a la xarxa de telefonia, per intentar dissuadir als lladres.

Aquest cas d'ús seria especialment interessant si es despleguessin sensors a una xarxa de gran cobertura, com a la xarxa Wi-Fi de la UdG, o encara millor a nivell d'una xarxa municipal. Per exemple la següent figura mostra una possible ubicació de sensors, al mateix lloc a on actualment hi ha desplegats punts d'accés de la xarxa Wi-Fi de la UdG, indicant-ne la cobertura:

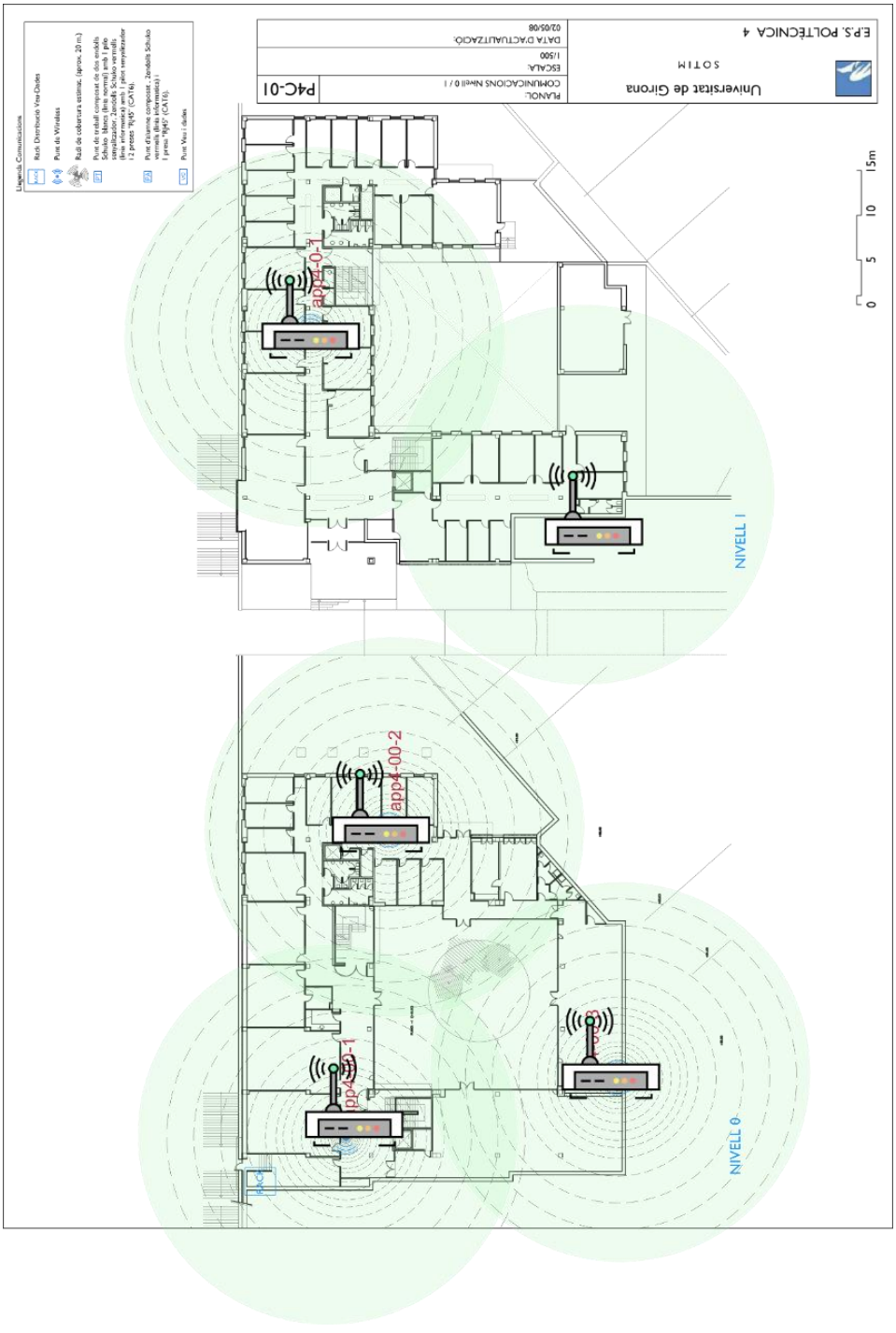


Figura 6-7 Exemple de desplegament de sensors al edifici PIV

(font de la imatge original: http://www.udg.edu/Portals/0/campus/mapesWifi/Politecnica%204.pdf)

Sempre que un estudiant o un ciutadà denunciés el robatori d'un dispositiu s'afegiria a una llista l'adreça MAC del mateix. Utilitzant el sistema desenvolupat en aquest projecte es podria programar una cerca periòdica que s'executés cada minut i cerqués el llistat d'adreces al trànsit capturat. De trobar algun resultat es podria generar una alerta per

indicar que s'havia vist trànsit amb l'adreça del dispositiu robat a un moment donat per un sensor determinat.

Sabent la ubicació física del sensor i analitzant la potència amb la que s'ha vist la trama amb l'adreça robada es podria acotar la zona física a la qual s'està utilitzant el dispositiu per posteriorment poder emprendre les accions pertinents. A més a més si la densitat de sensors fes que més d'un sensor captures el mateix trànsit, es podrien utilitzar tècniques de triangulació per afinar millor l'origen físic del trànsit i per tant del dispositiu robat.

7 Conclusions

Aquest capítol descriu tant l'assoliment dels objectius del projecte de final de carrera com els del sistema de captura, emmagatzemament i anàlisi de trànsit Wi-Fi desenvolupat.

L'apartat treball futur recull totes aquelles ampliacions i millores que s'han detectat durant la realització d'aquest projecte, i que no s'han dut a terme però sobre les quals es desitja continuar treballant.

7.1 Conclusions desenvolupament sistema de monitoratge

S'ha assolit amb èxit l'objectiu prioritari i principal: analitzar, dissenyar i implementar un sistema de captura, emmagatzemament i anàlisi de trànsit Wi-Fi, orientat a la detecció i resposta d'incidents de seguretat.

El sistema desenvolupat ha pogut cobrir els requeriments plantejats inicialment, entre els que destaquen:

- Requeriments captura de trànsit
  - Possibilitat d'especificar el canal o canals de captura que es corresponen amb les freqüències utilitzades pels protocols IEEE 802.11
  - Possibilitat de capturar múltiples bandes de freqüència utilitzades pels protocols IEEE 802.11 de forma simultània
  - Possibilitat de realitzar captures simultàniament en diferents zones de cobertura
  - Generació de fitxers amb les dades de captura

Solució: El disseny i la implementació d'una arquitectura distribuïda de sensors ha atorgat molta flexibilitat i permès assolir els requeriments de captura citat. L'ús de diferents sensors permet cobrir diferents zones i treballar en diferents freqüències simultàniament.

- Requeriments de processament
  - Processar i interpretar els fitxers generats en el procés de captura per tal d'extreure'n:
    - Detall de les xarxes Wi-Fi
    - Detall de les estacions i d'altres elements que realitzin comunicacions IEEE 802.11
    - Generació d'alertes de seguretat en base al trànsit capturat

  Solució: S'han aprofitat les capacitats d'eines existents per extreure la informació rellevant del trànsit capturat. De la mateixa forma s'ha utilitzat un motor de detecció d'intrusions per la generació d'alertes de seguretat.

- Emmagatzemament
  - Permetre la inserció de les dades processades
  - Permetre la modificació i/o eliminació de dades
  - Indexar les dades de forma adient per a la seva consulta

  Solució: La utilització d'un servidor de cerca en comptes d'una base de dades de tipus relacional ha permès dissenyar i implementar un procés d'indexació capaç d'emmagatzemar de forma eficient el trànsit capturat pels sensors.

- Anàlisi
  - Oferir una interfície d'usuari mitjançant la qual sigui possible explorar i consultar les dades emmagatzemades
  - Oferir una interfície o API per facilitar la integració amb altres eines
  - Disposar de cerques i visualitzacions prefixades que permetin començar a explorar les dades o veure'n ràpidament un resum de les mateixes

  Solució: L'ús d'una interfície d'usuari altament integrada amb el sistema de cerca utilitzat, i el disseny i implementació de cerques i visualitzacions, ha donat com a resultat un sistema d'anàlisi amigable i potent.

Per altra banda, al llarg del cicle de vida del projecte s'han detectat altres requeriments, especialment no funcionals, que no s'han assolit amb el grau desitjat donat l'abast del projecte i la priorització dels requeriments funcionals. Es tracta per exemple de la *paquetització* de la solució implementada per facilitar-ne el desplegament i d'altres aspectes que s'han recollit al apartat de treballs futurs.

7.2 Conclusions personals

Aquest projecte ha suposat un exercici molt interessant al permetre abordar una problemàtica gens trivial i amb un abast considerable, que a la seva vegada ha permès posar de manifest els coneixements adquirits al llarg dels estudis i l'experiència adquirida en el món laboral.

Així mateix s'han adquirit gran quantitat de coneixements nous, no només relacionats amb l'àmbit tècnic del projecte, sinó també relacionats amb com afrontar, planificar i gestionar problemes de certa magnitud.

7.3 Treball futur

El desenvolupament d'eines i sistemes amb un abast considerable, com el d'aquest projecte, fa que constantment sorgeixin millores o aspectes a modificar. Forma part de la naturalesa del desenvolupament, ja que és freqüent que fins i tot els requeriments dels usuaris vagin canviant amb el temps, i més amb un sistema d'aquestes característiques que pretén funcionar més com un framework, que com una solució final i única per a un sol problema.

Tot i el grau de satisfacció i de compliment dels objectius i requeriments inicials, s'han detectat diferents punts de millora, funcionalitats o aspectes en els que seria interessant continuar treballant.

A continuació es citen els més rellevants:

- Millores en el desplegament de la solució: S'han realitzat proves per empaquetar el programari desenvolupat i els seus requeriments, facilitant-ne la distribució, instal·lació i configuració. Per manca de temps no s'ha pogut donar per acabat aquest aspecte i seria de gran utilitat, especialment fer-ho per aquells dispositius més estesos (per exemple empaquetant el codi dels sensors en paquets de *OpenWRT*).
- Inspecció de dades encapsulades en les trames IEEE 802.11: L'anàlisi del trànsit capturat s'ha centrat en el protocol IEEE 802.11, especialment en la informació continguda a les trames de control i gestió. La dissecció de les trames de dades i la interpretació dels protocols encapsulats en les mateixes podrien aportar molta informació addicional de gran utilitat per a la detecció i la investigació de incidents de seguretat, o d'altres casos d'ús.
- Dissenyar i implementar una arquitectura distribuïda per a tot el sistema i no només per al procés de captura: Les solucions de servidor de cerca utilitzades permeten el funcionament en mode distribuït. S'han realitzat proves preliminars configurant nodes d'aquests servidors de cerca als sensors amb més capacitat de

càlcul. Aquestes proves han estat molt satisfactòries, però per manca de temps no s'han pogut implementar a la solució presentada en aquest projecte.

- Proves unitàries i millores de la qualitat del codi desenvolupat: Donat que no es disposaven de requeriments inicials, ni experiència en alguns àmbits pels quals s'han desenvolupat eines pròpies, aquest desenvolupament s'ha dut a terme seguint un procés similar al del prototipatge. Cal revisar el codi desenvolupat, implementar tests unitaris i realitzar altres modificacions per disposar de codi fàcil de mantenir i revisar.
- Optimitzacions i millores de seguretat en el programari del servidor de cerca i d'anàlisi: S'ha realitzat una instal·lació bàsica d'Elasticsearch i Kibana. Abans de realitzar un desplegament de la solució caldria revisar-ne la configuració i seguir les millors pràctiques per optimitzar-ne el funcionament i per assegurar-ne l'entorn.

8 Bibliografia

[1] Escola Politècnica Superior, "Guia dels projectes/treballs de final de carrera de les enginyeries informàtiques." pp. 1–8.

[2] L. A. N. Man, S. Committee, and I. Computer, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications IEEE Computer Society, vol. 2012, no. March. 2012.

[3] K. Hutchison, "Wireless Intrusion Detection Systems," 2004.

[4] J. Murray, "An Inexpensive Wireless IDS using Kismet and OpenWRT," 2009.

[5] Dragorn, "Kismet Wireless," 2016. [Online]. Available: https://www.kismetwireless.net/.

[6] ZeroMQ, "20/ZRE · ZeroMQ RFC," 2016. [Online]. Available: http://rfc.zeromq.org/spec:20/ZRE/.

[7] ZeroMQ, "ØMQ - The Guide - ØMQ - The Guide," 2016. [Online]. Available: http://zguide.zeromq.org/.

[8] Elastic.co, "Elasticsearch Reference [2.4] | Elastic," 2016. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html.

[9] H. Akdoğan, Elasticsearch indexing. 2015.

[10] C. Gormley and Z. Tong, Elasticsearch the definitive guide : a distributed real-time search and analytics engine. 2015.

[11] Elastic.co, "Kibana User Guide [4.6] | Elastic," 2016. [Online]. Available: https://www.elastic.co/guide/en/kibana/current/index.html.

[12] Y. Gupta, Kibana Essentials. Packt Publishing, 2015.

[13] OpenWrt, "OpenWrt," 2016. [Online]. Available: https://openwrt.org/.

[14] Carhuatocto, R. Everything generates data | Holistic Security and Technology. 2016. [Online]. Available: https://holisticsecurity.wordpress.com/2016/02/02/everything-generates-data-capturing-wifi-anonymous-traffic-raspberrypi-wso2-part-i/

Annexos

A. Sensors: plantilla de configuració servidor Kismet

Aquesta es la plantilla base amb la que es configura el servidor Kismet dels sensors.

```
# Kismet config file
# Most of the "static" configs have been moved to here -- the command line
# config was getting way too crowded and cryptic.  We want functionality,
# not continually reading --help!

# Version of Kismet config
version=2009-newcore

# Name of server (Purely for organizational purposes)
# If commented out, defaults to host name of system
# servername=Kismet Server

# Prefix of where we log (as used in the logtemplate later)
logprefix=/data/kismet

# Do we process the contents of data frames?  If this is enabled, data
# frames will be truncated to the headers only immediately after frame type
# detection.  This will disable IP detection, etc, however it is likely
# safer (and definitely more polite) if monitoring networks you do not own.
# hidedata=true

# Do we allow plugins to be used?  This will load plugins from the system
# and user plugin directiories when set to true (See the README for the default
# plugin locations).
allowplugins=true

# See the README for full information on the new source format
# ncsource=interface:options
# for example:
# AUTOCONFIGURE changes this
ncsource=wlan0
# ncsource=wifi0:type=madwifi
# ncsource=wlan0:name=intel,hop=false,channel=11

```

```
35. # Comma-
    separated list of sources to enable.  This is only needed if you defined
36. # multiple sources and only want to enable some of them.  By default, all defin
    ed
37. # sources are enabled.
38. # For example, if sources with name=prismsource and name=ciscosource are define
    d,
39. # and you only want to enable those two:
40. # enablesources=prismsource,ciscosource
41.
42. # Control which channels we like to spend more time on.  By default, the list
43. # of channels is pulled from the driver automatically.  By setting preferred ch
    annels,
44. # if they are present in the channel list, they'll be set with a timing delay s
    o that
45. # more time is spent on them.  Since 1, 6, 11 are the common default channels,
    it makes
46. # sense to spend more time monitoring them.
47. # For finer control, see further down in the config for the channellist= direct
    ives.
48. preferredchannels=1,6,11
49.
50. # How many channels per second do we hop?  (1-10)
51. channelvelocity=3
52.
53. # By setting the dwell time for channel hopping we override the channelvelocity

54. # setting above and dwell on each channel for the given number of seconds.
55. #channeldwell=10
56.
57. # Channels are defined as:
58. # channellist=name:ch1,ch2,ch3
59. # or
60. # channellist=name:range-start-end-width-offset,ch,range,ch,...
61. #
62. # Channels may be a numeric channel or a frequency
63. #
64. # Channels may specify an additional wait period.  For common default channels,

65. # an additional wait period can be useful.  Wait periods delay for that number

66. # of times per second - so a configuration hopping 10 times per second with a
67. # channel of 6:3 would delay 3/10ths of a second on channel 6.
68. #
69. # Channel lists may have up to 256 channels and ranges (combined).  For power

70. # users scanning more than 256 channels with a single card, ranges must be used
    .
71. #
72. # Ranges are meant for "power users" who wish to define a very large number of

73. # channels.  A range may specify channels or frequencies, and will automaticall
    y
74. # sort themselves to cover channels in a non-overlapping fashion.  An example
75. # range for the normal 802.11b/g spectrum would be:
76. #
77. # range-1-11-3-1
78. #
79. # which indicates starting at 1, ending at 11, a channel width of 3 channels,
80. # incrementing by one.  A frequency based definition would be:
81. #
82. # range-2412-2462-22-5
83. #
```

```
84. # since 11g channels are 22 mhz wide and 5 mhz apart.
85. #
86. # Ranges have the flaw that they cannot be shared between sources in a non-
    overlapping
87. # way, so multiple sources using the same range may hop in lockstep with each o
    ther
88. # and duplicate the coverage.
89. #
90. # channellist=demo:1:3,6:3,11:3,range-5000-6000-20-10
91.
92. # Default channel lists
93. # These channel lists MUST BE PRESENT for Kismet to work properly.  While it is

94. # possible to change these, it is not recommended.  These are used when the sup
    ported
95. # channel list can not be found for the source; to force using these instead of

96. # the detected supported channels, override with channellist= in the source def
    inition
97. #
98. # IN GENERAL, if you think you want to modify these, what you REALLY want to do
     is
99. # copy them and use channellist= in the packet source.
100. channellist=IEEE80211b:1:3,6:3,11:3,2,7,3,8,4,9,5,10
101. channellist=IEEE80211a:36,40,44,48,52,56,60,64,149,153,157,161,165
102. channellist=IEEE80211ab:1:3,6:3,11:3,2,7,3,8,4,9,5,10,36,40,44,48,52,56,60,64,
    149,153,157,161,165
103.
104. # Client/server listen config
105. listen=tcp://127.0.0.1:2501
106. # People allowed to connect, comma seperated IP addresses or network/mask
107. # blocks.  Netmasks can be expressed as dotted quad (/255.255.255.0) or as
108. # numbers (/24)
109. allowedhosts=127.0.0.1
110. # Maximum number of concurrent GUI's
111. maxclients=5
112. # Maximum backlog before we start throwing out or killing clients.  The
113. # bigger this number, the more memory and the more power it will use.
114. maxbacklog=5000
115.
116. # Server + Drone config options.  To have a Kismet server export live packets

117. # as if it were a drone, uncomment these.
118. # dronelisten=tcp://127.0.0.1:3501
119. # droneallowedhosts=127.0.0.1
120. # dronemaxclients=5
121. # droneringlen=65535
122.
123. # OUI file, expected format 00:11:22<tab>manufname
124. # IEEE OUI file used to look up manufacturer info.  We default to the
125. # wireshark one since most people have that.
126. ouifile=/etc/manuf
127. ouifile=/usr/share/wireshark/wireshark/manuf
128. ouifile=/usr/share/wireshark/manuf
129. ouifile=/Applications/Wireshark.app/Contents/Resources/share/wireshark/manuf
130.
131. # Do we have a GPS?
132. gps=false
133. # Do we use a locally serial attached GPS, or use a gpsd server, or
134. # use a fixed virtual gps?
135. # (Pick only one)
136. # gpstype=gpsd
137. # Host:port that GPSD is running on.  This can be localhost OR remote!
```

```
# gpshost=localhost:2947

# gpstype=serial
# What serial device do we look for the GPS on?
# gpsdevice=/dev/rfcomm0

# gpstype=virtual
# gpsposition=100,-50
# gpsaltitude=1234

# Do we lock the mode?  This overrides coordinates of lock "0", which will
# generate some bad information until you get a GPS lock, but it will
# fix problems with GPS units with broken NMEA that report lock 0
gpsmodelock=false
# Do we try to reconnect if we lose our link to the GPS, or do we just
# let it die and be disabled?
gpsreconnect=true

# Do we export packets over tun/tap virtual interfaces?
tuntap_export=false
# What virtual interface do we use
tuntap_device=kistap0

# Packet filtering options:
# filter_tracker - Packets filtered from the tracker are not processed or
#                  recorded in any way.
# filter_export  - Controls what packets influence the exported CSV, network,
#                  xml, gps, etc files.
# All filtering options take arguments containing the type of address and
# addresses to be filtered.  Valid address types are 'ANY', 'BSSID',
# 'SOURCE', and 'DEST'.  Filtering can be inverted by the use of '!' before
# the address.  For example,
# filter_tracker=ANY(!"00:00:DE:AD:BE:EF")
# has the same effect as the previous mac_filter config file option.
# filter_tracker=...
# filter_dump=...
# filter_export=...
# filter_netclient=...

# Alerts to be reported and the throttling rates.
# alert=name,throttle/unit,burst
# The throttle/unit describes the number of alerts of this type that are
# sent per time unit.  Valid time units are second, minute, hour, and day.
# Burst describes the number of alerts sent before throttling takes place.
# For example:
# alert=FOO,10/min,5
# Would allow 5 alerts through before throttling is enabled, and will then
# limit the number of alerts to 10 per minute.
# A throttle rate of 0 disables throttling of the alert.
# See the README for a list of alert types.
alert=ADHOCCONFLICT,5/min,1/sec
alert=AIRJACKSSID,5/min,1/sec
alert=APSPOOF,10/min,1/sec
alert=BCASTDISCON,5/min,2/sec
alert=BSSTIMESTAMP,5/min,1/sec
alert=CHANCHANGE,5/min,1/sec
alert=CRYPTODROP,5/min,1/sec
alert=DISASSOCTRAFFIC,10/min,1/sec
alert=DEAUTHFLOOD,5/min,2/sec
alert=DEAUTHCODEINVALID,5/min,1/sec
alert=DISCONCODEINVALID,5/min,1/sec
```

```
alert=DHCPNAMECHANGE,5/min,1/sec
alert=DHCPOSCHANGE,5/min,1/sec
alert=DHCPCLIENTID,5/min,1/sec
alert=DHCPCONFLICT,10/min,1/sec
alert=NETSTUMBLER,5/min,1/sec
alert=LUCENTTEST,5/min,1/sec
alert=LONGSSID,5/min,1/sec
alert=MSFBCOMSSID,5/min,1/sec
alert=MSFDLINKRATE,5/min,1/sec
alert=MSFNETGEARBEACON,5/min,1/sec
alert=NULLPROBERESP,5/min,1/sec
#alert=PROBENOJOIN,5/min,1/sec

# Controls behavior of the APSPOOF alert.  SSID may be a literal match (ssid=) or
# a regex (ssidregex=) if PCRE was available when kismet was built.  The allowed
# MAC list must be comma-
separated and enclosed in quotes if there are multiple
# MAC addresses allowed.  MAC address masks are allowed.
# AUTOCONFIGURE changes this
apspoof=Foo1:ssidregex="(?i:foobar)",validmacs=00:11:22:33:44:55
apspoof=Foo2:ssid="Foobar",validmacs="00:11:22:33:44:55,aa:bb:cc:dd:ee:ff"

# Known WEP keys to decrypt, bssid,hexkey.  This is only for networks where
# the keys are already known, and it may impact throughput on slower hardware.
# Multiple wepkey lines may be used for multiple BSSIDs.
# wepkey=00:DE:AD:C0:DE:00,FEEDFACEDEADBEEF0102030405060708090

# Is transmission of the keys to the client allowed?  This may be a security
# risk for some.  If you disable this, you will not be able to query keys from
# a client.
allowkeytransmit=true

# How often (in seconds) do we write all our data files (0 to disable)
writeinterval=60

# Do we use sound?
# Not to be confused with GUI sound parameter, this controls wether or not the
# server itself will play sound.  Primarily for headless or automated systems.
enablesound=false
# Path to sound player
soundbin=play

sound=newnet,true
sound=newcryptnet,true
sound=packet,true
sound=gpslock,true
sound=gpslost,true
sound=alert,true

# Does the server have speech? (Again, not to be confused with the GUI's speech)
enablespeech=false
# Binary used for speech (if not in path, full path must be specified)
speechbin=flite
# Specify raw or festival; Flite (and anything else that doesn't need formatting
```

```
# around the string to speak) is 'raw', festival requires the string be wrapped in
# SayText("...")
speechtype=raw

# How do we speak?  Valid options:
# speech    Normal speech
# nato      NATO spellings (alpha, bravo, charlie)
# spell     Spell the letters out (aye, bee, sea)
speechencoding=nato

speech=new,"New network detected s.s.i.d. %1 channel %2"
speech=alert,"Alert %1"
speech=gpslost,"G.P.S. signal lost"
speech=gpslock,"G.P.S. signal O.K."

# How many alerts do we backlog for new clients?  Only change this if you have
# a -very- low memory system and need those extra bytes, or if you have a high
# memory system and a huge number of alert conditions.
alertbacklog=50

# File types to log, comma seperated.  Built-in log file types:
# alert             Text file of alerts
# gpsxml            XML per-packet GPS log
# nettxt            Networks in text format
# netxml            Networks in XML format
# pcapdump          tcpdump/wireshark compatible pcap log file
# string            All strings seen (increases CPU load)
logtypes=pcapdump,gpsxml,netxml,nettxt,alert

# Format of the pcap dump (PPI or 80211)
pcapdumpformat=ppi
# pcapdumpformat=80211

# Default log title
logdefault=Kismet

# logtemplate - Filename logging template.
# This is, at first glance, really nasty and ugly, but you'll hardly ever
# have to touch it so don't complain too much.
#
# %p is replaced by the logging prefix + '/'
# %n is replaced by the logging instance name
# %d is replaced by the starting date as Mon-DD-YYYY
# %D is replaced by the current date as YYYYMMDD
# %t is replaced by the starting time as HH-MM-SS
# %i is replaced by the increment log in the case of multiple logs
# %l is replaced by the log type (pcapdump, strings, etc)
# %h is replaced by the home directory

logtemplate=%p%n-%D-%t-%i.%l

# Where state info, etc, is stored.  You shouldnt ever need to change this.
# This is a directory.
configdir=%h/.kismet/
```

Figura 0-1 Plantilla de configuració de Kismet server

B. Eina de monitoratge de canvis en sistema de fitxers

El sistema desenvolupat es basa en monitorar canvis al sistema de fitxers en diverses ocasions. Quan un sensor disposa de dades noves que ha de transmetre al servidor central, i quan aquest rep les dades. Amb aquesta aproximació ens estalviem comunicacions entre processos i/o sistemes evitant-ne les problemàtiques associades.

Per aquest propòsit s'ha desenvolupat una utilitat que funcioni de forma molt genèrica, per poder ser reutilitzada per diferents propòsits. Aquesta eina ha estat desenvolupada en llenguatge *bash scripting*, per fer-la fàcilment portable i compatible amb gran quantitat de sistemes. S'ha anomenat *supervise.sh*.

El seu funcionament bàsic es pot veure a l'ajuda de la pròpia eina:

```
$ ./supervise.sh --help

 Help using ./supervise.sh:

  -d --dir      [arg]  Directory to monitor. Required.
  -c --command  [arg]  Command to execute on changes. Default="echo"
  -e --events   [arg]  Events to monitor.
                       Default="close_write,create,moved_to,modify"
  -i --include  [arg]  Only include files with matching file extension.
                       Default="$"
  -x --exclude  [arg]  Exclude files matching pattern. Default="'(\.git)'"
  -t --temp     [arg]  Location of tempfile. Default="/tmp/supervise"
  -v                   Enable verbose mode, print script as it is executed
  -g --debug           Enable debug mode
  -h --help            Print this help
  -n --no-color        Disable color output

supervise.sh:

  Supervises a directory for changes and executes actions in response.
```

Figura 0-2 Ajuda de la utilitat supervise.sh

A continuació s'inclou el seu codi font:

```
#!/usr/bin/env bash

#                                                 _
#                ___ _   _ _ __   ___ _ ____   _(_)___  ___
#               / __| | | | '_ \ / _ \ '__\ \ / / / __|/ _ \
#               \__ \ |_| | |_) |  __/ |   \ V /| \__ \  __/
#               |___/\__,_| .__/ \___|_|    \_/ |_|___/\___|
#                         |_|
#
# supervises a directory or file for changes and executes actions in response
#
```

```
#
#  author: Pau Ochoa
#
#
# Usage:
#
#  LOG_LEVEL=7 ./supervise.sh -d "/tmp/directory" -c "command file"
#

#  based on the http://bash3boilerplate.sh script template

# Exit on error. Append || true if you expect an error.
set -o errexit
# Exit on error inside any functions or subshells.
set -o errtrace
# Do not allow use of undefined vars. Use ${VAR:-} to use an undefined VAR
set -o nounset
# Catch errors if command1 fails but command2 succeeds in `command1 | command2`
set -o pipefail
# Turn on traces, useful while debugging but commented out by default
# set -o xtrace

# Set magic variables for current file, directory, os, etc.
__dir="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
__file="${__dir}/$(basename "${BASH_SOURCE[0]}")"
__base="$(basename ${__file} .sh)"

# Define the environment variables (and defaults) that this script depends on
LOG_LEVEL="${LOG_LEVEL:-6}" # 7 = debug -> 0 = emergency
NO_COLOR="${NO_COLOR:-}"    # true = disable color. otherwise autodetected


### Generic functions
################################################################################


function _fmt ()      {
  local color_debug="\x1b[35m"
  local color_info="\x1b[32m"
  local color_notice="\x1b[34m"
  local color_warning="\x1b[33m"
  local color_error="\x1b[31m"
  local color_critical="\x1b[1;31m"
  local color_alert="\x1b[1;33;41m"
  local color_emergency="\x1b[1;4;5;33;41m"
  local colorvar=color_$1

  local color="${!colorvar:-$color_error}"
  local color_reset="\x1b[0m"
  if [ "${NO_COLOR}" = "true" ] || [[ "${TERM:-
}" != "xterm"* ]] && [[ "${TERM:-}" != "screen"* ]] || [ -t 1 ]; then
    # Don't use colors on pipes or non-recognized terminals
    color=""; color_reset=""
  fi
  echo -e "$(date -u +"%Y-%m-
%d %H:%M:%S UTC") ${color}$(printf "[%9s]" ${1})${color_reset}";
}

function emergency () { echo "$(_fmt emergency) ${@}" 1>&2 || true; exit 1; }
function alert ()     { [ "${LOG_LEVEL}" -
ge 1 ] && echo "$(_fmt alert) ${@}" 1>&2 || true; }
```

```
function critical ()  { [ "${LOG_LEVEL}" -ge 2 ] && echo "$(_fmt critical) ${@}" 1>&2 || true; }
function error ()     { [ "${LOG_LEVEL}" -ge 3 ] && echo "$(_fmt error) ${@}" 1>&2 || true; }
function warning ()   { [ "${LOG_LEVEL}" -ge 4 ] && echo "$(_fmt warning) ${@}" 1>&2 || true; }
function notice ()    { [ "${LOG_LEVEL}" -ge 5 ] && echo "$(_fmt notice) ${@}" 1>&2 || true; }
function info ()      { [ "${LOG_LEVEL}" -ge 6 ] && echo "$(_fmt info) ${@}" 1>&2 || true; }
function debug ()     { [ "${LOG_LEVEL}" -ge 7 ] && echo "$(_fmt debug) ${@}" 1>&2 || true; }

function help () {
  echo "" 1>&2
  echo " ${@}" 1>&2
  echo "" 1>&2
  echo "  ${__usage:-No usage available}" 1>&2
  echo "" 1>&2
  echo " ${__helptext:-}" 1>&2
  echo "" 1>&2
  exit 0
}

function print_runtime_info () {
  debug "__file: ${__file}"
  debug "__dir: ${__dir}"
  debug "__base: ${__base}"
  debug "OSTYPE: ${OSTYPE}"

  debug "arg_d: '${arg_d}'"
  debug "arg_c: '${arg_c}'"
  debug "arg_e: '${arg_e}'"
  debug "arg_i: '${arg_i}'"
  debug "arg_x: '${arg_x}'"
  debug "arg_t: '${arg_t}'"

  debug "arg_v: ${arg_v}"
  debug "arg_g: ${arg_g}"
  debug "arg_h: ${arg_h}"
  debug "arg_n: ${arg_n}"
}

function cleanup_before_exit () {
  info "Cleaning up..."
  info "Clean up done."
}
trap cleanup_before_exit EXIT


### Functions
################################################################################


# Run inotifywait forever
function supervise () {
  info "monitoring '${arg_d}' for '${arg_e}' events ..."
  inotifywait -m -q -r -e "${arg_e}" --exclude ${arg_x} "${arg_d}" | grep "${arg_i}" --line-buffered |
    while read path action file; do
      info "[+] new event: '${path}${file}' - '${action}'"
      exe_command=(${arg_c})
      exe_command+=(${path}${file})
```

```
      info "[*]  executing: ${exe_command[@]}"
      ${exe_command[@]} || true
    done
}

### Parse commandline options
#############################################################################

# Commandline options. This defines the usage page, and is used to parse cli
# opts & defaults from. The parsing is unforgiving so be precise in your syntax
read -r -d '' __usage <<-'EOF' || true # exits non-zero when EOF encountered
  -d --dir      [arg]  Directory to monitor. Required.
  -c --command  [arg]  Command to execute on changes. Default="echo"
  -e --events   [arg]  Events to monitor. Default="close_write,create,moved_to,modify"
  -i --include  [arg]  Only include files with matching file extension. Default="$"
  -x --exclude  [arg]  Exclude files matching pattern. Default="'(\.git)'"
  -t --temp     [arg]  Location of tempfile. Default="/tmp/supervise"
  -v                   Enable verbose mode, print script as it is executed
  -g --debug           Enable debug mode
  -h --help            Print this help
  -n --no-color        Disable color output
EOF
read -r -d '' __helptext <<-'EOF' || true # exits non-zero when EOF encountered
 supervise.sh:

   Supervises a directory for changes and executes actions in response.
EOF

# Translate usage string -> getopts arguments, and set $arg_<flag> defaults
while read line; do
  # fetch single character version of option string
  opt="$(echo "${line}" |awk '{print $1}' |sed -e 's#^-##')"

  # fetch long version if present
  long_opt="$(echo "${line}" |awk '/\-\-/ {print $2}' |sed -e 's#^--##')"
  long_opt_mangled="$(sed 's#-#_#g' <<< $long_opt)"

  # map long name back to short name
  varname="short_opt_${long_opt_mangled}"
  eval "${varname}=\"${opt}\""

  # check if option takes an argument
  varname="has_arg_${opt}"
  if ! echo "${line}" |egrep '\[.*\]' >/dev/null 2>&1; then
    init="0" # it's a flag. init with 0
    eval "${varname}=0"
  else
    opt="${opt}:" # add : if opt has arg
    init=""  # it has an arg. init with ""
    eval "${varname}=1"
  fi
  opts="${opts:-}${opt}"

  varname="arg_${opt:0:1}"
  if ! echo "${line}" |egrep '\. Default=' >/dev/null 2>&1; then
    eval "${varname}=\"${init}\""
  else
```

```
    match="$(echo "${line}" |sed 's#^.*Default=\(\)#\1#g')"
    eval "${varname}=\"${match}\""
  fi
done <<< "${__usage}"

# Allow long options like --this
opts="${opts}-:"

# Reset in case getopts has been used previously in the shell.
OPTIND=1

# start parsing command line
set +o nounset # unexpected arguments will cause unbound variables
               # to be dereferenced
# Overwrite $arg_<flag> defaults with the actual CLI options
while getopts "${opts}" opt; do
  [ "${opt}" = "?" ] && help "Invalid use of script: ${@} "

  if [ "${opt}" = "-" ]; then
    # OPTARG is long-option-name or long-option=value
    if [[ "${OPTARG}" =~ .*=.* ]]; then
      # --key=value format
      long=${OPTARG/=*/}
      long_mangled="$(sed 's#-#_#g' <<< $long)"
      # Set opt to the short option corresponding to the long option
      eval "opt=\"\${short_opt_${long_mangled}}\""
      OPTARG=${OPTARG#*=}
    else
      # --key value format
      # Map long name to short version of option
      long_mangled="$(sed 's#-#_#g' <<< $OPTARG)"
      eval "opt=\"\${short_opt_${long_mangled}}\""
      # Only assign OPTARG if option takes an argument
      eval "OPTARG=\"\${@:OPTIND:\${has_arg_${opt}}}\""
      # shift over the argument if argument is expected
      ((OPTIND+=has_arg_${opt}))
    fi
    # we have set opt/OPTARG to the short value and the argument as OPTARG if
    it exists
  fi
  varname="arg_${opt:0:1}"
  default="${!varname}"

  value="${OPTARG}"
  if [ -z "${OPTARG}" ] && [ "${default}" = "0" ]; then
    value="1"
  fi

  eval "${varname}=\"${value}\""
  debug "cli arg ${varname} = ($default) -> ${!varname}"
done
set -o nounset # no more unbound variable references expected

shift $((OPTIND-1))

[ "${1:-}" = "--" ] && shift


### Command-line argument switches (like -g for debugmode, -
    h for showing help)
################################################################################

```

```
241. # debug mode
242. if [ "${arg_g}" = "1" ]; then
243.   set -o xtrace
244.   LOG_LEVEL="7"
245.   print_runtime_info
246. fi
247. 
248. # verbose mode
249. if [ "${arg_v}" = "1" ]; then
250.   set -o verbose
251. fi
252. 
253. # no color mode
254. if [ "${arg_n}" = "1" ]; then
255.   NO_COLOR="true"
256. fi
257. 
258. # help mode
259. if [ "${arg_h}" = "1" ]; then
260.   # Help exists with code 1
261.   help "Help using ${0}:"
262. fi
263. 
264. 
265. ### Validation. Error out if the things required for your script are not prese
    nt
266. ################################################################################
    
267. 
268. [ -z "${arg_d:-}" ]     && error      "Setting a directory with -d or --
    directory is required"
269. [ -z "${LOG_LEVEL:-}" ] && emergency "Cannot continue without LOG_LEVEL. "
270. 
271. 
272. ### Runtime
273. ################################################################################
    
274. 
275. # Start supervising...
276. supervise
```

Figura 0-3 Codi font de la utilitat supervise.sh

C. Eina de processament de fitxers netxml

Per tal de facilitar la indexació dels fitxers de tipus netxml generats pels sensors, s'ha desenvolupat una petita utilitat que converteix el format XML d'origen en un format JSON. A més afegeix unes capçaleres a cada entrada JSON, per poder realitzar una indexació massiva al servidor de cerca, segons ho esperat per Elasticsearch.

Com a paràmetre d'entrada rep el *path* i nom complert d'un fitxer netxml, i en genera dos fitxers resultants, un amb les xarxes Wi-Fi i un altre amb els clients, ambdós en format JSON.

A continuació s'adjunta el codi font d'aquesta utilitat:

```
1.  #!/usr/bin/python
2.
3.  import xml.etree.ElementTree as etree
4.  import json
5.  import os
6.  import sys
7.
8.  #
9.  # Performs netxml to JSON conversion of input files
10. #
11. #
12.
13. #  author: Pau Ochoa
14.
15. def process_netxml_files():
16.     input_file_name = sys.argv[1]
17.
18.     try:
19.         doc = etree.parse(input_file_name)
20.     except:
21.         print("[!] Unable to open file: '%s'." % input_file_name)
22.         exit()
23.
24.     networks, clients = parse_netxml(doc)
25.
26.     file_dump(networks, input_file_name+'.json', "network")
27.     file_dump(clients, input_file_name+'.json', "client")
28.
29.
30. def parse_netxml(doc):
31.     network_list = []
32.     client_list = []
33.
34.     for network in doc.getiterator("wireless-network"):
35.         bssid = network.find('BSSID').text
36.         network_type = network.attrib["type"]
37.         first_seen = network.attrib["first-time"]
38.         last_seen = network.attrib["last-time"]
39.         channel = network.find('channel').text
```

```
40.
41.         ssid = network.find('SSID')
42.         essid = ""
43.         cloaked = ""
44.         ssid_type = ""
45.         if ssid is not None:
46.             essid = ssid.find('essid').text
47.             cloaked = ssid.find('essid').attrib["cloaked"]
48.             ssid_type = ssid.find('type').text
49.
50.         manufacturer = network.find('manuf').text
51.
52.         encryption_list = []
53.         encryption = ""
54.         encryption_elements = network.getiterator('encryption')
55.         for element in encryption_elements:
56.             encryption_list.append(element.text)
57.
58.         encryption = ", ".join(encryption_list)
59.
60.         llc_packets = ""
61.         data_packets = ""
62.         crypt_packets = ""
63.         total_packets = ""
64.         fragmented_packets = ""
65.         retried_packets = ""
66.         packets = network.find('packets')
67.         if packets is not None:
68.             llc_packets = packets.find('LLC').text
69.             data_packets = packets.find('data').text
70.             crypt_packets = packets.find('crypt').text
71.             total_packets = packets.find('total').text
72.             fragmented_packets = packets.find('fragments').text
73.             retried_packets = packets.find('LLC').text
74.
75.         datasize = network.find('datasize').text
76.
77.         max_signal = ""
78.         min_signal = ""
79.         last_signal = ""
80.         power = network.find('snr-info')
81.         if power is not None:
82.             max_signal = power.find('max_signal_dbm').text
83.             min_signal = power.find('min_signal_dbm').text
84.             last_signal = power.find('last_signal_dbm').text
85.
86.         clients = parse_associated_clients(network, bssid)
87.
88.         # assumes that [] is False
89.         if clients != []:
90.             client_list.extend(clients)
91.
92.         parsed_network = dict(bssid=bssid,
93.                               type=network_type,
94.                               first_seen=first_seen,
95.                               last_seen=last_seen,
96.                               channel=channel,
97.                               essid=essid,
98.                               cloaked=cloaked,
99.                               ssid_type=ssid_type,
100.                               manufacturer=manufacturer,
101.                               encryption=encryption,
102.                               llc_packets=llc_packets,
```

```
                                    data_packets=data_packets,
                                    crypt_packets=crypt_packets,
                                    total_packets=total_packets,
                                    fragmented_packets=fragmented_packets,
                                    retried_packets=retried_packets,
                                    datasize=datasize,
                                    max_signal=max_signal,
                                    min_signal=min_signal,
                                    last_signal=last_signal)
            network_list.append(parsed_network)

    return network_list, client_list

def parse_associated_clients(network, bssid):
    client_list = []
    clients = network.getiterator('wireless-client')
    for client in clients:
        mac = client.find('client-mac').text
        manufacturer = client.find('client-manuf').text
        channel = client.find('channel').text
        client_type = "client"
        subtype = client.attrib["type"]
        first_seen = client.attrib["first-time"]
        last_seen = client.attrib["last-time"]
        # should iterate over findall SSID (multiple SSIDs...)
        probed_essids = []
        for ssid in client.iter('SSID'):
            if ssid.find('type').text == "Probe Request":
                if ssid.find('ssid') is not None:
                    probed_essids.append(ssid.find('ssid').text)

        if mac is not None:
            parsed_client = dict(mac=mac,
                                 bssid=bssid,
                                 manufacturer=manufacturer,
                                 channel=channel,
                                 type=client_type,
                                 subtype=subtype,
                                 first_seen=first_seen,
                                 last_seen=last_seen,
                                 probed_essids=", ".join(probed_essids))
            client_list.append(parsed_client)

    return client_list

def print_networks(networks):
    for network in networks:
        print(json.dumps(network))

def print_clients(clients):
    for client in clients:
        print(json.dumps(client))

def file_dump(elements, outfile, json_type):
    with open(outfile, 'a') as out_file:
        for element in elements:
            # using jq before calling curl XPOST we will save file space but
            # we should test processing costs
            header = '{"index": {"_type": "' + json_type + '"}}\n'
            out_file.write(header)
            json.dump(element, out_file)
            out_file.write('\n')

```

```
165.if __name__ == "__main__":
166.    process_netxml_files()
```

Figura 0-4 Codi font de la utilitat netxml_parser.py

D. Eina de processament de fitxers d'alerta

Per tal de facilitar la indexació dels fitxers d'alertes generats pels sensors, s'ha desenvolupat una petita utilitat que els converteix en format JSON. A més afegeix unes capçaleres a cada entrada JSON, per poder realitzar una indexació massiva al servidor de cerca, segons ho esperat per Elasticsearch.

Com a paràmetre d'entrada rep el *path* i nom complert d'un fitxer d'alertes, i en genera un fitxer de sortida en format JSON.

A continuació s'adjunta el codi font d'aquesta utilitat:

```
1.  #!/usr/bin/python
2.
3.  import json
4.  import sys
5.
6.  #
7.  # Performs alert to JSON conversion of input files
8.  #
9.  #
10.
11. #  author: Pau Ochoa
12.
13. def file_dump(elements, outfile, json_type):
14.     with open(outfile, 'a') as out_file:
15.         for element in elements:
16.             # using jq before calling curl XPOST we will save file space but we
     should test processing costs
17.             header = '{"index": {"_type": "' + json_type + '"}}\n'
18.             out_file.write(header)
19.             json.dump(element, out_file, sort_keys=True)
20.             out_file.write('\n')
21.
22. def process_alert_file(filename):
23.     alerts = []
24.
25.     with open(filename, 'r') as in_file:
26.         for alert in in_file:
27.             split_alert = alert.split()
28.             date = " ".join(split_alert[:5])
29.             alert_type = split_alert[5]
30.             channel = split_alert[6]
31.             mac1 = split_alert[7]
32.             mac2 = split_alert[8]
33.             mac3 = split_alert[9]
34.             mac4 = split_alert[10]
35.             message = " ".join(split_alert[11:])
36.             parsed_alert = dict(date=date,
37.                                 type=alert_type,
38.                                 channel=channel,
39.                                 mac1=mac1,
40.                                 mac2=mac2,
```

```
                                    mac3=mac3,
                                    mac4=mac4,
                                    message=message)
            # print(json.dumps(parsed_alert, indent=2))
            # print(alert)
            alerts.append(parsed_alert)

    file_dump(alerts, filename+'.json', "alert")

if __name__ == "__main__":
    process_alert_file(sys.argv[1])
```

Figura 0-5 Codi font de la utilitat alert_parser.py

E. Sensors: modificació firmware

S'ha modificat el *firmware* original pel firmware obert OpenWRT. Aquest procés s'ha executat sense problemes a destacar seguint les pautes genèriques de la guia oficial del projecte OpenWRT[13]:

- https://wiki.openwrt.org/doc/howto/generic.flashing

Concretament s'ha utilitzat el firmware OEM original per carregar la imatge base d'OpenWRT, utilitzant les funcionalitats d'actualització de les que disposava. S'ha fet servir com a basa del sistema la versió *Chaos Calmer trunk r45157 d'OpenWRT*.

Donat que els dispositius utilitzats disposen d'una capacitat d'emmagatzemament molt limitada s'ha utilitzat el que es coneix com a *Rootfs* en un sistema d'emmagatzematge extern per ampliar-la. D'aquesta manera ha estat possible instal·lar el codi desenvolupat en aquest projecte, els seus requeriments i a més a més disposar de més capacitat per emmagatzemar el trànsit capturat. S'ha utilitzat una memòria usb de 128 Gb com a sistema d'emmagatzemament pels sensors del fabricant TP-Link i una memòria microSD també de 128 Gb pels del fabricant Gl.inet.

Per fer aquest procés s'han seguit els següents passos:

- Actualització de paquets i instal·lació de dependències inicials

```
$ opkg update ; opkg install block-mount kmod-fs-ext4 kmod-usb-storage-extras
```

- Preparar el dispositiu:

```
$ mount /dev/sda1 /mnt ; tar -C /overlay -cvf - . | tar -C /mnt -
xf - ; umount /mnt
```

- Creació de fitxer "/etc/fstab" inicial:

```
$ block detect > /etc/config/fstab; \
sed -
i s/option$'\t'enabled$'\t'\'0\'/option$'\t'enabled$'\t'\'1\'/ /etc/config/fsta
b; \
sed -i s#/mnt/sda1#/overlay# /etc/config/fstab; \
cat /etc/config/fstab;
```

- Configurar les particions al fitxer /etc/fstab:

```
config 'global'
        option  anon_swap       '0'
        option  anon_mount      '0'
        option  auto_swap       '1'
        option  auto_mount      '1'
        option  delay_root      '5'
        option  check_fs        '0'

config 'mount'
        option  target  '/overlay'
        option  uuid    'c91232a0-c50a-4eae-adb9-14b4d3ce3de1'
```

```
        option  fstype  'ext4'
        option  enabled '1'

config 'swap'
        option  uuid    '08b4f0a3-f7ab-4ee1-bde9-55fc2481f355'
        option  enabled '1'

config 'mount'
        option  target  '/data'
        option  uuid    'c1068d91-863b-42e2-bcb2-b35a241b0fe2'
        option  enabled '1'
```

- Fer una prova de muntatge de la partició a "/overlay":

```
$ mount /dev/sda1 /overlay
```

- Reiniciar el dispositiu i validar que les particions tenen els punts de muntatge adequats.

F. netxml XML DTD

El propòsit d'un DTD és el de definir l'estructura d'un document XML. A continuació es mostra el DTD del format netxml generat pels sensors i posteriorment processat abans de ser indexat al servidor de cerca:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!-- Kismet network/cisco dump DTD
     version 3.1.0
     dragorn@kismetwireless.net
-->

<!ELEMENT detection-run ( comment?,wireless-network* )>
    <!ATTLIST detection-run kismet-version CDATA "na"
              start-time CDATA "na"
              end-time CDATA "na">
<!ELEMENT comment (#PCDATA)>
<!ELEMENT wireless-
network ( SSID?,BSSID?,info?,channel?,maxrate?,maxseenrate?,carrier*,encoding*,
packets?,gps-info?,ip-address?,datasize?,wireless-client*,cisco* )>
    <!ATTLIST wireless-network number CDATA "0"
              type ( infrastructure | ad-
hoc | probe | data | turbocell | unknown ) "unknown"
              wep ( true | false ) "false"
              cloaked ( true | false ) "false"
              first-time CDATA "na"
              last-time CDATA "na">
<!ELEMENT SSID (#PCDATA)>
<!ELEMENT BSSID (#PCDATA)>
<!ELEMENT info (#PCDATA)>
<!ELEMENT channel (#PCDATA)>
<!ELEMENT maxrate (#PCDATA)>
<!ELEMENT maxseenrate (#PCDATA)>
<!ELEMENT carrier (#PCDATA)>
<!ELEMENT encoding (#PCDATA)>
<!ELEMENT packets (LLC,data,crypt,weak,total,ivdupe)>
    <!ELEMENT LLC (#PCDATA)>
    <!ELEMENT data (#PCDATA)>
    <!ELEMENT crypt (#PCDATA)>
    <!ELEMENT weak (#PCDATA)>
    <!ELEMENT total (#PCDATA)>
    <!ELEMENT ivdupe (#PCDATA)>
<!ELEMENT gps-info (min-lat?,min-lon?,min-alt?,min-spd?,max-lat?,max-lon?,max-
alt?,max-spd?)>
    <!ATTLIST gps-info unit ( english | metric ) "english">
    <!ELEMENT min-lat (#PCDATA)>
    <!ELEMENT min-lon (#PCDATA)>
    <!ELEMENT min-alt (#PCDATA)>
    <!ELEMENT min-spd (#PCDATA)>
    <!ELEMENT max-lat (#PCDATA)>
    <!ELEMENT max-lon (#PCDATA)>
    <!ELEMENT max-alt (#PCDATA)>
    <!ELEMENT max-spd (#PCDATA)>
<!ELEMENT ip-address (ip-range)>
    <!ATTLIST ip-address type (none | arp | udp | tcp | dhcp) "none">
<!ELEMENT ip-range (#PCDATA)>
```

```
<!ELEMENT datasize (#PCDATA)>
<!ELEMENT wireless-client (client-mac?,client-packets?,client-gps-info?,client-maxrate?,client-maxseenrate?,client-encoding*,client-datasize?,client-ip-address?)>
    <!ATTLIST wireless-client number CDATA "0"
              type ( fromds | tods | interds | established | unknown ) "unknown"
              wep (true | false ) "false"
              first-time CDATA "na"
              last-time CDATA "na">
    <!ELEMENT client-mac (#PCDATA)>
    <!ELEMENT client-packets (client-data,client-crypt,client-weak)>
        <!ELEMENT client-data (#PCDATA)>
        <!ELEMENT client-crypt (#PCDATA)>
        <!ELEMENT client-weak (#PCDATA)>
    <!ELEMENT client-gps-info (client-min-lat?,client-min-lon?,client-min-alt?,client-min-spd?,client-max-lat?,client-max-lon?,client-max-alt?,client-max-spd?)>
        <!ATTLIST client-gps-info unit ( english | metric ) "english">
        <!ELEMENT client-min-lat (#PCDATA)>
        <!ELEMENT client-min-lon (#PCDATA)>
        <!ELEMENT client-min-alt (#PCDATA)>
        <!ELEMENT client-min-spd (#PCDATA)>
        <!ELEMENT client-max-lat (#PCDATA)>
        <!ELEMENT client-max-lon (#PCDATA)>
        <!ELEMENT client-max-alt (#PCDATA)>
        <!ELEMENT client-max-spd (#PCDATA)>
    <!ELEMENT client-maxrate (#PCDATA)>
    <!ELEMENT client-maxseenrate (#PCDATA)>
    <!ELEMENT client-encoding (#PCDATA)>
    <!ELEMENT client-channel (#PCDATA)>
    <!ELEMENT client-datasize (#PCDATA)>
    <!ELEMENT client-ip-address (#PCDATA)>
        <!ATTLIST client-ip-address type (none | arp | udp | tcp | dhcp) "none">
<!ELEMENT cisco (cdp-device-id,cdp-capability,cdp-interface,cdp-ip,cdp-platform,cdp-software)>
    <!ATTLIST cisco number CDATA "0">
    <!ELEMENT cdp-device-id (#PCDATA)>
    <!ELEMENT cdp-capability EMPTY>
        <!ATTLIST cdp-capability level1 (true | false) "false"
                  igmp-forward (true | false) "false"
                  netlayer (true | false) "false"
                  level2-switching (true | false) "false"
                  level2-sourceroute (true | false) "false"
                  level2-transparent (true | false) "false"
                  level3-routing (true | false) "false">
    <!ELEMENT cdp-interface (#PCDATA)>
    <!ELEMENT cdp-ip (#PCDATA)>
    <!ELEMENT cdp-platform (#PCDATA)>
    <!ELEMENT cdp-software (#PCDATA)>

<!--
     Copied from HTML 3.2 DTD, with modifications (removed CDATA)
     http://www.w3.org/TR/REC-html32.html#dtd
     ============== BEGIN ==================
-->
<!--
     Character Entities for ISO Latin-1

     (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
```

```
    ISO 8879, provided this notice is included in all copies.
    This has been extended for use with HTML to cover the full
    set of codes in the range 160-255 decimal.
-->
<!-- Character entity set. Typical invocation:
    <!ENTITY % ISOlat1 PUBLIC
      "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
    %ISOlat1;
-->
   <!ENTITY nbsp   " "> <!-- no-break space -->
   <!ENTITY iexcl  "¡"> <!-- inverted exclamation mark -->
   <!ENTITY cent   "¢"> <!-- cent sign -->
   <!ENTITY pound  "£"> <!-- pound sterling sign -->
   <!ENTITY curren "¤"> <!-- general currency sign -->
   <!ENTITY yen    "¥"> <!-- yen sign -->
   <!ENTITY brvbar "¦"> <!-- broken (vertical) bar -->
   <!ENTITY sect   "§"> <!-- section sign -->
   <!ENTITY uml    "¨"> <!-- umlaut (dieresis) -->
   <!ENTITY copy   "©"> <!-- copyright sign -->
   <!ENTITY ordf   "ª"> <!-- ordinal indicator, feminine -->
   <!ENTITY laquo  "«"> <!-- angle quotation mark, left -->
   <!ENTITY not    "¬"> <!-- not sign -->
   <!ENTITY shy    ""> <!-- soft hyphen -->
   <!ENTITY reg    "®"> <!-- registered sign -->
   <!ENTITY macr   "¯"> <!-- macron -->
   <!ENTITY deg    "°"> <!-- degree sign -->
   <!ENTITY plusmn "±"> <!-- plus-or-minus sign -->
   <!ENTITY sup2   "²"> <!-- superscript two -->
   <!ENTITY sup3   "³"> <!-- superscript three -->
   <!ENTITY acute  "´"> <!-- acute accent -->
   <!ENTITY micro  "µ"> <!-- micro sign -->
   <!ENTITY para   "¶"> <!-- pilcrow (paragraph sign) -->
   <!ENTITY middot "·"> <!-- middle dot -->
   <!ENTITY cedil  "¸"> <!-- cedilla -->
   <!ENTITY sup1   "¹"> <!-- superscript one -->
   <!ENTITY ordm   "º"> <!-- ordinal indicator, masculine -->
   <!ENTITY raquo  "»"> <!-- angle quotation mark, right -->
   <!ENTITY frac14 "¼"> <!-- fraction one-quarter -->
   <!ENTITY frac12 "½"> <!-- fraction one-half -->
   <!ENTITY frac34 "¾"> <!-- fraction three-quarters -->
   <!ENTITY iquest "¿"> <!-- inverted question mark -->
   <!ENTITY Agrave "À"> <!-- capital A, grave accent -->
   <!ENTITY Aacute "Á"> <!-- capital A, acute accent -->
   <!ENTITY Acirc  "Â"> <!-- capital A, circumflex accent -->
   <!ENTITY Atilde "Ã"> <!-- capital A, tilde -->
   <!ENTITY Auml   "Ä"> <!-- capital A, dieresis or umlaut mark -->
   <!ENTITY Aring  "Å"> <!-- capital A, ring -->
   <!ENTITY AElig  "Æ"> <!-- capital AE diphthong (ligature) -->
   <!ENTITY Ccedil "Ç"> <!-- capital C, cedilla -->
   <!ENTITY Egrave "È"> <!-- capital E, grave accent -->
   <!ENTITY Eacute "É"> <!-- capital E, acute accent -->
   <!ENTITY Ecirc  "Ê"> <!-- capital E, circumflex accent -->
   <!ENTITY Euml   "Ë"> <!-- capital E, dieresis or umlaut mark -->
   <!ENTITY Igrave "Ì"> <!-- capital I, grave accent -->
   <!ENTITY Iacute "Í"> <!-- capital I, acute accent -->
   <!ENTITY Icirc  "Î"> <!-- capital I, circumflex accent -->
   <!ENTITY Iuml   "Ï"> <!-- capital I, dieresis or umlaut mark -->
   <!ENTITY ETH    "Ð"> <!-- capital Eth, Icelandic -->
   <!ENTITY Ntilde "Ñ"> <!-- capital N, tilde -->
   <!ENTITY Ograve "Ò"> <!-- capital O, grave accent -->
   <!ENTITY Oacute "Ó"> <!-- capital O, acute accent -->
   <!ENTITY Ocirc  "Ô"> <!-- capital O, circumflex accent -->
   <!ENTITY Otilde "Õ"> <!-- capital O, tilde -->
```

```
167.    <!ENTITY Ouml   "Ö"> <!-- capital O, dieresis or umlaut mark -->
168.    <!ENTITY times  "×"> <!-- multiply sign -->
169.    <!ENTITY Oslash "Ø"> <!-- capital O, slash -->
170.    <!ENTITY Ugrave "Ù"> <!-- capital U, grave accent -->
171.    <!ENTITY Uacute "Ú"> <!-- capital U, acute accent -->
172.    <!ENTITY Ucirc  "Û"> <!-- capital U, circumflex accent -->
173.    <!ENTITY Uuml   "Ü"> <!-- capital U, dieresis or umlaut mark -->
174.    <!ENTITY Yacute "Ý"> <!-- capital Y, acute accent -->
175.    <!ENTITY THORN  "Þ"> <!-- capital THORN, Icelandic -->
176.    <!ENTITY szlig  "ß"> <!-- small sharp s, German (sz ligature) -->
177.    <!ENTITY agrave "à"> <!-- small a, grave accent -->
178.    <!ENTITY aacute "á"> <!-- small a, acute accent -->
179.    <!ENTITY acirc  "â"> <!-- small a, circumflex accent -->
180.    <!ENTITY atilde "ã"> <!-- small a, tilde -->
181.    <!ENTITY auml   "ä"> <!-- small a, dieresis or umlaut mark -->
182.    <!ENTITY aring  "å"> <!-- small a, ring -->
183.    <!ENTITY aelig  "æ"> <!-- small ae diphthong (ligature) -->
184.    <!ENTITY ccedil "ç"> <!-- small c, cedilla -->
185.    <!ENTITY egrave "è"> <!-- small e, grave accent -->
186.    <!ENTITY eacute "é"> <!-- small e, acute accent -->
187.    <!ENTITY ecirc  "ê"> <!-- small e, circumflex accent -->
188.    <!ENTITY euml   "ë"> <!-- small e, dieresis or umlaut mark -->
189.    <!ENTITY igrave "ì"> <!-- small i, grave accent -->
190.    <!ENTITY iacute "í"> <!-- small i, acute accent -->
191.    <!ENTITY icirc  "î"> <!-- small i, circumflex accent -->
192.    <!ENTITY iuml   "ï"> <!-- small i, dieresis or umlaut mark -->
193.    <!ENTITY eth    "ð"> <!-- small eth, Icelandic -->
194.    <!ENTITY ntilde "ñ"> <!-- small n, tilde -->
195.    <!ENTITY ograve "ò"> <!-- small o, grave accent -->
196.    <!ENTITY oacute "ó"> <!-- small o, acute accent -->
197.    <!ENTITY ocirc  "ô"> <!-- small o, circumflex accent -->
198.    <!ENTITY otilde "õ"> <!-- small o, tilde -->
199.    <!ENTITY ouml   "ö"> <!-- small o, dieresis or umlaut mark -->
200.    <!ENTITY divide "÷"> <!-- divide sign -->
201.    <!ENTITY oslash "ø"> <!-- small o, slash -->
202.    <!ENTITY ugrave "ù"> <!-- small u, grave accent -->
203.    <!ENTITY uacute "ú"> <!-- small u, acute accent -->
204.    <!ENTITY ucirc  "û"> <!-- small u, circumflex accent -->
205.    <!ENTITY uuml   "ü"> <!-- small u, dieresis or umlaut mark -->
206.    <!ENTITY yacute "ý"> <!-- small y, acute accent -->
207.    <!ENTITY thorn  "þ"> <!-- small thorn, Icelandic -->
208.    <!ENTITY yuml   "ÿ"> <!-- small y, dieresis or umlaut mark -->
209.
210.<!--
211.     Copied from HTML 3.2 DTD, with modifications (removed CDATA)
212.     http://www.w3.org/TR/REC-html32.html#dtd
213.     ================= END ==================
214.-->
```

Figura 0-6 netxml DTD

Cal destacar que el DTD no està especialment ben format i que hi ha camps que no estan explícitament definits, però s'ha decidit utilitzar-lo al tractar-se del DTD definit pel programari Kismet Wireless que va implementar el format netxml inicialment.

G. *Dockerfile* del servidor de cerca i d'anàlisi

Per la creació, desplegament i execució dels servidors de cerca (Elasticsearch) i d'anàlisi (Kibana) s'ha utilitzat un fitxer Docker, basat en el proporcionat per Sébastien Pujadas[36]:

```
# Dockerfile for ELK stack
# Elasticsearch 2.3.5, Logstash 2.3.4, Kibana 4.5.4

# Build with:
# docker build -t <repo-user>/elk .

# Run with:
# docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -p 5000:5000 -it --name elk <repo-user>/elk

FROM phusion/baseimage
MAINTAINER Sebastien Pujadas http://pujadas.net
ENV REFRESHED_AT 2016-08-20

###############################################################################

#                                INSTALLATION
###############################################################################

### install prerequisites (cURL, gosu)

ENV GOSU_VERSION 1.8

ARG DEBIAN_FRONTEND=noninteractive
RUN set -x \
 && apt-get update -qq \
 && apt-get install -qqy --no-install-recommends ca-certificates curl \
 && rm -rf /var/lib/apt/lists/* \
 && curl -L -o /usr/local/bin/gosu "https://github.com/tianon/gosu/releases/download/$GOSU_VERSION/gosu-$(dpkg --print-architecture)" \
 && curl -L -o /usr/local/bin/gosu.asc "https://github.com/tianon/gosu/releases/download/$GOSU_VERSION/gosu-$(dpkg --print-architecture).asc" \
 && export GNUPGHOME="$(mktemp -d)" \
 && gpg --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-keys B42F6819007F00F88E364FD4036A9C25BF357DD4 \
 && gpg --batch --verify /usr/local/bin/gosu.asc /usr/local/bin/gosu \
 && rm -r "$GNUPGHOME" /usr/local/bin/gosu.asc \
 && chmod +x /usr/local/bin/gosu \
 && gosu nobody true \
 && apt-get clean \
 && set +x


### install Elasticsearch

```

[36] https://github.com/spujadas/elk-docker

```
ENV ES_VERSION 2.3.5
ENV ES_GID 991
ENV ES_UID 991

RUN curl http://packages.elasticsearch.org/GPG-KEY-elasticsearch | apt-key add -
RUN echo deb http://packages.elasticsearch.org/elasticsearch/2.x/debian stable main > /etc/apt/sources.list.d/elasticsearch-2.x.list

RUN groupadd -r elasticsearch -g ${ES_GID} \
 && useradd -r -s /usr/sbin/nologin -M -c "Elasticsearch service user" -u ${ES_UID} -g elasticsearch elasticsearch \
 && apt-get update -qq \
 && apt-get install -qqy \
        elasticsearch=${ES_VERSION} \
        openjdk-8-jdk \
 && apt-get clean

### install Logstash

ENV LOGSTASH_VERSION 2.3.4
ENV LOGSTASH_HOME /opt/logstash
ENV LOGSTASH_PACKAGE logstash-${LOGSTASH_VERSION}.tar.gz
ENV LOGSTASH_GID 992
ENV LOGSTASH_UID 992

RUN mkdir ${LOGSTASH_HOME} \
 && curl -O https://download.elasticsearch.org/logstash/logstash/${LOGSTASH_PACKAGE} \
 && tar xzf ${LOGSTASH_PACKAGE} -C ${LOGSTASH_HOME} --strip-components=1 \
 && rm -f ${LOGSTASH_PACKAGE} \
 && groupadd -r logstash -g ${LOGSTASH_GID} \
 && useradd -r -s /usr/sbin/nologin -d ${LOGSTASH_HOME} -c "Logstash service user" -u ${LOGSTASH_UID} -g logstash logstash \
 && mkdir -p /var/log/logstash /etc/logstash/conf.d \
 && chown -R logstash:logstash ${LOGSTASH_HOME} /var/log/logstash

ADD ./logstash-init /etc/init.d/logstash
RUN sed -i -e 's#^LS_HOME=$#LS_HOME='$LOGSTASH_HOME'#' /etc/init.d/logstash \
 && chmod +x /etc/init.d/logstash

### install Kibana

ENV KIBANA_VERSION 4.5.4
ENV KIBANA_HOME /opt/kibana
ENV KIBANA_PACKAGE kibana-${KIBANA_VERSION}-linux-x64.tar.gz
ENV KIBANA_GID 993
ENV KIBANA_UID 993

RUN mkdir ${KIBANA_HOME} \
 && curl -O https://download.elasticsearch.org/kibana/kibana/${KIBANA_PACKAGE} \
 && tar xzf ${KIBANA_PACKAGE} -C ${KIBANA_HOME} --strip-components=1 \
 && rm -f ${KIBANA_PACKAGE} \
 && groupadd -r kibana -g ${KIBANA_GID} \
 && useradd -r -s /usr/sbin/nologin -d ${KIBANA_HOME} -c "Kibana service user" -u ${KIBANA_UID} -g kibana kibana \
 && mkdir -p /var/log/kibana \
 && chown -R kibana:kibana ${KIBANA_HOME} /var/log/kibana

ADD ./kibana-init /etc/init.d/kibana
RUN sed -i -e 's#^KIBANA_HOME=$#KIBANA_HOME='$KIBANA_HOME'#' /etc/init.d/kibana \
```

```
 && chmod +x /etc/init.d/kibana


################################################################################

#                                   CONFIGURATION
################################################################################
#

### configure Elasticsearch

ADD ./elasticsearch.yml /etc/elasticsearch/elasticsearch.yml


### configure Logstash

# certs/keys for Beats and Lumberjack input
RUN mkdir -p /etc/pki/tls/certs && mkdir /etc/pki/tls/private
ADD ./logstash-forwarder.crt /etc/pki/tls/certs/logstash-forwarder.crt
ADD ./logstash-forwarder.key /etc/pki/tls/private/logstash-forwarder.key
ADD ./logstash-beats.crt /etc/pki/tls/certs/logstash-beats.crt
ADD ./logstash-beats.key /etc/pki/tls/private/logstash-beats.key

# filters
ADD ./01-lumberjack-input.conf /etc/logstash/conf.d/01-lumberjack-input.conf
ADD ./02-beats-input.conf /etc/logstash/conf.d/02-beats-input.conf
ADD ./10-syslog.conf /etc/logstash/conf.d/10-syslog.conf
ADD ./11-nginx.conf /etc/logstash/conf.d/11-nginx.conf
ADD ./30-output.conf /etc/logstash/conf.d/30-output.conf

# patterns
ADD ./nginx.pattern ${LOGSTASH_HOME}/patterns/nginx
RUN chown -R logstash:logstash ${LOGSTASH_HOME}/patterns


### configure logrotate

ADD ./elasticsearch-logrotate /etc/logrotate.d/elasticsearch
ADD ./logstash-logrotate /etc/logrotate.d/logstash
ADD ./kibana-logrotate /etc/logrotate.d/kibana
RUN chmod 644 /etc/logrotate.d/elasticsearch \
 && chmod 644 /etc/logrotate.d/logstash \
 && chmod 644 /etc/logrotate.d/kibana

################################################################################
#                                     START
################################################################################

ADD ./start.sh /usr/local/bin/start.sh
RUN chmod +x /usr/local/bin/start.sh

EXPOSE 5601 9200 9300 5000 5044
VOLUME /var/lib/elasticsearch

CMD [ "/usr/local/bin/start.sh" ]
```

Figura 0-7 Dockerfile Elasticsearch i Kibana

L'autor original d'aquesta imatge Docker l'ha documentat de forma molt complerta a:

- http://elk-docker.readthedocs.io/

Seguint aquesta documentació i amb el fitxer Docker es pot reproduir fàcilment el servidor de cerca i d'anàlisi utilitzat a l'entorn de proves d'aquest projecte.