


Universitat de Girona
Escola Politècnica Superior

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Implementació i experimentació amb un sistema de detecció d'intrusos en una xarxa d'una PIME

Document: Memòria

Alumne: Adrià Descamps Vilà

Tutor: Antonio Bueno Delgado

Departament: Arquitectura i Tecnologia de Computadors

Àrea: Arquitectura i Tecnologia de Computadors

Convocatòria (Setembre/2016)

Implementació i experimentació amb un sistema de detecció d'intrusos en una xarxa d'una PIME

ÍNDEX

| | | |
|-------|--|----|
| 1 | Introducció..... | 1 |
| 1.1 | Propòsits | 1 |
| 1.2 | Objectius inicials..... | 2 |
| 1.3 | Replantejament del projecte | 2 |
| 2 | Estudi de viabilitat..... | 3 |
| 3 | Metodologia..... | 4 |
| 3.1 | Metodologia d'implementació..... | 4 |
| 3.1.1 | Estudi de les característiques | 4 |
| 3.1.2 | Cerca d'eines | 4 |
| 3.1.3 | Comparativa i decisió..... | 4 |
| 3.1.4 | Estudi de l'eina..... | 4 |
| 3.1.5 | Instal·lació i configuració..... | 5 |
| 3.1.6 | Comprovació de funcionament | 5 |
| 3.1.7 | Documentació..... | 5 |
| 3.2 | Metodologia de desenvolupament | 5 |
| 3.2.1 | Estudi del projecte | 5 |
| 3.2.2 | Identificació de les tasques a realitzar | 6 |
| 3.2.3 | Programació de la tasca | 6 |
| 3.2.4 | Comprovació de la tasca | 6 |
| 3.2.5 | Documentació..... | 6 |
| 4 | Planificació..... | 7 |
| 4.1 | Planificació Inicial | 8 |
| 4.2 | Planificació Final | 8 |
| 5 | Marc de Treball i Conceptes Previs..... | 9 |
| 5.1 | Conceptes previs | 9 |
| 5.1.1 | IDS..... | 9 |
| 5.1.2 | Capes OSI..... | 9 |
| 5.1.3 | Payload..... | 10 |
| 5.1.4 | Buffer | 10 |
| 5.1.5 | URI..... | 11 |
| 5.1.6 | Màquina Virtual | 11 |
| 5.1.7 | IDE..... | 11 |

| | | |
|--------|---|----|
| 5.1.8 | Framework | 11 |
| 5.1.9 | Repositori | 11 |
| 5.1.10 | Regles | 12 |
| 5.1.11 | Commit..... | 11 |
| 5.2 | Marc de treball implementació ids..... | 12 |
| 5.2.1 | Eines utilitzades | 12 |
| 5.3 | Marc de treball desenvolupament | 13 |
| 5.3.1 | Eines utilitzades | 13 |
| 5.3.2 | Llenguatges..... | 14 |
| 6 | Requisits del sistema | 16 |
| 6.1 | Requisits Implementació del Sistema | 16 |
| 6.2 | Requisits Desenvolupament..... | 16 |
| 7 | Estudis i Decisions | 17 |
| 7.1 | IDS..... | 17 |
| 7.1.1 | SNORT | 17 |
| 7.1.2 | SURICATA..... | 19 |
| 7.1.3 | SNORT++ (3.0) [9] | 21 |
| 7.1.4 | Taula comparativa | 23 |
| 7.1.5 | Decisió..... | 24 |
| 7.2 | Visualitzador de contingut | 27 |
| 7.2.1 | BASE | 27 |
| 7.2.2 | Snorby | 28 |
| 7.2.3 | Elastic Stack | 29 |
| 7.2.4 | Taula comparativa | 31 |
| 7.2.5 | Decisió..... | 31 |
| 7.3 | Replantejament del projecte | 33 |
| 7.4 | Desenvolupament | 35 |
| 7.5 | IDE..... | 36 |
| 7.5.1 | Eclipse | 36 |
| 7.5.2 | Netbeans | 36 |
| 7.5.3 | IntelliJ | 37 |
| 7.5.4 | Taula comparativa | 38 |
| 7.5.5 | Decisió..... | 38 |
| 8 | Anàlisi i disseny del sistema..... | 39 |

| | | |
|-------|---|----|
| 8.1 | Anàlisi funcional | 39 |
| 8.1.1 | Configuració..... | 39 |
| 8.1.2 | Visualització de dades | 40 |
| 8.2 | Anàlisi estructural | 41 |
| 8.3 | Disseny de les tasques | 42 |
| 8.3.1 | Disseny inicial | 43 |
| 8.3.2 | Modificar l'acció d'una regla | 44 |
| 8.3.3 | Modificar l'acció de les regles d'una categoria | 44 |
| 8.3.4 | Transformar múltiples regles d'una categoria en múltiples Rulesets..... | 45 |
| 8.3.5 | Transformar totes les regles de múltiples categories..... | 45 |
| 8.3.6 | Disseny final..... | 46 |
| 9 | Implementació i proves | 47 |
| 9.1 | Sistema Operatiu | 47 |
| 9.1.1 | Creació de la màquina virtual | 48 |
| 9.1.2 | Instal·lació del Sistema Operatiu | 48 |
| 9.2 | Suricata | 49 |
| 9.2.1 | Prerequisits..... | 49 |
| 9.2.2 | Descàrrega i instal·lació..... | 49 |
| 9.2.3 | Configuració..... | 49 |
| 9.2.4 | Línia de comandes [12] | 50 |
| 9.3 | Oinkmaster..... | 52 |
| 9.3.1 | Instal·lació..... | 52 |
| 9.3.2 | Configuració..... | 52 |
| 9.3.3 | Actualització automàtica de regles | 53 |
| 9.4 | Elasticsearch..... | 54 |
| 9.4.1 | Prerequisits..... | 54 |
| 9.4.2 | Instal·lació..... | 54 |
| 9.4.3 | Configuració..... | 54 |
| 9.5 | Logstash | 55 |
| 9.5.1 | Instal·lació..... | 55 |
| 9.5.2 | Configuració..... | 55 |
| 9.6 | Kibana | 56 |
| 9.6.1 | Prerequisits..... | 56 |
| 9.6.2 | Instal·lació..... | 56 |

| | | |
|--------|---|----|
| 9.6.3 | Connexió de Kibana amb Elasticsearch..... | 56 |
| 9.6.4 | Configuració..... | 57 |
| 9.7 | Scirius..... | 59 |
| 9.7.1 | Introducció | 59 |
| 9.7.2 | Instal·lació i configuració..... | 59 |
| 9.7.3 | Enllaçar amb Elasticsearch | 60 |
| 9.7.4 | Enllaçar amb Kibana..... | 60 |
| 10 | Implantació i resultats | 62 |
| 10.1 | Regla | 62 |
| 10.1.1 | Visualització d'una regla..... | 62 |
| 10.1.2 | Modificació d'una regla..... | 63 |
| 10.2 | Categoria..... | 63 |
| 10.2.1 | Modificació de totes les regles d'una categoria | 64 |
| 10.2.2 | Modificació de múltiples regles..... | 64 |
| 10.3 | Font..... | 66 |
| 10.3.1 | Modificació de totes les regles de múltiples categories | 66 |
| 11 | Conclusions | 68 |
| 12 | Treball futur | 69 |
| 13 | Bibliografia | 71 |
| 14 | Índex de figures i taules..... | 74 |
| 14.1 | Índex de figures | 74 |
| 14.2 | Índex de taules | 74 |
| 15 | Annexos | 75 |
| 15.1 | Planificació inicial | 75 |
| 15.2 | Planificació final..... | 76 |
| 15.3 | Creació de la màquina virtual..... | 77 |
| 15.4 | Instal·lació del Sistema Operatiu..... | 78 |
| 15.4.1 | Creació de la màquina virtual | 78 |
| 15.4.2 | Instal·lació del Sistema Operatiu | 80 |
| 15.5 | Suricata | 83 |
| 15.5.1 | Prerequisits..... | 83 |
| 15.5.2 | Descàrrega i instal·lació..... | 84 |
| 15.5.3 | Configuració..... | 85 |
| 15.5.4 | Línia de comandes | 88 |

| | | |
|---------|---|-----|
| 15.6 | Elasticsearch..... | 90 |
| 15.6.1 | Prerequisits..... | 90 |
| 15.6.2 | Instal·lació..... | 90 |
| 15.6.3 | Configuració..... | 91 |
| 15.7 | Logstash | 93 |
| 15.7.1 | Instal·lació..... | 93 |
| 15.7.2 | Configuració..... | 93 |
| 15.8 | Kibana | 96 |
| 15.8.1 | Prerequisits..... | 96 |
| 15.8.2 | Instal·lació..... | 96 |
| 15.8.3 | Connexió de Kibana amb Elasticsearch..... | 97 |
| 15.8.4 | Configuració..... | 98 |
| 15.9 | Scirius..... | 100 |
| 15.9.1 | Introducció | 100 |
| 15.9.2 | Instal·lació i configuració..... | 100 |
| 15.9.3 | Enllaçar amb Elasticsearch | 102 |
| 15.9.4 | Enllaçar amb Kibana..... | 102 |
| 15.10 | Oinkmaster | 103 |
| 15.10.1 | Instal·lació..... | 103 |
| 15.10.2 | Configuració..... | 103 |
| 15.10.3 | Actualització automàtica de regles | 105 |

1 INTRODUCCIÓ

Vivim en un món en constant evolució i les xarxes de telecomunicacions no en són una excepció. En els darrers anys hem pogut veure com la quantitat de persones que tenen accés a la xarxa està augmentant de forma exponencial i com la tecnologia que s'utilitza està millorant i canviant.

Aquest increment tan elevat en la utilització de les xarxes ve donat, en part, per una gran facilitat en l'accés a aquestes. Això ens porta a que hi ha molts usuaris nous que entren en aquest món diàriament. El fet que cada cop hi hagi més persones connectades, així com la facilitat que hi ha en fer-ho, ens porta a un tipus d'usuari cada cop menys preparat i menys conscient del que suposa estar connectat. Un exemple podria ser el fet que cada cop s'accedeix a la xarxa més jove, fet que provoca que fins i tot infants tinguin accés a tot tipus d'informació i siguin susceptibles de rebre múltiples atacs.

També hi ha gent dedicada únicament a atacar de manera indiscriminada a objectius d'arreu del món ja que els interessos econòmics en el camp de les telecomunicacions són enormes. Aquesta gent utilitza, entre d'altres, programari maligne per infectar dispositius i, un cop infectats i sota el seu control, realitzen atacs de diferents tipus a objectius de la xarxa.

Un Sistema de Detecció d'Intrusos (IDS) ens ofereix seguretat i control sobre la nostra xarxa. Tant en un ambient empresarial com particular ens serveix per detectar tot el que hi entra i surt i permet controlar-ne el seu nivell de vulnerabilitat. Un dels sistemes per detectar aquests atacs i comportaments semblants, tant si es reben com si un usuari de la xarxa es troba infectat i en realitza, és la utilització dels IDS. Gràcies a l'anàlisi del comportament de la xarxa que efectuen són capaços de detectar aquests comportaments i et permet actuar-hi en conseqüència.

Per a l'administrador de sistemes o el tècnic de telecomunicacions encarregat de la xarxa un IDS és una eina molt útil per controlar l'acció dels usuaris i actuar en cas de veure un comportament perillós, tant per part d'un usuari com per part d'un agent exterior a la xarxa local.

1.1 PROPÒSITS

Després de fer les pràctiques en un ambient empresarial i, entre altres feines, treballar en contacte constant amb la xarxa de telecomunicacions d'una empresa mitjana, he après molt respecte el seu funcionament. També he conegut els avantatges i inconvenients que aquestes presenten i com això afecta als usuaris finals, és a dir, la gent que utilitza aquesta xarxa.

Amb l'experiència obtinguda i juntament amb una de les branques que sempre m'ha cridat l'atenció de la informàtica com és la seguretat, he volgut fer un projecte que tingui una aplicació pràctica i que, al mateix temps, generi interès.

La proposta del projecte es basa, doncs, en controlar per mitjà del Sistema de Detecció d'Intrusos la seguretat en la nostra xarxa. En aquesta proposta es busca estudiar la interacció de les xarxes de telecomunicacions amb la seguretat que aquestes tenen i com, per mitjà d'un IDS, es pot arribar a controlar i augmentar de forma important el seu nivell de seguretat.

1.2 OBJECTIUS INICIALS

Aquest projecte busca obtenir un sistema que analitzi la nostra xarxa i permeti fer canvis en la configuració de forma còmode i senzilla.

També ens interessa tenir una interfície gràfica on es puguin visualitzar totes les dades recollides, les diferents alertes que detecti el sistema i se'ns mostri de forma clara i entenedora.

Els objectius són:

1. Estudiar, implementar i analitzar un Sistema de Detecció d'Intrusos.
2. Estudiar, crear o implementar i analitzar una interfície gràfica per visualitzar totes les dades recopilades amb l'IDS.
3. Estudiar, crear o implementar i analitzar un sistema de notificacions a temps real per avisos d'alertes a l'administrador de sistemes o el tècnic de telecomunicacions encarregat de la xarxa i de l'IDS.
4. Estudiar, crear o implementar i analitzar una interfície gràfica per a gestionar i configurar d'una forma més còmode i senzilla el sistema IDS escollit.

1.3 REPLANTEJAMENT DEL PROJECTE

Durant la realització del projecte, un cop acabats els objectius 1 i 2, vaig descobrir un projecte, SELKS [1], que utilitzava les mateixes eines que jo havia escollit per fer el meu i, a més a més, incorporava una interfície gràfica de gestió del Sistema de Detecció d'Intrusos.

En aquest moment vaig decidir fer un pivotatge del meu projecte, ja que no veia cap sentit en reinventar la roda fent exactament el mateix que ja estava fet.

Part de la decisió, també, va ser presa en base a que un possible desenvolupament per part meua d'una eina de característiques similars mai arribaria a ser com una eina desenvolupada per experts en la matèria, i amb més de tres anys d'experiència en aquest projecte.

Això va provocar que hagués de replantejar-me els objectius que m'havia proposat a l'inici d'aquest projecte. Tot i la troballa, no vaig voler canviar-los totalment i vaig decidir intentar millorar el seu projecte.

Aquesta decisió m'implicava diverses coses, passava d'estar treballant en un projecte propi i ser-ne el responsable a estar treballant sobre un projecte comú en què, al ser qui s'hi afegia un cop aquest ja estava desenvolupat, m'hi havia d'adaptar i passar a ser un actor secundari. Aquest fet afectava la manera de treballar, ja que ja no era qui prenia les decisions sinó que s'havien de comentar i discutir, i esperar que decidissin els coordinadors del projecte.

En el punt en el que em trobava, vaig decidir prescindir dels objectius 3 i 4 de forma temporal i centrar-me en els següent:

5. Familiaritzar-me en profunditat amb el sistema i les eines utilitzades en el projecte.
6. Estudiar, analitzar i entendre el projecte.
7. Crear o millorar una funcionalitat en el projecte.

2 ESTUDI DE VIABILITAT

El desenvolupament del projecte en termes de viabilitat es podria separar en dos grans apartats, la viabilitat econòmica en quant al Hardware que necessita i la viabilitat econòmica i funcional en quant al software.

Per realitzar aquest projecte és necessari un maquinari amb unes característiques importants. Això és degut a que el funcionament del projecte utilitza unes capacitats de computació i anàlisi molt grans. Normalment un equip que complís els paràmetres necessaris per a ser utilitzat en aquest projecte costaria milers d'Euros, i en el meu cas no és una opció fer una inversió d'aquesta magnitud.

Per resoldre aquest problema, que és bàsic per a poder implementar el sistema ja que sinó no es podria fer la part pràctica del projecte, he aconseguit que el Parc Científic i Tecnològic de la UdG, on treballa actualment, em permeti utilitzar temporalment un equip amb les característiques suficients per a poder posar en pràctica tot el sistema que es vol estudiar i implementar en aquest projecte. L'equip en qüestió es troba situat a les instal·lacions del Parc Científic i Tecnològic de la UdG, on es tenen les sales degudament condicionades per a aquest tipus de dispositius ja que necessiten una temperatura i humitat constants i jo no tinc la possibilitat de tenir un espai així.

Un altre punt important és quin sistema s'utilitzarà per a realitzar el projecte i quins softwares es faran servir. En el mercat n'hi ha de moltes classes, des de sistemes propietaris i amb cost econòmic a sistemes de codi lliure i gratuïts. En el nostre cas ens interessarà un sistema gratuït degut a la impossibilitat de fer cap inversió econòmica en aquest aspecte. A més a més, dins dels diferents sistemes gratuïts es donarà prioritat als que tinguin el codi lliure, ja que et permet fer modificacions en cas de necessitar-ho. Tot i que aquest últim punt queda molt lligat a la viabilitat funcional que ens proporcionin els diferents sistemes ja que aquests han de ser compatibles amb l'equip que disposem i han de complir amb les nostres necessitats.

3 METODOLOGIA

3.1 METODOLOGIA D'IMPLEMENTACIÓ

Per a aquesta part s'ha seguit una mateixa metodologia per a tots els diferents blocs que s'han treballat. Els diferents passos que s'han seguit ens han permès treballar d'una forma més estructurada i consistent.

3.1.1 Estudi de les característiques

El primer que s'ha fet és analitzar quines són les característiques que ha de complir l'eina o sistema que es vol utilitzar. Per fer-ho, es fa una llista de les més destacades i es fa la cerca en concordança amb aquestes.

En línies generals, les característiques que s'han buscat en tots els casos han estat que siguin gratuïtes, ja que no tenim pressupost per aconseguir-ne d'una altra forma i, a ser possible, que siguin de codi lliure, ja que ens proporciona un marge de modificació en cas de necessitar una funcionalitat específica. També ens interessa que es trobi en desenvolupament o que tingui un suport darrere, ja que si s'hi troba algun error, es pugui aconseguir una solució en versions pròximes. Finalment, es comprova la comunitat que utilitza aquesta eina o sistema, ja que és un dels indicadors més importants del seu funcionament.

3.1.2 Cerca d'eines

Un cop sabem quines característiques ha de complir l'eina que s'està buscant, es comença a fer una recerca de les que es troben disponibles.

Per fer-ho, primer es busquen totes les eines que facin la funció que es necessita i, un cop es trobin, es comprova quines d'aquestes compleixen les característiques especificades en l'apartat anterior.

Es fa un resum de cada eina amb les característiques específiques que té i si destaca en algun punt respecte les altres eines.

3.1.3 Comparativa i decisió

Amb les diferents eines explicades, podem fer una comparativa objectiva de quina és la que s'ajusta més a les nostres necessitats.

Es comprova per cada característica quin valor té cada eina. D'aquesta manera es pot tenir una visió molt més objectiva.

Per decidir quina és l'eina o el sistema a escollir, s'estudien les diferents comparatives fetes i, juntament amb els comentaris de la comunitat sobre aquests, es decideix quin es farà servir.

3.1.4 Estudi de l'eina

Ja amb la decisió de quina eina s'utilitza, es fa un estudi a fons d'aquesta. S'estudien totes les funcionalitats que ofereix i com s'implementen.

3.1.5 Instal·lació i configuració

Ja amb tota la informació relacionada amb l'eina, es procedeix a la seva instal·lació.

Per fer-ho, primer es comproven els requisits que existeixen per a l'eina en qüestió. Un cop instal·lats, es comproven els passos a seguir per fer la instal·lació i s'efectuen.

Amb l'eina instal·lada, s'estudien les diferents opcions que es poden configurar i quines s'ajusten més a les nostres necessitats.

Amb les diferents opcions de configuració analitzades, es decideix quins valors s'hi assignen per tal que el funcionament de l'eina sigui el desitjat.

3.1.6 Comprovació de funcionament

Una part important és comprovar que la instal·lació s'ha realitzat de forma correcta i que la configuració entrada es comporta de forma adient.

Per fer-ho, es comprova una per una les diferents funcionalitats que representa que ha de tenir l'eina, d'aquesta manera s'eviten problemes en fases posteriors del projecte.

3.1.7 Documentació

L'última part és documentar tots els passos que s'han seguit per aconseguir un correcte funcionament de l'eina.

La forma que s'ha utilitzat per fer-ho és separar tots els passos en blocs diferents, documentant cada bloc en un apartat del projecte.

3.2 METODOLOGIA DE DESENVOLUPAMENT

En aquesta part del projecte s'ha seguit una metodologia diferent a l'anterior. Primer s'ha seguit unes pautes globals per el projecte en general i després s'ha seguit unes pautes específiques per a cadascuna de les tasques que s'han realitzat.

3.2.1 Estudi del projecte

Abans de començar a analitzar el projecte en si, el primer que hem de fer és comprovar quins llenguatges utilitza i, en cas de ser necessari, aprendre'ls i practicar la seva programació.

Tot seguit, el que s'ha de fer, ja que es tracta d'un projecte començat, és estudiar-ne la seva estructura i les diferents funcionalitats de les que disposa.

Per fer-ho, primer es comprova el seu funcionament per entendre les diferents funcionalitats que té. Un cop comprovat i analitzat el seu funcionament, es mira la seva estructura de fitxers i es comprova si té relació amb les funcionalitats vistes.

Després es mira si es troben diferents blocs dins el projecte en què es pugui separar l'eina que facilitin el seu desenvolupament, i en cas de ser així, s'estudia l'estructura de cada bloc.

3.2.2 Identificació de les tasques a realitzar

Amb el projecte ja mínimament interioritzat, comencem a pensar en quines poden ser les tasques a realitzar per millorar-ne la funcionalitat.

En aquest punt, s'està en contacte de forma constant amb els creadors i desenvolupadors de l'eina, per tal de poder ajudar en el desenvolupament d'aquesta i que les millores aplicades siguin considerades per afegir-se en pròximes versions del projecte.

Un cop es decideix quina tasca efectuar, i ho convenim amb els creadors del projecte, comencem a pensar com s'implementarà.

3.2.3 Programació de la tasca

Per a cada tasca realitzada s'han seguit un conjunt de passos a l'hora de programar-la. Primer de tot, és necessari pensar quines seran les parts del projecte afectades ja que és per on s'haurà de començar a implementar.

El primer que fem és dissenyar les estructures i funcionalitats necessàries per portar a terme la tasca en qüestió. Per fer-ho, s'analitza la secció del projecte on s'haurà de fer la implementació i es comença a decidir quines de les estructures existents ens són útils, i quines s'hauran de crear.

Un cop decidit això, comencem a programar les funcionalitats anteriors, vigilants de no tocar cap part de codi que no sigui estrictament necessari, ja que podria portar a perdre funcionalitats ja existents.

El mètode seguit per a realitzar les tasques ha estat dividir aquestes en petits punts que es poguessin anar realitzant de forma independent, d'aquesta manera es té un control molt més gran de qualsevol error de programació.

3.2.4 Comprovació de la tasca

Un cop acabada la realització de la tasca, hem procedit a realitzar una comprovació detallada de totes les funcionalitats que se suposava que havia de tenir.

Aquestes comprovacions les fem per no trobar-nos que quan hem realitzat totes les tasques indicades, hi ha alguna funcionalitat concreta que no funciona com s'esperava i fa que el conjunt de la feina feta tampoc ho faci.

3.2.5 Documentació

Un cop s'ha realitzat la implementació i la comprovació d'una tasca, es procedeix a realitzar la documentació d'aquesta.

El fet de realitzar la documentació un cop s'acaba cada tasca es deu a que és molt més fàcil explicar com s'ha realitzat un cop es té fresc en memòria.

4 PLANIFICACIÓ

Amb una metodologia de treball fixada com tenim en aquest projecte, la planificació s'ajusta a aquesta.

Com que era un projecte que començava de zero, sense tenir experiència prèvia en els camps que es volien estudiar i implementar, hi ha una gran part de la planificació que s'ha reservat per als estudis de les eines que es voldran utilitzar.

De la mateixa manera, en les fases inicials del projecte, com que era una època en què es tenia menys temps de dedicació degut a la feina i a les classes de la universitat, les tasques s'han programat amb un marge més ampli de temps.

En aquest projecte tindrem una planificació inicial, que es va fer al començament d'aquest, i una planificació final. Això és degut al replantejament que s'ha fet del projecte, que ha afectat de manera clara a molta part de planificació feta.

La planificació inicial que es va fer del projecte assignava la primera part d'aquest a l'estudi i implementació de les eines necessàries, i a la segona part al desenvolupament de la interfície web i al sistema de notifikacions. Així tenim que els dos primers objectius inicials del projecte es farien durant els primers mesos, i els dos últims durant els mesos finals, tenint l'estiu per a dedicar-hi més hores.

La fase de documentació es realitzarà durant tot l'espai de temps que dura el projecte, d'aquesta manera només es necessitarà un temps molt menor per a fer les revisions i correccions finals del projecte.

Després de fer el replantejament del projecte, la planificació inicial va quedar obsoleta, el que ens va suposar refer-la des del punt en què ens trobàvem.

Es mostrarà el temps real dedicat per a tots els apartats anteriors al replantejament del projecte, ja que en el moment de refer la planificació ja es sabia el temps dedicat a cadascun d'ells.

Degut al temps que quedava per a la data d'entrega del projecte, moltes de les tasques s'han realitzat en el mateix espai de temps i optimitzant-lo molt per a poder acabar-les a temps.

S'ensenyarà una mostra de la planificació en els dos apartats, en cas de voler-les veure de forma més detallada es trobaran en l'annex.

4.1 PLANIFICACIÓ INICIAL

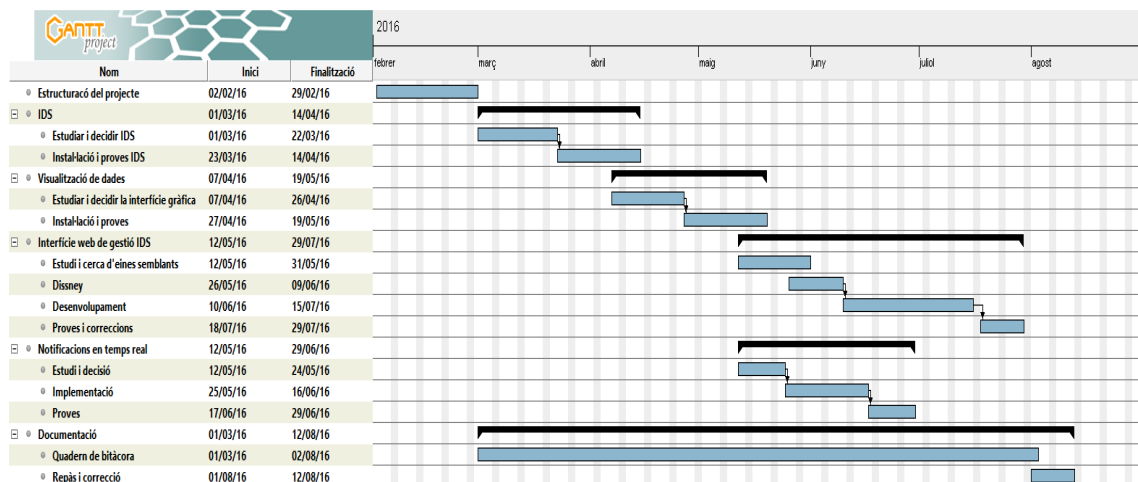


Figura 1- Planificació Inicial

4.2 PLANIFICACIÓ FINAL

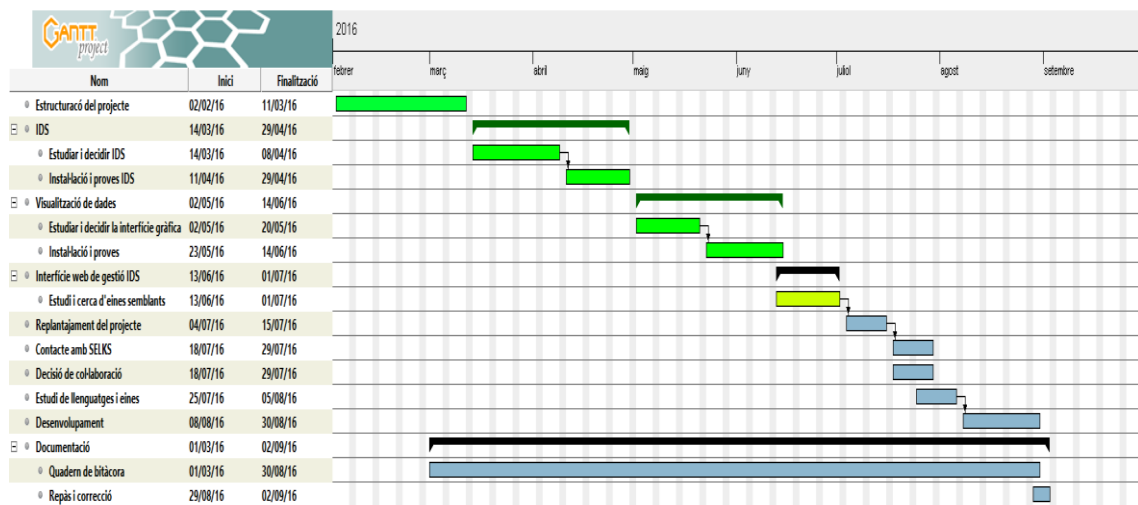


Figura 2- Planificació Final

Podem observar que les tasques ja realitzades es troben indicades de color verd i les que es planifica realitzar es troben indicades amb el color blau neutre.

5 MARC DE TREBALL I CONCEPTES PREVIS

5.1 CONCEPTES PREVIS

Es descriuran un seguit de conceptes que s'utilitzaran durant la documentació del projecte.

5.1.1 IDS

IDS significa *Intrusion Detection System*, Sistema de Detecció d'Intrusos. Es tracta d'un sistema de gestió de seguretat per a ordinadors i xarxes. Un sistema de detecció d'intrusos recull i analitza informació de diferents àrees en un ordinador o en una xarxa per identificar possibles forats de seguretat, els quals inclouen tant intrusions (atacs des de fora l'organització) com un mal ús dels recursos (atacs des de dins la pròpia organització).

Les funcions més rellevants d'un sistema d'aquestes característiques són:

- Monitoratge i anàlisi de les activitats dels usuaris i del sistema
- Anàlisi de les configuracions i vulnerabilitats del sistema
- Avaluació del sistema i de la integritat dels fitxers
- Habilitat de reconèixer patrons típics d'atacs
- Anàlisi de patrons amb una activitat anormal
- Seguiment de violacions de polítiques d'usuari

Existeixen dos tipus diferents d'aquests sistemes, els basats en la xarxa i els basats en l'amfitrió.

Basat en la Xarxa

Aquests tipus de sistemes intenten identificar comportaments no autoritzats, il·lícits i anòmals basant-se únicament en el trànsit de la xarxa. Utilitzen tant un embut de xarxa, un *port span* com un *hub* per capturar paquets que travessin una xarxa designada. Utilitzant les dades capturades, processa i marca qualsevol trànsit sospitós. Al contrari que un sistema de prevenció d'intrusos, un IDS no bloqueja de forma activa el trànsit de xarxa, el seu rol és passiu; reunir, identificar, registrar i alertar.

Basat en l'amfitrió (HIDS)

Normalment referits com a HIDS, *Host based Intrusion Detection System* (IDS basat en l'Amfitrió) intenten identificar comportaments no autoritzats, il·lícits i anòmals en un dispositiu concret. Els HIDS normalment requereixen un agent instal·lat a cada sistema, monitoritzant i alertant de les activitats del SO i les aplicacions. L'agent instal·lat utilitza una combinació de signatures, regles i heurístiques per identificar activitat no autoritzada.

5.1.2 Capes OSI

El model OSI, *Open System Interconnection* o Sistema d'Interconnexió Obert, té la tasca de separar les interconnexions en el que s'estructura com una pila vertical que consisteix de 7 capes.

Capa 1 – Física

Aquesta capa té la funció de transmetre el flux de bits, ja siguin impulsos elèctrics com senyals de llum o ràdio, a través de la xarxa a nivell elèctric i mecànic.

Capa 2 – Enllaç de Dades

A la Capa 2, els paquets són codificats i descodificats a bits. Subministra el coneixement i gestió del protocol de transmissió i gestiona errors a la capa física, el control del flux i la sincronització de trames.

Capa 3 – Xarxa

La capa de Xarxa proveeix de tecnologies de commutació i enrutament, creant camins lògics, coneguts com circuits virtuals, transmetent dades de node a node.

Capa 4 – Transport

Proporciona una transmissió transparent de dades entre extrems dels nodes, i és responsable de la recuperació d'errors punt a punt i del control del flux. També assegura que la transmissió de dades es realitza de forma completa.

Capa 5 – Sessió

Aquesta capa estableix, gestiona i acaba connexions entre aplicacions.

Capa 6 – Presentació

Aquesta capa proporciona independència en les diferències de representació de dades traduint del format d'aplicació al de xarxa i viceversa. Treballa per transformar les dades en un format que la capa d'aplicació accepti.

Capa 7 – Aplicació

Suporta aplicacions i processos finals d'usuari. S'identifiquen els punts de la comunicació, la qualitat de servei, l'autenticació i la privacitat de l'usuari, juntament amb altres requisits en les dades.

5.1.3 Payload

És la part de les dades transmeses que inclou explícitament el missatge. No inclou informació com ara les capçaleres o les metadades.

5.1.4 Buffer

Es tracta d'una regió de l'emmagatzematge físic de memòria utilitzat per a guardar dades de forma temporal mentre es mouen d'una localització a una altra. Un exemple seria quan dos processos es transmeten dades dins un mateix ordinador.

5.1.5 URI

Uniform Resource Identifier és una cadena de caràcters que s'utilitza per identificar un recurs de la xarxa. Un URI normalment descriu:

- El mecanisme utilitzat per accedir a un recurs
- L'ordinador específic on es troba allotjat el recurs
- El nom específic del recurs en l'ordinador

5.1.6 Màquina Virtual

Una màquina virtual és un Sistema Operatiu o un entorn d'aplicació que es troba instal·lat en un software, el qual imita el Hardware dedicat. L'usuari final té la mateixa experiència en una màquina virtual que el que tindria utilitzant maquinari dedicat.

5.1.7 IDE

Integrated Development Environment, o Entorn de Desenvolupament Integrat, és un tipus d'aplicació que proporciona unes característiques completes per als programadors per al desenvolupament de software.

Un IDE normalment consta de com a mínim un editor de codi font, eines de desenvolupament automàtiques i un depurador, o *debugger*.

5.1.8 Framework

Es tracta d'un entorn universal i reutilitzable que proporciona funcionalitats específiques com a part d'una plataforma de software major per tal de facilitar el desenvolupament d'aplicacions de software. Algunes característiques que diferencien un *framework* d'una llibreria són:

- El control del flux del programa no és dictat per qui fa la crida sinó pel *framework*.
- El *framework* té un comportament per defecte.
- Es pot ampliar per part de l'usuari sobreescrivint selectivament part del codi d'usuari per aconseguir funcionalitats específiques.
- El codi del *framework*, per norma general, no ha de ser modificat, tot i acceptar complements implementats per part dels usuaris.

5.1.9 Commit

En el camp de la informàtica, un *commit* és el fet de fer permanents un conjunt de canvis en les dades tractades.

En els projectes de desenvolupament distribuït de software, es sol realitzar per a compartir el codi quan s'ha acabat una tasca.

5.1.10 Repositori

Podem entendre com a repositori de dues maneres diferents.

- Lloc d'emmagatzematge des del qual s'aconsegueixen paquets de software per a instal·lar-los en un ordinador.

- En un sistema de control de versions, un repositori és una estructura de dades que guarda les metadades per a un conjunt de fitxers o l'estructura d'un directori. Algunes de les metadades que es guarden són:
 - Un històric dels canvis en el repositori.
 - El conjunt de *commits* realitzats.
 - Un conjunt de referències als *commits*, anomenats *heads*.

5.1.11 Regles

Les regles són el nucli del funcionament dels sistemes IDS. Gràcies a les regles, o signatures, aquests sistemes són capaços de analitzar i comprovar els paquets que trobem a la xarxa.

Les regles estan formades per tres parts, l'acció, la capçalera i les opcions.

5.2 MARC DE TREBALL IMPLEMENTACIÓ IDS

5.2.1 Eines utilitzades

VMWare vSphere Client

És una aplicació que ens proporciona la capacitat de connectar-nos a un servidor VMWare de forma remota sense la necessitat d'accedir directament al servidor físicament.

Amb aquesta aplicació es poden crear, modificar i eliminar màquines virtuals. Proporciona molta comoditat per als administradors de sistemes ja que es tenen totes les màquines virtuals centralitzades des d'un sol punt.

També permet veure les diferents característiques de cada màquina virtual així com canviar-les en cas de necessitat, el que ens dóna molta versatilitat per adaptar cada màquina virtual a les necessitats específiques de cada moment.

PuTTY

PuTTY és un emulador de consola, una consola per ports sèrie i una aplicació de transferències de fitxers per xarxa gratuït i de codi lliure. Suporta un conjunt de protocols de xarxa, incloent SSH i Telnet. També permet connectar-se a ports sèrie per connexions locals.

Nano

Nano és un editor de text per a sistemes basats en Unix el qual utilitza la interfície de la línia de comandes.

Respecte als altres editors del mateix estil, proporciona una major senzillesa a l'hora de fer-lo servir i incorpora característiques noves.

Consola de Cisco - CLI

La interfície de línia de comandes de Cisco IOS (CLI) és la interfície d'usuari principal utilitzada per configurar, monitoritzar i mantenir els dispositius Cisco. Aquesta interfície d'usuari permet executar directament comandes Cisco IOS, ja sigui des de la consola d'un dispositiu com utilitzant un mètode d'accés remot.

Escriptori remot de Windows

El client de servei de terminal de Windows, altrament conegut com RDC o *Remote Desktop Connection*, és l'aplicació per als serveis d'escriptoris remots. Permet als usuaris connectar-se de forma remota a un ordinador que tingui funcionant aquest servei. Mostra la interfície d'escriptori o la interfície d'usuari del sistema remot com si fos accedit de forma local.

5.3 MARC DE TREBALL DESENVOLUPAMENT

5.3.1 Eines utilitzades

Navegador web

Un navegador web és l'eina amb la qual s'accedeix als recursos del World Wide Web, més conegut com WWW. Aquests recursos poden ser una pàgina web, una imatge, un vídeo o qualsevol altre tipus de contingut. Tot i que aquesta és la seva major utilitat, per aquest projecte s'utilitzarà el navegador web per a mostrar el contingut d'un servidor privat de forma remota i també el contingut local.

El navegador que es farà servir és el Mozilla Firefox, tot i que també es faran comprovacions amb el Google Chrome, ja que són els dos navegadors més estesos actualment i que tenen unes característiques més avançades.

IntelliJ IDEA

IntelliJ és un IDE basat en Java per al desenvolupament de software.

Aquesta eina té una interfície d'usuari molt intuïtiva, la qual et detecta el llenguatge i et marca les paraules claus d'aquest per a que sigui molt més còmode i senzill programar.

També incorpora un sistema de control de versions. Aquesta característica et permet fer tot allò que es faria amb una eina especialitzada de forma completament automàtica amb un simple clic. A més a més, es troba completament integrat a l'aplicació, el que suposa poder fer totes les tasques necessàries des d'un mateix programa.

També incorpora un conjunt de connectors, o *plug-ins*, per a facilitar el desenvolupament de software de diferents llenguatges així com la utilització de *Frameworks* per a aquest desenvolupament.

Finalment, entre moltes altres característiques, cal destacar la possibilitat de depurar, o *debug*, un programa. Permet executar el software de forma seqüencial i línia a línia per tal de trobar un possible error.

Control de versions: Git i GitHub

Git

Git és un sistema de control de versions per al desenvolupament de software.

La seva essència és facilitar la feina de treballar en un mateix projecte en equip. Es basa en un sistema de flux de treball distribuït, amb la possibilitat de crear branques pròpies on es facin els canvis i després combinar-les amb la branca principal, d'aquesta manera només es modifica el codi de forma local.

Cada versió que es publica es guarda de forma independent, tot i que en el cas que hi hagi arxius que no s'hagin modificat, simplement es guarda un enllaç al fitxer de la versió anterior en comptes de el fitxer sencer, el que permet una major velocitat de comunicació.

Aporta una gran integritat en les dades ja que cada repositori local és una còpia completa del central.

GitHub

GitHub és un servei d'allotjament de repositoris basats en Git.

Ofereix totes les característiques de Git però amb una interfície gràfica per a gestionar-ho, així com un historial de totes les accions fetes sobre un repositori. A més a més, permet la interacció dels usuaris per a fer comentaris del codi, exposar problemes trobats en un projecte o demanar ajuda sobre qüestions relacionades amb aquests.

Una de les característiques més importants de GitHub és la possibilitat de crear una branca d'un repositori públic al teu propi repositori.

5.3.2 Llenguatges

Python

Python és un llenguatge de programació orientat a objectes, interactiu i interpretat. Incorpora mòduls, excepcions, tipus de dades dinàmiques a molt alt nivell i classes. Combina una important potència amb una sintaxis molt clara, el que facilita llegir i entendre el programa.

Ve amb una extensa llibreria estàndard que suporta una gran quantitat de tasques de programació com ara connectar-se a servidors, llegir i modificar fitxers, etc. A més a més, és fàcilment extensible amb la incorporació de mòduls implementats en llenguatges compilats com ara C o C++.

És també útil com a un llenguatge per a aplicacions que necessitin una interfície programable.

Finalment, Python és portable; funciona amb la majoria de sistemes operatius com Unix, Mac, Windows i Linux.

Django Framework

Django és un framework Web basat en Python d'alt nivell que proporciona un desenvolupament ràpid i un disseny clar. S'encarrega de la majoria de molèsties del desenvolupament Web, per tal que el desenvolupador es centri en escriure l'aplicació sense la necessitat de reinventar la roda.

És un framework gratuït i de codi lliure.

Va ser dissenyat per ajudar a portar a terme les aplicacions de la forma més ràpida possible. Es pren molt seriosament l'apartat de la seguretat, ajudant a evitar errors comuns en aquest àmbit.

També cal destacar la capacitat de ser molt escalable en la gestió de les peticions cap al servidor web.

JavaScript

JavaScript, o JS, és un llenguatge lleuger, interpretat i orientat a objectes amb funcions que es comporten com aquests, i, tot i ser conegut com a llenguatge de *scripting* per a pàgines Web, també s'utilitza en molts entorns que no inclouen un navegador. És un llenguatge de *scripting* dinàmic que suporta estils de programació orientats a objectes, imperatius i funcionals.

Conté un conjunt de llibreries estàndard d'objectes i un conjunt bàsic d'elements com ara operadors, estructures de control i declaracions. El nucli de JavaScript es pot ampliar completant-lo amb objectes addicionals com pot ser el *Client-side JavaScript* o el *Server-side JavaScript*.

SQL

Structured Query Language (SQL) és un llenguatge de programació declaratiu que serveix per a accedir a bases de dades relacionals i permet especificar diferents operacions en aquestes.

Algunes de les operacions que permet efectuar són crear, modificar i esborrar estructures de dades, llegir, inserir, modificar i esborrar registres o finalitzar o rebutjar transaccions.

Una de les característiques interessants que ofereix és la integració amb altres llenguatges de programació. Permet utilitzar la sintaxis d'aquests llenguatges per a fer crides SQL a bases de dades.

jQuery

jQuery és una llibreria de JavaScript dissenyada per simplificar les accions realitzades al costat del client en la programació web. S'utilitza per aconseguir utilitzar de forma molt més senzilla accions com la manipulació de documents HTML, la gestió d'esdeveniments o les crides AJAX.

6 REQUISITS DEL SISTEMA

6.1 REQUISITS IMPLEMENTACIÓ DEL SISTEMA

Per a la primera part del projecte es necessitava una màquina amb una capacitat computacional alta ja que la feina que havia de realitzar era bàsicament capturar i analitzar paquets capturats en una xarxa.

Per aquesta feina s'ha utilitzat un servidor amb les característiques següents:

- **Marca:** Dell Inc.
- **Model:** PowerEdge R310
- **Nuclis:** 4 CPUs x 2.393 GHz
- **Processador:** Intel Xenon CPU X3430 @ 2.40 GHz
- **Memòria:** 12 GB
- **Capacitat de disc:** 224.75 GB

El que es recomana utilitzar en casos d'eines que serveixen per capturar i analitzar paquets de xarxa és un màquina amb un mínim de 8 GB de memòria, un processador potent i una capacitat d'emmagatzematge important.

En el nostre cas, com que les proves que es realitzaran no seran en un entorn de molt de trànsit de dades, no necessitem una gran capacitat de processament i una gran quantitat d'emmagatzematge per lo que amb les característiques que ens ofereix aquest servidor és suficient.

6.2 REQUISITS DESENVOLUPAMENT

Per a la part de desenvolupament de software hem utilitzat una màquina més convencional. Les característiques de l'ordinador utilitzat són:

- **Marca:** Acer
- **Model:** Aspire V5-571PG
- **Nuclis:** 2 x 2501 MHz
- **Processador:** Intel Core i7-3537U CPU @ 2.00 GHz – 4 Processadors Lògics
- **Sistema:** 64 bits
- **Memòria:** 8 GB
- **Capacitat de disc:** 750 GB
- **Sistema Operatiu:** Ubuntu 16.04 LTS

Els requisits mínims per a poder realitzar les tasques d'aquest apartat del projecte segurament fossin menors de les mostrades, però aquesta és la màquina de la qual disposava.

Indiquem el Sistema Operatiu utilitzat ja que era necessari la utilització d'aquest per les característiques de les tasques a realitzar.

7 ESTUDIS I DECISIONS

7.1 IDS

La idea que es tenia per fer aquest projecte era estudiar, implementar, configurar i analitzar un sistema de detecció i prevenció d'intrusos. Amb aquesta idea al cap, es va començar a buscar les diferents opcions que existien.

En la cerca d'aquestes opcions ens hem trobat diferents tipus de IDPS, des de programari lliure a privatiu, gratuït a comercial, etc.

Com que aquest és un projecte universitari, sense recursos per fer inversions en material o programari es buscarà un software que sigui gratuït. A més a més, és interessant que sigui de codi lliure ja que d'aquesta manera es podrà modificar, en cas de ser necessari, algun dels seus components per adaptar-lo a les nostres necessitats, ja que amb programari privatiu no es sol tenir aquesta opció.

Un altre punt a tenir en compte és la possibilitat d'aquest software de no ser només un IDS sinó que, ja sigui originalment o gràcies a algun complement, es pugui utilitzar com a IPS ja que ens interessa no només detectar anomalies en la xarxa sinó també que s'actui de forma automàtica per tal que el nivell de risc sigui el mínim possible.

A l'hora d'analitzar quin IDS/IDPS ens interessa més es tindrà en compte el *roadmap*, les previsions de futur del programari, ja que ens indicarà si a curt termini es disposarà de característiques interessants per als nostres interessos que faria que ens decantéssim per un programari o un altre en base si es tenen o no aquestes característiques.

En cas de dubtar quin IDS escollir, un altre punt important serà la comunitat que hi hagi darrere de cada sistema ja que és molt important tenir un suport d'aquesta per tal d'entendre com funciona el sistema i en cas de tenir algun problema, trobar-ne una solució de forma ràpida i eficaç.

7.1.1 SNORT

Va ser la primera eina que va sorgir que s'acostava a la idea de IDS. Amb el pas dels anys s'ha convertit en l'estàndard de-facto per IDS i, fins i tot, pels IPS. És un sistema de detecció i prevenció d'intrusos de codi lliure capaç de fer un anàlisi del trànsit en temps real i de registrar paquets.

És un sistema que ha estat en desenvolupament durant molt de temps, i això sol ser un indicatiu de que ha sabut renovar-se i adaptar-se a les noves necessitats dels usuaris. Es poden desenvolupar característiques noves o millorar-ne les existents per tal que la funcionalitat de l'eina segueixi essent l'esperada.

Snort és el sistema de detecció d'intrusos més estès, però també ha millorat les seves capacitats i han aconseguit que funcioni també com a sistema de detecció d'intrusos, el que implica que ja no només és un sistema passiu de detecció i alerta sinó que té la capacitat d'actuar en els casos en què abans només alertava.

Per detectar les diferents alertes s'utilitzen regles. Snort suporta diferents formats de regles sobre les que actua; cada format prové d'un origen diferent i solen acabar sent fonts complementàries d'informació. El format principal que utilitza són les regles VRT, pròpies de Snort, les quals les desenvolupen un conjunt d'experts en seguretat d'arreu del món. [2] També accepta el format de les regles SO (Shared Object), les quals es troben escrites en el llenguatge C i permeten detectar un conjunt molt més gran de condicions de les que pot arribar a detectar les regles anteriors. [3] Finalment accepta les regles Emerging Threats, que són desenvolupades per experts com les VRT, tot i que no s'ajusten a un sistema en concret sinó que volen ser més universals.

Aquesta eina funciona en un mode *single-thread*, el que significa que només utilitza un fil d'execució de la màquina on es trobi instal·lat.

Per registrar la sortida d'informació, Snort ofereix tres formats diferents. El primer en el que pot registrar els esdeveniments que ocorren és en el format de text pla, és a dir, text llegible per les persones. També pot registrar la informació en el format de base de dades, el que permet accedir a la informació en forma de peticions, més útil en cas de necessitat d'obtenir la informació registrada segons paràmetres concrets. Finalment també permet registrar en format Unified2, propi de Snort.

Altres característiques que es poden destacar d'aquesta eina és la compatibilitat amb el protocol IPv6, el qual és el futur de les xarxes de telecomunicacions i que en un temps relativament baix ha d'implementar-se globalment. No incorpora cap accelerador de captura de paquets propi, sinó que funciona amb un únic que ve ja definit. Una característica interessant és el fet de poder executar anàlisis fora de línia.

Si ens fixem en la usabilitat, Snort no incorpora cap interfície gràfica pròpia. Tot i que existeixen diferents interfícies compatibles amb l'eina, cap d'elles és dels desenvolupadors d'aquesta.

Punts a favor

Com hem comentat anteriorment, és el sistema d'aquest tipus més estès globalment, i és per les virtuts que té.

És un sistema que pot ser instal·lat en qualsevol entorn de xarxa, el que significa que per poder-lo utilitzar no s'ha de pensar en cap modificació física ni lògica de la xarxa a la qual es vol implementar. A més a més, també és compatible amb varis dels Sistemes Operatius més estesos actualment, com poden ser Windows, la majoria de distribucions de Linux o Mac OS X. Per tant, si ens fixem en aquests dos fet podem observar el perquè és tan popular aquest sistema.

Snort és capaç de capturar i analitzar en temps real del trànsit de xarxa a la que està assignat. Això permet obtenir informació en temps real de la xarxa i tenir un temps d'actuació en cas de necessitat molt menor.

També cal destacar el fet que els sensors que utilitza Snort són modulars, el que significa que poden monitoritzar múltiples dispositius i màquines des d'una única localització física i lògica.

A més a més, un dels punts més interessants és que existeixen una gran quantitat d'eines i utilitats que funcionen conjuntament amb Snort que ofereixen una ampliació important de les característiques i funcionalitats bàsiques de les que disposa l'eina.

Roadmap

El Roadmap és el camí a seguir per als desenvolupadors de l'eina per seguir actualitzant i millorant les característiques del sistema. Algunes de les més importants són: [4]

- OpenAppId: Auto-detecció de serveis
- Descompressió Flash/PDF
- Suport SMTP/POP/IMAP PAF
- Tipus de fitxers ID
- Extracció de fitxers
- Suport a noms de fitxers en format Unicode

Finalment, cal destacar el fet que aquesta eina compti amb una comunitat formada per milions de persones. Això és molt important ja que fa que el producte es trobi en constant evolució i en cas de tenir qualsevol problema hi hagi un suport comunitari que intentarà solucionar-lo el més ràpid possible.

7.1.2 SURICATA

Suricata [5] és un Sistema de Detecció i Prevenció d'Intrusos d'alt rendiment i un motor de Monitoratge de Seguretat de la Xarxa. És de codi lliure i propietat d'una comunitat controlada per una fundació sense ànim de lucre, la Open Information Security Foundation (OISF). És desenvolupat per l'OISF i els membres que hi donen suport

Suricata és un Sistema de Detecció i Prevenció d'Intrusos basat en regles que utilitza conjunts de regles desenvolupades externament per monitoritzar el trànsit de xarxa i proveir alertes a l'administrador del sistema quan un esdeveniment sospitós ocorre.

Aquest sistema és propietat d'una comunitat controlada per una fundació sense ànim de lucre, la Open Information Security Foundation (OISF) [6]. És una eina de codi lliure i es troba desenvolupada per aquesta fundació i els diferents membres que hi donen suport. [7]

Tot i ser bastant recent, la primera versió estable va sortir el juliol de 2010, ha anat creixent i cada cop té una comunitat més gran. Té una quantitat de característiques que el diferencien dels seus competidors, alguna de les quals veurem durant la seva descripció.

Les seves característiques principals són:

- **Altament escalable**
És un sistema multi-threaded, això significa que fa ús de diferents fils d'execució. Aquesta característica ens indica que Suricata és capaç de fer servir tota la capacitat computacional del dispositiu o màquina al qual ha estat instal·lat, ja que fa servir tots els nuclis (*cores*) d'aquesta, optimitzant-ne el rendiment.
- **Identificació de protocols**
El sistema reconeix de forma automàtica els protocols més comuns. Aquest fet permet que les diferents regles utilitzades en l'anàlisi de la informació puguin ser escrites segons el protocol i no segons el port estàndard que aquest tingui.

- **Identificació de fitxers**

Suricata és capaç de detectar els tipus dels fitxers que passen per la xarxa en temps d'execució, per tant, mentre està fent la captura de paquets, és capaç d'analitzar el tipus de fitxer que passa en cada moment.

- **MD5 Checksum**

El càlcul del Checksum MD5 es fa al moment de la captura. Això serveix per poder mantenir-ne o bloquejar-ne l'accés per mitjà d'una llista de hash MD5.

- **Extracció de fitxers**

Un cop identificat un tipus de fitxer, el sistema té la capacitat d'extreure'n el contingut i guardar-lo per poder-ne fer un anàlisi posterior.

Punts a favor

Suricata és compatible amb la majoria de Sistemes Operatius que es troben actualment en el mercat. Els principals sistemes als quals és funcional l'eina són Windows, Mac OS X, la majoria de distribucions de Linux i FreeBSD i OpenBSD. Cal destacar que la majoria de mercat de Suricata es troba en les distribucions de Linux ja que solen ser els Sistemes Operatius més instal·lats en servidors i màquines de monitoratge de xarxa.

El sistema incorpora diferents motors com el de detecció d'intrusos, el de prevenció d'intrusos i el monitoratge de seguretat de la xarxa. A més a més, incorpora un motor de detecció i captura de paquets molt potent el qual li dóna un avantatge en el rendiment de l'eina.

També hi trobem un suport per a la descodificació de paquets segons el protocol que estava utilitzant. Així com el suport a la descodificació de paquets a nivell de la capa d'Aplicació del model OSI.

A més a més, té la capacitat de fer un filtratge de les alertes i d'analitzar el contingut en un mode fora de línia i disposa d'un sistema de reputació d'adreces IP per utilitzar durant l'anàlisi del contingut.

Suricata disposa de diferents formats de registre de la informació que captura i analitza. El format més simple de registre és el **pcap** ja que guarda la informació tal i com la rep, com si es tractés de text pla.

També és compatible amb el format **Unified2** desenvolupat pels creadors de Snort, el que implica que totes les eines que utilitzin el format de Snort per funcionar seran compatibles amb Suricata.

El format **fast.log** es tracta d'una taula on es guarden totes les alertes generades pel sistema de forma precisa i concreta.

Després hi ha el format **Eve.json** el qual utilitza el format JSON per registrar els esdeveniments i la informació, donant molta facilitat perquè es puguin utilitzar una gran quantitat d'eines de visualització de dades.

Finalment també dona la possibilitat de registrar tota la informació a un Servidor de Registres, o Syslog, extern per tenir còpies de seguretat de les dades.

Una de les característiques més importants és el fet de tenir la capacitat **multi-threading**, o poder utilitzar diferents fils d'execució de la màquina.

Roadmap [8]

La previsió de millores i correccions de Suricata es troba molt ben estructurat on és molt fàcil distingir els pròxims desenvolupaments de l'eina, les correccions de problemes que s'hagin trobat i les millores que es vulguin implementar. Aquest sistema permet que la comunitat ajudi en el desenvolupament de l'eina i facilita la incorporació de noves característiques que d'una altra forma no es podrien incloure properament.

Es troba separat en diferents seccions on a cadascuna d'aquestes s'indica els procediments que s'hi estan realitzant, l'estat d'aquests i la quantitat que ja s'han acabat. Entre les diferents seccions hi podem trobar les pròximes versions, amb les característiques noves en les que s'està treballant i la data de publicació, una secció relacionada amb la documentació que indica els aspectes en els que cal crear o millorar la informació que es té d'aquests o les seccions que fan referència a l'empaquetament de l'eina i a les preguntes freqüents. Finalment cal destacar les dues seccions més importants, la de millores que es preveuen tenir per la pròxima versió principal i les característiques que es preveuen tenir a llarg termini.

En la pròxima versió principal de Suricata es preveu la inclusió d'una gran quantitat de característiques i utilitats noves.

7.1.3 SNORT++ (3.0) [9]

SNORT++ és un sistema de detecció d'intrusos desenvolupat per el mateix grup de treball que desenvolupa Snort. Tot i això, és un sistema que s'ha treballat des de la base, sense agafar com a referència l'altre i amb la idea de desenvolupar característiques diferents i noves utilitats fent que, des de l'inici, es diferenciés molt de Snort.

Aquest programari es troba en estat Alpha, el que significa que tot just es troba en un estat inicial de desenvolupament i existeixen, per ara, moltes característiques i propietats que no es troben correctament implementades i poden donar errors a l'hora de utilitzar-les.

Snort++ té la capacitat d'utilitzar múltiples fils d'execució d'una màquina, és a dir, és *multi-thread*. Aquesta característica possibilita el processament de paquets en paral·lel utilitzant tota la capacitat de computació de la màquina on es trobi el sistema optimitzant d'aquesta manera els recursos i el temps del qual es disposen.

Incorpora una taula de configuracions i atributs compartida, el que dóna una millor compactació i senzillesa al sistema. També contribueix a aquesta senzillesa que s'utilitzi una configuració simple mitjançant *scripts*.

Una característica interessant que ofereix aquesta eina és que s'està desenvolupant amb la idea de que tots els components clau que la formen estiguin connectats entre si. Aquesta relació dels diferents components aporta una millor optimització de recursos que utilitza l'eina i disminueix el risc de fallada.

Snort++ varia de forma important respecte el seu antecessor en la detecció de les regles. Aquest sistema utilitza una detecció automàtica dels serveis dels paquets que captura i aconseguen tenir una configuració sense ports. Això és un avanç important de concepte ja que controla una quantitat important de la seguretat de la que abans no es podia tenir constància.

També és capaç de tenir un *buffer* de dimensions estàtiques per a les regles, el que implica tenir un major control en la quantitat de disc que utilitza el sistema.

Com molts altres sistemes i utilitats utilitza un generador automàtic de referències per a la documentació, el que implica que els desenvolupadors hagin d'afegir la informació relacionada a cada part de l'eina que creïn o modifiquin.

Finalment trobem un seguit de característiques que es troben en Snort, però encara no han estat desenvolupades per a Snort++. Tot i això, aquestes característiques s'esperen que es trobin implementades en la finalització de la versió Alpha d'aquest sistema.

- Captura de sessions.
- Alta disponibilitat i canal dual.
- Preprocessador dcerpc2.
- Preprocessador appid.
- Preprocessador sdf.

Roadmap

En aquest cas, les característiques que hi trobem no són per a la pròxima versió, que seria la publicació de la versió Alpha definitiva, sinó que estan encarades a versions posteriors del sistema. Per tant, no són característiques que s'implementarien en un període curt de temps.

- Utilització d'un mapa de xarxa compartit.
- Suport de processament de paquets en cadena.
- Suport de descàrrega de maquinari (hardware) i integració de dades planes.
- Reescriptura de mòduls crítics com ara reassemblatge TCP i inspecció HTTP.
- Suport d'un mode Proxy.
- Simplificar la gestió de memòria.
- Suport per Windows.

7.1.4 Taula comparativa

La següent taula mostra de forma estructurada quines són les diferències entre les diferents opcions.

| | SNORT | SURICATA | SNORT++ |
|--|--------------|-----------------|----------------|
| OPEN SOURCE | Si | Si | Si |
| GRATUÏT | Si | Si | Si |
| MODE IPS | Si | Si | Si |
| COMUNITAT | Molt gran | Petita | Gran |
| REGLES VRT | Si | Si | Si |
| REGLES SO | Si | No | Si |
| REGLES EMERGINGTHREADS | Si | Si | Si |
| MÚLTIPLES FILS D'EXECUCIÓ | No | Si | Si |
| REGISTRE AMB TEXT PLA | Si | Si | Si |
| REGISTRE EN BASE DE DADES | Si | Si | Si |
| SUPORT D'IPV6 | Si | Si | Si |
| ACCELERADOR DE CAPTURA DE PAQUETS | No | Si | No |
| ANÀLISI FORA DE LÍNIA | Si | Si | Si |
| INTERFÍCIE GRÀFICA | No | No | No |
| LINUX | Si | Si | Si |
| MAC OS | Si | Si | No |
| WINDOWS | Si | Si | No |
| IDENTIFICACIÓ DE SERVEIS | No | Si | Si |
| IDENTIFICACIÓ DE FITXERS | No | Si | No |
| EXTRACCIÓ DE FITXERS | No | Si | No |

Taula 1- Comparativa entre IDS

7.1.5 Decisió

Al final hem vist tres opcions, de les quals dues són un programari desenvolupat per el mateix grup, tot i ser projectes bastant diferents.

Tant Snort com Snort++ tenen una empresa com Cisco al darrere que hi dóna suport. A més a més, és el sistema més utilitzat al món, amb la major comunitat. Aquesta comunitat és molt activa i és de molta ajuda a l'hora de desenvolupar millores en cadascun dels dos projectes, respondre dubtes i solucionar problemes.

Snort ja porta molts anys funcionant i es té previsió que en duri molts més, el que ens indica que no ens quedaríem sense suport en un futur proper. Un exemple d'això és Snort++, ja que és un projecte de llarga durada que tot just es troba en fase Alpha i s'està desenvolupant per millorar molt notablement totes les característiques essencials que manquen a Snort, el que ens demostra que no es té cap previsió de deixar aquests projectes en un futur proper o a mig termini.

En aquest sentit Suricata és molt més recent ja que la primera versió estable va sortir el juliol de 2010. Tot i això, el fet que tingui una fundació com la OISF al darrere que hi dóna suport i no sigui un projecte desenvolupat per un grup de persones anònimes dóna una seguretat que aquest projecte no es quedarà sense suport a curt termini. A més a més, podem trobar a la seva pàgina que és un projecte desenvolupat a llarg termini i que tenen intenció de suportar-lo durant molt de temps. També dóna seguretat el fet que no només té la fundació OSIF al darrere, sinó un consorci d'organitzacions que hi donen suport.

Per tant, en aquest sentit tots tres projectes tenen una projecció de futur clara i un suport a llarg termini, pel que no s'haurà de patir perquè es quedin sense desenvolupament i suport en un futur proper.

Degut a l'antiguitat i la comunitat que posseeix Snort, aquest projecte té un conjunt molt gran de regles ja desenvolupades i provades. Això fa que es tingui accés amb molta facilitat a regles i opcions ja funcionals.

Snort++ és completament compatible amb les regles de Snort ja que s'ha desenvolupat a partir d'aquest. Això fa que totes les regles que s'han creat i comprovats fins ara es puguin seguir utilitzant sense cap tipus de problema i tota la feina feta per la comunitat segueixi sent completament funcional.

En el cas de Suricata tenim que utilitza un format de regles propi. Tot i això, els creadors d'aquest projecte, sabent que Snort era l'IDS de facto van fer que les regles escrites per funcionar-hi fossin compatibles amb Suricata. Amb això van aconseguir que, des del principi, tota la feina feta per la comunitat d'Snort fos útil per al seu projecte.

També va ser una manera d'atraure usuaris de Snort a provar la seva eina ja que no tenien que reescriure totes les regles sinó que només necessitaven instal·lar l'eina i aprofitar les regles que ja tenien funcionant.

Per tant, tenim que Snort té una gran quantitat de regles pròpies funcionals que la comunitat ha anat creant i comprovant. De la mateixa manera, Snort++ utilitza aquestes regles per funcionar.

Per altra banda, tenim que Suricata pot fer servir les regles dissenyades per funcionar amb Snort i, a més a més, té un conjunt de regles que serveixen per utilitzar les característiques pròpies de Suricata.

Tots tres projectes són de codi lliure i gratuïts, el que significa que qualsevol persona pot utilitzar-los de forma independent i lliure. A més a més, s'ajuden de la comunitat per millorar les seves característiques i prestacions.

En quant a característiques, Snort és el que té unes prestacions més bàsiques. Això ho volen solucionar amb el desenvolupament de Snort++ ja que és un canvi radical en la filosofia del projecte. Tot i això, s'estan desenvolupant en paral·lel i tant Snort com Snort++ tenen el seu propi *read map* per anar avançant.

Tot i que Snort++ tingui certes característiques molt importants implementades que Snort no incorpora, com pot ser el fet d'utilitzar múltiples fils d'execució en comptes de només fer-ne servir un, encara té moltes característiques que fa falta implementar.

En aquest cas, Suricata és el que té unes característiques més interessants. Si ens fixem en les previsions de millores de Snort i Snort++ trobem que la gran majoria de noves característiques que es volen desenvolupar en versions posteriors d'aquests projectes ja estan desenvolupades i integrades a versions estables i actuals de Suricata, com podrien ser l'extracció de fitxers, la captura i seguiment de sessions, detecció de paquets en cadena, suport per diferents plataformes com Windows, entre altres.

Si ens fixem en el rendiment de cada eina trobem que Snort és l'eina amb el rendiment més baix. Mentre que Snort només funciona amb un fil d'execució (single-threaded), tant Suricata com Snort++ funcionen amb múltiples fils d'execució (multi-threaded). Això significa que Snort només pot fer servir una CPU de la màquina on estigui instal·lat mentre que tant Suricata com Snort++ en poden utilitzar totes les CPUs. Aquest punt és molt important, ja que aquestes eines analitzen molt de trànsit de xarxa i això comporta un grau de computació molt alt, el qual utilitza la CPU per funcionar. Per tant, el fet d'utilitzar múltiples CPUs és un punt a favor molt important.

Comparant les tres eines que estem estudiant per utilitzar en aquest projecte veiem que, com que el rendiment és molt important, ens interessa fer servir Snort++ o Suricata ja que ambdós utilitzarien tot el potencial de la màquina on s'instal·lessin, mentre que amb Snort desaprofitaríem la majoria d'aquesta.

Dels tres projectes, tenim que Snort i Suricata tenen versions estables disponibles mentre que Snort++ tot just es troba en fase Alpha. Això indica que, tant les versions de Snort com la de Suricata, han estat provades d'una forma intensa abans de estar disponibles, el que ens assegura que hi trobarem molts pocs errors i no tindrem problemes d'estabilitat amb l'eina, és a dir, es podrà treballar correctament amb elles. En canvi, tenim que Snort++ està en fase Alpha, això significa que és una versió en estat primari de desenvolupament que no està provada a fons i que encara es troba en una fase on no s'han solucionat tots els errors importants.

Aquest és un punt molt important ja que ens interessa tenir una eina amb la qual tinguem la seguretat que ens durarà i no ens portarà problemes a l'hora de fer-la funcionar. En aquest cas, Snort++ no compleix aquest requisit, per tant no ens interessa per al nostre projecte.

Suricata és una eina que utilitza tot el potencial de la màquina on està instal·lada i té una versió estable sense errors importants. També és interessant poder provar totes les característiques que ens ofereix ja que tant amb Snort com amb Snort++ hauríem d'esperar a versions posteriors i no sabem quan les implementarien finalment.

A part, és una eina de codi lliure i gratuïta que està en fase de desenvolupament constant amb una comunitat molt menor a la de Snort però que va augmentant amb el temps.

És interessant el fet de poder utilitzar una eina bastant recent, la qual està en fase d'expansió i de la qual se'n pot aprendre molt. A més a més, amb aquest projecte es vol intentar contribuir-hi ja que al estar desenvolupada per una fundació sense ànim de lucre és important la contribució de la comunitat per millorar-ne les característiques i trobar possibles errors i com solucionar-los.

7.2 VISUALITZADOR DE CONTINGUT

Amb un sistema que captura i analitza paquets es sol tenir una gran quantitat de dades. Si féssim servir el sistema de forma autònoma ens seria gairebé impossible poder comprovar totes les coincidències i alertes que generi l'eina.

Per poder tenir un control sobre aquestes alertes es necessita un sistema de visualització de les dades, el qual tingui un processament d'aquestes prou important per poder tractar la quantitat d'informació que es recull.

Ens interessa que aquest sistema de visualització tingui una interfície gràfica ja que és la millor manera de veure i estudiar les dades. Hauria de tractar-se d'una interfície web, ja que permet l'accés remot des de qualsevol localització en cas de necessitat, així com l'accés local per mitjà d'un navegador.

Com que el sistema de detecció d'intrusos que hem escollit és capaç de registrar les dades capturades en diferents formats de sortida, ens dóna una quantitat major de possibilitats a l'hora de buscar una eina de visualització que hi sigui compatible.

Comprovarem i analitzarem les utilitats que compleixin aquestes característiques més esteses. Gràcies a que Suricata utilitza el registre d'informació de Snort, les eines de visualització gràfica de dades s'aquest sistema també seran compatibles.

7.2.1 BASE

És una interfície gràfica dedicada a mostrar els registres obtinguts amb Snort, per tant també és compatible amb Suricata, i enviar-los en una base de dades. El seu nom prové de *Basic Analysis and Security Engine*, que significa Motor Bàsic d'Anàlisi i Seguretat.

La última versió disponible d'aquesta eina és del juny de 2013, per tant, aquest projecte porta un temps considerable estancat. Això és indicatiu que el més probable és que no es rebin millores ni actualitzacions i si ens trobem amb algun problema no hi hagi una solució.

Té una documentació molt escassa i és difícil trobar-ne informació.

Algun exemple d'execució d'aquesta eina on es pot veure la seva interfície i alguna de les característiques de les que disposa.

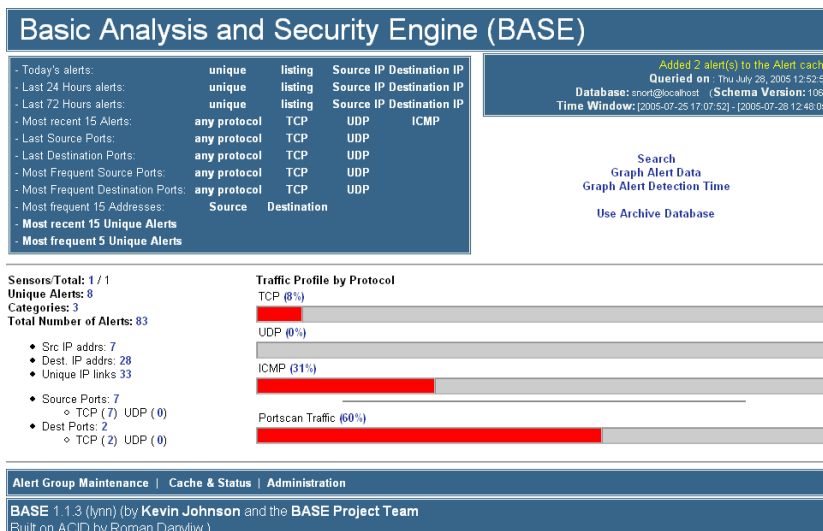


Figura 3- BASE (font de la imatge original: <http://www.oracle.com/technetwork/systems/articles/snort-base-fig1-145116.gif>)

7.2.2 Snorby

Snorby és una aplicació web basada en *Ruby on rails* per al monitoratge de seguretat en la xarxa amb compatibilitat amb els sistemes més populars actualment com poden ser Snort i Suricata.

És una eina gratuïta i de codi lliure, que proporciona una interfície agradable i, sobretot, funcional.

L'última actualització disponible d'aquesta eina és del maig de 2013, tot i que a la seva pàgina es pot veure que hi segueixen treballant mínimament ja que l'última modificació sobre el projecte va ser a inicis d'aquest any.

Disposa d'una documentació acceptable i, al ser un projecte de codi lliure i estar allotjat a GitHub, té un apartat per els diferents problemes que puguin sorgir amb l'aplicació per tal de poder-hi donar resposta.

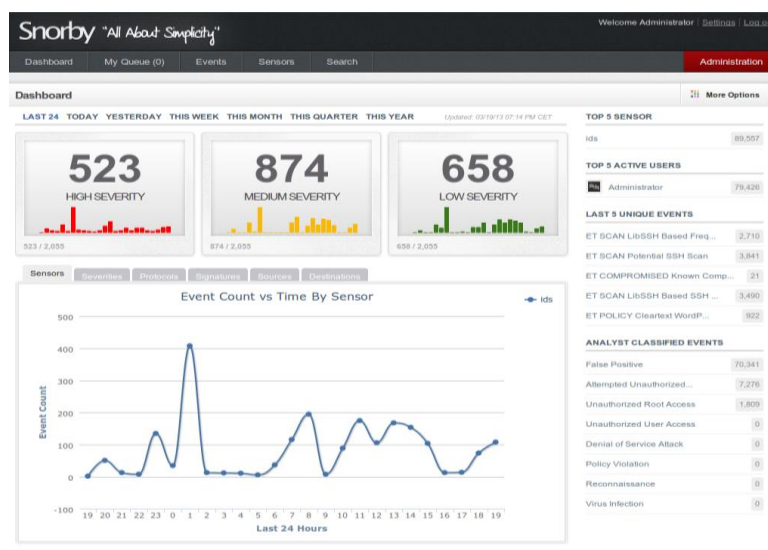


Figura 4- Snorby (font de la imatge original: <https://www.rivy.org/wp-content/uploads/2013/03/snorby-screenshot.png>)

7.2.3 Elastic Stack

Elastic Stack és un conjunt de tres eines que actuen juntes sobre els formats de sortida de Suricata per tal de crear les visualitzacions de dades.

El primer element, Elasticsearch, tracta les dades obtingudes dels registres de Suricata i les converteix al format necessari per a la seva utilització. Després Logstash tracta aquestes dades que ha convertit Elasticsearch i les envia cap a la tercera eina, Kibana, que és el motor de visualització. Kibana, amb les dades obtingudes de Logstash, crea els diagrames de visualització i les diferents gràfiques.

Aquest conjunt d'eines es troba en constant millora i actualització i es desenvolupa com a codi lliure, tot i ser propietat d'una empresa.

El llenguatge que utilitzen per al desenvolupament és el Java, tot i que es basa en Lucense per al tracte amb fitxers dels índex i fragments.

Elasticsearch

Elasticsearch és un motor de cerca distribuït RESTful i amb una alta disponibilitat. Cada índex es troba fragmentat amb un nombre configurable de fragments. Cada fragment pot tenir una o més rèpliques i les operacions de lectura i escriptura es realitzen en qualsevol d'aquestes.

Té suport per múltiples índex i es poden assignar múltiples tipus diferents a cadascun. La seva configuració és a nivell d'índex, on es poden assignar el nombre de fragments, l'emmagatzematge de cada índex...

Està orientat a documents, per tant no s'hauran de definir esquemes i estructures a l'inici, tot i que es podran configurar esquemes personalitzats.

Les operacions a nivell de documents únics són atòmiques, consistents, isolades i duradores, el que dóna una important consistència per les operacions.

La primera publicació d'Elasticsearch va ser al Febrer de 2010.

Logstash

Logstash és un motor de recopilació de dades de codi lliure amb capacitats de comunicació en temps real. Té la capacitat d'unificar dades de diferents orígens dinàmicament i normalitzar-la en les destinacions que es decideixi.

Als inicis va ser una innovació en la recopilació de registres, tot i que actualment les seves capacitats excedeixen àmpliament aquest ús. Qualsevol tipus d'esdeveniment pot ser enriquit i transformat amb una àmplia gamma de connectors d'entrada, de filtrat i de sortida conjuntament amb un conjunt de còdecs natiu que simplifiquen el procés de recaptació de dades.

Posseeix un canal de processament de dades escalable horitzontalment amb una sinèrgia molt forta amb Elasticsearch i Kibana.

És capaç de combinar, trobar i organitzar diferents entrades, filtres i sortides per tal de tenir un canal de processament de dades en harmonia.

Disposa de més de 200 connectors, i atorga la flexibilitat de crear i contribuir amb els propis.

Gestiona tot tipus de dades registrades.

- Tracta sense dificultat una multitud de registres webs com ara d'Apache.
- Captura molts altres formats de registre com Syslog, registres d'esdeveniments de Windows, registres de xarxes i Firewalls, entre altres.

Recull mètriques de diferents infraestructures i plataformes d'aplicacions per mitjà de TCP i UDP.

És capaç de transformar peticions HTTP en esdeveniments i crear-ne sota demanda fent peticions sobre variables HTTP.

- Captura de forma universal la salut, rendiment, mètriques i altres tipus de dades de les interfícies web de les aplicacions.

La primera publicació de Logstash va ser a l'agost de 2009.

Kibana

Kibana és una plataforma d'anàlisi i visualització de codi lliure dissenyada per treballar conjuntament amb Elasticsearch. S'utilitza per buscar, veure i interactuar amb les dades guardades en els índex d'Elasticsearch. De forma senzilla es pot efectuar anàlisis complexes de dades i visualitzar-ne els resultats en una varietat de gràfics, taules i mapes.

Aquesta eina fa fàcil entendre grans volums de dades, és simple i amb una interfície basada per navegadors que permet crear i compartir taulells de visualització que mostrin canvis en les peticions a Elasticsearch en temps real.

La primera versió de Kibana va ser al maig de 2013.

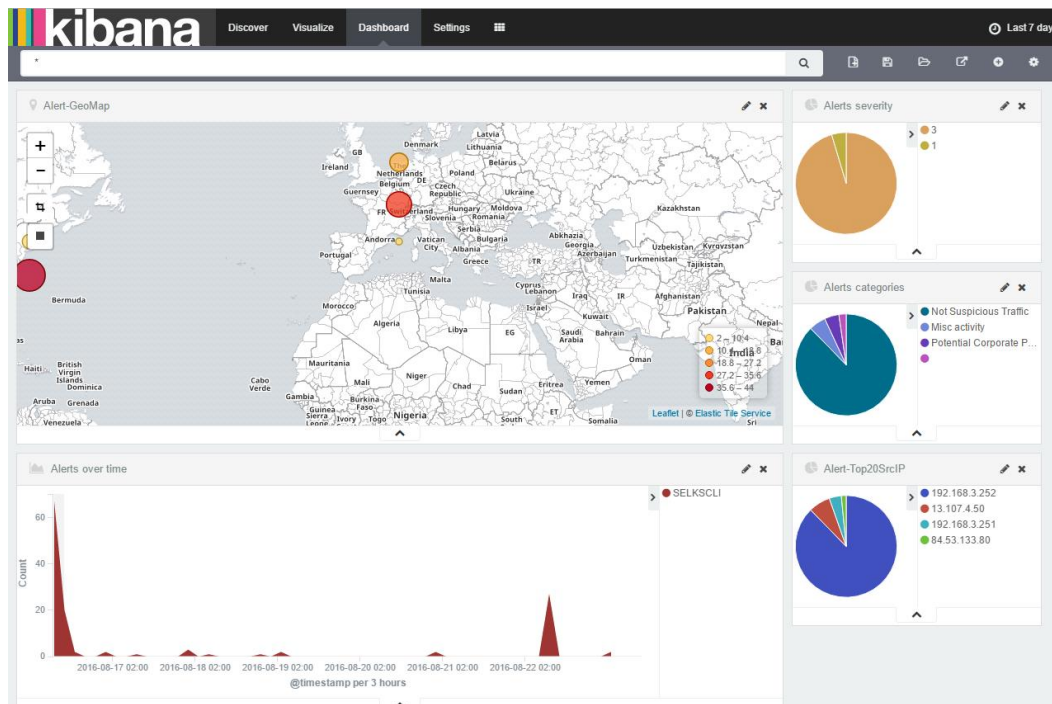


Figura 5- Kibana

7.2.4 Taula comparativa

| | BASE | SNORBY | ELASTIC STACK |
|---------------------------|------------------|------------------|----------------------|
| PRIMERA PUBLICACIÓ | Novembre de 2004 | Desembre de 2010 | Entre 2009 i 2013 |
| ÚLTIMA PUBLICACIÓ | Març de 2010 | Maig de 2013 | Actualitat |
| CODI LLIURE | Si | Si | Si |
| GRATUÏT | Si | Si | Si |
| COMUNITAT | Mitjana | Gran | Molt gran |
| DOCUMENTACIÓ | Molt poca | Bastant | Molta |
| ESTÈTICA | Antiga | Actual | Actual |
| CARACTERÍSTIQUES | Poques | Bastantes | Moltes |
| PERSONALITZACIÓ | Poca | Bastant | Molta |
| CONFIGURABLE | Si | Si | Si |

Taula 2- Comparativa d'eines de visualització

7.2.5 Decisió

Hem vist un total de tres opcions, les quals són les més utilitzades com a visualitzadors de dades per sistemes de captura.

La majoria han estat creats per a ser un complement de Snort, per tant han buscat implementar el màxim de característiques d'aquesta eina. És el cas de BASE i de Snorby, que tot i que siguin compatibles amb Suricata van ser dissenyats per funcionar principalment amb Snort.

Per altra banda tenim la Elastic Stack, composta per tres components i sense haver estat influïda per les característiques de Snorby en el seu desenvolupament. Això comporta que Kibana, l'eina purament de visualització de les dades, sigui molt més adaptable a les necessitat d'altres sistemes com Suricata.

BASE és l'eina més antiga de les tres, creada al 2004 mentre que la resta es va començar a desenvolupar molt més tard. Això es nota en la interfície que té i les característiques de les que disposa.

En contrast tenim Snorby i Kibana, la primera desenvolupada entre 2010 i 2013 i la segona des del 2013 fins l'actualitat. Tot i que no hi hagi una diferència molt notable d'anys, aquesta es pot notar en la interfície i, sobretot, en les característiques i el suport. Tant en un terme com en l'altre, Kibana segueix activa i millorant contínuament, mentre que Snorby s'ha quedat estancat en quant a desenvolupament des de fa uns anys.

Totes tres opcions es desenvolupen sota codi lliure i es distribueixen de forma gratuïta, per tant tenim l'opció d'utilitzar qualsevol d'aquestes sense problema.

El conjunt d'eines Elasticsearch, Logstash i Kibana ens proporcionen unes característiques més avançades i una interfície gràfica més agradable i actual. Es troba en constant desenvolupament i té una comunitat al darrere molt gran que dona suport a qualsevol problema que es pugui tenir. També és molt interessant la personalització que es pot aconseguir amb la creació de taulells a mida amb les característiques que es vulguin.

És important el fet que sigui la única eina que no hagi estat pensada exclusivament per Snort, el que ens proporciona una adaptabilitat molt major ja que totes les característiques de l'eina han estat pensades de forma general en comptes de ser concretes per un sistema.

Tot i ser una eina propietat d'una empresa privada, és de codi lliure i gratuïta. Això és una avantatge ja que, en cas de necessitat, tens un suport privat que saps que et resoldrà el problema que es tingui i no s'haurà de confiar en què la comunitat t'ho solucioni.

7.3 REPLANTEJAMENT DEL PROJECTE

Després de l'etapa inicial del projecte centrada en el sistema IDS i l'eina de visualització i abans de començar amb el següent objectiu, vaig decidir fer una cerca d'eines semblants a allò que volia desenvolupar per a agafar idees tant de disseny com de funcionalitats.

Aquest va ser el moment en què vaig trobar Scirius, un sistema d'interfície web per a gestionar Suricata. Investigant més a fons aquest sistema, vaig descobrir que formava part d'un projecte més ampli en el qual s'hi utilitzava Suricata com a IDS, Elasticsearch, Logstash i Kibana per a la visualització de les dades capturades i, finalment, la utilització d'una interfície de configuració de Suricata, Scirius.

Aquest projecte, anomenat SELKS com a acrònim de Suricata + Elasticsearch + Logstash + Kibana + Scirius, ha estat desenvolupat per un conjunt d'experts en xarxes informàtiques i seguretat i porten més de tres anys d'experiència en el seu desenvolupament.

En aquest moment vaig decidir replantejar-me el meu projecte i els objectius que m'havia marcat ja que no trobava adequat seguir fent un projecte després de descobrir-ne un altre amb unes característiques tan semblants. Tot i que el que estava fent m'estava agradant, no tenia la intenció de reinventar la roda amb el meu projecte.

Com que el treball que portava fet i el que em quedava per fer m'interessaven i m'agradaven, vaig decidir fer un canvi en els objectius inicials del projecte de tal forma que seguissin sent semblants, però en un àmbit diferent. Tenint una eina com Scirius que incloïa, entre moltes altres, la majoria de funcionalitats que jo havia pensat en desenvolupar i que es trobava desenvolupada sota codi lliure i en un repositori públic, vaig pensar en la possibilitat de millorar-ne les capacitats implementant alguna millora en les funcionalitats que oferia.

Abans de fer el canvi definitiu d'objectius i encarar el projecte cap a aquella direcció, em vaig posar en contacte amb els creadors del projecte SELKS i els hi vaig comentar la idea d'ajudar-los en el desenvolupament d'alguna de les seves eines.

No em van posar cap impediment en ajudar-los i, després de mantenir contacte amb varis dels desenvolupadors vaig prendre la decisió de modificar els objectius del projecte de forma definitiva.

Els nous objectius que em vaig proposar es troben en l'apartat d'introducció d'aquesta memòria.

El projecte SELKS està format per diverses utilitats i és desenvolupat per un conjunt de persones. Majoritàriament cada utilitat és desenvolupada per una persona, per lo que en cas de treballar amb ells, em comunicaria amb el coordinador de la utilitat en concret que hagués de modificar.

Treballar en aquest projecte ha implicat un nivell diferent de complexitat i aprenentatge.

El primer que vaig haver de fer va ser familiaritzar-me en profunditat amb el sistema i les eines que utilitzaven per al desenvolupament del projecte. L'aprenentatge d'ús d'un framework nou, la forma en què estructuraven i utilitzen la base de dades, l'estructura del projecte que utilitzaven, etc.

També vaig haver d'aprendre i adaptar-me a la metodologia de treball, desenvolupament i compartiment de codi i informació que utilitzaven.

Un dels punts més importants en què vaig notar durant el desenvolupament del projecte Scirius és el fet de treballar en equip. Deixes de ser una única persona que fas i apliques totes les decisions i passes a ser un conjunt de persones que discuteix com seria la millor manera de solucionar un problema o implementar una funcionalitat nova.

També és diferent el fet de dependre d'algú per prendre les decisions, ja que en comptes de decidir un mètode i començar-lo a aplicar, com que he sigut l'últim en arribar a aquest projecte i la persona amb menys experiència en l'àmbit, tota modificació important que volia efectuar, havia de parlar amb un dels coordinadors del projecte perquè donés el vistiplau.

7.4 DESENVOLUPAMENT

Per a la part de desenvolupament del projecte, el primer que es va fer va ser posar-nos en contacte amb els creadors i desenvolupadors de les eines que s'estaven utilitzant.

El primer del que es va parlar va ser de la possibilitat d'ajudar en el desenvolupament del seu projecte, sense especificar en quina eina ni de quina forma fer-ho. Es van mostrar molt oberts a rebre la meua col·laboració i, a partir d'aquí, es va començar a estudiar amb què podria treballar.

Després d'intercanviar impressions i d'explicar-los les meves propostes per al meu projecte i com, part d'aquestes, estaven ja implementades en el seu projecte, vam decidir que millorés alguna funcionalitat del projecte Scirius, l'eina encarregada de generar la interfície web que gestiona Suricata.

Amb el projecte amb el qual havia de treballar decidit, vaig començar a estudiar-ne les seves funcionalitats, característiques i estructura. Tot aquest estudi el vaig fer a partir de l'execució de l'eina i la prova de les seves funcions, sense tractar el codi del projecte.

Mentre mantenia la comunicació amb els creadors de Scirius, em vaig reciclar en el llenguatge Python, ja que l'havia utilitzat en funcionalitats molt bàsiques, i vaig començar a estudiar el *framework* que utilitzaven, el Django.

Quan vaig entendre el funcionament del llenguatge i el *framework*, vaig començar a estudiar l'estructura del codi del projecte. Aquest fet em va ajudar a entendre amb més precisió com s'utilitzava el llenguatge Python en aquest àmbit.

Amb una idea més a fons del projecte i una comprensió del llenguatge més elevada, vam començar a decidir quina modificació podria fer al projecte.

El primer en què vam convenir va ser en què em dedicués a crear una funcionalitat nova per a la eina ja que em donava més llibertat a l'hora de desenvolupar i permetia veure la feina feta de manera molt més nítida.

Després de valorar diverses opcions, juntament amb els desenvolupadors del projecte, em van proposar que afegís la funció de IPS, Sistema de Prevenció d'Intrusos, al projecte. Tot i que al principi em va semblar una responsabilitat molt gran i un nivell de complexitat important, al final em va semblar correcte i vaig començar a estudiar com s'hauria de fer.

Aquest canvi suposaria un gir molt important en la funcionalitat de l'eina Scirius, ja que deixaria de tenir la capacitat de configurar Suricata com a IDS per aconseguir tenir la capacitat de configurar Suricata com a IPS. Significaria també, deixar de tenir un sistema passiu que generés alertes per passar a tenir un sistema actiu que actués en els casos en què estigués configurat per fer-ho, donant un nivell de seguretat a la xarxa molt major.

7.5 IDE

Per a programar codi de forma més eficient, existeixen els programes IDE, els quals aporten unes facilitats en el desenvolupament de software que ajuden a que el desenvolupador només es centri amb el seu programa.

Entre les seves característiques generals trobem que incorporen un editor de textos amb estils, per a facilitar l'entesa del codi, un compilador integrat i un intèrpret, un depurador de codi, entre altres funcionalitats.

L'IDE que ens interessa utilitzar ha de complir uns requeriments mínims ja que el necessitem per unes funcions molt específiques.

Primer de tot, ha de ser compatible amb el llenguatge Python, que serà el que utilitzarem per al projecte. Normalment, aquestes eines amplien moltes de les seves característiques per mitjà de connectors i complements, per tant, no només ens fixarem en l'eina sola sinó també en els complements existents.

Una altra funcionalitat important que hauria de tenir és la integració dels sistemes de control de versions, ja que el projecte es troba allotjat a GitHub i la metodologia de desenvolupament és per mitjà de Git.

Finalment, seria interessant que l'IDE a escollir possibilités depurar el codi, el que significa que l'hauria de poder executar sense problemes. Això ens permetria trobar els errors de forma molt més ràpida i senzilla.

7.5.1 Eclipse

Eclipse és un entorn de desenvolupament integrat, IDE, utilitzat en el desenvolupament de software i el més utilitzat per a desenvolupar Java. És de codi programat en Java, el que ens indica que es podrà utilitzar en qualsevol Sistema Operatiu que es desitgi.

Tot i que és conegut sobretot per al desenvolupament de Java, també permet desenvolupar projectes en C, C++, Python, Perl, PHP i molts altres llenguatges de programació, per mitjà de connectors que ofereix el programa.

Va ser desenvolupat inicialment per IBM l'any 2004, tot i que actualment es troba suportat per una fundació independent sense ànim de lucre, la Eclipse Foundation.

7.5.2 Netbeans

Netbeans IDE és una eina de desenvolupament modular per a un gran rang de tecnologies de desenvolupament d'aplicacions. L'IDE base inclou un editor multi-llenguatge avançat, un depurador de codi, així com eines per a controls de versions i col·laboracions en desenvolupaments.

Ens proporciona un sistema de directoris estructurat i un conjunt de plantilles per a començar projectes amb diferents llenguatges de programació.

També ofereix una integració de les bases de dades, permetent la interacció amb aquestes des de dins el propi entorn.

També incorpora la possibilitat de millorar i ampliar les funcionalitats bases de l'eina per mitjà de connectors i complements.

Finalment, indicar que es tracta d'una eina gratuïta i de codi lliure.

7.5.3 IntelliJ

IntelliJ és un IDE desenvolupat per JetBrains. La primera publicació va ser a l'any 2001 i cada anys fins a l'actualitat han anat traient versions noves. Tot i no ser un software de codi lliure, ofereixen una versió gratuïta amb la qual es pot desenvolupar qualsevol aplicació sense necessitats específiques.

IntelliJ analitza el nostre codi, buscant connexions entre símbols a través de tots els fitxers del projecte. Utilitzant aquesta informació proporciona una assistència al codi en profunditat, una ràpida navegació, un anàlisi d'errors intel·ligent, i, també, refactoritzar.

Durant el desenvolupament ens proporciona un entorn de treball on el centre és l'editor, dreceres per a totes les funcionalitats, una interfície d'usuari ergonòmica o un depurador de codi que l'analitza línia a línia.

Algunes de les eines que incorpora IntelliJ per facilitar el desenvolupament són un control de versions automatitzat, on cada funció té la seva drecera, eines per a la compilació, empaquetat, execució de testos, implementació i altres activitats de forma automàtica. També ens proporciona un terminal amb les funcionalitats de Linux, una eina de gestió de bases de dades i una de servidors d'aplicacions.

A més a més, gràcies a la quantitat de connectors i complements dels que disposa, és compatible amb una àmplia gamma de llenguatges.

7.5.4 Taula comparativa

| | ECLIPSE | NETBEANS | INTELLIJ |
|--------------------------------------|----------------|-----------------|-----------------|
| COMPATIBLE AMB PYTHON | Si | Si | Si |
| INCORPORA CONTROL DE VERSIONS | Si | Si | Si |
| DEPURACIÓ DE CODI | Si | Si | Si |
| SUPORT ACTUAL | Si | Si | Si |
| PRIMERA PUBLICACIÓ | 2004 | 2004 | 2001 |
| ÚLTIMA PUBLICACIÓ | 2016 | 2016 | 2016 |
| CODI LLIURE | Si | Si | No |
| GRATUÏT | Si | Si | Si |

Taula 3- Comparativa IDEs

7.5.5 Decisió

Hem comparat els tres IDEs més coneguts que actualment es troben al mercat.

Les característiques generals entre tots tres són bastant semblants, sense que hi hagi cap gran diferència, però quan ens fixem en els petits detalls veurem que hi ha diferències notables.

Durant els últims anys he pogut utilitzar totes tres eines, per lo que cap d'elles m'és desconeguda. Això influeix clarament en la meua decisió, ja que sempre es té predilecció per un per sobre dels altres.

L'IDE que utilitzarem serà l'IntelliJ ja que, personalment, és el que m'ha proporcionat millor rendiment. Tot i això, si ens fixem en les característiques de tots tres IDEs, podem veure com aquest és el que té un acabat més perfeccionat amb unes funcionalitats molt ben estudiades i encarades al desenvolupament de software.

El fet que no sigui de codi lliure té gaire importància en aquest cas, ja que tinc la certesa, després d'estar utilitzant-lo durant anys, que no em trobaré cap complicació en el codi que hagués de requerir una modificació per part meua.

Entre altres, les característiques més destacables d'aquesta eina és la gran integració que té amb tots els llenguatges que accepta, la capacitat predictiva quan estàs desenvolupant o la capacitat de connectar totes les estructures de dades de tot el projecte, el que facilita enormement moure's d'un bloc a un altre del sistema que s'està desenvolupant.

També incorpora un depurador molt ben aconseguit, el qual és capaç d'analitzar i mostrar els valors de totes les variables línia a línia. Aquesta funcionalitat facilita enormement la cerca i detecció d'errors en el codi així com la solució d'aquests.

8 ANÀLISI I DISSENY DEL SISTEMA

En aquest apartat realitzarem un anàlisi de les funcionalitats i del codi de desenvolupament del projecte Scirius. Aquests anàlisis ens serviran per comprendre i aprendre l'estructura i el funcionament d'aquest projecte, i amb això, ens serà més fàcil desenvolupar o millorar alguna funcionalitat nova o existent.

La metodologia que seguirem és començar per l'anàlisi més genèric, i anar seguint fins arribar a un anàlisi més específic de la part que haurem de modificar durant el desenvolupament del projecte.

8.1 ANÀLISIS FUNCIONAL

Des d'un inici veiem que podem separar les funcions de l'eina en dues vessants molt diferenciades, la configuració i la visualització de dades.

8.1.1 Configuració

Aquesta vessant la podem separar en tres blocs, les fonts, els conjunts personalitzats de regles i Suricata. Tots tres blocs es troben relacionats entre ells i efectuar un canvi en un repercutirà en els altres dos.

Fonts

Les fonts de regles és des d'on Suricata extreu les regles que ha de comprovar, per tant és important que es trobin amb una bona configuració.

Permet afegir les regles que es vulguin amb diferents mètodes de càrrega, ja sigui per HTTP o per assignació d'un fitxer local. A més a més, podem carregar un fitxer de regles únic o un fitxer comprimit que en contingui un conjunt.

Un cop creada la font, ens permet efectuar-hi diverses accions com poden ser modificar la informació inicial, eliminar la font de la llista o actualitzar-la. Aquesta última acció implica que es puguin eliminar, modificar o crear regles de forma automàtica.

Dins d'aquest apartat també s'hi inclouen les categories que formen una font i les regles que formen cada categoria. Per cada categoria tenim l'opció d'habilitar-la o desactivar-la i per cada regla, a més a més d'aquestes dues accions, podem eliminar-la de la llista.

Rulesets

Els conjunts de regles, anomenats *Rulesets*, es componen de fonts.

Com en el cas anterior, es podran realitzar diferents accions. La primera de totes és la d'afegir un nou conjunt. Per fer-ho s'assignaran les fonts que es volen utilitzar per a aquests conjunt i s'hi indicarà un nom.

També podrem modificar, editar, eliminar i copiar el conjunt de regles que es vulgui, permetent tenir un grau de llibertat interessant.

És important configurar correctament els *Rulesets* ja que seran els que utilitzarà Suricata per decidir quines regles comprova durant l'anàlisi.

Suricata

Des de la interfície web podrem configurar les opcions principals de Suricata i realitzar certes accions referents a les regles.

La configuració general que ens permetrà fer és sobretot referent als fitxers de configuració general i els fitxers de regles i sobre quin *Ruleset* es vol utilitzar com a predeterminat a l'hora d'analitzar les regles.

També ens permetrà efectuar diferents operacions referents a aquest *Ruleset* predeterminat. Podrem actualitzar-lo, de tal forma que les regles es trobin al dia, generar el fitxer únic de regles on es trobaran totes les que s'hagin assignat al seu *Ruleset* i, finalment, podrem fer efectius aquests canvis en les regles.

8.1.2 Visualització de dades

La interfície gràfica té moltes parts on es mostra informació de diferents dades, ja sigui mitjançant gràfiques com per mitjà de valors numèrics.

Es pot separar el tipus de dades que es pot visualitzar entre dades referents a les alertes generades per Suricata, dades referents a la quantitat d'informació que s'ha recollit i els recursos que s'està utilitzant per fer-ho.

Alertes

A la pàgina principal de la interfície hi trobem un resum de les alertes i la quantitat que se n'ha generat durant un període de temps indicat. Aquesta informació és bastant semblant a la que trobem a l'apartat Suricata.

En cas de voler més informació de l'alerta que s'ha generat, podem anar a la regla que l'ha fet saltar i se'ns mostrarà una informació més detallada d'aquesta.

Informació recollida

Per tenir una idea de la quantitat d'informació que s'està tractant, hem d'anar a l'apartat d'Elasticsearch. Des d'allà podrem comprovar diferents punts que hi fan referència.

Se'ns mostra informació de els fragments de dades, dels nodes actius, del sistema de fitxers i dels documents utilitzats.

Entre altres informacions, podem observar el nombre de fragments que utilitza i en quin estat es troben, la quantitat de dades recollida per a l'anàlisi o el número de documents creats que han de ser tractats.

Recursos

Per comprovar els recursos que està utilitzant Suricata per capturar i analitzar paquets, podem anar al seu apartat i podrem observar diferents característiques.

Algunes de les característiques més importants que podem observar és la quantitat de memòria que està utilitzant la màquina, separat per protocols. Aquesta informació ens pot indicar algun índex de mal funcionament, o, en cas d'estar sempre utilitzant el màxim de la seva capacitat, la necessitat d'augmentar-la per millorar el rendiment.

També podem comprovar la velocitat en què Suricata captura els paquets o la informació recollida d'aquests és enviada a l'eina de visualització.

8.2 ANÀLISI ESTRUCTURAL

Un cop hem vist les diferents funcions que ofereix aquesta eina, ens mirarem quina estructura de desenvolupament té.

El primer que podem observar és que aquest projecte s'estructura sobre quatre blocs molt diferenciats:

- **Accounts:** Els comptes d'usuari i les accions relacionades
- **Rules:** Les regles i les accions que s'hi apliquen
- **Scirius:** La base de l'aplicació web
- **Suricata:** Totes les accions i tota la informació referent a Suricata

La tasca que haig de fer en aquest projecte és incorporar la funcionalitat de IPS. Aquesta funcionalitat es basa en l'estructura de les regles i les accions que es prenen, per tant, ens interessa centrar-nos en el bloc de **rules** del projecte ja que serà amb el que haurem de treballar.

L'apartat de les regles consta tant d'una part de codi referent a la interfície gràfica, com una part que implementa les diverses funcionalitats.

El centre d'aquest apartat és el fitxer **models.py** on hi trobem totes les classes referents a les estructures de dades que s'estan utilitzant.

L'estructura de dades que s'utilitza per a les regles és jeràrquica. Consta d'una font de regles, la qual té un conjunt de categories associades a ella. Cadascuna d'aquestes categories té un conjunt de regles associades a les quals té accés.

Un *Ruleset* pot estar format per diverses fonts, de la mateixa manera que una font pot ésser assignada a varis *Rulesets* al mateix temps.

També és important notar que una regla pot ser assignada a múltiples *Rulesets* i cadascuna d'aquestes assignacions és independent una de l'altra. Això significa que qualsevol canvi que pateixi la regla en un *Ruleset* no afectarà a la mateixa regla assignada a la resta de *Rulesets*.

L'estructura de relacions sencera que tenim en el bloc de **rules** és el següent:



Figura 6- Diagrama de classes Scirius/rules inicial

8.3 DISSENY DE LES TASQUES

En aquests apartat explicarem detalladament com s’ha dissenyat cada tasca realitzada i amb quines característiques s’ha fet.

Abans de començar amb el disseny del projecte, s’ha estudiat de forma intensa dues utilitats encarregades de gestionar les regles, Oinkmaster i Pulledpork. Aquesta feina ens ha servit per entendre com ho fan aquestes dues eines per modificar les regles, ja sigui per activar-les o desactivar-les com per modificar-ne el contingut intern.

Després de estudiar-ne el seu funcionament i la forma en què tracten les regles i les seves modificacions m’ha estat molt més fàcil pensar en un possible disseny per al desenvolupament del projecte.

8.3.1 Disseny inicial

El primer que hem fet és fer un disseny de les classes i mètodes necessaris per a la realització de les tasques posteriors. Abans de començar a implementar-les necessitàvem un conjunt d'estructures per a portar-les a terme.

Per a transformar les regles perquè realitzin accions diferents a la que se'ls hi assigna per defecte necessitàvem una classe que recollís aquesta informació. Aquesta classe havia de distingir la regla i el *Ruleset* concret als que feia referència, per tant, havia d'estar relacionada amb ambdues classes.

El que s'ha fet primer, doncs, és crear la classe *Transformation* per a que sigui l'encarregada de totes les modificacions en les accions d'una regla. Aquesta classe tindrà un **datatype** on es guardarà l'acció que realitza la regla, i dos relacions amb les classes *Rule* i *Ruleset*.

S'ha creat un mètode per a la modificació de l'acció dins la classe *Transformation*. Aquest mètode és l'única manera, un cop s'ha creat l'objecte, de canviar-ne el tipus guardat a la variable **datatype**.

Amb la classe intermèdia *Transformation* creada afegirem a la classe *Rule* una relació amb la classe *Ruleset* a través de la primera per a que es guardin totes les modificacions que es produeixin.

Creem un mètode a la classe *Rule* per a obtenir l'acció que s'aplicarà a la regla donat un *Ruleset* concret.

Pera transformar una regla s'utilitza un mètode de la classe *Rule*. Aquest mètode, amb una acció i un *Ruleset* donats, crea, en el cas de no existir, o modifica, en cas contrari, la regla a la que fa referència amb els valors de l'acció que se li han passat.

En els formularis de creació de *Ruleset* s'ha afegit que quan es creï una font, que, al mateix temps importa totes les seves categories, faci que per a cadascuna d'aquestes obtingui totes les regles que hi estan assignades. Per a cadascuna d'aquestes regles es crea un objecte *Transformation* amb el **datatype** per defecte '*alert*' que és el que ve assignat a les signatures.

En el cas de la creació d'un *Ruleset* per defecte s'efectuaran els mateixos canvis que en el cas anterior per a que la generació es realitzi de la forma correcta.

També hem modificat la forma en què es genera el fitxer de regles únic **scirius.rules** per tal que s'hi incorporin les regles amb les modificacions fetes segons el *Ruleset* que se li ha assignat. D'aquesta manera, si una regla per el *Ruleset* indicat ha estat modificada s'incorporarà al fitxer el valor de l'acció després de la modificació.

En la visualització d'una regla, en l'apartat on es mostra la informació d'aquesta a la secció *Status in Rulesets*, s'hi ha d'afegir una columna a la taula que es mostra amb el valor de l'acció que s'aplica a la regla per cada *Ruleset* a la que estigui assignada.

Per fer això, el primer que hem de fer és afegir un nou camp a la classe encarregada de generar aquesta taula i, també, indicant-li el nom que es mostrarà en la capçalera de la columna generada.

8.3.2 Modificar l'acció d'una regla

L'objectiu d'aquesta tasca és aconseguir modificar el valor de l'acció que realitzarà la regla en cas de coincidir. Això ho aconseguirem a partir de la classe *Transformation* i els diferents mètodes explicats en l'apartat anterior.

S'han afegit tres accions al panell d'accions que es troba a la pàgina d'una regla. Aquestes accions són *Alert*, *Allow* i *Drop*. Quan es seleccioni alguna d'aquestes opcions se'ns enviarà a una nova pàgina on es mostraran tots els *Rulesets* en els quals la regla està associada.

Per aplicar l'acció indicada, s'haurà de seleccionar els diferents *Rulesets* als quals se'ls hi vol incloure i clicar al botó *Submit*.

Per enllaçar l'opció a la pàgina de la regla amb la nova pàgina que se'ns mostrarà s'haurà de crear i assignar una URL única per a cada una de les opcions que hem indicat anteriorment.

Un cop es seleccionen els *Rulesets* als quals es vol aplicar les transformacions, per a cadascun d'ells, es modifica l'objecte *Transformation* assignat a la regla i el *Ruleset* en qüestió i es canvia el valor de l'acció del camp **datatype** per l'opció seleccionada anteriorment.

Aquesta tasca només es troba disponible per als usuaris amb privilegis d'administrador ja que comporta canvis en els objectes creats a la base de dades. En cas de no tenir els permisos necessaris no es deixarà executar la tasca i s'enviarà un missatge d'error.

Un cop s'ha acabat l'execució, se'ns redirigirà a la pàgina de la regla que s'ha modificat.

8.3.3 Modificar l'acció de les regles d'una categoria

Aquesta tasca busca poder modificar el valor de l'acció de totes les regles en una mateixa categoria. Per fer-ho haurem de modificar individualment totes les regles que compleixin els requisits necessaris.

S'han afegit tres entrades, *Alert*, *Allow* i *Drop*, en el panell d'accions dins la pàgina d'una categoria.

Quan es seleccioni una d'aquestes opcions, es mostrarà una pàgina que s'ha creat per a cada opció on hi trobarem un llistat de tots els *Rulesets* als quals es troba assignada la categoria que es vol modificar. Per a aplicar els canvis haurem de seleccionar un mínim d'un *Ruleset* i clicar el botó de *Submit*.

Per a aquestes accions s'ha hagut de crear una pàgina per a cadascuna de les opcions disponibles i enllaçar l'opció de la pàgina amb aquesta nova per mitjà de la creació d'un URL únic per a cada tipus de modificació.

Un cop s'ha cridat l'acció de modificar la categoria en un o varis *Rulesets* indicats, es procedeix a actuar sobre cada regla associada a la categoria per tal de canviar-ne l'objecte *Transformation* relacionat i assignar-li el valor de l'opció que s'ha seleccionat.

Aquest procés es realitzarà per a cada *Ruleset* que es trobés seleccionat a la pàgina.

Com en la tasca anterior, aquesta només la pot realitzar un usuari amb permisos d'administrador. Per tant, en cas de no tenir-los s'enviaria un missatge d'error i n'aturaria l'execució. Amb la fi de l'acció, se'ns redirigirà a la pàgina de la categoria modificada.

8.3.4 Transformar múltiples regles d'una categoria en múltiples Rulesets

En aquesta tasca es busca poder modificar el valor de l'acció de múltiples regles d'una categoria seleccionades de dins un llistat indicant, també, sobre quin dels *Rulesets* disponibles es volen aplicar aquests canvis.

El primer que hem fet ha estat afegir l'opció per realitzar aquesta tasca al panell d'accions de la pàgina d'una categoria. Si es selecciona aquesta opció, se'ns enviarà a una nova pàgina.

Aquesta pàgina consta d'un llistat dels *Rulesets* als quals estan assignades les regles de la categoria indicada, el llistat de regles que pertanyen a la categoria i, finalment, els botons per a realitzar les diferents accions sobre les regles i els *Rulesets* seleccionats. Per fer la transició de la pàgina de la categoria cap a la pàgina nova s'ha creat un URL únic que ens hi dirigirà.

Un cop s'activa una de les opcions el primer que es fa és comprovar que l'usuari que està intentant realitzar l'acció sigui l'administrador del sistema. En cas de ser així, es segueix amb l'execució, en cas contrari, s'atura i es mostra un missatge d'error.

Per a cada *Ruleset* seleccionat es modificarà les accions de totes les regles seleccionades. Això ho farem canviant el valor de l'acció de l'objecte *Transformation* relacionat amb el *Ruleset* i la regla indicats. Un cop acabat, se'ns redirigirà a la pàgina de la categoria modificada.

8.3.5 Transformar totes les regles de múltiples categories

Aquesta tasca ha d'aconseguir que totes les regles assignades a un conjunt de categories seleccionades siguin modificades per a canviar el tipus de l'acció que realitzen en diferents *Rulesets* seleccionats.

Per aconseguir això, el primer que fem és afegir una nova opció al panell d'acció de la font. Aquesta opció ens dirigirà cap a una nova pàgina.

Com en els casos anteriors, es necessitarà crear un URL únic per aquesta pàgina de forma que hi podem accedir de forma directa.

A la pàgina que es mostra, els *Rulesets* que es llisten són aquells que es troben assignats amb totes les categories de la font amb la qual s'està treballant. Per tant, amb totes les regles assignades a aquestes categories.

Un cop s'envia la petició de modificació de les categories, el primer que es fa és comprovar l'autoria d'aquesta. L'usuari que ha efectuar la petició ha de ser l'administrador del sistema ja

que efectuar canvis en els objectes de la base de dades només li és permès fer a ell. En cas de no ser així, s'aturaria l'acció i s'enviaria un missatge d'error.

Tot seguit es procedeix a executar la tasca. S'agafen tots els *Rulesets* seleccionats, i per a cadascun d'ells, s'agafa totes les categories seleccionades.

Per cada categoria que s'hagi seleccionat es modifiquen totes les regles, canviant el valor amb l'acció i el *Rulesets* actuals. Un cop s'ha realitzat la tasca, se'ns redirigirà a la pàgina de la font que hem modificat.

8.3.6 Disseny final

Amb totes les modificacions fetes ens queda una estructura de dades així:

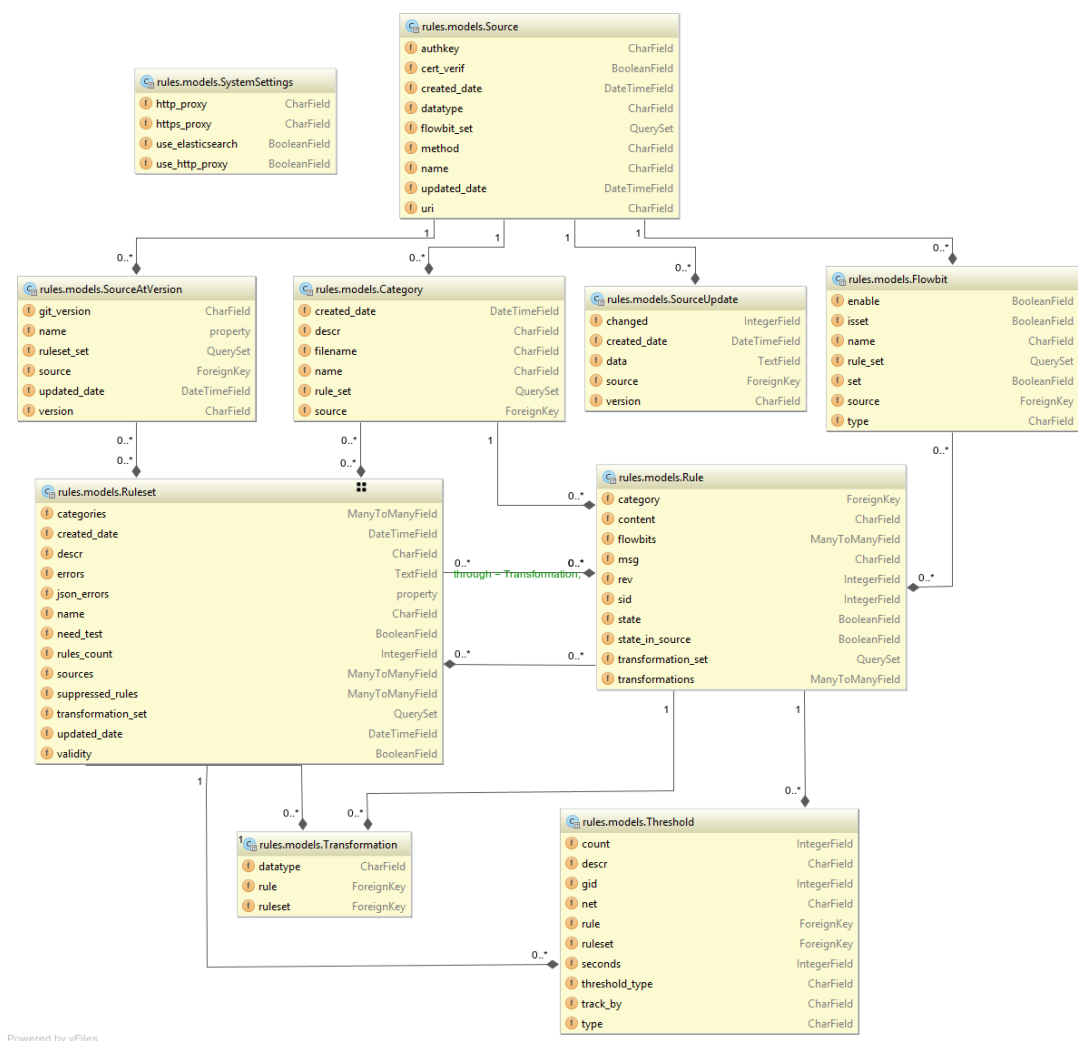


Figura 7- Diagrama de classes Scirius/rules final

El desenvolupament realitzat amb la meua col·laboració amb Scirius es pot seguir a <https://github.com/adescamps11/scirius>

En cas que les meves modificacions fossin incorporades a una nova versió, es trobaria a <https://github.com/StamusNetworks/scirius>

9 IMPLEMENTACIÓ I PROVES

En aquest apartat s'explicarà de forma detallada la instal·lació i els apartats de la configuració més importants dels diferents sistemes, eines i utilitats que s'han fet servir.

Suricata serà la primera eina que instal·larem al nostre servidor. A partir d'aquesta eina es decidiran aspectes del treball com ara el Sistema Operatiu o les posteriors eines complementaries que s'utilitzarà.

Per no fer molt llarga l'explicació, s'indicaran els passos i la configuració que s'ha de fer de forma molt general i en els annexos s'afegirà una guia molt més detallada de tot el procés.

9.1 SISTEMA OPERATIU

Primer de tot es decidirà quin Sistema Operatiu utilitzarem. Per fer-ho, mirarem quins són els Sistemes Operatius per els quals està disponible Suricata.

- Ubuntu
- Debian
- CentOS
- Fedora
- OpenSUSE
- FreeBSD
- Mac OS
- Windows

És important mirar si es farà servir un Sistema Operatiu basat en Linux, en Windows o en Mac OS. Com que Suricata ha estat desenvolupat des de l'inici i enfocat per entorns Linux, tot i haver desenvolupat posteriorment l'eina per a Windows i Mac OS, ens centrarem en les diferents distribucions de Linux compatibles amb l'eina.

Entre les diferents distribucions Linux per les quals està desenvolupat, s'ha decidit utilitzar Ubuntu Server.

Ubuntu és la distribució més coneguda i que, últimament, està atraient més comunitat. A més a més, és la distribució que, personalment, he utilitzat més i en tinc més coneixement.

La versió de Ubuntu que s'utilitzarà serà la de Ubuntu Server 16.04 LTS ja que per aquest projecte no necessitem interfície gràfica en el Sistema Operatiu i això ens proporciona un millor rendiment i una pèrdua menor de capacitat utilitzada pel propi sistema. La versió 16.04 LTS és la última versió estable i amb suport a llarg termini, 5 anys, disponible.

Un cop decidit el Sistema Operatiu que s'utilitzarà, es procedirà a instal·lar-lo com a màquina virtual al servidor.

Explicarem pas a pas com es crea una màquina virtual dins el sistema de VMWare i un cop creada la màquina virtual procedirem a la instal·lació del Sistema Operatiu.

9.1.1 Creació de la màquina virtual

Es crearà una màquina virtual amb una configuració personalitzada, el que ens permetrà modelar-la més adequadament a les nostres necessitats.

Es voldrà assignar uns valors personalitzats a la màquina per tal que s'adeqüi a les nostres necessitats.

Indicarem el Sistema Operatiu que es voldrà instal·lar a la nostra màquina i li assignarem el màxim de processadors disponibles ja que és un punt clau en el rendiment i funcionament del sistema.

S'afegirà dues interfícies de xarxa ja que se'n necessitarà una per a la gestió del sistema i una altra que rebí les dades que ha de capturar.

S'assignarà una quantitat important de disc a la màquina ja que guardarà tots els registres de les captures que faci el sistema i és una gran quantitat d'espai. Tot i això, es podrà ampliar la capacitat de disc assignat a la màquina un cop acabada la instal·lació.

Se'ns mostrarà un resum de les característiques assignades al sistema per poder comprovar que no hi ha hagut cap equivocació en el procés.

9.1.2 Instal·lació del Sistema Operatiu

Els primers passos es configura la regió, idioma i l'usuari administrador del sistema.

Es configuraran les diferents particions del sistema de forma manual, d'aquesta manera podem configurar les diferents parts segons les necessitats pròpies. Es separarà el sistema en quatre particions, la d'arrencada, la d'intercanvi de memòria, la partició arrel on s'hi trobarà la major part del sistema i la partició de variables, on es troben els registres, que serà la que es configurarà amb una major capacitat.

Tot seguit s'assignaran les diferents particions creades al Grup de Volums Lògics corresponents.

Finalment es configuraran les últimes característiques abans d'acabar la instal·lació com ara les actualitzacions automàtiques, el servei *OpenSSH Server* per a poder-nos connectar en remot i, en el cas d'utilitzar una màquina virtual, el servei *Virtual Machine Host*.

9.2 SURICATA

Abans de començar la instal·lació del servei com a tal tenim que comprovar quins requisits demana i si es troben instal·lat en el servidor.

En el nostre cas cap dels requisits que ens podem trobar estaran instal·lats ja que tenim el servidor amb la imatge d'instal·lació, el qual ve sense les utilitats necessàries.

9.2.1 Prerequisits

Abans de poder instal·lar Suricata, cal tenir un seguit de llibreries i utilitats les quals fa servir el sistema per funcionar. Les descripcions les podem trobar a <http://packages.ubuntu.com/>

Depenent de l'estat actual del sistema, pot tardar en completar aquest procés.

HTP

És un analitzador de seguretat per el protocol HTTP que utilitza Suricata, per defecte ve integrat amb l'eina.

Es pot comprovar i aconseguir la última versió aquí - <http://suricata-ids.org/download/>

Mode IPS

Per defecte, Suricata treballa com un IDS. Si es vol utilitzar com a programa IDS i IPS, s'haurà d'instal·lar un conjunt de paquets addicionals.

En aquests moments ja tindriem tots els requisits per poder instal·lar Suricata.

9.2.2 Descàrrega i instal·lació

Per obtenir Suricata ho farem mitjançant el paquet que proporciona la pròpia fundació que el desenvolupa. Un cop s'obté el paquet, es descomprimirà per tenir accés als seus arxius. [10]

9.2.3 Configuració

Preparació

Per preparar la configuració haurem de crear el directori on es guardaran tots els registres de Suricata. De la mateixa manera, s'ha de crear el directori on es situaran els diferents fitxers de configuració de l'eina.

Procedirem a copiar tots els fitxers de configuració del directori d'instal·lació al directori que hem creat per a Suricata.

Altrament, també es disposa d'una preparació automàtica.

El codi font de Suricata posseeix uns fitxers de configuració automàtica per defecte. S'instal·laran aquests fitxers de configuració per defecte.

Suricata no té sentit sense unes regles IDS. El fitxer Makefile ve amb una opció d'instal·lació de regles IDS. Aquesta comanda descarregarà el conjunt de regles creades per la comunitat actualment disponibles a EmergingThreats.net i les guardarà a **/etc/suricata/rules**

Configuració inicial [11]

El fitxer de configuració de Suricata es troba a `/etc/suricata/suricata.yaml`. Obrirem el fitxer amb un editor de text per modificar-ne les opcions. Començarem amb una configuració inicial.

Registre

La variable `default-log-dir` ha d'apuntar a la localització del fitxer de registre de Suricata.

Variables

A la secció `vars:`, podem trobar varies variables importants utilitzades per Suricata.

`HOME_NET` hauria d'indicar la xarxa local inspeccionada per Suricata.

`!$HOME_NET` s'assigna a `EXTERNAL_NET` i refereix a qualsevol altre xarxa a part de la local.

`XXX_PORTS` indica el número de port utilitzats per diferents serveis.

Tot i això, Suricata és capaç de detectar automàticament el trànsit HTTP sense tenir en compte els ports utilitzats, per tant no és una característica crítica especificat el valor de la variable `HTTP_PORTS` correctament.

Política basada en SO

La secció `host-os-policy` s'utilitza per defensar-se davant alguns dels atacs coneguts que exploten el comportament de la xarxa d'un sistema operatiu per evadir la seva detecció. Com a contra mesura, els IDS moderns han desenvolupat les anomenades inspeccions basades en l'objectiu ("target-based"), on el motor d'inspecció afina el seu algorisme de detecció basat en el sistema operatiu del trànsit a l'objectiu. Per això, si es coneix el Sistema Operatiu que utilitzen els hosts locals, es pot entrar aquesta informació a Suricata i incrementar potencialment el nivell de detecció. Aquí és on la secció `host-on-policy` és utilitzada.

Fils d'execució

A la secció `threading` podem especificar l'afinitat de CPU per a diferents fils d'execució de Suricata.

Per defecte, l'afinitat de CPU es troba desactivada el que significa que els fils d'execució de Suricata seran programats en qualsevol dels nuclis disponibles de la CPU.

Si no s'indica cap configuració en concret, Suricata crearà un fil d'execució de detecció per cada nucli de la CPU. Es pot ajustar aquest comportament especificant un valor a la variable `detect-thread-ratio: N`. D'aquesta manera es crearan N fils d'execució per cada nucli, per tant, N*M fils d'execució on M són el nombre de nuclis.

9.2.4 Línia de comandes [12]

Opcions

Suricata és una eina amb execució via la línia de comandes. Això ens aporta la possibilitat de poder executar aquesta eina amb diferents opcions i configuracions, indicant-ho a la línia d'execució.

Un exemple d'execució utilitzant diverses opcions és:

```
suricata -c /etc/suricata/suricata.yaml -i eth0 -D
```

Aquesta seria una execució bàsica de Suricata on s'especifica el fitxer de configuració que ha de fer servir l'eina, la interfície de xarxa que ha d'utilitzar per a capturar els paquets i s'indica, amb l'opció **-D**, que s'executi en mode dimoni, és a dir, com a procés en segon pla.

Tests unitaris

Suricata incorpora un seguit de tests unitaris per a la verificació del codi. Aquests tests es poden utilitzar mitjançant la línia de comandes indicant unes opcions concretes.

9.3 OINKMASTER

És possible descarregar i instal·lar les regles de forma manual, però existeix una manera molt més ràpida i còmode de fer-ho. Hi ha programes especials que s'utilitzen per descarregar i instal·lar regles, com serien Pulled Pork i Oinkmaster. En aquest projecte farem servir Oinkmaster ja que és el que es recomana des de la OISF, organització fundadora de Suricata.

9.3.1 Instal·lació

Per començar instal·larem Oinkmaster al nostre Sistema Operatiu.

Existeixen diferents tipus de regles, Emerging Threats (ET), Emerging Threats Pro i VRT. En aquest cas utilitzarem Emerging Threats.

9.3.2 Configuració

Oinkmaster necessita saber on es troben aquestes regles. Les podem trobar en el següent enllaç:

```
https://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz
```

Per afegir aquest enllaç a l'Oinkmaster, haurem de modificar el fitxer **/etc/oinkmaster.conf**.

El següent pas que farem serà crear un directori per les noves regles i copiar les que es trobin en el directori de Suricata en aquest.

En el nou directori de regles es podran trobar els fitxers **classification.config** i **reference.config**. Els directoris d'ambdós han de ser afegits al fitxer **suricata.yaml**.

S'afegeixen les noves localitzacions dels fitxers al lloc de les actualment presents.

Per comprovar si tot funciona correctament, executarem Suricata. En aquesta execució hi afegirem la opció **-init-errors-fatal** per comprovar que no existeixi cap problema.

Un cop estigui funcionant, afegirem la opció **-D** per executar-ho en mode dimoni i podem comprovar si ja hi ha registres al fitxer **/var/log/suricata/fast.log**

Emerging Threats conté més regles que les que es troben carregades a Suricata.

Es miren quines són aquelles que no es troben presents al fitxer **suricata.yaml** i s'hi afegeixen si es desitja. Per comprovar quines regles es troben actives i quines no es pot fer mirant el fitxer **/etc/suricata/suricata.yaml**.

Si es desactiva una regla col·locant un **#** al davant, la pròxima vegada que s'executi Oinkmaster es tornarà a activar. Es pot desactivar directament per Oinkmaster. Per fer-ho s'anirà al directori de regles de Suricata i es buscarà l'identificador de la regla o regles que es vulguin desactivar.

Per activar o desactivar una regla, un cop es sap el seu identificador, s'afegirà al fitxer de configuració d'Oinkmaster.

9.3.3 Actualització automàtica de regles

Com que interessa que les regles estiguin actualitzades, i Emerging Threats actualitza les seves regles diàriament, el que es vol és que les nostres també ho facin. Per fer això, en comptes d'executar la comanda anterior a diari, el que es fa és programar una tasca al **cron** del Sistema Operatiu perquè ho faci. [13]

S'afegeix una configuració per descarregar les regles cada dia a la mitjanit.

També afegirem una configuració per actualitzar les regles que utilitza Suricata sense parar el servei. D'aquesta manera no es perdrà cap paquet i en cap moment tindrem el procés parat.

9.4 ELASTICSEARCH

9.4.1 Prerequisits

L'únic prerequisit que ens demana Elasticsearch per a la instal·lació és tenir una versió de Java instal·lada. En el nostre cas instal·larem la versió més actual de Oracle Java 8, ja que és la que ens recomanen els desenvolupadors d'Elasticsearch. Tot i això, OpenJDK hauria de funcionar correctament com a alternativa de codi lliure.

Comprovem si tenim instal·lat el paquet, i en cas contrari procediríem a fer-ho.

Un cop tenim instal·lat Oracle Java 8, podem començar amb la instal·lació de Elasticsearch.

9.4.2 Instal·lació

Elasticsearch es pot instal·lar des d'un gestor de paquets afegint a la llista de fonts el paquet d'Elastic.

Primer de tot s'ha d'importar la clau pública GPG d'Elasticsearch, afegir la descripció del repositori i instal·lar Elasticsearch.

Ara que Elasticsearch es troba instal·lat en el nostre sistema, editarem el fitxer de configuració **/etc/elasticsearch/elasticsearch.yml** per fer-hi unes primeres modificacions inicials.

Es voldrà restringir l'accés exterior cap al nostre Elasticsearch, port 9200, de tal forma que des de fora no sigui possible llegir les nostres dades o apagar el clúster d'Elasticsearch des de l'API HTTP. Per fer-ho, buscarem la línia que conté **network.host**, traurem el comentari, i canviarem el seu valor per "**localhost**".

Volem que Elasticsearch s'iniciï al mateix moment que ho faci el servidor, per tant, haurem de modificar el servei d'arrencada d'aquest per tal que ho faci.

Un cop fet això, ja tindrem instal·lat Elasticsearch.

9.4.3 Configuració

Per modificar la configuració predeterminada d'Elasticsearch s'ha d'editar el fitxer **/etc/elasticsearch/elasticsearch.yml**

Primer de tot ens permet indicar un nom descriptiu per al nostre clúster, assignar un nom a cada node o indicar la localització dels fitxers on es troben les dades guardades i la localització on es guardaran els fitxers de registres de l'eina.

També es pot assignar una quantitat de memòria per a Elasticsearch i d'aquesta manera assegurar que té la quantitat mínima necessària.

El rendiment de l'eina baixa molt notablement quan el sistema està utilitzant l'intercanvi de memòria (*swapping memory*).

Com a últim punt important en la configuració, es pot indicar la direcció i el port pel qual es vol accedir a l'eina.

9.5 LOGSTASH

9.5.1 Instal·lació

Per instal·lar aquesta eina s'utilitzarà el repositori que ofereixen els seus desenvolupadors.

El primer que s'ha de fer és descarregar i instal·lar la Clau de Signatura Pública que ens ofereixen i afegir el repositori, i es pot procedir a instal·lar l'eina.

Un cop finalitzi l'operació ja tindrem acabada la instal·lació de Logstash al nostre sistema.

9.5.2 Configuració

Crearem el fitxer **logstash.conf** en el directori **/etc/logstash/conf.d/**, el qual serà el seu fitxer de configuracions.

El contingut d'aquesta eina s'estructura en tres blocs diferenciats, l'entrada d'informació, el filtrat d'aquestes dades i la sortida i posterior registre.

En el primer apartat, indicarem el fitxer de registre del qual es llegiran les dades, la localització de la base de dades pròpia de Logstash, el tipus de codificació del fitxer de lectura i se li assigna un tipus, al gust de l'administrador.

En el segon bloc, es configuren els diferents filtres segons el tipus que siguin (en aquest moment s'utilitza el que s'ha assignat al primer apartat).

Finalment es configurarà la sortida i registre de les dades. En aquest cas s'indicarà el nom del fitxer que es crearà per guardar-hi la informació i, segons si es tracta d'un esdeveniment o del registre de l'estat de l'eina es crearà amb un nom o amb un altre.

9.6 KIBANA

9.6.1 Prerequisits

Per instal·lar i configurar Kibana es necessiten alguns components i requisits.

Com s'ha comentat anteriorment, Kibana treballa conjuntament amb Elasticsearch, per tant es necessitarà tenir aquesta eina instal·lada amb la versió corresponent a la que s'estigui utilitzant per Kibana. En el nostre cas necessitarem una versió d'Elasticsearch igual o posterior a la 2.3.

També es necessitarà recuperar la informació d'Elasticsearch, per tant s'haurà de saber alguns paràmetres de la seva instal·lació:

- URL de la instància d'Elasticsearch a la qual es vulgui connectar.
- Quins índex d'Elasticsearch es volen buscar.

Degut a les característiques que ofereix Kibana, no es pot utilitzar en tots els navegadors existents. Els navegadors compatibles, els més moderns, són els següents:

- Internet Explorer 11+
- Firefox
- Chrome
- Safari (Mac)
- Safari (iOS)
- Chrome (Android)

9.6.2 Instal·lació

Per instal·lar Kibana es pot fer per dues vies diferents. Es pot obtenir el paquet corresponent a la plataforma a la que es vulgui instal·lar o es pot fer per mitjà del seu repositori.

En aquest cas s'utilitzarà la segona manera ja que proporciona una base estàndard d'instal·lació que ens servirà per orientar-nos millor en tot el conjunt de fitxers que comprenen l'eina.

Primer de tot ens descarregarem i instal·larem la Clau de Signatura Pública, en el nostre cas afegida anteriorment durant la instal·lació de Elasticsearch.

S'instal·la Kibana i configurarem el seu inici perquè es produeixi juntament amb l'inici d'arrencada del servidor.

Afegim l'adreça del servidor en el fitxer de configuració i reiniciem el servei.

9.6.3 Connexió de Kibana amb Elasticsearch

Abans de començar a utilitzar Kibana s'ha d'indicar quins índex d'Elasticsearch es voldran explorar.

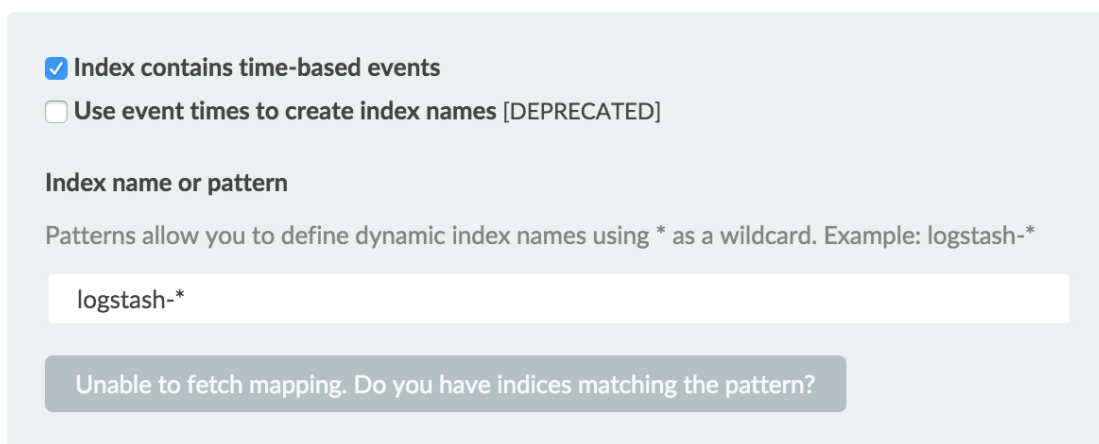
La primera vegada que s'accedeix a Kibana es demanarà que es configuri un *patró d'índex* que coincideixi amb el nom de un o varis dels índex propis.

Configuració d'un patró d'índex

El primer que s'ha de fer és accedir al navegador pel port 5601 per tal d'accedir a la interfície gràfica de Kibana. Si no s'ha canviat la configuració serà <http://localhost:5601>, en cas d'haver fet algun canvi tindrà la forma <http://DOMINIPROPI.com:5601>. Es mostrarà la següent pantalla per configurar el patró.

Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.



Index contains time-based events

Use event times to create index names [DEPRECATED]

Index name or pattern

Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*

logstash-*

Unable to fetch mapping. Do you have indices matching the pattern?

Figura 8- Inserir patrons a Kibana

S'ha d'especificar un patró que coincideixi amb un o varis dels índex d'Elasticsearch propis, com s'ha indicat abans.

Per defecte, Kibana fa el supòsit que s'està treballant amb dades portades a Elasticsearch per Logstash. En aquest cas, que és el nostre, es pot utilitzar **logstash-***, que ve per defecte, com al patró d'índex propi.

Es selecciona el camp índex que conté la data i hora que es vulgui utilitzar per efectuar comparacions basades en el temps. En cas que l'índex no tingués dades que incloguessin el temps s'hauria de desactivar el camp **Index contains time-based events**, però no és el nostre cas.

S'ha de clicar a **Create** per afegir el patró. Aquest primer patró que s'ha afegit automàticament es configurarà per a ser el patró per defecte. Per canviar el patró per defecte en cas de tenir-ne varis, s'ha de fer des de l'apartat **Settings > Indices**.

9.6.4 Configuració

Els apartats més importants de la configuració de Kibana s'explicaran tot seguit per tenir una noció inicial de les opcions que permet modificar aquesta eina.

La part més important de la configuració és on indiquem l'adreça i el port per els quals s'accedirà a l'eina. Es poden deixar per defecte o canviar-los. En el nostre cas farem la primera opció ja que qui accedeix a Kibana és una interfície intermèdia, Scirius, i per tant, no necessitem tenir-hi accés des de l'exterior.

Hem d'indicar la instància de Elasticsearch per a que Kibana sàpiga on ha de fer les peticions de les dades que s'han de mostrar.

Kibana utilitza índex per guardar cerques, visualitzacions i taulells. Si es vol utilitzar un índex propi es pot canviar el valor d'aquesta variable.

En cas de tenir protegit Elasticsearch amb una autenticació bàsica, es pot indicar el nom d'usuari i la paraula clau per tal que Kibana tingui accés a les dades que tracta.

És interessant poder posar un temps màxim de resposta a les peticions. Com que hi ha vegades que les peticions són molt grans, Elasticsearch pot tenir problemes per tractar tota la informació.

Depèn el tipus de peticions que es facin es pot assignar un valor o un altre. Si les peticions són de poques dades, es pot assignar un temps d'espera menor i així controlar més ràpidament si alguna cosa ha fallat mentre que si les peticions són de una gran quantitat de dades lo més interessant és assignar un valor alt ja que de per si tardarà a tractar-les i enviar-les.

Moltes vegades és interessant assignar l'identificador del procés a un fitxer, d'aquesta manera si es vol actuar sobre aquest només s'ha de mirar l'identificador i es sabrà quin procés és.

Una altra característica interessant per els administradors és el poder tenir tots els registres de l'aplicació, ja siguin de caràcter informatiu com indicant que hi ha hagut errors en l'execució. Aquests registres donen la possibilitat d'actuar un cop s'ha produït una fallada ja que et permeten saber què ha fallat i perquè.

9.7 SCIRIUS

9.7.1 Introducció

Scirius és una interfície web dedicada a la gestió dels diferents conjunts de regles de Suricata. Gestiona els fitxers de regles i actualitza els fitxers associats. A més a més permet visualitzar de forma gràfica informació referent a les regles i les alertes generades.

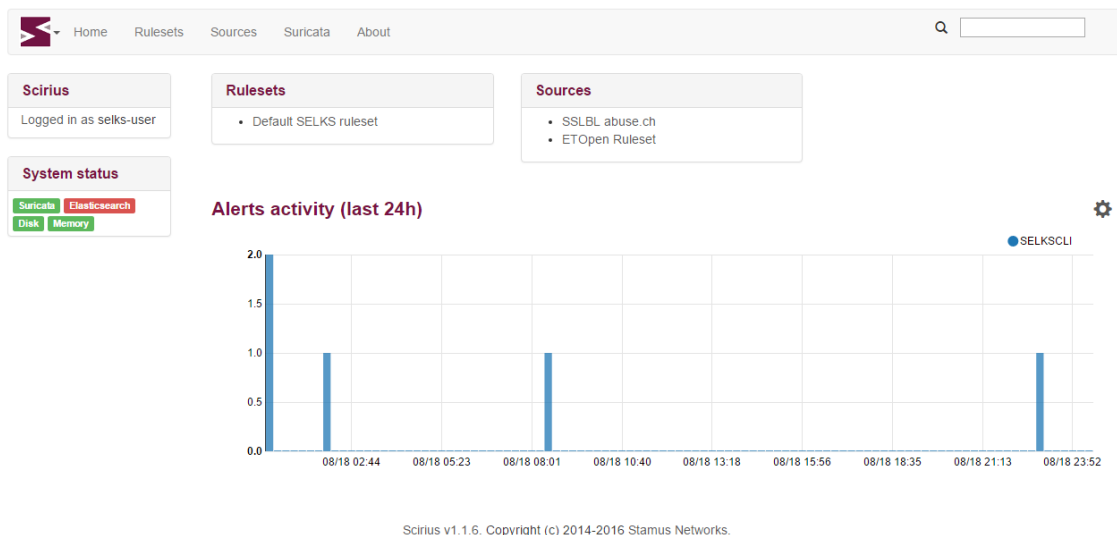


Figura 9- Pàgina inicial Scirius

9.7.2 Instal·lació i configuració

Instal·lació

L'aplicació Scirius es troba escrita en Django, la qual es pot instal·lar com qualsevol altre aplicació d'aquest tipus. El procediment es descriu tot seguit.

Dependències

Scirius utilitza els mòduls de Django següents:

- tables2
- south
- bootstrap3
- requests
- revproxy

La manera més senzilla per a instal·lar aquestes dependències és per mitjà de la utilitat **pip**.

Una de les utilitats que ens ofereix Scirius és un conjunt de Scripts, com ara el *suri_reloader* que gestiona el reinici de Suricata, el qual necessitarà *pyinotify*.

És possible, segons la distribució que s'utilitzi, que es necessiti un GitPython recent.

I també es necessitarà el mòdul respectiu a les bases de dades.

Execució

Ens dirigirem fins al directori on es troba el codi i sincronitzarem la base de dades a utilitzar.

Per defecte es demanarà una autenticació, per tant serà necessari crear un compte amb privilegis d'administrador en el moment en què es demani.

La forma més simple de provar Scirius és executar el servidor de proves de Django.

Configuració de Suricata

Scirius genera un sol fitxer de signatures amb totes les regles activades. Quan s'editi Suricata, s'ha de configurar el directori on es vol que es generi aquest fitxer i els arxius del conjunt de regles associats siguin copiats.

Aquesta eina no tocarà el fitxer de configuració general de Suricata, per tant s'ha d'actualitzar perquè apunti on les dades són configurades per Scirius.

Per interaccionar amb Scirius, es necessita detectar quan el fitxer **/etc/suricata/rules/scirius.reload** és creat. Quan sigui el cas, s'ha de reiniciar o recarregar Suricata i eliminar el fitxer **reload** un cop s'hagi efectuat.

9.7.3 Enllaçar amb Elasticsearch

En el nostre cas, que utilitzem Suricata amb enregistrament Eve i utilitzem la utilitat d'Elasticsearch, es pot obtenir informació sobre signatures i també informació sobre Suricata.

Per configurar la connexió amb Elasticsearch, es pot editar el fitxer **settings.py** o crear un **local_settings.py** dins el directori de **scirius** per tal de configurar la característica. Elasticsearch s'activarà si la variable **USE_ELASTICSEARCH** s'assigna a **True** en el fitxer anomenat anteriorment.

9.7.4 Enllaçar amb Kibana

En el cas d'utilitzar l'eina de visualització Kibana, és possible obtenir enllaços dels diferents taulers clicant sobre la icona superior esquerra.

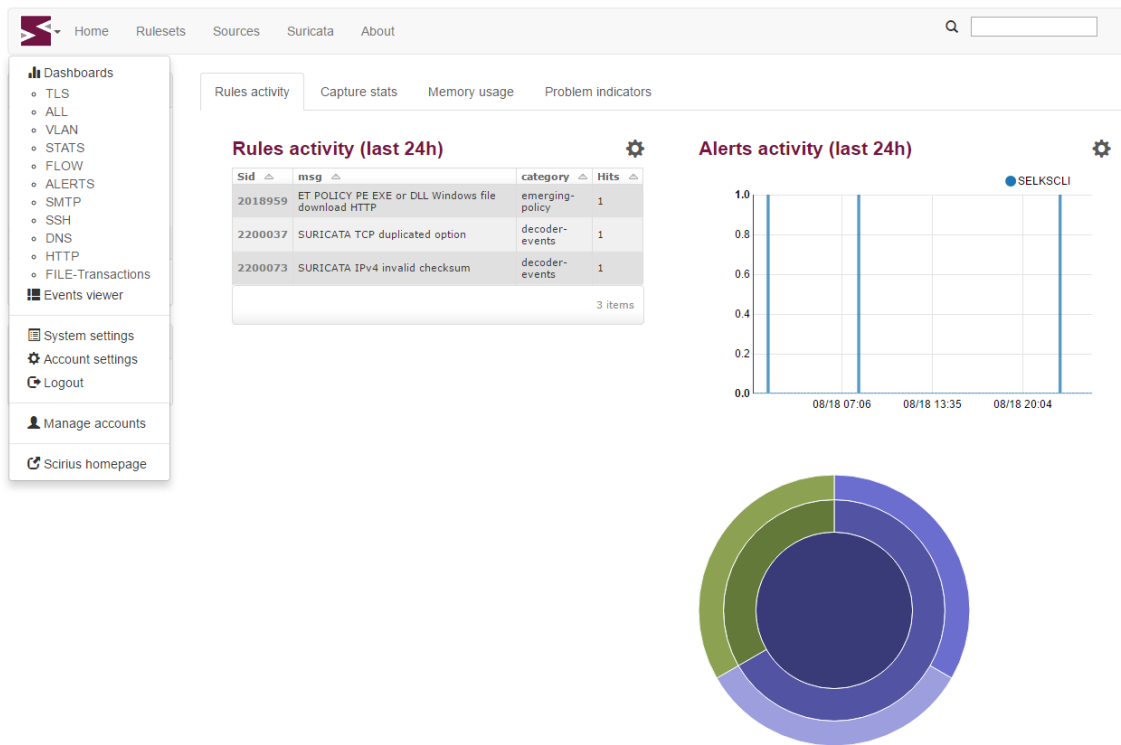


Figura 10- Pàgina de Suricata a Scirius

Per activar aquesta característica, s'ha d'editar el fitxer de configuracions **local_settings.py**.

10 IMPLANTACIÓ I RESULTATS

La implantació dels requeriments del projecte s'ha realitzat de forma pautada, de tal manera que quan s'ha acabat el desenvolupament d'una tasca, aquesta s'implantava i es comprovaven els resultats.

Tot i que la millor opció hagués sigut poder implantar el desenvolupament realitzat en un escenari de producció, no ens ha estat possible. Això és degut a que el desenvolupament ha estat en forma de col·laboració amb un projecte ja començat i s'ha seguit la metodologia que s'hi utilitzava. Per poder tenir els canvis realitzats a la nova versió del projecte, primer ha de passar un seguit de controls i normalitzacions per part dels coordinadors del projecte.

Per poder implantar i veure els resultats de les tasques, s'ha utilitzat un escenari local.

10.1 REGLA

S'ha modificat lleugerament la pàgina principal d'una regla, afegint tres opcions a la barra lateral *Actions* referents a les modificacions que podem efectuar sobre una regla. Quan es selecciona una d'aquestes opcions, se'ns dirigeix cap a una pàgina per efectuar-la.

The screenshot shows the Scirius web interface. At the top, there is a navigation bar with links for Home, Sources, Rulesets, Suricata, and About. A search bar is located on the right. The main content area displays the rule configuration for 'ET SNMP Cisco Non-Trap PDU request on SNMPv1 trap port'. On the left, there is a sidebar with the rule ID '2002880', revision '8', and available status 'True'. Below this, there is an 'Action' menu with options: Disable rule, Enable rule, Threshold rule, Suppress rule, Delete generated alerts, Alert rule, Allow rule, and Drop rule. The 'Path' is listed as 'ET Rules / emerging-snmp'. At the bottom left, there is a 'System status' section showing 'Suricata' and 'Elasticsearch' as active, and 'Disk' and 'Memory' as healthy. The main content area shows 'Rules info' and three charts: 'Hits by host (last 24h)', 'Source IP (last 24h)', and 'Destination IP (last 24h)', all of which show 'No data for period.'. A 'References' section on the right lists CVE: 2004-0714, Bugtraq: 10186, and a URL to the rule documentation. At the bottom, the footer text reads 'Scirius v1.1.10. Copyright (c) 2014-2016 Stamus Networks.'

Figura 11- Regla a Scirius

10.1.1 Visualització d'una regla

Encara en la pàgina d'una regla, hem canviat la taula en què es visualitzen els *Rulesets* per tal que es pugui veure quina acció realitzarà la regla en cas de coincidir quan pertanyi al *Ruleset* indicat.

Status in rulesets

| Name <small>▲</small> | Status in ruleset <small>▲</small> | Threshold <small>▲</small> | Operational status <small>▲</small> | Action in ruleset <small>▲</small> |
|-----------------------|------------------------------------|----------------------------|-------------------------------------|------------------------------------|
| Prova | Active | No | Unknown | alert |
| Ruleset 1 | Active | No | Unknown | allow |
| Ruleset2 | Active | No | Unknown | drop |

3 items

Figura 12- Informació d'una regla a Scirius

S'hi ha afegit una columna extra que ens proporciona la informació necessària.

10.1.2 Modificació d'una regla

Quan es selecciona una de les opcions *Alert*, *Allow* o *Drop* del panell lateral de la pàgina, se'ns dirigeix a una de nova per a realitzar l'acció seleccionada.

En aquesta pàgina podem seleccionar sobre quins *Rulesets* es vol realitzar l'acció. Només es mostraran els *Rulesets* als quals pertany la regla que es vol modificar.

Rule 2002880

System status
Suricata Elasticsearch
Disk Memory

Allow rule 2002880 in

Rulesets

Prova

Ruleset 1

Ruleset2

Allow rule

Figura 13- Modificar una regla a Scirius

10.2 CATEGORIA

A la pàgina d'una categoria s'hi ha realitzat modificacions semblants que les mostrades anteriorment. S'ha afegit quatre opcions al panell lateral referents a les accions que es poden realitzar:

- Alert category
- Allow category
- Drop category
- Edit rules

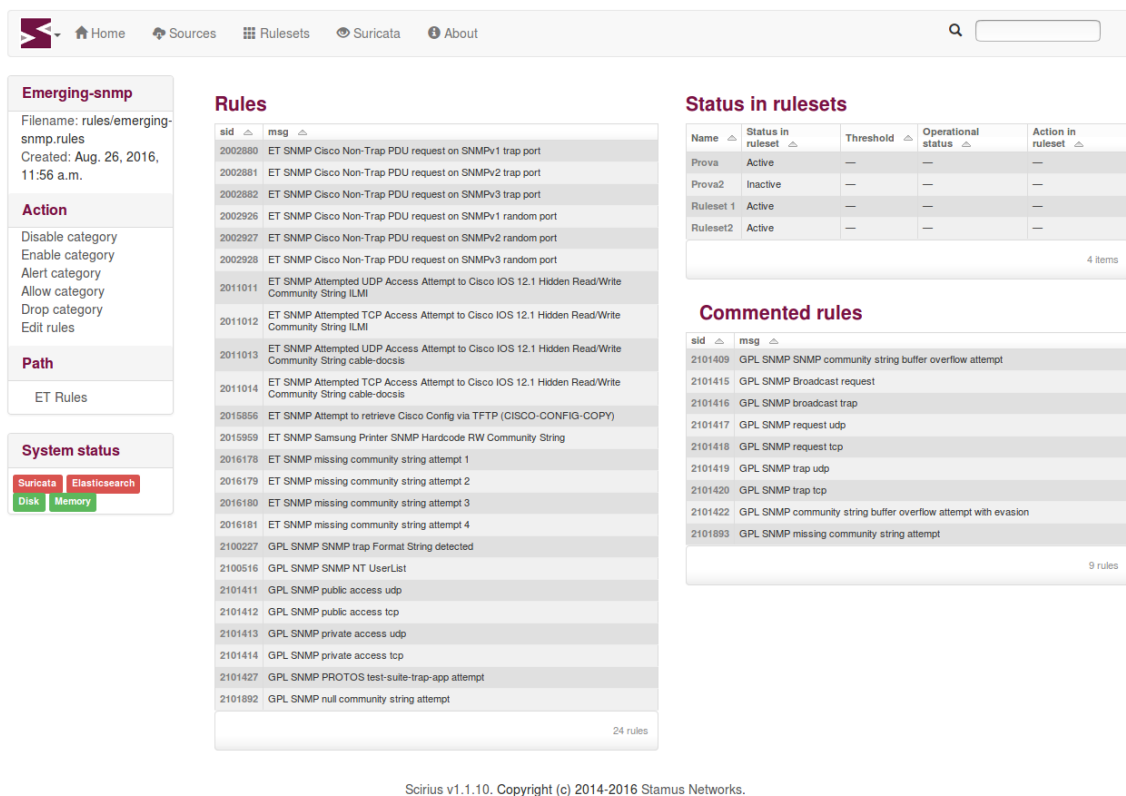


Figura 14- Pàgina d'una regla a Scirius

10.2.1 Modificació de totes les regles d'una categoria

Quan es selecciona una de les opcions referents a realitzar una acció sobre la categoria, com ara *Alert*, *Allow* o *Drop*, es tractarà la petició de forma semblant.

Se'ns dirigirà a una pàgina nova, on se'ns mostraran diferents *Rulesets* amb els quals està relacionada la categoria. Es podran seleccionar múltiples amb la mateixa petició, enviant-la clicant al botó que es mostra sota el llistat.

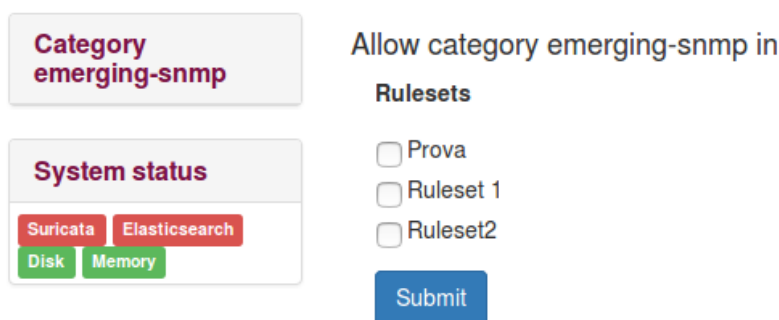


Figura 15- Modificar categoria a Scirius

10.2.2 Modificació de múltiples regles

Per a poder seleccionar quines regles són les que es volen modificar d'una categoria concreta, haurem d'utilitzar l'opció de *Edit rules* de tal forma que ens enviarà a una nova pàgina de d'on podrem realitzar la petició desitjada.

En aquesta pàgina hi trobem primer de tot un llistat amb tots els *Rulesets* que es troben relacionats amb la categoria que es vol modificar. Es farà la petició de modificació per cadascun dels diferents *Rulesets* d'aquesta llista.

Edit rules in emerging-snmp

Choose an operation in the actions below:

Choose rulesets

Rulesets

- Prova
- Ruleset 1
- Ruleset2

Figura 16- Seleccionar Rulesets de varies regles

Després ens trobem amb el llistat de totes les regles que formen la categoria. Es podran seleccionar múltiples regles que es vulguin modificar.

Choose rules

| <input type="checkbox"/> | sid | msg |
|--------------------------|---------|---|
| <input type="checkbox"/> | 2002880 | ET SNMP Cisco Non-Trap PDU request on SNMPv1 trap port |
| <input type="checkbox"/> | 2002881 | ET SNMP Cisco Non-Trap PDU request on SNMPv2 trap port |
| <input type="checkbox"/> | 2002882 | ET SNMP Cisco Non-Trap PDU request on SNMPv3 trap port |
| <input type="checkbox"/> | 2002926 | ET SNMP Cisco Non-Trap PDU request on SNMPv1 random port |
| <input type="checkbox"/> | 2002927 | ET SNMP Cisco Non-Trap PDU request on SNMPv2 random port |
| <input type="checkbox"/> | 2002928 | ET SNMP Cisco Non-Trap PDU request on SNMPv3 random port |
| <input type="checkbox"/> | 2011011 | ET SNMP Attempted UDP Access Attempt to Cisco IOS 12.1 Hidden Read/Write Community String ILM1 |
| <input type="checkbox"/> | 2011012 | ET SNMP Attempted TCP Access Attempt to Cisco IOS 12.1 Hidden Read/Write Community String ILM1 |
| <input type="checkbox"/> | 2011013 | ET SNMP Attempted UDP Access Attempt to Cisco IOS 12.1 Hidden Read/Write Community String cable-docst |
| <input type="checkbox"/> | 2011014 | ET SNMP Attempted TCP Access Attempt to Cisco IOS 12.1 Hidden Read/Write Community String cable-docst |
| <input type="checkbox"/> | 2015856 | ET SNMP Attempt to retrieve Cisco Config via TFTP (CISCO-CONFIG-COPY) |

Figura 17- Llistat de regles per modificar

Finalment hi trobem els tres botons que fan referència a les tres modificacions possibles que pot rebre una regla.

| | | |
|--------------------------|---------|--|
| <input type="checkbox"/> | 2101409 | GPL SNMP SNMP community string buffer overflow attempt |
| <input type="checkbox"/> | 2101411 | GPL SNMP public access udp |
| <input type="checkbox"/> | 2101412 | GPL SNMP public access tcp |
| <input type="checkbox"/> | 2101413 | GPL SNMP private access udp |
| <input type="checkbox"/> | 2101414 | GPL SNMP private access tcp |
| <input type="checkbox"/> | 2101415 | GPL SNMP Broadcast request |
| <input type="checkbox"/> | 2101416 | GPL SNMP broadcast trap |

Page 1 of 2 25 of 33 rules

Figura 18- Accions per modificar múltiples regles

10.3 FONT

Com en els casos anteriors, s'ha afegit una opció per a modificar múltiples categories d'una font.

ET Rules

Created: Aug. 26, 2016, 11:56 a.m.
Updated: Aug. 26, 2016, 11:56 a.m.

Action

Changelog
Update
Edit
Delete
Edit categories

System status

Suricata Elasticsearch
Disk Memory

Categories

| name <small>▲</small> | descr <small>▲</small> | date created <small>▲</small> |
|----------------------------|------------------------|-------------------------------|
| emerging-snmp | — | 08/26/2016 11:56 a.m. |
| emerging-icmp | — | 08/26/2016 11:56 a.m. |
| emerging-user_agents | — | 08/26/2016 11:56 a.m. |
| emerging-web_specific_apps | — | 08/26/2016 11:56 a.m. |
| emerging-inappropriate | — | 08/26/2016 11:56 a.m. |
| emerging-activex | — | 08/26/2016 11:56 a.m. |
| emerging-icmp_info | — | 08/26/2016 11:56 a.m. |
| emerging-smtp | — | 08/26/2016 11:56 a.m. |
| emerging-dos | — | 08/26/2016 11:56 a.m. |
| drop | — | 08/26/2016 11:56 a.m. |
| emerging-web_client | — | 08/26/2016 11:56 a.m. |
| emerging-malware | — | 08/26/2016 11:56 a.m. |
| dshield | — | 08/26/2016 11:56 a.m. |
| emerging-attack_response | — | 08/26/2016 11:56 a.m. |
| emerging-imap | — | 08/26/2016 11:56 a.m. |
| decoder-events | — | 08/26/2016 11:56 a.m. |
| emerging-voip | — | 08/26/2016 11:56 a.m. |
| tls-events | — | 08/26/2016 11:56 a.m. |
| emerging-deleted | — | 08/26/2016 11:56 a.m. |
| botcc.portgrouped | — | 08/26/2016 11:56 a.m. |
| emerging-worm | — | 08/26/2016 11:56 a.m. |
| emerging-rpc | — | 08/26/2016 11:56 a.m. |
| emerging-current_events | — | 08/26/2016 11:56 a.m. |
| emerging-policy | — | 08/26/2016 11:56 a.m. |
| emerging-tftp | — | 08/26/2016 11:56 a.m. |

Page 1 of 3
25 of 53 categories

Figura 19- Pàgina d'una categoria a Scirius

10.3.1 Modificació de totes les regles de múltiples categories

En aquest cas, quan es selecciona l'opció *Edit categories*, se'ns envia a una nova pàgina per a realitzar la petició de modificació.

Com en el cas anterior, la pàgina consta d'un llistat principal dels *Rulesets* que es troben relacionats amb totes les categories, amb l'opció de seleccionar-ne diferents per a realitzar-hi la modificació sobre més d'un.

Hi trobem un llistat amb totes les categories que formen la font, les quals es poden seleccionar de forma múltiple.

Edit categories in ET Rules

Choose an operation in the actions below:

Choose rulesets

Rulesets

- Prova
- Ruleset 1
- Ruleset2

Choose categories

| <input type="checkbox"/> | name <small>△</small> | descr <small>△</small> | date created <small>△</small> |
|--------------------------|----------------------------|------------------------|-------------------------------|
| <input type="checkbox"/> | emerging-snmp | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-icmp | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-user_agents | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-web_specific_apps | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-inappropriate | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-activex | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-icmp_info | — | 08/26/2016 11:56 a.m. |

Figura 20- Llistat de Rulesets i categories

Finalment trobarem al final de la pàgina els tres botons per a realitzar la petició de modificació de totes les regles de les categories seleccionades sobre els *Rulesets* indicats.

| | | | |
|--------------------------|-------------------------|---|-----------------------|
| <input type="checkbox"/> | botcc.portgrouped | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-worm | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-rpc | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-current_events | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-policy | — | 08/26/2016 11:56 a.m. |
| <input type="checkbox"/> | emerging-tftp | — | 08/26/2016 11:56 a.m. |

Page 1 of 3 25 of 53 categories

Figura 21- Accions per modificar múltiples categories

11 CONCLUSIONS

A l'inici d'aquest projecte em vaig marcar un objectiu a assolir que centraven la planificació del treball.

Com s'ha exposat ja en la introducció, els objectius inicials es van veure alterats i això va afectar la planificació del que restava de treball. Tot i que el temps era just, vaig decidir agafar com a base el projecte trobat i millorar-ne les funcionalitats.

Després de posar-me en contacte amb els creadors d'aquest projecte, i decidir quina funcionalitat afegiria al projecte, vaig començar a desenvolupar-la.

Comprovant els objectius finals marcats durant el replantejament del projecte, podem assegurar que s'han complert.

S'ha pogut estudiar i implementar Suricata com a eina IDS, la qual és una de les noves tecnologies que es troben actualment en aquest sector i està desenvolupada en codi lliure. Ofereix unes característiques molt interessants que els seus competidors encara no han aconseguit igualar.

Per visualitzar totes les dades obtingudes i analitzades per Suricata, s'ha utilitzat el conjunt d'eines Elasticsearch, Logstash i Kibana. Aquest conjunt d'eines ofereix una capacitat de tractat, desplaçament i visualització de les dades molt gran. A més a més, compta amb una gran comunitat ja que es pot utilitzar combinats amb moltes tecnologies diferents.

En els objectius inicials es va decidir dissenyar una interfície web per a gestionar l'eina IDS seleccionada, en el nostre cas Suricata. Després del replantejament del projecte es va desestimar aquest objectiu ja que la utilització de Scirius ens oferia moltes més funcionalitats de les que jo hagués pogut desenvolupar.

Es va canviar aquest objectiu per a utilitzar Scirius com a interfície web base i millorar-ne les seves funcionalitats. Això va provocar que hagués de reciclar-me en el llenguatge de Python i aprendre el funcionament del *Framework* Django.

S'han obtingut els resultats finals esperats, amb tots els objectius proposats assolits.

La única part negativa és no haver pogut realitzar les proves en un escenari d'una empresa mitjana real amb la informació fiable que es generaria per analitzar els resultats. Tot i això, ho deixo com a treball futur amb la intenció de realitzar-ho.

Personalment aquest projecte m'ha fet créixer, des del principi he estat aprenent tecnologies noves. Tot i els entrebancs trobats i els girs que ha fet el projecte, m'ha servit per aprendre a utilitzar un llenguatge i un *Framework* nou i unes eines diferents.

També m'ha agradat molt treballar com a col·laborador [14] d'un projecte *Open Source*, veient la metodologia, la forma de comunicar-se, de desenvolupar i el bon ambient amb el que m'he trobat a l'hora de compartir idees per a millores de la seva eina.

Com a reflexió final, s'han assolit els objectius i he après molt realitzant aquest treball de fi de grau.

12 TREBALL FUTUR

El treball futur el podem separar en dos apartats molt diferenciats, el que fa referència a la instal·lació i utilització de l'eina de captura de paquets i el que fa referència al desenvolupament de l'eina de gestió utilitzada.

El que m'hagués agradat fer si hagués tingut temps hagués sigut realitzar proves amb l'eina en dos entorns suficientment diferents.

El primer entorn seria aplicar el projecte en una empresa mitjana, amb més d'una xarxa local i amb connexions des de l'exterior a serveis interns de l'empresa. En aquest cas es podrien dissenyar un conjunt de regles que controlessin un seguit de paràmetres, controlar les connexions internes de les xarxes locals, controlar les connexions exteriors i controlar tots els serveis que oferís l'empresa. En aquest cas, interessaria provar Suricata amb la funció d'IPS, ja que en cas de rebre atacs des de l'exterior es voldria poder-los bloquejar de forma automàtica.

El segon entorn seria una petita empresa amb una sola xarxa local i sense rebre connexions des de l'exterior. En aquest cas no faria falta utilitzar Suricata amb el mode IPS ja que en ser un escenari molt més tancat i fàcil de protegir, no és probable rebre els atacs com en l'escenari anterior.

El conjunt de regles dissenyat seria encarat més a problemes interns que en possibles amenaces externes.

El treball futur per a la millora de l'eina de gestió el podríem separar en dos blocs, el de millores de funcionalitats ja existents i el de creació de noves funcionalitats.

Com a millores de les funcionalitats existents, tinc pensat realitzar uns canvis en el disseny en què es canvia l'acció de les regles que pertanyen a una categoria en concret, ja sigui en una categoria individual o quan es fa de forma múltiple.

El que vull millorar és el fet que quan s'actualitza una categoria pot ser que a aquesta se li assignin noves regles. Si es produís aquest cas, les noves regles tindrien l'acció per defecte que els hi ve donada i no l'acció que se li ha assignat a la categoria en qüestió.

Fer aquest canvi implica canviar l'estructura i el disseny de tot el bloc de regles del projecte, el que significa una feina important i bastant llarga en qüestió de temps.

Una altra funcionalitat que seria interessant incorporar a llarg termini, ja que significaria un grau de feina molt elevat, seria incloure notificacions a temps real a les alertes que rep el sistema. D'aquesta manera, l'administrador del sistema estaria al corrent de totes les accions que passessin a temps real i milloraria dràsticament el temps de resposta en cas de tenir una fallada important.

Una de les característiques que seria important incorporar és que aquestes notificacions es poguessin assignar per regles individuals, o per categories senceres. Així només es rebrien les notificacions que es volguessin.

Una altra funcionalitat interessant per a incorporar al projecte seria posar un temps màxim d'acció en les accions automàtiques que realitzen les regles. És a dir, si Suricata ha detectat una alerta i la regla a la que fa referència té assignada l'acció de *Drop*, és a dir, bloquejar, que no ho faci de forma indefinida sinó ho bloquegi durant un espai de temps. D'aquesta manera obtindríem un sistema molt més independent de l'administrador, ja que per a molt problemes puntuals es solucionaria sense la seva actuació amb aquesta característica nova.

Un pas més dins aquesta funcionalitat seria que cada cop que es detectés una alerta en una mateixa regla, el temps de bloqueig s'anés augmentant, de tal manera que si és un problema persistent, s'acabarà bloquejant de forma indefinida fins a l'actuació de l'administrador.

Totes aquestes millores descrites són idees per al projecte Scirius, amb el qual tinc la intenció de seguir col·laborant [14]. Després d'aconseguir millorar-lo per a que fos capaç de configurar Suricata en el mode IPS, i per tant, afegir valor en el concepte d'aquest projecte, m'agradaria implementar les diferents millores que tinc pensades de forma conjunta amb els coordinadors de Scirius.

Finalment, també voldria fer un manual d'usuari amb totes les funcionalitats que ofereix el projecte. La idea seria fer-lo de forma gràfica amb les explicacions i comentaris en anglès, ja que serviria com a manual per el projecte en el qual he col·laborat i el podrien incloure en la seva pròpia wiki.

13 BIBLIOGRAFIA

- [1] Stamus Network, 30 08 2016. [En línia]. Available: <https://github.com/StamusNetworks/SELKS>.
- [2] Talos, «Snort,» Snort, [En línia]. Available: <https://www.snort.org/talos>. [Últim accés: 15 04 2016].
- [3] J. Esler, «Snort,» Cisco, 22 2 2016. [En línia]. Available: <http://blog.snort.org/2011/02/snort-shared-object-rules.html> . [Últim accés: 12 04 2016].
- [4] J. Esler, «amazonaws,» [En línia]. Available: https://s3.amazonaws.com/snort-org/www/assets/227/Open_Source_Community_Meeting.pdf. [Últim accés: 04 2016].
- [5] OISF, «Suricata,» OISF, [En línia]. Available: <https://suricata-ids.org/>. [Últim accés: 11 04 2016].
- [6] OISF, «Suricata-ids,» Suricata, [En línia]. Available: <https://idsips.wordpress.com/about/oisf/>. [Últim accés: Març 2016].
- [7] «oisf,» [En línia]. Available: <https://oisf.net/consortium-members/> . [Últim accés: Març 2016].
- [8] OISF, «oisf,» Open Info Sec Foundation, [En línia]. Available: <https://redmine.openinfosecfoundation.org/projects/suricata/roadmap> . [Últim accés: Agost 2016].
- [9] Snort, «Snort,» Cisco, [En línia]. Available: <https://www.snort.org/snort3>. [Últim accés: Març 2016].
- [10] D. Nanni, «Xmodulo,» 3 Setembre 2015. [En línia]. Available: <http://xmodulo.com/install-suricata-intrusion-detection-system-linux.html> . [Últim accés: Abril 2016].
- [11] OISF, «openinfosecfoundation,» Open Info Sec Foundation, [En línia]. Available: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Basic_Setup . [Últim accés: Abril 2016].
- [12] OISF, «openinfosecfoundation,» Open Info Sec Foundation, [En línia]. Available: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Command_Line_Options . [Últim accés: Agost 2016].
- [13] PacNOG, «Network Startup Resource Center,» Juliol 2015. [En línia]. Available: <https://web.nsrc.org/workshops/2015/pacnog17-ws/raw-attachment/wiki/Track2Agenda/ex-suricata-rules.htm> . [Últim accés: Agost 2016].

- [14 Stamus Networks, «stamus-networks,» Stamus Networks, [En línia]. Available:] <https://www.stamus-networks.com/open-source/>. [Últim accés: Agost 2016].
- [15 «GitHub,» GitHub, [En línia]. Available: <https://github.com>. [Últim accés: 2016].]
- [16 «Stackoverflow,» Stackoverflow, [En línia]. Available: <https://stackoverflow.com>. [Últim accés: 2016].]
- [17 «python,» Python Software Foundation, [En línia]. Available:] <https://docs.python.org/2/tutorial/appetite.html>. [Últim accés: Agost 2016].
- [18 J. Ullrich, «Sans,» 29 Maig 2013. [En línia]. Available:] <https://isc.sans.edu/diary/Running+Snort+on+VMWare+ESXi/15899>. [Últim accés: Març 2016].
- [19 N. Dietrich, «Sublime Robots,» 28 Desembre 2014. [En línia]. Available:] <http://sublimerobots.com/2014/12/promiscuous-mode-esxi/>. [Últim accés: Març 2016].
- [20 «Sans,» Sans Institute, [En línia]. Available: <https://www.sans.org/security-resources/idfaq/>. [Últim accés: Març 2016].]
- [21 J. Schreiber, «Alienvault,» Alienvault, 13 Gener 2014. [En línia]. Available:] <https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview>. [Últim accés: Febrer 2016].
- [22 SearchSecurity, «Search Security,» Tech Target, [En línia]. Available:] <http://searchsecurity.techtarget.com/tutorial/Intrusion-detection-and-prevention-learning-guide>. [Últim accés: Març 2016].
- [23 UpGuard, «Upguard,» [En línia]. Available: <https://www.upguard.com/articles/top-free-network-based-intrusion-detection-systems-ids-for-the-enterprise>. [Últim accés: Abril 2016].
- [24 P. Manev, «Pevma,» Pevma, 26 Març 2014. [En línia]. Available:] https://pevma.blogspot.com.es/2014/03/suricata-and-grand-slam-of-open-source_26.html. [Últim accés: Maig 2016].
- [25 A. Lahmadi, «HAL,» 5 Octubre 2015. [En línia]. Available: <https://hal.inria.fr/hal-01212015/document>. [Últim accés: Maig 2016].]
- [26 M. Anicas, «Digital Ocean,» Digital Ocean, 10 Març 2015. [En línia]. Available:] <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-14-04>. [Últim accés: Juny 2016].

- [27 Suricata, «openinfosecfoundation,» Open Info Sec Foundation, [En línia]. Available:
] https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Rules.
[Últim accés: Agost 2016].

14 ÍNDEX DE FIGURES I TAULES

14.1 ÍNDEX DE FIGURES

| | |
|---|----|
| Figura 1- Planificació Inicial | 8 |
| Figura 2- Planificació Final | 8 |
| Figura 3- BASE..... | 28 |
| Figura 4- Snorby..... | 28 |
| Figura 5- Kibana | 30 |
| Figura 6- Diagrama de classes Scirius/rules inicial..... | 42 |
| Figura 7- Diagrama de classes Scirius/rules final..... | 46 |
| Figura 8- Inserir patrons a Kibana..... | 57 |
| Figura 9- Pàgina inicial Scirius..... | 59 |
| Figura 10- Pàgina de Suricata a Scirius..... | 61 |
| Figura 11- Regla a Scirius..... | 62 |
| Figura 12- Informació d'una regla a Scirius | 63 |
| Figura 13- Modificar una regla a Scirius | 63 |
| Figura 14- Pàgina d'una regla a Scirius..... | 64 |
| Figura 15- Modificar categoria a Scirius..... | 64 |
| Figura 16- Seleccionar Rulesets de varies regles | 65 |
| Figura 17- Llistat de regles per modificar..... | 65 |
| Figura 18- Accions per modificar múltiples regles..... | 65 |
| Figura 19- Pàgina d'una categoria a Scirius | 66 |
| Figura 20- Llistat de Rulesets i categories | 67 |
| Figura 21- Accions per modificar múltiples categories | 67 |
| Figura 22- Planificació inicial ampliada | 75 |
| Figura 23- Planificació final ampliada | 76 |

14.2 ÍNDEX DE TAULES

| | |
|--|----|
| Taula 1- Comparativa entre IDSs | 23 |
| Taula 2- Comparativa d'eines de visualització..... | 31 |
| Taula 3- Comparativa IDEs | 38 |

15 ANNEXOS

15.1 PLANIFICACIÓ INICIAL

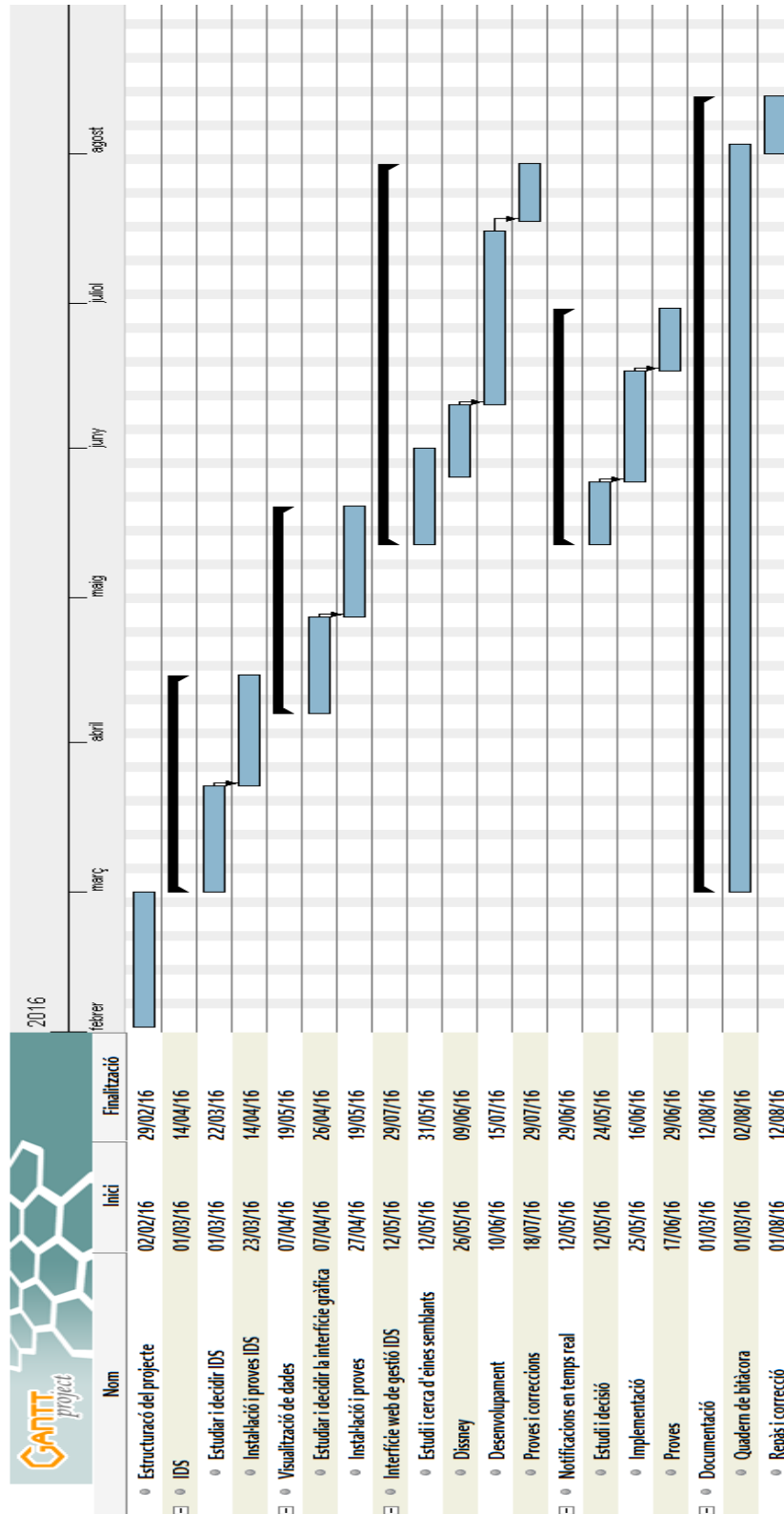


Figura 22- Planificació inicial ampliada

15.2 PLANIFICACIÓ FINAL

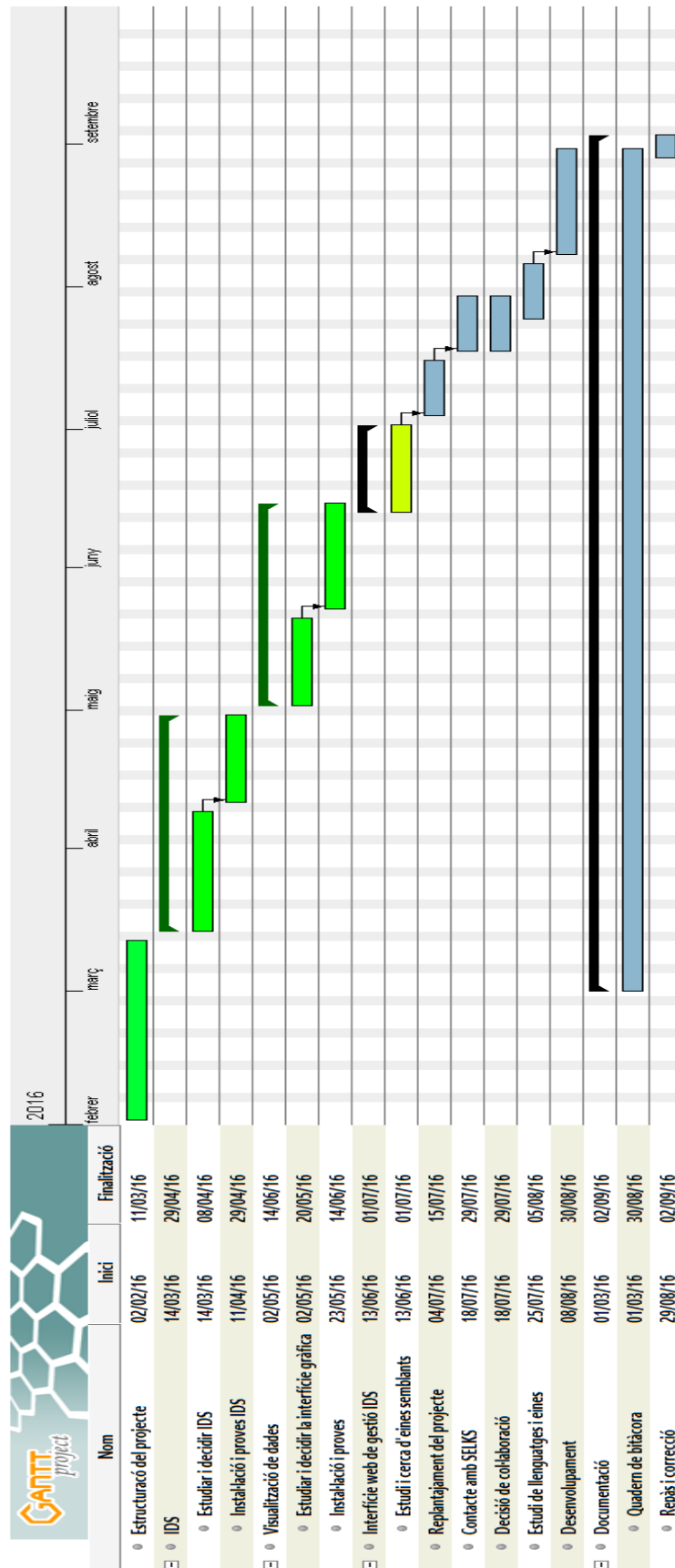


Figura 23- Planificació final ampliada

15.3 CREACIÓ DE LA MÀQUINA VIRTUAL

Per crear la màquina virtual

- Anar a **File** → **New** → **Virtual Machine** [CLICK]

Com que es voldrà assignar uns valors personalitzats a la màquina per tal que s'adeqüi a les nostres necessitats

- Seleccionar **Custom** [SEGÜENT]

Indicarem el Sistema Operatiu que es voldrà instal·lar a la nostra màquina

- Seleccionar **Linux** i **Ubuntu Linux 64-bits** [SEGÜENT]

Assignarem el màxim de processadors disponibles ja que és un punt clau en el rendiment i funcionament del sistema

- Seleccionar màxim de **Number of virtual sockets** [SEGÜENT]

S'afegirà dues interfícies de xarxa ja que se'n necessitarà una per a la gestió del sistema i una altra que rebí les dades que ha de capturar

- Seleccionar **2** en el nombre de **NICs**
- Seleccionar **VM Network** com a primera interfície
- Seleccionar **Promiscuous** com a segona interfície [SEGÜENT]

S'assignarà una quantitat important de disc a la màquina ja que guardarà tots els registres de les captures que faci el sistema i és una gran quantitat d'espai. Tot i això, es podrà ampliar la capacitat de disc assignat a la màquina un cop acabada la instal·lació.

- Seleccionar la capacitat de disc que es vol assignar al sistema [SEGÜENT]

Se'ns mostrarà un resum de les característiques assignades al sistema per poder comprovar que no hi ha hagut cap equivocació en el procés.

- [FINALITZAR]

15.4 INSTAL·LACIÓ DEL SISTEMA OPERATIU

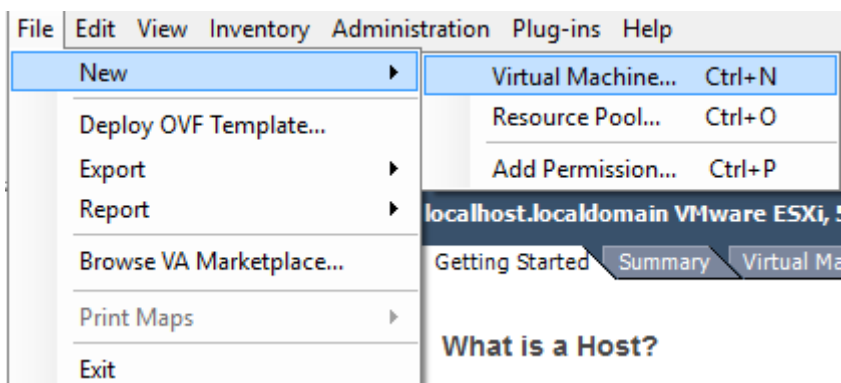
Es mostrarà pas a pas com es crea la màquina virtual i s'hi instal·la el Sistema Operatiu amb suport gràfic d'imatges per a un millor seguiment.

15.4.1 Creació de la màquina virtual

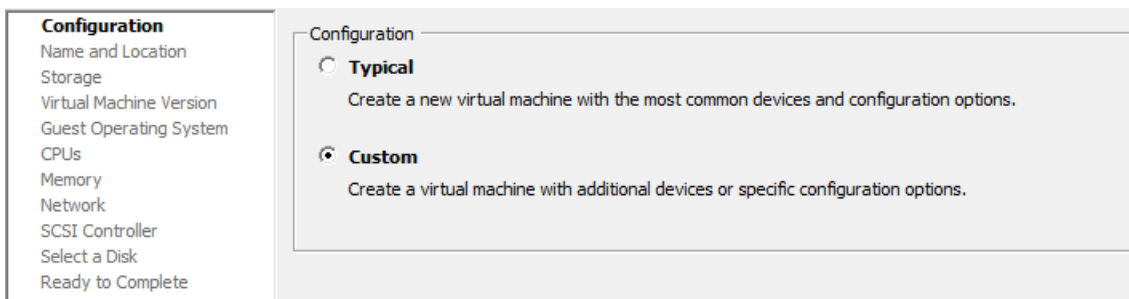
Es crearà una màquina virtual amb una configuració personalitzada, el que ens permetrà modelar-la més adequadament a les nostres necessitats.

Per crear la màquina virtual

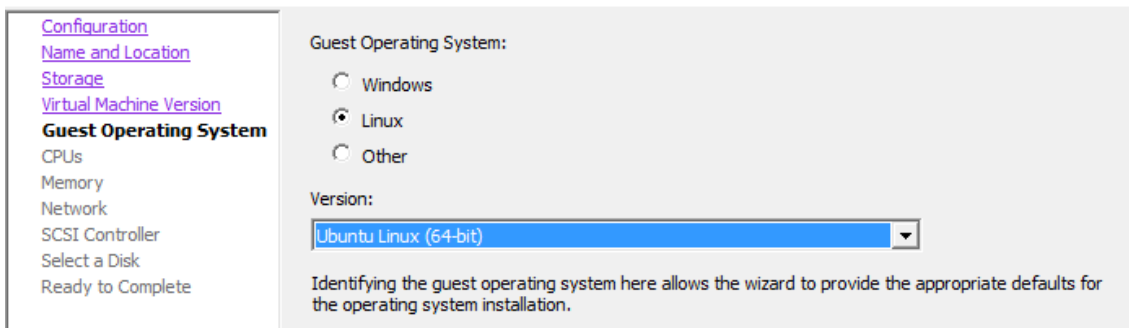
- Anar a **File** → **New** → **Virtual Machine** [CLICK]



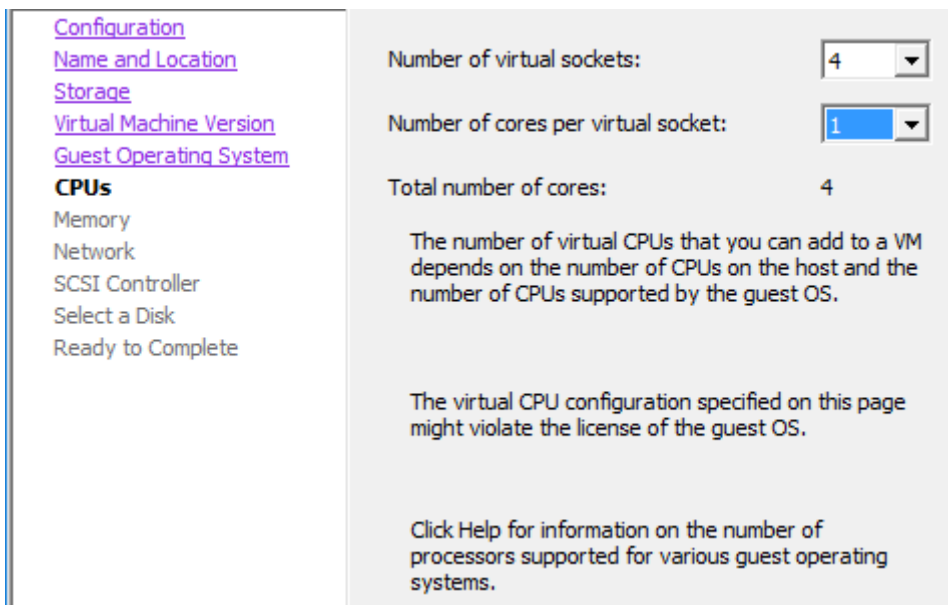
- Seleccionar **Custom** [SEGÜENT]



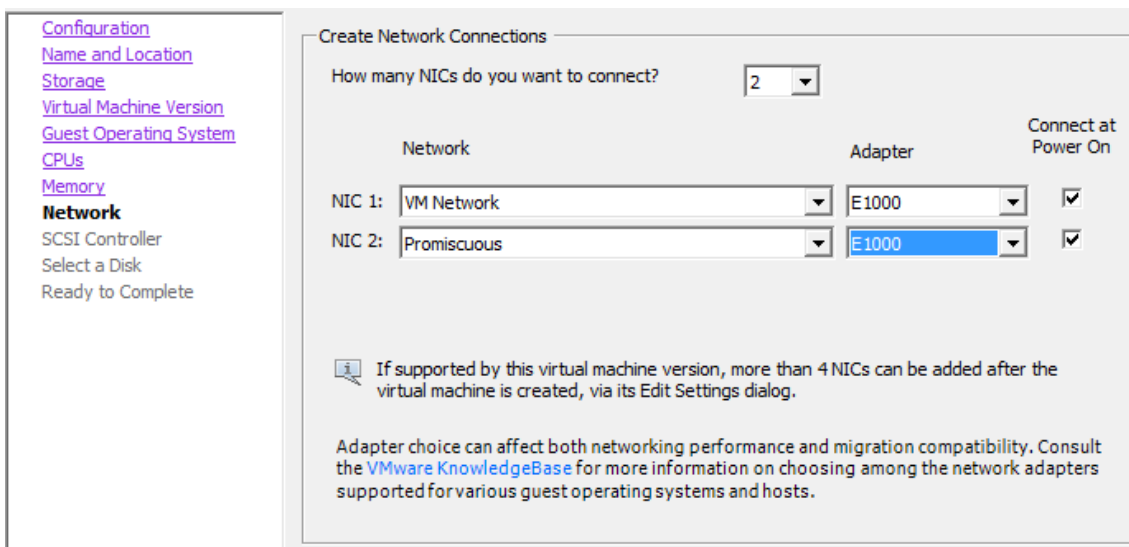
- Seleccionar **Linux i Ubuntu Linux 64-bits** [SEGÜENT]



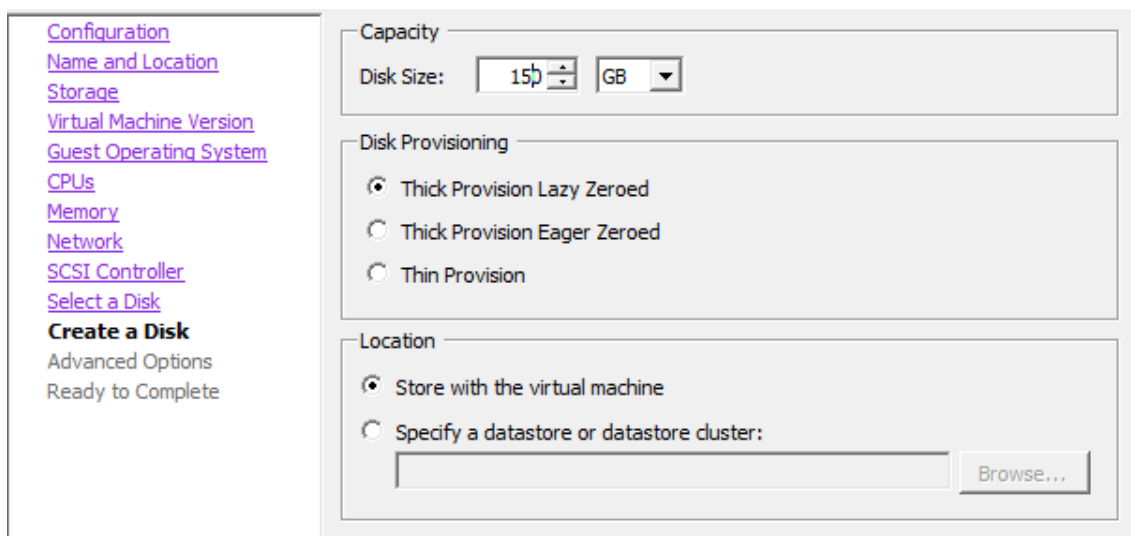
- Seleccionar màxim de **Number of virtual sockets** [SEGÜENT]



- Seleccionar **2** en el nombre de **NICs**
- Seleccionar **VM Network** com a primera interfície
- Seleccionar **Promiscuous** com a segona interfície [SEGÜENT]



- Seleccionar la capacitat de disc que es vol assignar al sistema [SEGÜENT]



- [FINALITZAR]

15.4.2 Instal·lació del Sistema Operatiu

En cas de fer la instal·lació com a una màquina virtual dins un Sistema VMWare, haurem de fer un pas previ a la pròpia instal·lació del SO.

Ens haurem de situar a la màquina creada i engegar-la. Un cop es trobi funcionant ens situarem a la pestanya **Console** que trobem a la dreta del panell.

- Anar a **CD/DVD drive 1** → **Connect to ISO imatge on local disk** [CLICK]

Seleccionarem el fitxer ISO que es vol instal·lar, en aquest cas el fitxer del SO Ubuntu Server. Per descarregar-lo es pot fer des de <http://www.ubuntu.com/download/server>

Amb el fitxer seleccionat, ja podem començar amb la instal·lació pròpia del SO.

Els primers passos es configura la regió, idioma i l'usuari administrador del sistema.

1. Seleccionar **English** [ENTER]
2. Seleccionar **Install Ubuntu Server** [ENTER]
3. Seleccionar **English** [ENTER]
4. Seleccionar **Other** [ENTER]
5. Seleccionar **Europe** [ENTER]
6. Seleccionar **Spain** [ENTER]
7. Seleccionar **United States** [ENTER]
8. Seleccionar **No** [ENTER]
9. Seleccionar **Spanish** [ENTER]
10. Seleccionar **Spanish - Catalan** [ENTER]
11. Seleccionar la primera NIC [ENTER]
12. Escriure **IDS** [ENTER]
13. Escriure **Suricata** [ENTER]
14. Escriure la contrasenya [ENTER]

15. Seleccionar **Yes** [ENTER]
16. Seleccionar **Yes** [ENTER]

Es configuraran les diferents particions del sistema de forma manual, d'aquesta manera podem configurar les diferents parts segons les necessitats pròpies. Es separarà el sistema en quatre particions, la d'arrencada, la d'intercanvi de memòria, la partició arrel on s'hi trobarà la major part del sistema i la partició de variables, on es troben els registres, que serà la que es configurarà amb una major capacitat.

1. Seleccionar **Manual** [ENTER]
2. Seleccionar **SCSI3 (0,0,0) (sda)** [ENTER]
3. Seleccionar **Yes** per crear una taula de particions nova [ENTER]
4. Seleccionar **pri/log FREE SPACE** [ENTER]
5. Seleccionar **Create a new Partition** [ENTER]
6. Escriure **200 MB** [ENTER] (NOTA: Aquesta serà la partició de **/boot**)
7. Seleccionar **Primary** [ENTER]
8. Seleccionar **Beginning** [ENTER]
9. Seleccionar **Use as: Ext4 journaling file System** [ENTER]
10. Seleccionar **Ext2 file System** [ENTER]
11. Seleccionar **Mount point: /** [ENTER]
12. Seleccionar **/boot – static files of the boot loader** [ENTER]
13. Seleccionar **Bootable flag: off** [ENTER] (NOTA: Canvia el flag a **on**)
14. Seleccionar **Done setting up the partition** [ENTER]
15. Seleccionar **Configure the Logical Volume Manager** [ENTER]
16. Seleccionar **Yes a write change to disks and configure LVM,** [ENTER]
17. Seleccionar **Create volume group** [ENTER]
18. Escriure **LVG** [ENTER]
19. Seleccionar **/dev/sda FREE SPACE, [ESPAI]** [ENTER]
20. Seleccionar **Yes a write change to disks and configure LVM,** [ENTER]
21. Seleccionar **Create logical volume** [ENTER]
22. Seleccionar **LVG** [ENTER]
23. Escriure **swap** [ENTER]
24. Escriure **20GB** [ENTER] (NOTA: El doble del valor de la memòria RAM)
25. Seleccionar **Create logical volume** [ENTER]
26. Seleccionar **LVG** [ENTER]
27. Escriure **root** [ENTER]
28. Escriure **20GB** [ENTER]
29. Seleccionar **Create logical volume** [ENTER]
30. Seleccionar **LVG** [ENTER]
31. Escriure **var** [ENTER]
32. Escriure **115GB** [ENTER]
33. Seleccionar **Finish** [ENTER]

Tot seguit s'assignaran les diferents particions creades al Grup de Volums Lògics corresponents.

1. Seleccionar **#1 20.0 GB** sota **LVM VG LVG, LV swap**, [ENTER]
2. Seleccionar **Use as: do not use** [ENTER]
3. Seleccionar **swap area** [ENTER]
4. Seleccionar **Done setting up the partition** [ENTER]
5. Seleccionar **#1 20.0 GB** sota **LVM VG LVG, LV root**, [ENTER]
6. Seleccionar **Use as: do not use** [ENTER]
7. Seleccionar **Ext4 journaling file system** [ENTER]
8. Seleccionar **Mount point: none** [ENTER]
9. Seleccionar **/ - the root file system** [ENTER]
10. Seleccionar **Done setting up the partition** [ENTER]
11. Seleccionar **#1** sota **LVM VG LVG, LV var**, [ENTER]
12. Seleccionar **Use as: do not use** [ENTER]
13. Seleccionar **Ext4 journaling file system** [ENTER]
14. Seleccionar **Mount point: none** [ENTER]
15. Seleccionar **/var** [ENTER]
16. Seleccionar **Label: none** [ENTER]
17. Escriure **var** [ENTER]
18. Seleccionar **Done setting up the partition** [ENTER]
19. Seleccionar **Finish partitioning and write changes to disk** [ENTER]
20. Seleccionar **Yes to write changes to disk**, [ENTER]

Finalment es configuraran les últimes característiques abans d'acabar la instal·lació.

1. Seleccionar **Continue** [ENTER]
2. Seleccionar **No automàtic updates** [ENTER] (NOTA: Les actualitzacions es revisaran manualment abans de ser instal·lades)
3. Seleccionar **Virtual Machine Host** [ESPAI]
4. Seleccionar **OpenSSH Server** [ESPAI][ENTER] (NOTA: Ens permetrà accedir en remot al servidor)
5. Seleccionar **Yes** [ENTER]
6. Seleccionar **Continue** [ENTER]

15.5 SURICATA

15.5.1 Prerequisits

Abans de poder instal·lar Suricata, cal tenir un seguit de llibreries i utilitats les quals fa servir el sistema per funcionar:

libpcre3: Llibreria d'Expressions Regulars compatibles amb Perl 5 – Fitxers d'execució

libpcre3-dbg: Llibreria d'Expressions Regulars compatibles amb Perl 5 – Fitxers de debug

libpcre3-dev: Llibreria d'Expressions Regulars compatibles amb Perl 5 – Fitxers de desenvolupament

build-essential: Llista d'informació de paquets de build-essential

autoconf: Constructor d'*scripts* de configuració automàtic

automake: Eina per generar *GNU Standards-compliant Makefiles*

libtool: Llibreria genèrica de suport d'*scripts*

libpcap-dev: Llibreries de desenvolupament i fitxers de capçaleres per libpcap2

libnet1-dev: Fitxers de desenvolupament per libnet

libyaml-0-2: Llibreria que emet i analitza ràpidament YAML 1.1

libyaml-dev: Llibreria que emet i analitza ràpidament YAML 1.1 (Desenvolupament)

zlib1g: Llibreria de compressió en temps d'execució

zlib1g-dev: Llibreria de compressió - desenvolupament

libcap-ng-dev: Fitxers de desenvolupament i capçaleres per libcap-ng

libcap-ng0: Llibreria alternativa per capacitats POSIX

make: Utilitat per una compilació de forma directa

libmagic-dev: Llibreria de determinació de tipus de fitxer utilitzant números "màgics"

libjansson-dev: Llibreria en C per codificar, descodificar i manipular dades en JSON

libjansson4: Llibreria en C per codificar, descodificar i manipular dades en JSON

pkg-config: Gestiona, compila i enllaça senyals per llibreries

Es poden instal·lar tots els paquets anteriors amb una sola comanda:

```
sudo apt-get -y install libpcre3 libpcre3-dbg libpcre3-dev build-essential autoconf automake libtool libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libcap-ng-dev libcap-ng0 make libmagic-dev libjansson-dev libjansson4 pkg-config
```

Depenent de l'estat actual del sistema, pot tardar en completar aquest procés.

HTP

És un analitzador de seguretat per el protocol HTTP que utilitza Suricata. Per defecte ve integrat amb l'eina, però en cas de necessitar-ho el mètode d'instal·lació és el següent:

```
wget      https://github.com/OISF/libhttp/releases/download/0.5.19/http-0.5.19.tar.gz
tar -xvzf libhttp-0.5.19.tar.gz
cd libhttp-0.5.19
./configure
make
make install
```

Es pot comprovar i aconseguir la última versió aquí - <http://suricata-ids.org/download/>

Mode IPS

Per defecte, Suricata treballa com un IDS. Si es vol utilitzar com a programa IDS i IPS, s'haurà d'instal·lar els següents paquets addicionals:

libnetfilter-queue-dev: Fitxers de desenvolupament per libnetfilter-queue1

libnetfilter-queue1: Llibreria per Netfilter netlink-queue

libnfnetlink-dev: Fitxers de desenvolupament per libnfnetlink0

libnfnetlink0: Llibreria de Netfilter netlink

Per instal·lar aquests paquets utilitzarem la següent comanda:

```
sudo apt-get -y install libnetfilter-queue-dev libnetfilter-queue1
libnfnetlink-dev libnfnetlink0
```

En aquests moments ja tindriem tots els requisits per poder instal·lar Suricata.

15.5.2 Descàrrega i instal·lació

Per obtenir Suricata ho farem mitjançant el paquet que proporciona la pròpia fundació que el desenvolupa. Un cop s'obté el paquet, es descomprimirà perquè siguin es tingui accés als seus arxius.

```
VER=3.0.1
wget      "http://www.openinfosecfoundation.org/download/suricata-$VER.tar.gz"
tar -xvzf "suricata-$VER.tar.gz"
cd "suricata-$VER"
```

Si es vol instal·lar Suricata amb capacitats de IPS, s'ha d'utilitzar la següent comanda:

```
./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --
localstatedir=/var
```

En cas de només voler instal·lar Suricata com a IDS, utilitzar la comanda:

```
./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
```

Seguim amb la instal·lació:

```
make
sudo make install
sudo ldconfig
```

15.5.3 Configuració

Preparació

Per preparar la configuració haurem de crear el directori on es guardaran tots els registres de Suricata.

```
sudo mkdir /var/log/suricata
```

De la mateixa manera, s'ha de crear el directori on es situaran els diferents fitxers de configuració de l'eina.

```
sudo mkdir /etc/suricata
```

Procedirem a copiar tots els fitxers de configuració del directori d'instal·lació al directori que hem creat per a Suricata, **/etc/suricata**.

Ens col·locarem al directori de instal·lació i entrarem les comandes següents:

```
sudo cp classification.config /etc/suricata
sudo cp reference.config /etc/suricata
sudo cp suricata.yaml /etc/suricata
```

Altrament, també es disposa d'una preparació automàtica.

El codi font de Suricata posseeix uns fitxers de configuració automàtica per defecte. S'instal·laran aquests fitxers de configuració per defecte de la següent manera:

```
sudo make install-conf
```

Suricata no té sentit sense unes regles IDS. El fitxer Makefile ve amb una opció d'instal·lació de regles IDS. Per instal·lar-les, executarem:

```
sudo make install-rules
```

La comanda d'instal·lació anterior descarregarà el conjunt de regles creades per la comunitat actualment disponibles a EmergingThreats.net i les guardarà a **/etc/suricata/rules**

Configuració inicial

El fitxer de configuració de Suricata es troba a **/etc/suricata/suricata.yaml**. Obrirem el fitxer amb un editor de text per modificar-ne les opcions:

```
sudo nano /etc/suricata/suricata.yaml
```

Començarem amb una configuració inicial.

Registre

La variable *default-log-dir* ha d'apuntar a la localització del fitxer de registre de Suricata:

```
# The default logging directory. Any log or output file will be
# placed here if its not specified with a full path name. This can be
# overridden with the -l command line parameter.
default-log-dir: /var/log/suricata/
```

Variables

A la secció *vars:*, podem trobar varies variables importants utilitzades per Suricata.

HOME_NET hauria d'indicar la xarxa local inspeccionada per Suricata.

"*!\$HOME_NET*" s'assigna a *EXTERNAL_NET* i refereix a qualsevol altre xarxa a part de la local.

XXX_PORTS indica el número de port utilitzats per diferents serveis.

Tot i això, Suricata és capaç de detectar automàticament el trànsit HTTP sense tenir en compte els ports utilitzats, per tant no és una característica crítica especificat el valor de la variable *HTTP_PORTS* correctament.

```
# Holds variables that would be used by the engine.
vars:
# Holds the address group vars that would be passed in a Signature.
# These would be retrieved during the Signature address parsing stage.
address-groups:
HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
EXTERNAL_NET: "!$HOME_NET"
HTTP_SERVERS: "$HOME_NET"
SMTP_SERVERS: "$HOME_NET"
SQL_SERVERS: "$HOME_NET"
DNS_SERVERS: "$HOME_NET"
TELNET_SERVERS: "$HOME_NET"
AIM_SERVERS: "$EXTERNAL_NET"
DNP3_SERVER: "$HOME_NET"
DNP3_CLIENT: "$HOME_NET"
MODBUS_CLIENT: "$HOME_NET"
MODBUS_SERVER: "$HOME_NET"
ENIP_CLIENT: "$HOME_NET"
ENIP_SERVER: "$HOME_NET"

# Holds the port group vars that would be passed in a Signature.
# These would be retrieved during the Signature port parsing stage.
port-groups:
HTTP_PORTS: "80"

SHELLCODE_PORTS: "!80"

ORACLE_PORTS: 1521

SSH_PORTS: 22

DNP3_PORTS: 20000

MODBUS_PORTS: 502
```

Política basada en SO

La secció *host-os-policy* s'utilitza per defensar-se davant alguns dels atacs coneguts que exploten el comportament de la xarxa d'un sistema operatiu per evadir la seva detecció. Com a contra mesura, els IDS moderns han desenvolupat les anomenades inspeccions basades en l'objectiu ("target-based"), on el motor d'inspecció afina el seu algorisme de detecció basat en el sistema operatiu del trànsit a l'objectiu. Per això, si es coneix el Sistema Operatiu que utilitzen els hosts locals, es pot entrar aquesta informació a Suricata i incrementar potencialment el nivell de detecció. Aquí és on la secció *host-on-policy* és utilitzada.

```
# Host specific policies for defragmentation and TCP stream
# reassembly. The host OS lookup is done using a radix tree, just
# like a routing table so the most specific entry matches.
host-os-policy:
  # These are Windows machines.
  windows: [192.168.3.0/24,0.0.0.0/0]
  bsd: []
  bsd-right: []
  old-linux: []
  # Make the default policy Linux.
  linux: [192.168.3.252/32]
  old-solaris: []
  solaris: ["::1"]
  hpux10: []
  hpux11: []
  irix: []
  macos: []
  vista: []
  windows2k3: []
```

Fils d'execució

A la secció *threading* podem especificar l'afinitat de CPU per a diferents fils d'execució de Suricata.

Per defecte, l'afinitat de CPU es troba desactivada el que significa que els fils d'execució de Suricata seran programats en qualsevol dels nuclis disponibles de la CPU.

Si no s'indica cap configuració en concret, Suricata crearà un fil d'execució de detecció per cada nucli de la CPU. Es pot ajustar aquest comportament especificant un valor a la variable *detect-thread-ratio*: *N*. D'aquesta manera es crearan *N* fils d'execució per cada nucli, per tant, *N*M* fils d'execució on *M* són el nombre de nuclis.

```
# Suricata is multi-threaded. Here the threading can be influenced.
threading:
  # On some cpu's/architectures it is beneficial to tie individual threads
  # to specific CPU's/CPU cores. In this case all threads are tied to CPU0,
  # and each extra CPU/core has one "detect" thread.
  set-cpu-affinity: no

  #
  # By default Suricata creates one "detect" thread per available CPU/CPU core.
  # This setting allows controlling this behaviour. A ratio setting of 2 will
  # create 2 detect threads for each CPU/CPU core. So for a dual core CPU this
  # will result in 4 detect threads. If values below 1 are used, less threads
  # are created. So on a dual core CPU a setting of 0.5 results in 1 detect
  # thread being created. Regardless of the setting at a minimum 1 detect
  # thread will always be created.
  #
  detect-thread-ratio: 1.5
```

15.5.4 Línia de comandes

Opcions

Suricata és una eina amb execució via la línia de comandes. Això ens aporta la possibilitat de poder executar aquesta eina amb diferents opcions i configuracions, indicant-ho a la línia d'execució. Les diferents opcions de les que disposa Suricata són:

- c Aquesta és l'opció més important. Després de -c s'ha d'indicar el camí cap a la localització del fitxer de configuració general suricata.yaml
- i Després de l'opció -i s'ha d'indicar la targeta de xarxa que es vol utilitzar per a capturar-hi paquets. Implica la captura de paquets mitjançant libpcap en el mode en temps real pcap.
- r Després de l'opció -r s'indica el camí del fitxer de tipus pcap en el qual els diferents paquets capturats es guarden. D'aquesta manera es té l'opció de d'inspeccionar els paquets en mode fora de línia o en mode pcap.
- s Amb l'opció -s es pot assignar un fitxer amb signatures, el qual serà carregat juntament amb les regles indicades en el fitxer de configuració general.
- l Amb aquesta opció es pot indicar el directori per defecte dels registres. En cas de tenir assignat un valor al **default-log-dir** en el fitxer suricata.yaml, aquest no serà utilitzat si s'indica la opció -l en la línia d'execució. Utilitzarà el valor indicat en amb l'opció -l.
- D Per norma general, si s'executa Suricata des de la consola, l'execució la manté ocupada. No es pot utilitzar per altres propòsits, i quan es tanca la consola, Suricata para l'execució.
Si s'executa el Suricata com a dimoni, amb la opció -D, aquest s'executa en segon pla. Això dóna la possibilitat de seguir fent servir la consola mentre Suricata s'executa.

--list-app-layer-protos : Llistat de protocols de la capa d'aplicació suportats.

--list-keywords[=all | cvs | <keyword>] : Llistat de paraules claus implementades pel motor.

--list-runmodes Es llisten tots els modes d'execució possibles.

--runmode (en combinació amb les opcions -i / -o)

Amb aquesta opció es pot indicar el mode d'execució que es vol utilitzar. Aquesta opció de la línia de comandes sobreescriu l'opció del mode d'execució indicat en el fitxer de configuració general.

Tests unitaris

Suricata incorpora un seguit de tests unitaris per a la verificació del codi. Aquests tests es poden utilitzar mitjançant la línia de comandes indicant unes opcions concretes:

- u** Amb aquesta opció s'executen tots els tests unitaris per a comprovar la validesa del codi de Suricata.

- U** Amb aquesta opció es pot indicar quins tests unitaris concrets es volen realitzar. Aquesta opció utilitza REGEX.

- list-unittests** Mostra un llistat amb tots els tests unitaris possibles.

- fatal-unittests** Permet executar els tests unitaris però en cas que un dels tests fallés, aturaria de forma immediata l'execució el que permet veure directament on s'ha produït el problema.

15.6 ELASTICSEARCH

15.6.1 Prerequisites

L'únic requisit que ens demana Elasticsearch per a la instal·lació és tenir una versió de Java instal·lada. En el nostre cas instal·larem la versió més actual de Oracle Java 8, ja que és la que ens recomanen els desenvolupadors d'Elasticsearch. Tot i això, OpenJDK hauria de funcionar correctament com a alternativa de codi lliure.

Comprovem si tenim instal·lat el paquet, i en cas contrari procediríem a fer-ho.

```
java -version
echo $JAVA_HOME
```

Per instal·lar el JDK de Oracle Java 8 afegirem el repositori de Oracle Java al apt:

```
sudo add-apt-repository -y ppa:webupd8team/java
```

Actualitzarem el llistat de paquets de la base de dades:

```
sudo apt-get update
```

Instal·lem l'última versió de Oracle Java 8 i acceptem la llicència que se'ns mostrarà:

```
sudo apt-get -y install oracle-java8-installer
```



Un cop tenim instal·lat Oracle Java 8, podem començar amb la instal·lació de Elasticsearch.

15.6.2 Instal·lació

Elasticsearch es pot instal·lar des d'un gestor de paquets afegint a la llista de fonts el paquet d'Elastic.

Primer de tot s'ha d'importar la clau pública GPG d'Elasticsearch:

```
wget -q0 - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

Si la línia de comandes queda sense fer res, és possible que estigui esperant la contrasenya de l'usuari per poder utilitzar la comanda **sudo**.

Afegirem la descripció del repositori.

```
echo "deb http://packages.elastic.co/elasticsearch/2.x/debian stable
main" | sudo tee -a /etc/apt/sources.list.d/elasticsearch-2.x.list
```

Actualitzem la base de dades de paquets apt:

```
sudo apt-get update
```

Instal·lem Elasticsearch:

```
sudo apt-get -y install elasticsearch
```

Ara que Elasticsearch es troba instal·lat en el nostre sistema, editarem el fitxer de configuració **/etc/elasticsearch/elasticsearch.yml** per fer-hi unes primeres modificacions inicials:

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

Es voldrà restringir l'accés exterior cap al nostre Elasticsearch, port 9200, de tal forma que des de fora no sigui possible llegir les nostres dades o apagar el clúster d'Elasticsearch des de l'API HTTP. Per fer-ho, buscarem la línia que conté **network.host**, traurem el comentari, i canviarem el seu valor per **"localhost"** per tal que quedi de la següent manera:

```
network.host: localhost
```

Tot seguit iniciarem el servei:

```
sudo service elasticsearch restart
```

Volem que Elasticsearch s'iniciï al mateix moment que ho faci el servidor. Haurem de modificar el servei d'arrencada del servidor per tal que ho faci:

```
sudo update-rc.d elasticsearch defaults 95 10
```

Un cop fet això, ja tindrem instal·lat Elasticsearch.

15.6.3 Configuració

Per modificar la configuració predeterminada d'Elasticsearch s'ha d'editar el fitxer **/etc/elasticsearch/elasticsearch.yml**

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

Primer de tot ens permet indicar un nom descriptiu per al nostre clúster.

```
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
# cluster.name: my-application
```

També podem assignar un nom a cada node.

```
# ----- Node -----  
#  
# Use a descriptive name for the node:  
#  
# node.name: node-1
```

Es pot indicar la localització dels fitxers on es troben les dades guardades i la localització on es guardaran els fitxers de registres de l'eina.

```
# ----- Paths -----  
#  
# Path to directory where to store the data (separate multiple locations by comma):  
#  
# path.data: /path/to/data  
#  
# Path to log files:  
#  
# path.logs: /path/to/logs
```

També es pot assignar una quantitat de memòria per a Elasticsearch i d'aquesta manera assegurar que té la quantitat mínima necessària.

El rendiment de l'eina baixa molt notablement quan el sistema està utilitzant l'intercanvi de memòria (swapping memory).

```
# ----- Memory -----  
#  
# Lock the memory on startup:  
#  
# bootstrap.mlockall: true  
#  
# Make sure that the `ES_HEAP_SIZE` environment variable is set to about half the  
memory  
# available on the system and that the owner of the process is allowed to use  
this limit.  
#  
# Elasticsearch performs poorly when the system is swapping the memory.  
#
```

Com a últim punt important en la configuració, es pot indicar la direcció i el port pel qual es vol accedir a l'eina.

```
# ----- Network -----  
#  
# Set the bind address to a specific IP (IPv4 or IPv6):  
#  
# network.host: 192.168.0.1  
#  
# Set a custom port for HTTP:  
#  
# http.port: 9200  
#
```

15.7 LOGSTASH

15.7.1 Instal·lació

Per instal·lar aquesta eina s'utilitzarà el repositori que ofereixen els seus desenvolupadors.

El primer que s'ha de fer és descarregar i instal·lar la Clau de Signatura Pública que ens ofereixen:

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

Tot seguit s'afegeix el repositori a la llista pròpia:

```
echo "deb https://packages.elastic.co/logstash/2.3/debian stable main" | sudo tee -a /etc/apt/sources.list
```

S'ha d'actualitzar el llistat de repositoris amb la comanda:

```
sudo apt-get update
```

I es pot procedir a instal·lar l'eina:

```
sudo apt-get install logstash
```

Un cop finalitzi l'operació ja tindrem acabada la instal·lació de Logstash al nostre sistema.

15.7.2 Configuració

Crearem el fitxer **logstash.conf** en el directori **/etc/logstash/conf.d/**:

```
sudo touch /etc/logstash/conf.d/logstash.conf
```

Editarem el fitxer creat anteriorment i hi afegirem el següent contingut:

```
sudo nano /etc/logstash/conf.d/logstash.conf
```

El contingut d'aquesta eina s'estructura en tres blocs diferenciats, l'entrada d'informació, el filtrat d'aquestes dades i la sortida i posterior registre.

En el primer apartat, indicarem el fitxer de registre del qual es llegiran les dades, la localització de la base de dades pròpia de Logstash, el tipus de codificació del fitxer de lectura i se li assigna un tipus, al gust de l'administrador.

```
input {
  file {
    path => ["/var/log/suricata/eve.json"]
    #sincedb_path => ["/var/lib/logstash/"]
    sincedb_path => ["/var/cache/logstash/sinceds/since.db"]
    codec => json
    type => "SELKS"
  }
}
```

En el segon bloc, es configuren els diferents filtres segons el tipus que siguin (en aquest moment s'utilitza el que s'ha assignat al primer apartat).

Una configuració bastant estàndard seria la següent:

```
filter {
  if [type] == "SELKS" {

    date {
      match => [ "timestamp", "ISO8601" ]
    }

    ruby {
      code => "if event['event_type'] == 'fileinfo';
event['fileinfo']['type']=event['fileinfo']['magic'].to_s.split(',')[0]; end;"
    }

    metrics {
      meter => [ "eve_insert" ]
      add_tag => "metric"
      flush_interval => 30
    }
  }

  if [http] {
    useragent {
      source => "[http][http_user_agent]"
      target => "[http][user_agent]"
    }
  }
  if [src_ip] {
    geoip {
      source => "src_ip"
      target => "geoip"
      #database => "/opt/logstash/vendor/geoip/GeoLiteCity.dat"
      add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
      add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
    }
    mutate {
      convert => [ "[geoip][coordinates]", "float" ]
    }
  }
  if ![geoip.ip] {
    if [dest_ip] {
      geoip {
        source => "dest_ip"
        target => "geoip"
        #database => "/opt/logstash/vendor/geoip/GeoLiteCity.dat"
        add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
        add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
      }
      mutate {
        convert => [ "[geoip][coordinates]", "float" ]
      }
    }
  }
}
```

Finalment es configurarà la sortida i registre de les dades. En aquest cas s'indicarà el nom del fitxer que es crearà per guardar-hi la informació i, segons si es tracta d'un esdeveniment o del registre de l'estat de l'eina es crearà amb un nom o amb un altre.

```
output {
  if [event_type] and [event_type] != 'stats' {
    elasticsearch {
      hosts => "127.0.0.1"
      index => "logstash-%{event_type}-%{+YYYY.MM.dd}"
    }
  } else {
    elasticsearch {
      hosts => "127.0.0.1"
      index => "logstash-%{+YYYY.MM.dd}"
    }
  }
}
```

15.8 KIBANA

15.8.1 Prerequisits

Per instal·lar i configurar Kibana es necessiten alguns components i requisits.

Com s'ha comentat anteriorment, Kibana treballa conjuntament amb Elasticsearch, per tant es necessitarà tenir aquesta eina instal·lada amb la versió corresponent a la que s'estigui utilitzant per Kibana. En el nostre cas necessitarem una versió d'Elasticsearch igual o posterior a la 2.3.

També es necessitarà recuperar la informació d'Elasticsearch, per tant s'haurà de saber alguns paràmetres de la seva instal·lació:

- URL de la instància d'Elasticsearch a la qual es vulgui connectar.
- Quins índex d'Elasticsearch es volen buscar.

Degut a les característiques que ofereix Kibana, no es pot utilitzar en tots els navegadors existents. Els navegadors compatibles, els més moderns, són els següents:

| | IE9 | IE11+ | Firefox | Chrome | Safari (Mac) | Safari (iOS) | Chrome (Android) |
|------------|-----|-------|---------|--------|--------------|--------------|------------------|
| Kibana 3.1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Kibana 4.x | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

15.8.2 Instal·lació

Per instal·lar Kibana es pot fer per dues vies diferents. Es pot obtenir el paquet corresponent a la plataforma a la que es vulgui instal·lar o es pot fer per mitjà del seu repositori.

En aquest cas s'utilitzarà la segona manera ja que proporciona una base estàndard d'instal·lació que ens servirà per orientar-nos millor en tot el conjunt de fitxers que comprenen l'eina.

Primer de tot ens descarregarem i instal·larem la Clau de Signatura Pública, en el nostre cas afegida anteriorment durant la instal·lació de Elasticsearch:

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

Afegirem la descripció del repositori al fitxer **/etc/apt/source.list**:

```
echo "deb http://packages.elastic.co/kibana/4.5/debian stable main" | sudo tee -a /etc/apt/sources.list
```

S'actualitza el llistat de repositoris i paquets per tenir accés al que s'ha afegit anteriorment:

```
sudo apt-get update
```

S'instal·la Kibana amb la següent comanda:

```
sudo apt-get install kibana
```

Configurarem l'inici de Kibana perquè es produeixi juntament amb l'inici d'arrencada del servidor:

```
sudo update-rc.d kibana defaults 95 10
```

Afegim l'adreça del servidor en el fitxer de configuració modificant el fitxer

```
sudo nano /opt/kibana/config/kibana.yml
```

de la següent manera:

```
# The host to bind the server to.  
server.host: "192.168.3.252"
```

Reiniciem el servei:

```
sudo /bin/systemctl daemon-reload  
sudo /bin/systemctl enable kibana.service
```

15.8.3 Connexió de Kibana amb Elasticsearch

Abans de començar a utilitzar Kibana s'ha d'indicar quins índex d'Elasticsearch es voldran explorar.

La primera vegada que s'accedeix a Kibana es demanarà que es configuri un *patró d'índex* que coincideixi amb el nom de un o varis dels índex propis.

Configuració d'un patró d'índex

El primer que s'ha de fer és accedir al navegador pel port 5601 per tal d'accedir a la interfície gràfica de Kibana. Si no s'ha canviat la configuració serà <http://localhost:5601>, en cas d'haver fet algun canvi tindrà la forma <http://DOMINIPROPI.com:5601>. Es mostrarà la següent pantalla per configurar el patró.

Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

Index contains time-based events

Use event times to create index names [DEPRECATED]

Index name or pattern

Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*

Unable to fetch mapping. Do you have indices matching the pattern?

S'ha d'especificar un patró que coincideixi amb un o varis dels índex d'Elasticsearch propis, com s'ha indicat abans.

Per defecte, Kibana fa el supòsit que s'està treballant amb dades portades a Elasticsearch per Logstash. En aquest cas, que és el nostre, es pot utilitzar **logstash-***, que ve per defecte, com al patró d'índex propi.

Es selecciona el camp índex que conté la data i hora que es vulgui utilitzar per efectuar comparacions basades en el temps. En cas que l'índex no tingués dades que incloguessin el temps s'hauria de desactivar el camp **Index contains time-based events**, però no és el nostre cas.

S'ha de clicar a **Create** per afegir el patró. Aquest primer patró que s'ha afegit automàticament es configurarà per a ser el patró per defecte. Per canviar el patró per defecte en cas de tenir-ne varis, s'ha de fer des de l'apartat **Settings > Indices**.

15.8.4 Configuració

Els apartats més importants de la configuració de Kibana s'explicaran tot seguit per tenir una noció inicial de les opcions que permet modificar aquesta eina.

La part més important de la configuració és on indiquem l'adreça i el port pels quals s'accedirà a l'eina. Es poden deixar per defecte o canviar-los. En el nostre cas farem la primera opció ja que qui accedeix a Kibana és una interfície intermèdia, Scirius, i per tant, no necessitem tenir-hi accés des de l'exterior.

```
# Kibana is served by a back end server. This controls which port to use.
# server.port: 5601

# The host to bind the server to.
#server.host: "127.0.0.1"
```

Hem d'indicar la instància de Elasticsearch per a que Kibana sàpiga on ha de fer les peticions de les dades que s'han de mostrar.

```
# The Elasticsearch instance to use for all your queries.
# elasticsearch.url: "http://localhost:9200"
```

Kibana utilitza índex per guardar cerques, visualitzacions i taulells. Si es vol utilitzar un índex propi es pot canviar el valor d'aquesta variable.

```
# Kibana uses an index in Elasticsearch to store saved searches, visualizations
# and dashboards. It will create a new index if it doesn't already exist.
# kibana.index: ".kibana"
```

En cas de tenir protegit Elasticsearch amb una autenticació bàsica, es pot indicar el nom d'usuari i la paraula clau per tal que Kibana tingui accés a les dades que tracta.

```
# If your Elasticsearch is protected with basic auth, these are the user
# credentials used by the Kibana server to perform maintenance on the kibana_index
# at startup. Your Kibana users will still need to authenticate with Elasticsearch
# (which is proxied through the Kibana server)
# elasticsearch.username: "user"
# elasticsearch.password: "pass"
```

És interessant poder posar un temps màxim de resposta a les peticions. Com que hi ha vegades que les peticions són molt grans, Elasticsearch pot tenir problemes per tractar tota la informació.

Depèn el tipus de peticions que es facin es pot assignar un valor o un altre. Si les peticions són de poques dades, es pot assignar un temps d'espera menor i així controlar més ràpidament si alguna cosa ha fallat mentre que si les peticions són de una gran quantitat de dades lo més interessant és assignar un valor alt ja que de per si tardarà a tractar-les i enviar-les.

```
# Time in milliseconds to wait for responses from the back end or elasticsearch.  
# This must be > 0  
# elasticsearch.requestTimeout: 30000
```

Moltes vegades és interessant assignar l'identificador del procés a un fitxer, d'aquesta manera si es vol actuar sobre aquest només s'ha de mirar l'identificador i es sabrà quin procés és.

```
# Set the path to where you would like the process id file to be created.  
# pid.file: /var/run/kibana.pid
```

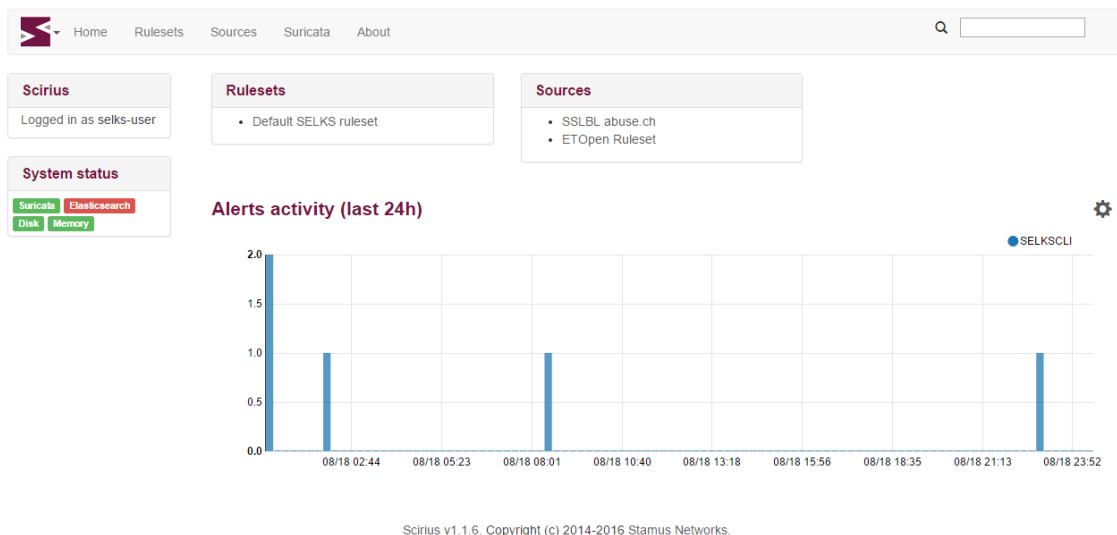
Una altra característica interessant per els administradors és el poder tenir tots els registres de l'aplicació, ja siguin de caràcter informatiu com indicant que hi ha hagut errors en l'execució. Aquests registres donen la possibilitat d'actuar un cop s'ha produït una fallada ja que et permeten saber què ha fallat i perquè.

```
# If you would like to send the log output to a file you can set the path below.  
# logging.dest: stdout
```

15.9 SCIRIUS

15.9.1 Introducció

Scirius és una interfície web dedicada a la gestió dels diferents conjunts de regles de Suricata. Gestiona els fitxers de regles i actualitza els fitxers associats. A més a més permet visualitzar de forma gràfica informació referent a les regles i les alertes generades.



15.9.2 Instal·lació i configuració

Instal·lació

L'aplicació Scirius es troba escrita en Django, la qual es pot instal·lar com qualsevol altre aplicació d'aquest tipus. El procediment es descriu tot seguit.

Dependències

Scirius utilitza els mòduls de Django següents:

- tables2
- south
- bootstrap3
- requests
- revproxy

La manera més senzilla per a instal·lar aquestes dependències és per mitjà de la utilitat **pip**. Per a les utilitats de Python, llenguatge que utilitza Django, i pip s'utilitzarà:

```
apt-get install python-pip python-dev
```

Tot seguit es poden instal·lar Django i les diferents dependències.

```
pip install -r requirements.txt
```

Una de les utilitats que ens ofereix Scirius és un conjunt de Scripts, com ara el `suri_reloader` que gestiona el reinici de Suricata, el qual necessitarà `pyinotify`.

```
pip install pyinotify
```

És possible, segons la distribució que s'utilitzi, que es necessiti un GitPython recent:

```
pip install gitpython==0.3.1-beta2
```

I també es necessitarà el mòdul respectiu a les bases de dades.

```
pip install gitdb
```

Execució

Ens dirigirem fins al directori on es troba el codi i entrarem la comanda:

```
python manage.py syncdb
```

Per defecte es demanarà una autenticació, per tant serà necessari crear un compte amb privilegis de superusuari en el moment en què es demani.

La forma més simple de provar Scirius és executar el servidor de proves de Django:

```
python manage.py runserver
```

Configuració de Suricata

Scirius genera un sol fitxer de signatures amb totes les regles activades. Quan s'editi Suricata, s'ha de configurar el directori on es vol que es generi aquest fitxer i els arxius del conjunt de regles associats siguin copiats.

Aquesta eina no tocarà el fitxer de configuració general de Suricata, per tant s'ha d'actualitzar perquè apunti on les dades són configurades per Scirius. En cas de només tenir regles generades per aquesta eina, s'hauria de tenir un fitxer de configuració com el següent:

```
# Set the default rule path here to search for the files.  
# if not set, it will look at the current working dir  
default-rule-path: /etc/suricata/rules  
rule-files:  
- scirius.rules
```

Per interaccionar amb Scirius, es necessita detectar quan el fitxer `/etc/suricata/rules/scirius.reload` és creat. Quan sigui el cas, s'ha de reiniciar o recarregar Suricata i eliminar el fitxer `reload` un cop s'hagi efectuat.

Una manera de fer-ho és utilitzant `suri_reloader` disponible al directori `suricata/scripts`. La forma de utilització pot ser:

```
suri_reloader -p /etc/suricata/rules -l /var/log/suri-reload.log -D
```

15.9.3 Enllaçar amb Elasticsearch

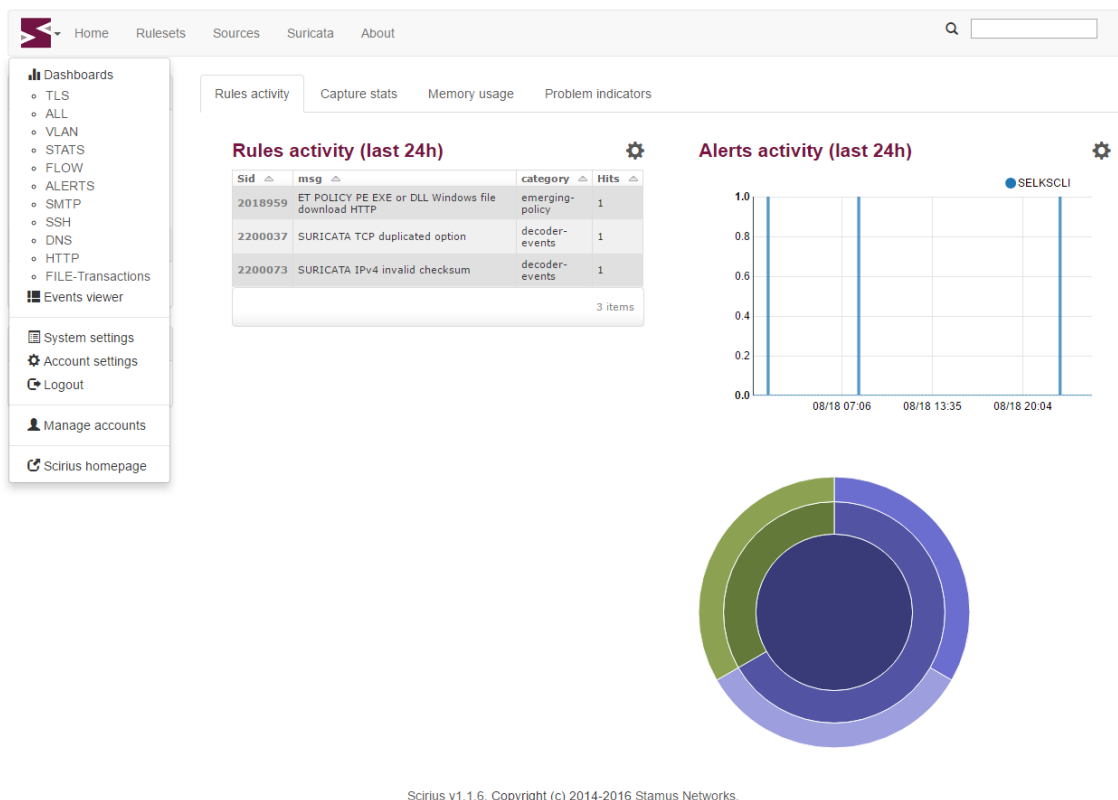
En el nostre cas, que utilitzem Suricata amb enregistrament Eve i utilitzem la utilitat d'Elasticsearch, es pot obtenir informació sobre signatures i també informació sobre Suricata.

Per configurar la connexió amb Elasticsearch, es pot editar el fitxer **settings.py** o crear un **local_settings.py** dins el directori de **scirius** per tal de configurar la característica. Elasticsearch s'activarà si la variable **USE_ELASTICSEARCH** s'assigna a **True** en el fitxer anomenat anteriorment.

```
USE_ELASTICSEARCH = True
ELASTICSEARCH_ADDRESS = "localhost:9200"
ELASTICSEARCH_2X = True
```

15.9.4 Enllaçar amb Kibana

En el cas d'utilitzar l'eina de visualització Kibana, és possible obtenir enllaços dels diferents taulers clicant sobre la icona superior esquerra.



Per activar aquesta característica, s'ha d'editar el fitxer de configuracions **local_settings.py**:

```
KIBANA_VERSION=4
KIBANA_INDEX = ".kibana"
KIBANA_URL = "http://localhost:5601"
USE_KIBANA = True
KIBANA_PROXY = True
```

15.10 OINKMASTER

És possible descarregar i instal·lar les regles de forma manual, però existeix una manera molt més ràpida i còmode de fer-ho. Hi ha programes especials que s'utilitzen per descarregar i instal·lar regles, com serien Pulled Pork i Oinkmaster. En aquest projecte farem servir Oinkmaster ja que és el que es recomana des de la OISF, organització fundadora de Suricata.

15.10.1 Instal·lació

Per començar instal·larem Oinkmaster al nostre Sistema Operatiu.

```
sudo apt-get install oinkmaster
```

Existeixen diferents tipus de regles, Emerging Threats (ET), Emerging Threats Pro i VRT. En aquest cas utilitzarem Emerging Threats.

15.10.2 Configuració

Oinkmaster necessita saber on es troben aquestes regles. Les podem trobar en el següent enllaç:

```
https://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz
```

Per afegir aquest enllaç a l'Oinkmaster, haurem de modificar el fitxer **/etc/oinkmaster.conf**.

```
sudo nano /etc/oinkmaster.conf
```

Col·locarem un **#** davant l'enllaç existent i afegirem el nou.

Per tancar l'editor clicarem **ctrl+x** seguit de **y** i **enter**.

El següent pas que farem serà crear un directori per les noves regles.

```
sudo mkdir /etc/suricata/rules
```

Seguidament entrarem:

```
cd /etc  
sudo oinkmaster -C /etc/oinkmaster.conf -o /etc/suricata/rules
```

En el nou directori de regles es podran trobar els fitxers **classification.config** i **reference.config**. Els directoris d'ambdós han de ser afegits al fitxer **suricata.yaml**.

```
sudo nano /etc/suricata/suricata.yaml
```

S'afegeixen les noves localitzacions dels fitxers al lloc de les actualment presents.

```
classification-file: /etc/suricata/rules/classification.config  
reference-config-file: /etc/suricata/rules/reference.config
```

Per comprovar si tot funciona correctament, executarem Suricata. En aquesta execució hi afegirem la opció **-init-errors-fatal** per comprovar que no existeixi cap problema.

```
suricata -c /etc/suricata/suricata.yaml -i ens33 -init-errors-fatal
```

Un cop estigui funcionant, afegirem la opció `-D` per executar-ho en mode dimoni.

```
suricata -D -c /etc/suricata/suricata.yaml -i ens33
```

Un cop es troba funcionant, podem comprovar si ja hi ha registres al fitxer `/var/log/suricata/fast.log`

```
tail -f /var/log/suricata/fast.log
```

Emerging Threats conté més regles que les que es troben carregades a Suricata. Per comprovar quines són les disponibles en el directori de regles, fem:

```
ls /etc/suricata/rules/*.rules
```

Es miren quines són aquelles que no es troben presents al fitxer `suricata.yaml` i s'hi afegeixen si es desitja. Per comprovar quines regles es troben actives i quines no es pot fer mirant el fitxer exposat anteriorment:

```
sudo nano /etc/suricata/suricata.yaml
```

Si es desactiva una regla col·locant un `#` al davant, la pròxima vegada que s'executi Oinkmaster es tornarà a activar. Es pot desactivar directament per Oinkmaster. Per fer-ho s'anirà al següent directori:

```
cd /etc/suricata/rules
```

i es buscarà el sid de la regla o regles que es vulguin desactivar.

Seguidament entrarem al següent fitxer:

```
sudo nano /etc/oinkmaster.conf
```

i al final de tot s'hi escriurà:

```
disablesid 2010495
```

En comptes de 2010495, s'escriurà el sid de la regla que es vol desactivar. És possible desactivar múltiples regles escrivint els seus sid separats per comes.

Per activar alguna regla que es troba desactivada per defecte, s'ha de fer exactament el mateix, amb la diferència que la línia que s'ha d'entrar és:

```
enablesid: 2010495
```

on 2010495 és el sid de la regla que volem activar.

Si es vol actualitzar el conjunt de regles del que es disposa, s'utilitzarà la comanda:

```
sudo oinkmaster -C /etc/oinkmaster.conf -o /etc/suricata/rules
```

15.10.3 Actualització automàtica de regles

Com que interessa que les regles estiguin actualitzades, i Emerging Threats actualitza les seves regles diàriament, el que es vol és que les nostres també ho facin. Per fer això, en comptes d'executar la comanda anterior a diari, el que es fa és programar una tasca al **cron** del Sistema Operatiu perquè ho faci.

```
crontab -e
```

S'afegeix una configuració per descarregar les regles cada dia a la mitjanit.

```
0 0 * * * oinkmaster -C /etc/oinkmaster.conf -o /etc/suricata/rules
```

També afegirem una configuració per actualitzar les regles que utilitza Suricata sense parar el servei. D'aquesta manera no es perdrà cap paquet i en cap moment tindrem el procés parat. Per fer-ho afegirem la següent configuració al crontab:

```
5 0 * * * kill -USR2 $(sudo cat /var/run/suricata.pid)
```

En aquest cas hem utilitzat una comanda dins una altra comanda. Amb la comanda **cat** aconseguim obtenir l'identificador del procés de Suricata, que cada cop que s'executi serà diferent. Amb l'identificador, podem utilitzar la comanda **kill** que ens permetrà fer una recàrrega de les regles que utilitza Suricata.