Treball final de grau Seguretat informàtica

Detecció de vulnerabilitats i intrusions a les xarxes de computadors

Daniel Manau Donas

Grau d'Enginyeria Informàtica

Seguretat Informàtica

Consultora: Cristina Pérez Solà

Professora: Helena Rifà Pous

03/01/2018

Universitat Oberta de Catalunya

uoc.edu

FITXA DEL TREBALL FINAL

Títol del treball:	*Detecció de vulnerabilitats i intrusions a les xarxes de computadors*
Nom de l'autor:	*Daniel Manau Donas*
Nom del consultor/a:	*Cristina Pérez Solà*
Nom del PRA:	*Helena Rifà Pous*
Data de lliurament (mm/aaaa):	*01/2018*
Titulació o programa:	*Grau d'Enginyeria Informàtica*
Àrea del Treball Final:	*Seguretat informàtica*
Idioma del treball:	*Català*
Paraules clau	*Seguretat en xarxes, vulnerabilitats, intrusions*

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

El problema a resoldre en aquest projecte és la detecció de les possibles vulnerabilitats que poden existir en els diferents equips d'una organització.

Cada vegada és més habitual que les organitzacions ofereixin serveis de diferents tipus mitjançant internet. Aquest tipus de serveis públics requereixen una correcta protecció per tal d'evitar accessos no autoritzats als equips i a la xarxa interna.

Per protegir aquests equips exposats necessitarem les funcions d'un sistema detector de vulnerabilitats, el qual realitzarà diferents anàlisis sobre els equips existents a la xarxa client. També inclourem un sistema de detecció d'intrusions (IDS). Aquest sistema IDS registrarà tota activitat que sigui configurada com a no autoritzada.

L'objectiu final és protegir el màxim possible les xarxes de computadors de l'organització de possibles atacs d'intrusió.

Aquestes eines seran implementades en un dispositiu Raspberry Pi 3 i mitjançant un entorn simulat realitzarem les diferents proves de vulnerabilitats.

El resultat final de les proves realitzades demostra que en la majoria d'equips que s'utilitzen en entorns reals de treball presenten greus deficiències de seguretat. Aquestes poden ser solucionades amb una correcta gestió de les actualitzacions del

fabricant del sistema operatiu i auditories periòdiques de seguretat.

Abstract (in English, 250 words or less):

In this project the problem to resolve is the correct detection of possible vulnerabilities that may exist in the different devices of an organization.

Today is very common offer different types of services through the internet in any type of organization. These types of public services require the correct protection to prevent unauthorized access to devices and the internal network.

To protect these exposed devices, we will need the functions of a vulnerability detection system, which perform different analytics on the existing computers. We will also include an intrusion detection system (IDS). This IDS will record any activity that is configured as unauthorized.

The finally goal is to protect as much as possible the computer networks of the organization of possible intrusion attacks.

These tools will be implemented on a Raspberry Pi 3 device and through a simulated environment we will perform different tests of vulnerabilities.

The final result of the tests carried out shows that in the majority of equipment that is used in real work environment they present serious security deficiencies. These can be solved with a correct management of the updates of the manufacturer of the operating system and periodically security scans.

Índex

Llistat de figures

1. Introducció

1.1 Problema a resoldre

El problema a resoldre en aquest projecte és la detecció i gestió de les possibles vulnerabilitats que poden existir en els diferents equips d'una organització que es troben accessibles des de l'exterior.

Avui dia cada vegada és més habitual que les organitzacions ofereixin als seus clients i empleats serveis de diferents tipus mitjançant internet. Aquest tipus de serveis públics requereixen una correcta protecció per tal d'evitar accessos no autoritzats als equips i a la xarxa interna.

Per protegir aquests equips exposats a internet necessitarem les funcions d'un sistema detector de vulnerabilitats, el qual realitzarà diferents anàlisis sobre els equips existents a la xarxa client. D'aquesta forma podrem dur a terme auditories de seguretat que ens reportarà un llistat de les debilitats trobades i que exposen la seguretat de la xarxa de computadors.

Segons les deteccions del detector de vulnerabilitats estudiarem les actuacions corresponents per tal d'evitar els riscos. A part, inclourem un sistema de detecció d'intrusions (IDS). Aquest sistema IDS registrarà tota activitat que sigui configurada com a no autoritzada, i en cas de detectar alguna intrusió, poder implementar les solucions corresponents per tal de defensar-se de l'atac.

L'objectiu final és protegir el màxim possible les xarxes de computadors de l'organització de possibles atacs d'intrusió.

1.2 Objectius

Per tal d'oferir una solució al problema plantejat en l'apartat anterior en aquest projecte és pretén integrar en un equip un sistema que ofereixi una solució de detecció d'intrusions i un sistema detector de vulnerabilitats. D'aquesta forma obtindrem un dispositiu que amb les personalitzacions oportunes pugui oferir els serveis plantejats a qualsevol organització que ho requereixi. Aquest equip romandrà destinat a la seu del client i disposarà de gestió remota.

També crearem un entorn simulat per poder comprovar el correcte funcionament del sistema implementat. Aquest entorn estarà format per una màquina virtual amb Windows Server 2012 R2, que realitzarà les funcions habituals de servidor, i també per una altra màquina virtual amb Windows 7 que realitzarà les funcions de client.
Per últim, i mitjançant els resultats del detector de vulnerabilitats sobre l'entorn simulat, estudiarem quines solucions s'han d'implementar per protegir els equips de possibles atacs.

1.3 Metodologia

Com a punt inicial del projecte ens documentarem amb tota la informació relacionada disponible sobre els sistemes detectors de vulnerabilitats i dels detectors d'intrusions. També recopilarem informació sobre possibles vulnerabilitats del nostre sistema simulat.

A continuació crearem el sistema simulat en dos màquines virtuals. Decidirem quines intrusions es volen detectar i crearem les regles adequades per a implementar-les al IDS.

Una vegada el nostre dispositiu sigui funcional comprovarem la funcionalitat del IDS amb les regles aplicades i elaborarem una auditoria de seguretat mitjançant el detector de vulnerabilitats. Per últim, realitzarem una proposta d'accions a dur a terme per tal de solucionar les vulnerabilitats detectades.

1.3.1 Recursos necessaris

Els recursos necessaris inclouen una màquina virtual del servidor amb el sistema simulat vulnerable Windows Server 2012 R2 i una altra amb el sistema simulat d'un equip punt de treball amb Windows 7 . Per tal de generar aquest entorn de treball necessitarem un equip, que en aquest projecte serà un Macbook Pro amb processador i5 de 2,4 Ghz, 8 Gb de RAM 1600Mhz DDR3 i disc SSD de 256 GB.

Per una altra banda l'equip a utilitzar per a la instal·lació del sistema detector de vulnerabilitats i el IDS serà una Raspberry Pi 3 [1]. Aquest equip ofereix els requeriments necessaris per dur a terme les funcions exposades i ofereix unes mides reduïdes a un cost molt baix. En aquest equip instal·larem una distribució adaptada per a processadors ARM de Kali Linux [2].

1.4 Tasques a realitzar

Les tasques que caldrà realitzar per assolir el projecte són les següents:

- Recerca d'informació eina Snort [3] (manuals i documentació específica).
- Documentació eines detector de vulnerabilitats (Nessus [4] i OpenVas [5]) per tal de decidir quina eina implementarem.
- Creació entorn simulat. Aquest estarà format per les màquines virtuals (servidor i client) amb el sistema vulnerable simulat.
- Preparació equip Raspberry Pi amb distribució Kali Linux.
- Instal·lació i posada en marxa de les eines escollides en el sistema Kali Linux.

- Elecció grup de regles Snort corresponents i implementar entorn gràfic.
- Realització proves alertes d'intrusió Snort.
- Detecció de les vulnerabilitats existents en el sistema simulat.
- Realització de proposta per tal de solucionar les vulnerabilitats detectades.
- Configuració d'un servei de connexió remota per tal de gestionar el dispositiu remotament.

Simultàniament a la realització de les tasques s'anirà documentant tota la informació per a la realització dels documents a entregar.

1.5 Planificació temporal

A continuació podem veure la planificació prevista per dur a terme el projecte:

Nom	Inici	Finalització	Durada
PAC 1 Pla de treball	20/09/17	06/10/17	13
Problema a resoldre	20/09/17	22/09/17	3
Objectius	25/09/17	25/09/17	1
Metodologia	26/09/17	26/09/17	1
Recursos necessaris	27/09/17	27/09/17	1
Tasques a realitzar	28/09/17	29/09/17	2
Estat de l'art	02/10/17	03/10/17	2
Planificació temporal	04/10/17	06/10/17	3
PAC 2	09/10/17	03/11/17	20
Documentació Snort	09/10/17	11/10/17	3
Documentació detectors vulnerabilitats	12/10/17	18/10/17	5
Sistema escollit	19/10/17	19/10/17	1
Preparació Raspberry amb Kali Linux	17/10/17	19/10/17	3
Instalació i posada en marxa eines escollides	20/10/17	25/10/17	4
Creació entorn simulat (servidor i client)	26/10/17	01/11/17	5
Reestructurar memòria	02/11/17	03/11/17	2

▼ PAC 3	06/11/17	01/12/17	20
Selecció regles Snort i entorn gràfic	06/11/17	10/11/17	5
Proves d'intrussió	13/11/17	15/11/17	3
Detecció vulnerabilitats	16/11/17	20/11/17	3
Proposta solució vulnerabilitats	21/11/17	24/11/17	4
Connexió remota	27/11/17	28/11/17	2
Reestructurar memòria	29/11/17	01/12/17	3
▼ PAC 4	04/12/17	03/01/18	23
Finalització apartats pendents	04/12/17	15/12/17	10
Restructuració i finalització memòria	18/12/17	03/01/18	13
▼ Presentació virtual	04/01/18	11/01/18	6
Disseny i creació presentació virtual	04/01/18	11/01/18	6

Figura 1. Taula planificació tasques

- PAC 1 i PAC 2

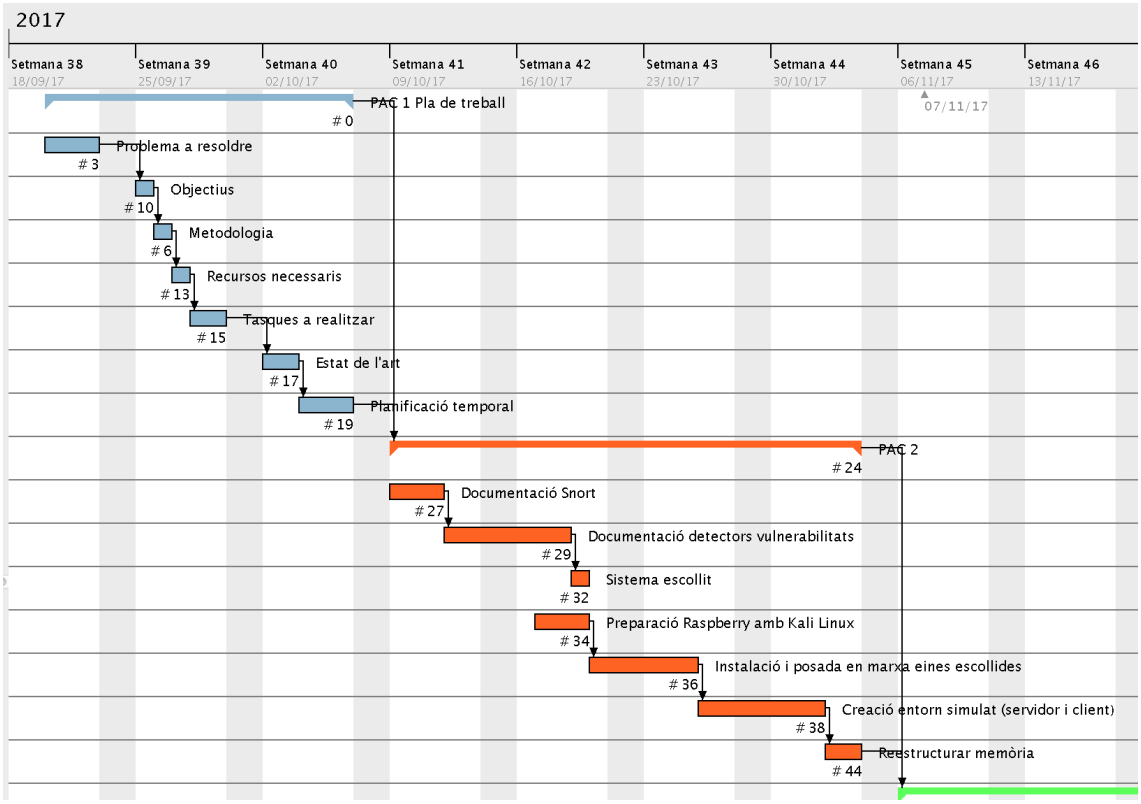


Figura 2. Diagrama de Gantt PAC 1 i PAC 2

- PAC 3 i PAC 4

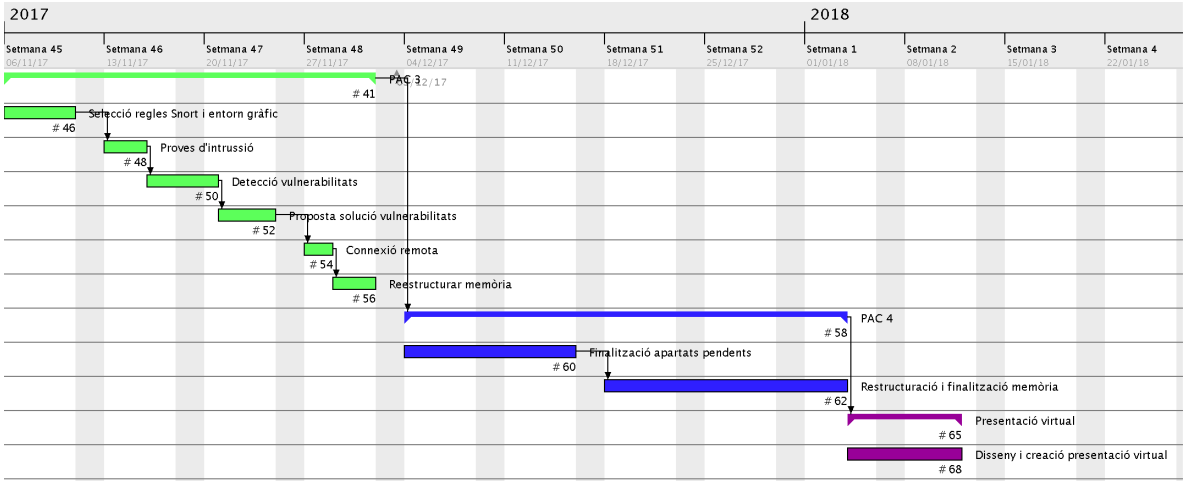


Figura 3. Diagrama de Gantt PAC 3 i PAC 4

1.6 Estat de l'art

Els entorns Kali Linux [2] estan preparats per a la realització de proves de penetració i d'auditories de seguretat amb les més de 600 eines que incorpora per defecte. Aquest entorn disposa de versions per a tot tipus de processadors de 32 bit, 64 bit i ARM. Aqueta distribució Linux disposa d'una gran comunitat d'usuaris i desenvolupadors que ofereix tota mena d'informació i millores. A la mateixa pàgina web podem trobar gran quantitat d'informació.

Snort [3] és una de les solucions de programari més utilitzades per a la identificació d'intrusions. Està disponible sota llicència OpenSource i funciona en plataformes Windows / Linux / UNIX. Disposa d'una gran quantitat de filtres ja definits i gràcies al seu constant creixement disposa de constants actualitzacions. Aquest IDS disposa d'una comunitat que ens ofereix una multitud de webs on trobar exemples de regles i configuracions.

També podem afirmar l'existència d'un gran número d'eines detectores de vulnerabilitats. D'entre aquestes eines destaquem les més utilitzades, una de pagament sota llicència (Nessus) i una altra de codi lliure (OpenVas). Nessus inicialment va estar disponible sota llicència OpenSource i es va convertir en tot un referent en eines de detecció de vulnerabilitats. En l'actualitat es va convertir a llicència privativa.
OpenVas és una variant de la eina Nessus quan aquesta va canviar el tipus de llicenciament. Disposa d'una comunitat important que aporta noves actualitzacions i a diferència de Nessus aquesta eina ofereix una versió per a processadors ARM.

Existeixen diversos projectes d'implementació del sistema IDS Snort en el dispositiu Raspberry Pi. Entre ells destaquem el projecte: Raspberry pi amb Kali Linux i Snort [6]

per obtenir un IDS portable i a baix cost. També destaquem un altre projecte que converteix una Raspberry Pi en un IDS amb Snort [7].

En resum, les eines esmentades reben el suport d'una gran comunitat d'usuaris que faciliten la publicació d'actualitzacions i ,per tant, ens facilitarà la realització del projecte plantejat per a la protecció de vulnerabilitats i detecció d'intrusions amb Snort.

2. Programari

2.1 Sistemes detectors de vulnerabilitats

Els sistemes detectors de vulnerabilitats són una de les eines més utilitzades tant per administradors de sistemes com per possibles atacants. Aquestes eines disposen d'un conjunt d'utilitats que permeten comprovar si un sistema és vulnerable a un ampli conjunt de problemes de seguretat. Una vegada detecta alguna de les possibles vulnerabilitats s'encarregarà d'informar de la seva existència i les seves possibles solucions. Dins dels detectors de seguretat destaquem les eines Nessus i OpenVas.

2.1.1 Nessus

Figura 4. Logo Nessus vulnerability scanner

2.1.1.1 Descripció

Nessus [4] és una eina basada en el model client-servidor que compta amb el seu propi protocol de comunicació. De la mateixa forma que els altres escàners de vulnerabilitats existents, les tasques d'exploració i proves d'atacs contra objectius són realitzats per el servidor (nessusd). Les altres tasques de control, generació d'informes i presentació
són gestionats per el client (nessus). D'aquesta forma Nessus ens permet una exploració proactiva de les deficiències de seguretat dels equips de la nostra xarxa. La major part de les alertes que reportarà Nessus es troben relacionades amb les següents deficiències:

- Utilització de servidors (daemons) no actualitzats que presenten deficiències de seguretat conegudes.
- Deficiències de seguretat relacionades amb una incorrecta configuració dels servidors.

- Deficiències de seguretat relacionades amb la implementació de la pila TCP/IP del equip remot.
- Utilització d'aplicacions des de servidors webs configurats incorrectament.
- Instal·lació de back doors, trojans, DDoS dèmons o altres serveis perillosos en sistemes de producció.

Les proves de vulnerabilitats, disponibles com a un llarg llistat de plugins, són escrits en NASL, un llenguatge de scripting optimitzat per a interaccions personalitzades a les xarxes.

En la seva última versió es troba disponible per als sistemes operatius: Microsoft Windows, macOS, Linux, FreeBSD, GPG Keys en les plataformes i386 i x86-64.

2.1.1.2 Història

Renaud Deraison va iniciar el projecte Nessus el 1998 per oferir a la comunitat un escàner remot de seguretat de codi obert. El 2005, Tenable Security, empresa de la que Renaud era cofundador, va canviar a llicència privativa del Nessus 3. En el 2008, Tenable va revisar la llicència per permetre als usuaris domèstics accedir sense restriccions als plugins i també va crear una llicència d'ús comercial.

2.1.2 OpenVas

Figura 5. Logo OpenVas

2.1.2.1 Descripció

OpenVas [5] (Open Vulnerability Assessment System) és una suite d'eines especialitzades en l'escaneig i gestió de vulnerabilitats de seguretat en sistemes informàtics. Les característiques principals són:

- Escaneig simultani de diferents nodes
- Suport SSL
- Suport WMI
- Escaneig automàtic programat
- Servidor web integrat
- Multiplataforma

Des de la versió 4.0 OpenVas permet l'actualització continua de la base de proves de vulnerabilitats de xarxa. Amb la versió 7 es va suprimir el mòdul d'administrador i el client d'escriptori per centrar-se en el seu client web.

Openvas és multiplataforma com Nessus, però en aquest cas també suporta processadors d'arquitectura ARM.

2.1.2.2 Història

OpenVas va ser inicialment anomenat GNessUS, per ser una variant de l'escàner de seguretat Nessus quan aquest va canviar el tipus de llicenciament. Va ser proposat com a sistema per dur a terme proves de penetració, pentester a l'organització Portcullis Computer Security. Després va ser anunciat com una solució de programari lliure per Tim Brown en Slashdot. Actualment és l'opció de programari lliure més completa gràcies a les continues actualitzacions de la seva base de dades de vulnerabilitats.

2.2 Sistemes detectors d'intrusions

Els sistemes IDS [9] (Intrusion Detection System) analitzen tots els datagrames IP que circulen en la xarxa amb l'objectiu de detectar el trànsit no autoritzat. Es poden implementar com a simples detectors de paquets (sniffers) per al monitoratge del trànsit d'una petita xarxa fins com a un sistema de detecció d'intrusos en temps real. Un dels IDS més reconeguts i utilitzats actualment és Snort.

2.2.1 Snort

Figura 6. Logo Snort

2.2.1.1 Descripció

Snort [3] és un NIDS de seguretat passiva, ja que, es limita a capturar els paquets i registrar-los. És open source i gràcies a la seva popularitat disposa d'una gran comunitat que col·labora en el seu desenvolupament. Algunes característiques rellevants:

- Registre en servidors de fitxers o de bases de dades de tot el trànsit capturat.
- Client IDS lleuger
- Multiplataforma
- Ofereix regles de detecció creades per la comunitat

Snort contrastarà tot el trànsit capturat amb les regles de detecció establertes per el usuari. Aquestes regles poden ser de creació pròpia, que no són res més que un conjunt de requisits on indiquem que ha d'activar una alarma si es compleixen.

2.2.1.2 Història

El 1998 va ser desenvolupat per Marty Roesch per a distribucions GNU/Linux amb el nom d'APE.
L'any 1999 és va introduir l'analitzador de signatures com a nova funcionalitat. D'aquesta forma es va començar a utilitzar com a IDS.
L'autor Marty Roesch va dedicar-se a partir d'aquell moment a temps complet en el desenvolupament de noves funcionalitats que milloressin la capacitat de configuració i facilitessin l'ús de Snort.

Marty va aconseguir fundar Sourcefire amb el finançament aconseguit. Actualment es distribueix la versió 3.0 Alpha 4.

Snort continua sent de codi lliure i promet seguir sent-ho per sempre.

2.3 Sistema escollit

Una vegada analitzats els diferents programaris possibles per dur a terme el nostre projecte, escollirem els que millor s'adaptin al maquinari escollit. Com s'indica en l'apartat "1.3.1 Recursos Necessaris" el maquinari escollit és un equip Raspberry Pi 3. Aquest dispositiu disposa d'un processador de quatre nuclis de 1,2 Ghz ARM de 64bits Cortex-A53 i 1 GB RAM.

L'arquitectura ARM del processador ens limitarà a la utilització de programari adaptat a aquests processadors.

Amb les característiques tècniques del maquinari escollit optarem per a la implementació en el projecte del següent programari:

- Sistema operatiu: Kali Linux Custom ARM for Raspberry Pi 2/3 (ref. [8]).
- Sistema detector de vulnerabilitats: OpenVas 8.0
- Sistema IDS: Snort

3. Preparació equips

3.1 Preparació dispositiu

Una vegada escollides les eines a implementar en el dispositiu realitzarem les instal·lacions oportunes. Primer realitzarem la instal·lació del sistema operatiu Kali Linux (Annex 8.1). Una vegada tenim la Raspberry operativa amb una IP assignada procedirem a instal·lar les eines OpenVas (Annex 8.2) i Snort (Annex 8.3).

A partir d'aquest moment podrem accedir al client OpenVas mitjançant la ip de l'equip i el port 9392. Snort també serà funcional quan configurem un grup de regles. A més, implementarem el framework Snorby per tal de disposar d'una interfície gràfica per visualitzar les alertes.

3.2 Preparació entorn simulat

Per a la creació de l'entorn simulat optarem per a la creació de dues màquines virtuals. Aquestes màquines han de representar els sistemes i configuracions més comuns en les organitzacions. D'aquesta manera l'estudi de les vulnerabilitats d'aquests equips seran més ajustades a la realitat. Una de les màquines virtuals realitzarà les funcions de servidor de l'organització. Aquest servidor funcionarà amb Windows Server 2012 R2, realitzarà funcions d'administració de directori actiu i DNS. També compartirà una unitat de xarxa per als treballadors a c:\Share.

L'altra màquina virtual amb Windows 7 Enterprise realitzarà les funcions d'equip client. Aquest equip disposarà d'algunes eines ofimàtiques habituals i tindrà accés a la unitat de xarxa compartida.

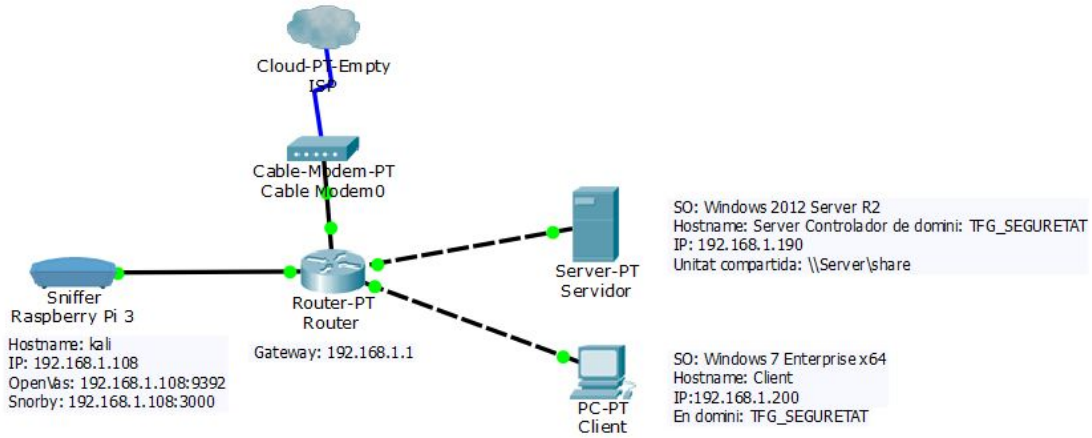


Figura 7. Topologia de xarxa dels equips

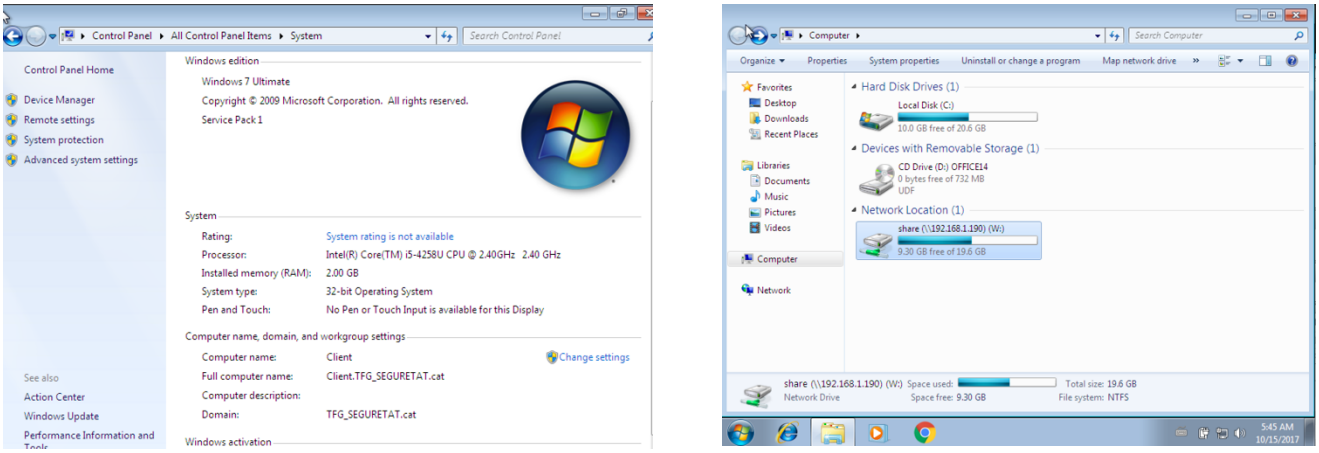


Figura 8. Configuracions equip client amb Windows 7 Enteprise

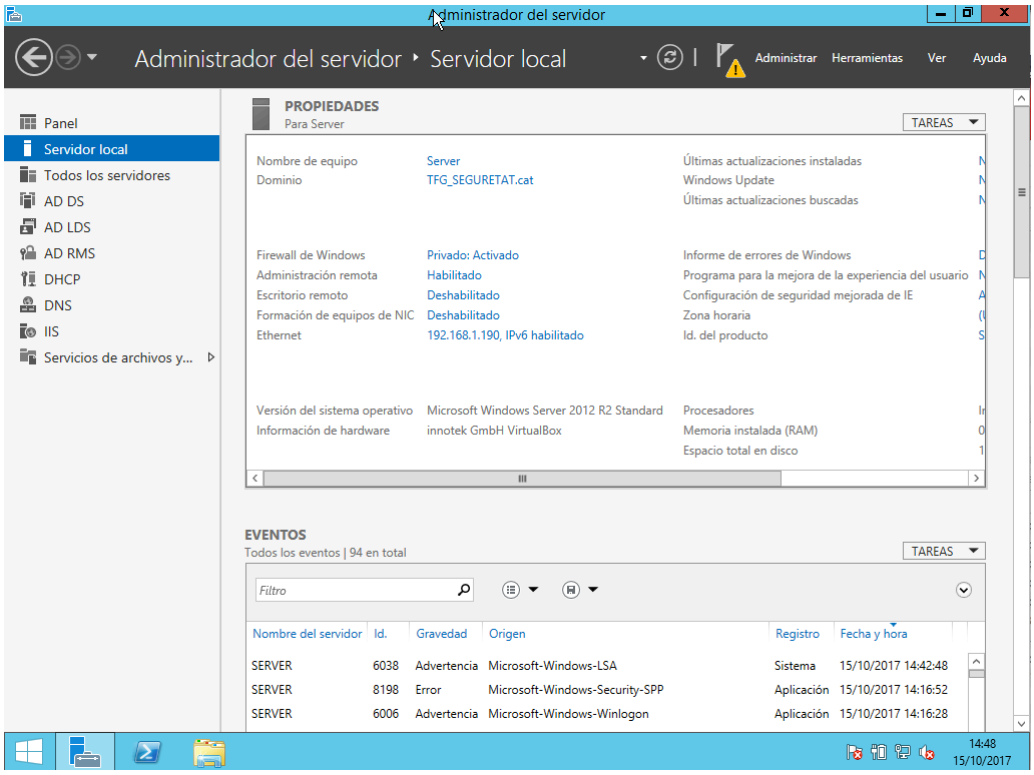


Figura 9. Configuració Domini al Servidor Windows Server 2012 R2

4. Personalització Snort

4.1 Regles Snort

Per a la utilització de regles optarem per les que ofereix la mateixa comunitat snort.org. Aquestes regles s'actualitzen regularment amb les noves amenaces a detectar. Per aquest motiu és molt recomanable actualitzar el més sovint possible l'arxiu de regles de Snort.
Aquesta tasca l'automatitzarem amb la utilització del programari PulledPork[18]. Aquesta aplicació s'encarrega de carregar les noves regles que descarrega directament des de la web oficial de Snort i aplicar-les en el nostre dispositiu.

Primerament configurarem l'arxiu snort.conf de Snort:

```
wget https://www.snort.org/downloads/snort/snort-2.9.11.tar.gz
cd /usr/local/src && wget --no-check-certificate https://www.snort.org/documents/185
-O snort.conf
tar xvfz snort-2.9.11.tar.gz
cd snort-2.9.11
./configure --enable-sourcefire; make; sudo make install

mkdir /usr/local/etc/snort /usr/local/etc/snort/rules /var/log/snort
/var/log/barnyard2 /usr/local/lib/snort_dynamicrules
touch /usr/local/etc/snort/rules/white_list.rules
/usr/local/etc/snort/rules/black_list.rules /usr/local/etc/snort/sid-msg.map

groupadd snort && useradd -g snort snort

cp /usr/local/src/snort-2.9.11/etc/*.conf* /usr/local/etc/snort
cp /usr/local/src/snort-2.9.11/etc/*.map /usr/local/etc/snort
cp /usr/local/src/snort.conf /usr/local/etc/snort
mkdir /var/log/snort
chown snort:snort /var/log/snort
nano /usr/local/etc/snort/snort.conf
```

Dins de snort.conf modificarem aquestes línies:

```
var RULE_PATH rules
var SO_RULE_PATH so_rules
var PREPROC_RULE_PATH preproc_rules
var WHITE_LIST_PATH rules
var BLACK_LIST_PATH rules

output unified2: filename snort.log, limit 128
```

Després necessitarem registrar-nos com a usuaris a snort.org per obtenir el nostre Oinkcode (codi d'usuari):

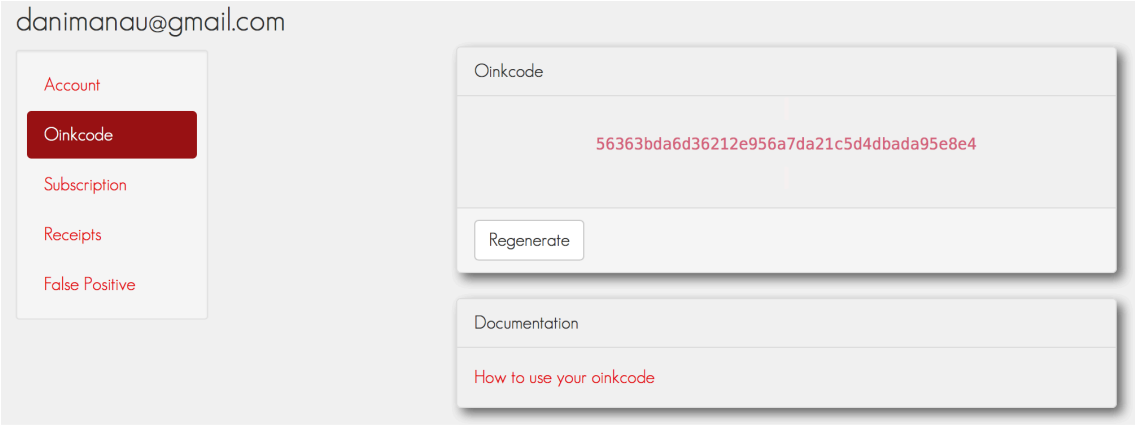


Figura 10. Registre a snort.org

A continuació les comandes per a la instal·lació i configuració de PulledPork:

```
cd /usr/local/src
git clone https://github.org/shirkdog/pulledpork
cd pulledpork
cp pulledpork.pl /urs/local/bin && cp etc/*.conf /usr/local/etc/snort
```

A continuació ens cal estar registrats i disposar del Oinkcode:

```
nano /usr/local/etc/snort/pulledpork.conf
```

Substituirem a la línia 19 <oinkcode> per el nostre codi.
A la línia 133 indiquem la nostra distribució:
distro=Debian-6-0

```
chmod +x /usr/local/bin/pulledpork.pl
mkdir /usr/local/etc/snort/rules/iplists
```

Per actualitzar les nostres regles per les més actualitzades:

```
/usr/local/src/pulledpork/pulledpork.pl -c /usr/local/etc/snort/pulledpork.conf -T -l
```

```
[root@kali:/usr/local/src/pulledpork# /usr/local/bin/pulledpork.pl -c /usr/local/etc/snort/pulledpork.conf -T -l

    https://github.com/shirkdog/pulledpork
     ,_____ ____
     `----,\    )
      `--==\\  /    PulledPork v0.7.3 - Making signature updates great again!
       `--==\\/
     .-~~~~-.Y|\\_  Copyright (C) 2009-2017 JJ Cummings, Michael Shirk
  @_/        /  66\_  and the PulledPork Team!
    |    \   \   _(")
     \   /-| ||'--'  Rules give me wings!
      \_\  \_\\
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Checking latest MD5 for snortrules-snapshot-29110.tar.gz....
Rules tarball download of snortrules-snapshot-29110.tar.gz....
        They Match
        Done!
Checking latest MD5 for community-rules.tar.gz....
Rules tarball download of community-rules.tar.gz....
        They Match
        Done!
IP Blacklist download of https://talosintelligence.com/documents/ip-blacklist....
Reading IP List...
Checking latest MD5 for opensource.gz....
Rules tarball download of opensource.gz....
        They Match
        Done!
Prepping rules from snortrules-snapshot-29110.tar.gz for work....
        Done!
Prepping rules from community-rules.tar.gz for work....
        Done!
Prepping rules from opensource.gz for work....
        Done!
Reading rules...
Writing Blacklist File /usr/local/etc/snort/rules/iplists/default.blacklist....
Writing Blacklist Version 1717724772 to /usr/local/etc/snort/rules/iplistsIPRVersion.dat....
Setting Flowbit State....
        Enabled 8 flowbits
        Done
Writing /usr/local/etc/snort/rules/snort.rules....
        Done
Generating sid-msg.map....
        Done
Writing v1 /usr/local/etc/snort/sid-msg.map....
        Done
Writing /var/log/sid_changes.log....
        Done
Rule Stats...
        New:-------32928
        Deleted:---0
        Enabled Rules:----10467
        Dropped Rules:----0
        Disabled Rules:---22461
        Total Rules:------32928
IP Blacklist Stats...
        Total IPs:-----2471

Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!
```

Figura 11. Exemple d'actualització de regles amb PulledPork

4.2 Entorn gràfic Snort

Per facilitar la visualització i gestió de totes les dades que recull Snort implementarem un dels diversos entorns gràfics desenvolupats per la comunitat.
Escollirem el GUI que sigui més adequat per al nostre maquinari, ja que, disposem d'uns recursos més limitats que amb un ordinador/servidor habitual.
Aquest és el cas del GUI Snorby que és compatible amb el maquinari de Raspberry Pi i requereix pocs recursos per al seu funcionament.

Snorby

Per a la seva instal·lació primer instal·larem totes les dependències:

```
apt-get install libmysqlclient-dev postgresql-server-dev-9.4  libpq-dev
wget https://ftp.psu.ac.th/pub/snort/libdnet-1.12.tgz
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz

tar xvfvz libdnet-1.12.tgz
cd libdnet-1.12
./configure "CFLAGS=-fPIC"
make && make install && make check
ln -s /usr/local/lib/libdnet.1.0.1 /usr/lib/libdnet.1

cd /usr/local/src
tar xvfz daq-2.0.6.tar.gz
cd daq-2.0.6
./configure
make && make install
```

Ara iniciem la instal·lació de Snorby:

```
wget https://github.com/Snorby/snorby/archive/master.zip
gem install bundler
bundle install
bundle exec rake snorby:setup
```

Configurem l'arxiu /snorby/config/database.yml:

```
# Snorby Database Configuration
#
# Please set your database password/user below
# NOTE: Indentation is important.
#
snorby: &snorby
  adapter: mysql
  username: snorby # usuari de mysql
  password: snorby # el mateix password del nostre usuari
  host: 127.0.0.1 # IP del equip on es troba la BBDD

development:
  database: snorby
  <<* snorby

production:
  database: snorby
  <<* snorby
```

Canviem el nom l'arxiu /snorby/config/snorby_config.yml.example a snorby/config/snorby_config.yml

Finalment, podrem iniciar snorby per realitzar proves. Abans caldrà que MySQL estigui iniciat:

```
/etc/init.d/mysql start
cd /snorby
bundle exec rails server -e production
```

Una vegada iniciat podrem accedir a la interfície de snorby conectant-nos a http://192.168.1.108:300 amb les credencials snorby@example.com / snorby.

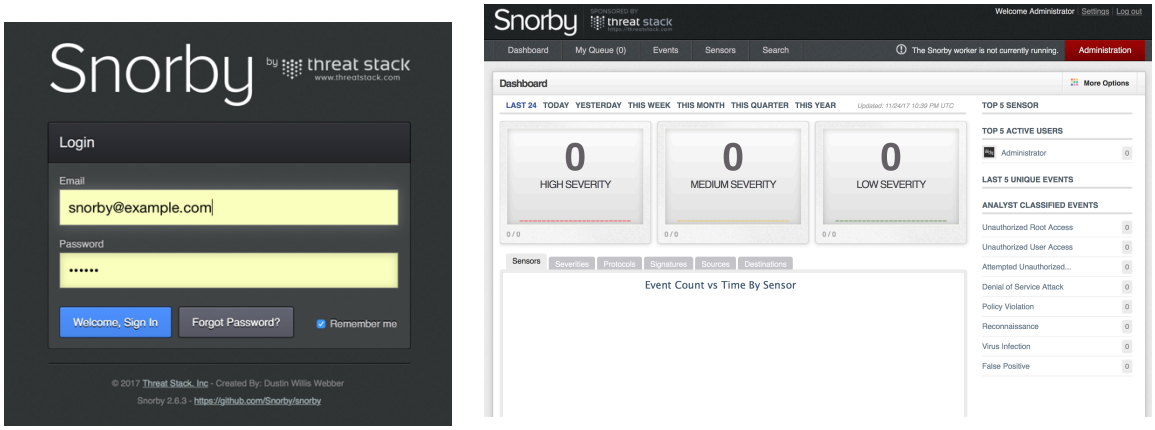


Figura 12. Accés a Snorby i el seu dashboard.

Barnyard2

Realitzarem una altra instal·lació necessària per moure les alertes generades per Snort a la BBDD i també per a millorar-ne el rendiment, es tracta del programari Barnyard2 [17].
En entorns on el IDS Snort ha de processar una gran quantitat de dades és molt possible que el rendiment es vegi afectat i finalment descarti alguns paquets de dades. Això es produeix per que Snort no processa el següent paquet fins que no finalitza l'escriptura de l'alerta a la base de dades. Aquest procés és lent, ja que, requereix una connexió TCP i un insert a la BBDD per cada alerta.
Per generar alertes més ràpidament Snort utilitza el format unified2 [17] per escriure les dades en un arxiu local.
Per per tal de traslladar aquestes dades a la base de dades utilitzarem Barnyard2 [17], que és un intèrpret open source del fitxers de sortida en format unified2.

Per instal·lar i configurar Barnyard2 utilitzarem les següents comandes:

```
cd /usr/local/src && wget https://github.com/firnsy/barnyard2/archive/master.tar.gz
tar -zxf master.tar.gz && cd barnyard2-*
autoreconf -fvi -I ./m4 && ./configure --with-mysql --with-mysql-
libraries=/usr/lib/arm-linux-gnueabihf/ && make && make install
mv /usr/local/etc/barnyard2.conf /usr/local/etc/snort
cp schemas/create_mysql /usr/local/src
nano /usr/local/etc/snort/barnyard2.conf
```

Configurarem l'arxiu barnyard2.conf modificant les línies 27, 28, 29 i 30:

```
# set the appropiate paths to the file(s) your Snort process is using.
config reference_file:      /usr/local/etc/snort/reference.config
config classification_file: /usr/local/etc/snort/classification.config
config gen_file:            /usr/local/etc/snort/gen-msg.map
config sid_file:            /usr/local/etc/snort/sid-msg.map
```

Per últim afegim al arxiu barnyard2.conf una última línia amb les dades de la BBDD:

```
output database: log, mysql, user=snorby password=snorby dbname=snorby host=127.0.0.1
```

A continuació configurarem la base de dades on s'emmagatzemen les dades recollides per Snort.

```
sudo service mysql start
mysql –u root –p

create database snorby;
grant CREATE, INSERT, SELECT, DELETE, UPDATE on snorby.* to snorby@localhost;
SET PASSWORD FOR snorby@localhost=PASSWORD('snorby');
use snorby;
source /usr/local/src/create_mysql
```

Podem observar les taules creades:

```
show tables;
```

```
MariaDB [snorby]> show tables
    -> ;
+--------------------+
| Tables_in_snorby   |
+--------------------+
| agent_asset_names  |
| aggregated_events  |
| asset_names        |
| caches             |
| classifications    |
| data               |
| delayed_jobs       |
| detail             |
| encoding           |
| event              |
| events_with_join   |
| favorites          |
| icmphdr            |
| iphdr              |
| lookups            |
| notes              |
| notifications      |
| opt                |
| reference          |
| reference_system   |
| schema             |
| search             |
| sensor             |
| settings           |
| severities         |
| sig_class          |
| sig_reference      |
| signature          |
| tcphdr             |
| udphdr             |
| users              |
+--------------------+
31 rows in set (0.00 sec)
```

Figura 13. Taules creades a la BBDD per a Snorby

Com a últim pas per iniciar el sistema Snort amb Snorby cal iniciar els diferents components en el següent ordre:

```
#comprovem actualitzacions de regles
/usr/local/src/pulledpork/pulledpork.pl -c /usr/local/etc/snort/pulledpork.conf -T –l

/etc/init.d/mysql start                                    # iniciem la BBDD

snort -c /usr/local/etc/snort/snort.conf -i eth0           # iniciem Snort en mode IDS

#iniciem Barnyard2
barnyard2 -c /usr/local/etc/snort/barnyard2.conf -d /var/log/snort -f snort.log -w
/var/log/snort/barnyard2.waldo


cd /snorby                                                 #anem al directori Snorby
bundle exec rails server -e production                     #iniciem Snorby
```

Una vegada tots els serveis estiguin en marxa podrem observar que a Snorby ja apareixen dades sobre les alertes detectades per Snort:

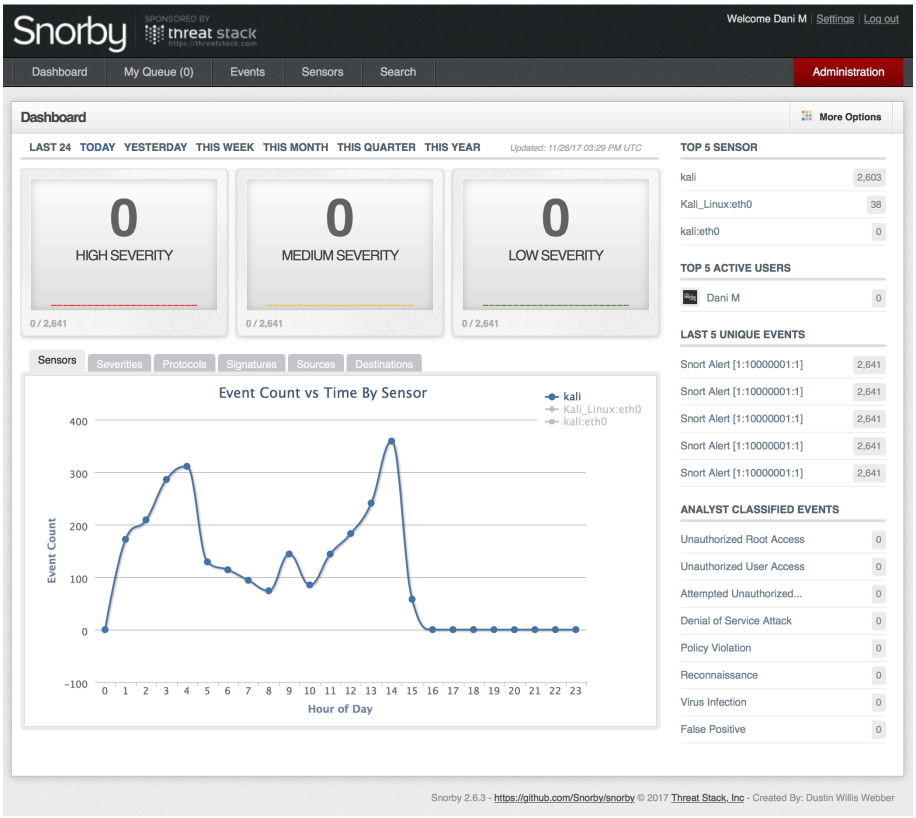


Figura 14. Dashboard Snorby amb alertes Snort

5. Estudi vulnerabilitats

5.1 Detecció vulnerabilitats

Una vegada tenim el nostre dispositiu amb OpenVas connectat a la mateixa xarxa que els dispositius a escanejar ja podem iniciar la detecció a cada equip. Per això entrarem al client web mitjançant la ip 192.168.1.108:9392 i entrem al sistema.

Una vegada al dashboard anirem al menú: SCANS > TASKS > NEW TASK.

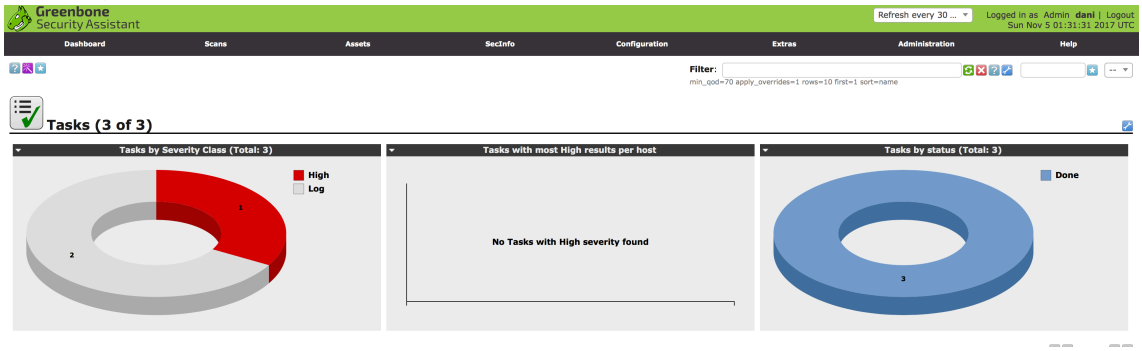


Figura 15. Dashboard OpenVas

Iniciarem analitzant les vulnerabilitats de l'equip simulat client amb Windows 7 i amb ip 192.168.1.200. Dins de la nova tasca crearem un nou "Target".

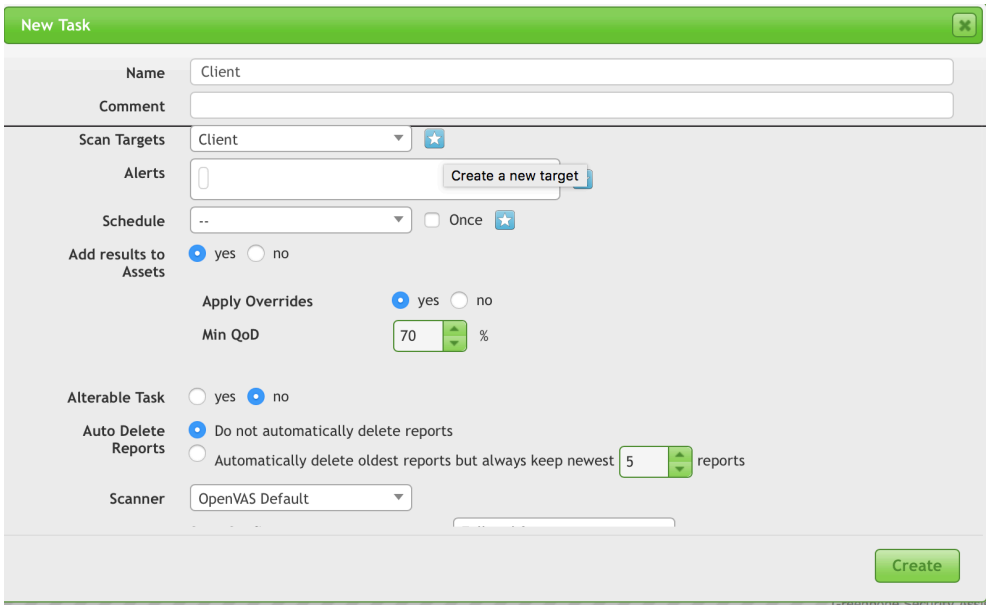


Figura 16. Configuració tasca nova

En aquesta finestra configurarem "Create a new target" amb la ip del equip de la següent manera:

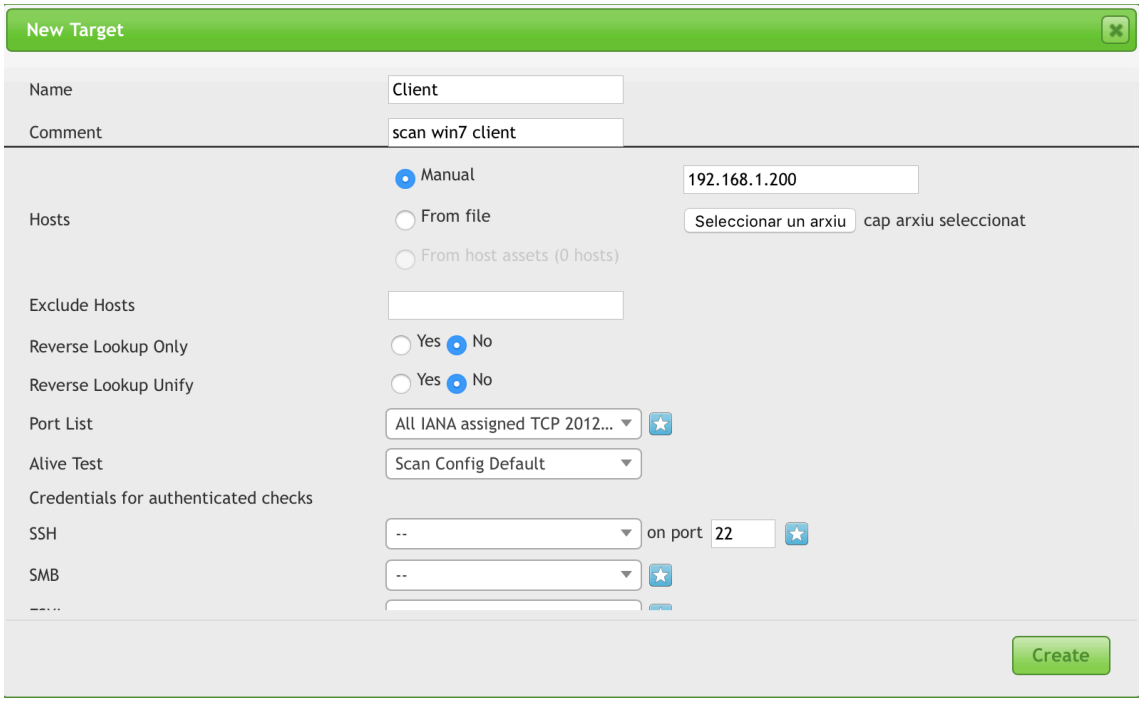


Figura 17. Configuració New Target

Deixem per defecte els valors del llistat de ports IANA ja que és el més complet. Una vegada creada la tasca ja serà accessible des del llistat de tasques:

Figura 18. Llistat de tasques creades

Podem iniciar la tasca, aturar-la, esborrar-la o modificar-la des del menú d'accions disponibles a la dreta. Una vegada complert podrem accedir al report corresponent.

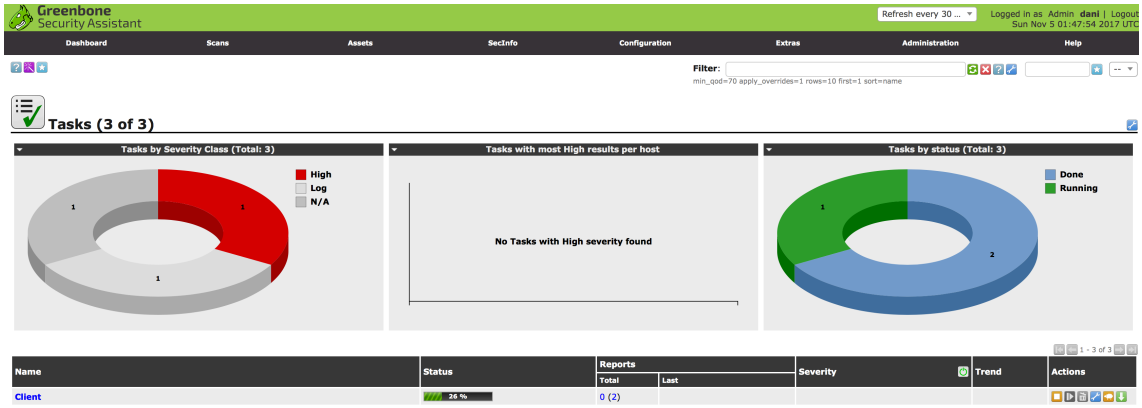


Figura 19. Llistat tasques amb les diferents accions i percentatge completat

Una vegada finalitzi la tasca ja podrem accedir als resultats:

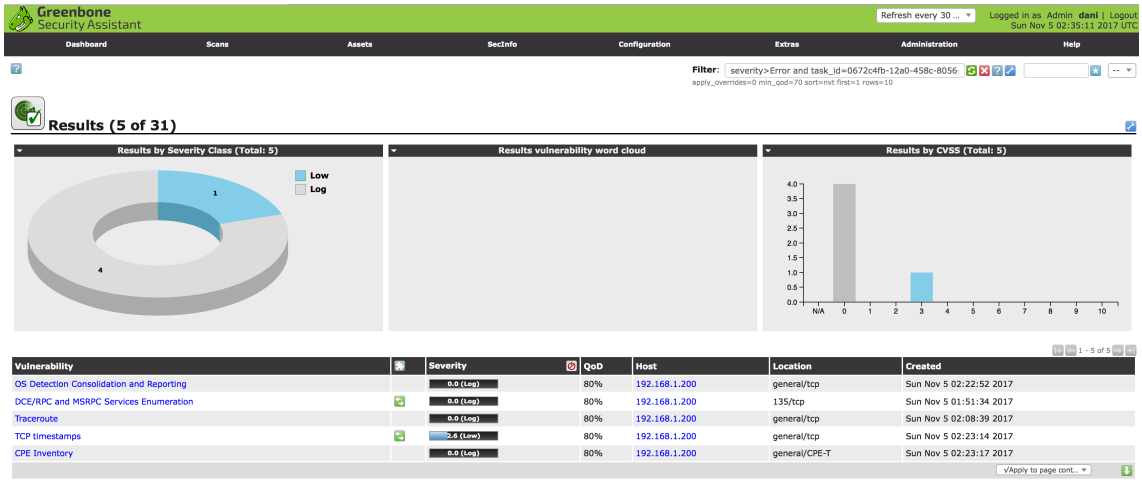


Figura 20. Resultats escaneig Client Windows 7 Enterprise

Després d'analitzar els resultats obtinguts podem concloure que quatre dels cinc avisos ens informen que un possible atacant serà capaç d'identificar quin sistema estem utilitzant mitjançant el seu fingerprint. Aquestes alertes són inherents al sistema operatiu i no podrem oferir cap solució per a la seva ocultació a un possible atacant. Per aquest motiu el sistema OpenVas les classifica com alertes tipus log:

- OS Detection Consolidation and Reporting
- DCE/RPC and MSRPC Services Enumeration
- Traceroute
- CPE Inventory

Per una altra banda l'anàlisi ens informa d'una vulnerabilitat considerada "Low" o lleu. En aquest cas el sistema atacant podrà calcular el "uptime" (temps d'activitat) de l'equip. En la seva descripció ens informa de la seva afectació i possibles solucions:

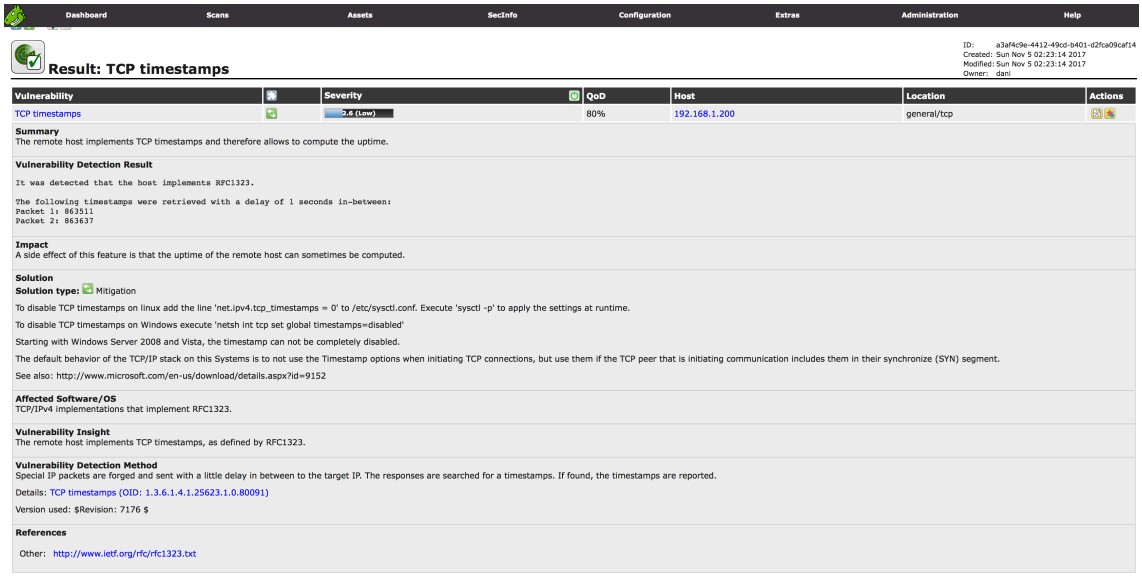
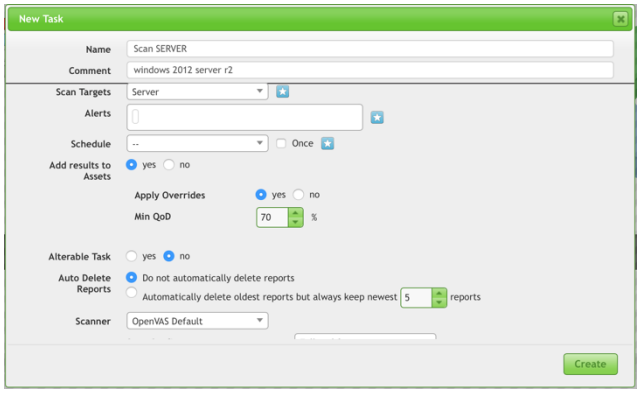


Figura 21. Vulnerabilitat TCP timestamps

En la següent part de l'anàlisi de vulnerabilitats escanejarem el servidor simulat amb Windows 2012 server. En aquest cas tornarem a crear una nova tasca, però en aquest cas indicarem la ip del servidor (192.168.1.190) a la finestra "New target".



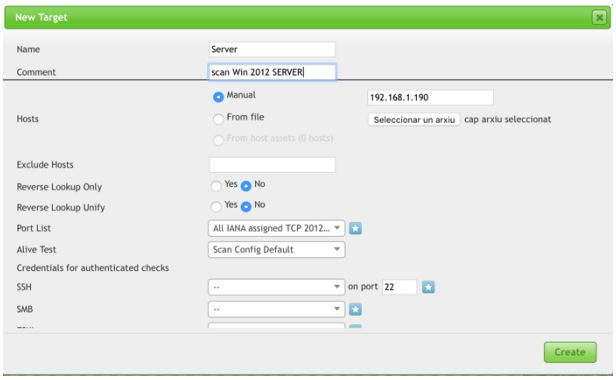


Figura 22. Nova tasca amb per analitzar el servidor

És recomanable identificar correctament les diferents tasques per a poder reutilitzar-les sobre els equips desitjats. Una vegada creada la tasca podrem accedir als resultats i a les diferents accions des del llistat de tasques.

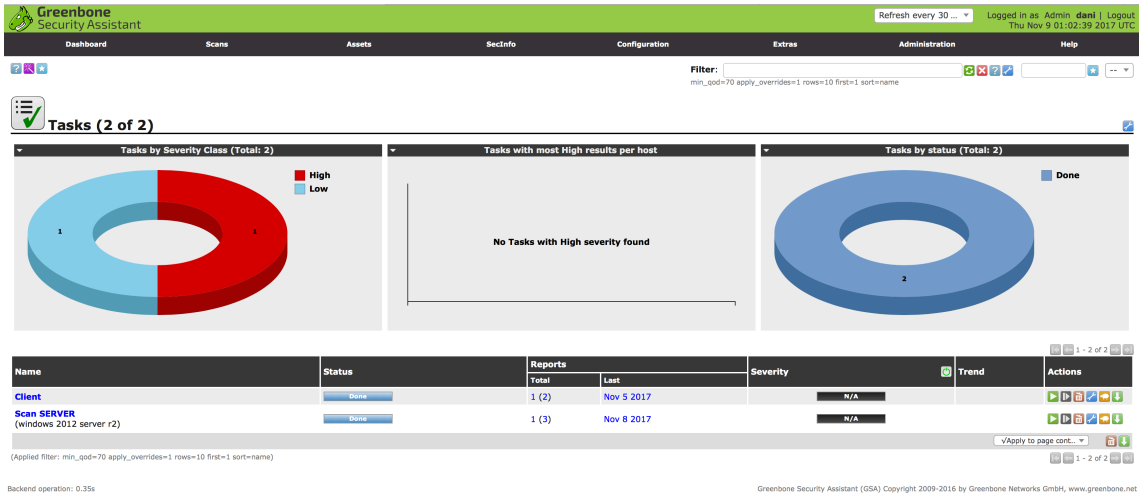


Figura 23. Llistat de tasques

Comprovem que els resultats de l'anàlisi en el servidor ens informa de dos vulnerabilitats greus i tres vulnerabilitats més de nivell moderat.

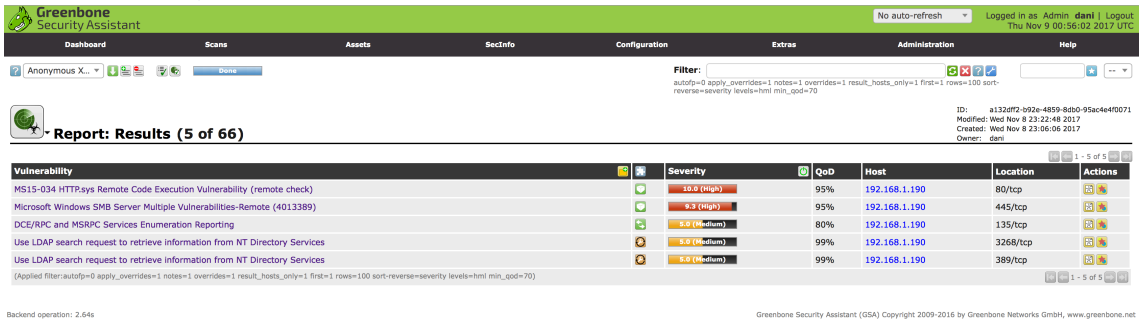


Figura 24. Resultats anàlisi vulnerabilitats Servidor

La primera vulnerabilitat greu ens informa d'un error reportat per el desenvolupador del sistema operatiu, Microsoft, que disposa de solució mitjançant actualització.

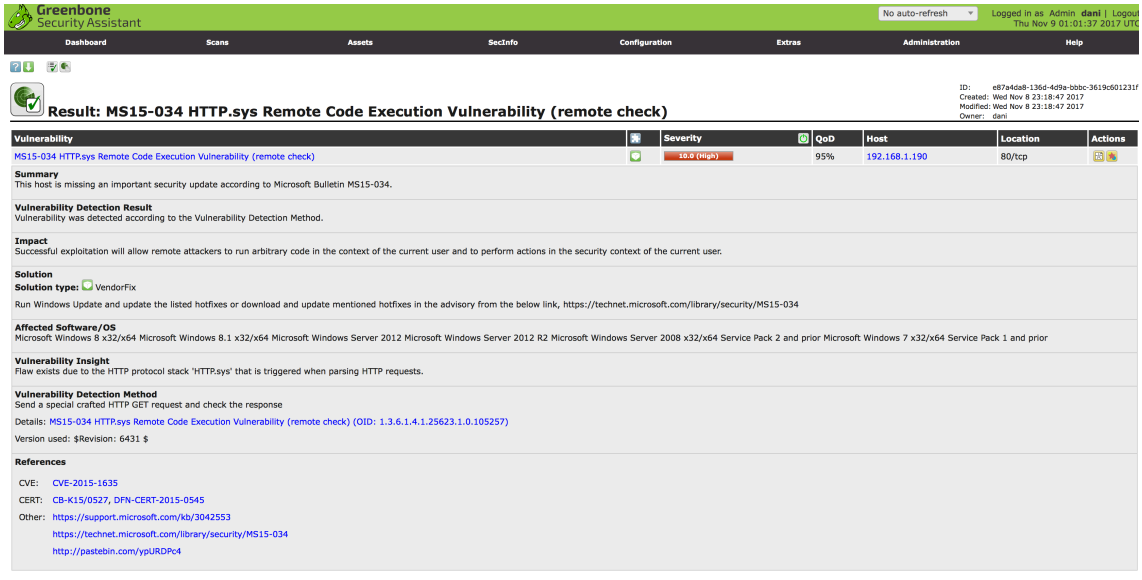


Figura 25. Vulnerabilitat MS15-034 HTTP.sys execució de codi remot

El segon cas greu també reporta una vulnerabilitat reportada per Microsoft que permetria als atacants executar codi en el servidor mitjançant exploit que afecta el servidor Samba.

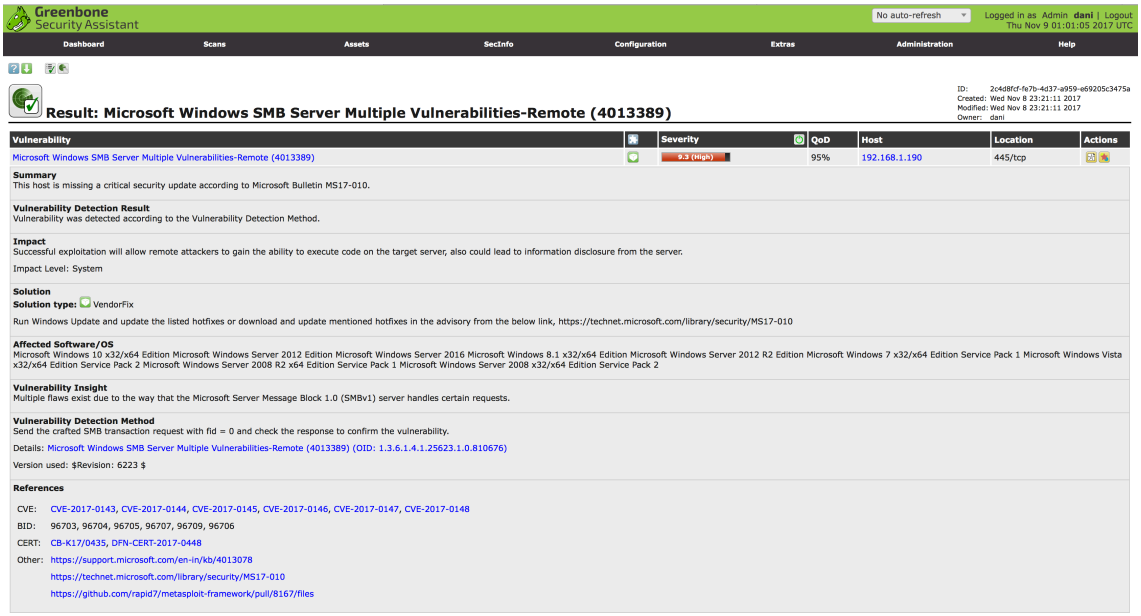


Figura 26. Vulnerabilitats SMB Server 4013389

El següent avís de seguretat de grau mitjà està relacionat amb la possibilitat d'obtenció d'informació a partir d'una consulta LDAP.

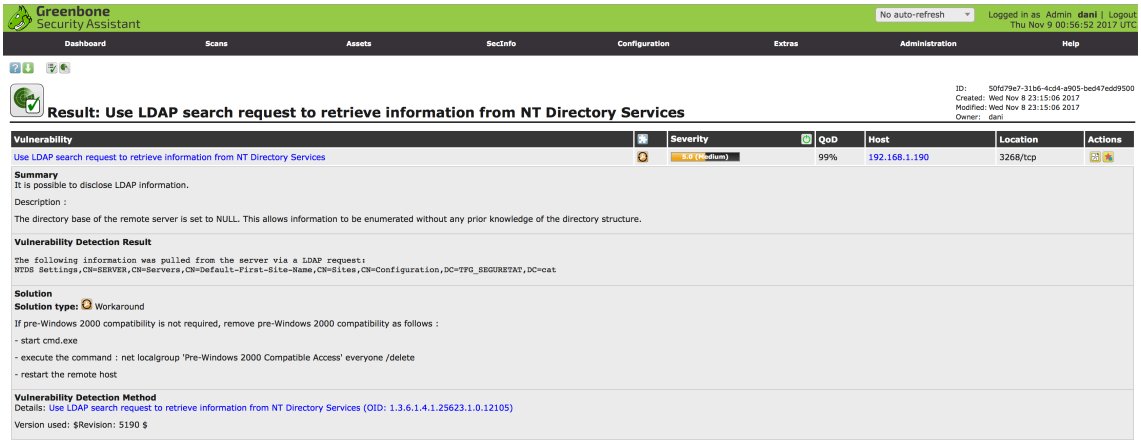


Figura 27. Vulnerabilitat divulgació informació LDAP

Aquesta segona vulnerabilitat correspon al mateix tipus de feblesa (obtenir informació mitjançant consulta LDAP). En aquest cas aquestes consultes es realitzen pel port 389 TCP.

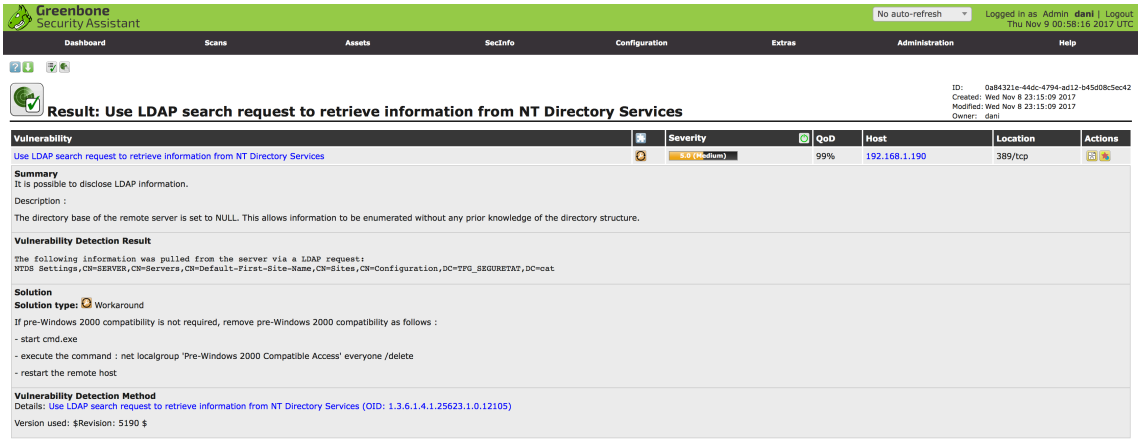


Figura 28. Vulnerabilitat divulgació informació LDAP 2

Per últim, els serveis RPC i DCE en funcionament en el servidor poden ser enumerats mitjançant una connexió al port 135 i les consultes corresponents.

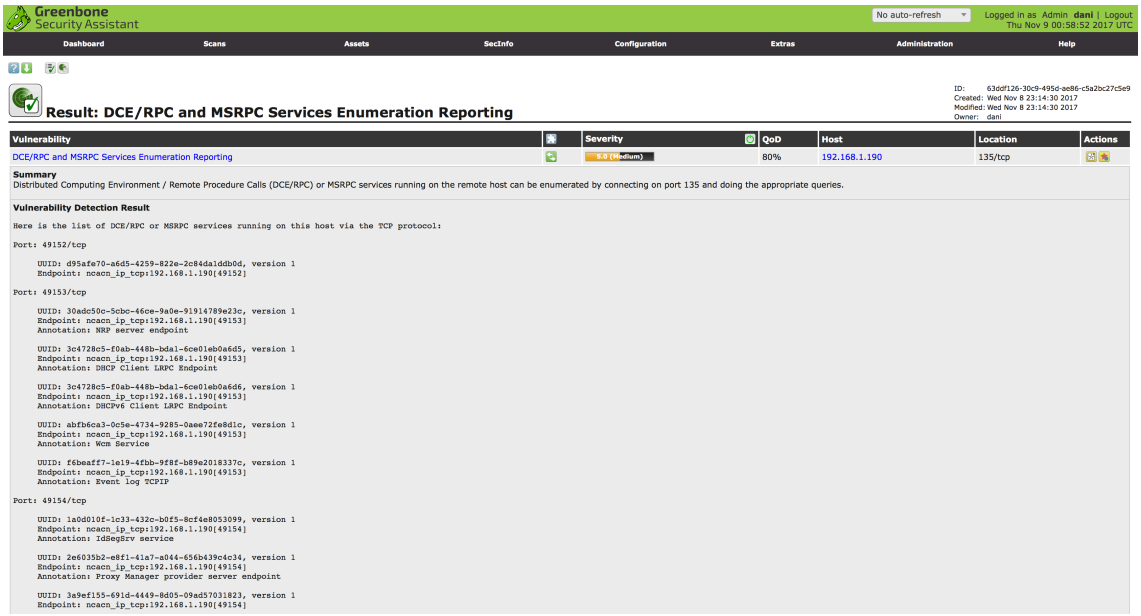


Figura 29. Vulnerabilitat enumeració serveis DCE/RPC

Figura 30. Vulnerabilitat enumeració serveis DCE/RPC 2

6.2 Proposta solució vulnerabilitats

Una vegada examinades les vulnerabilitats detectades en els dos equips simulats proposarem un llistat d'accions per dur a terme per tal de corregir aquestes debilitats.

- **Equip simulat Client (Windows 7 Enterprise)**

  - Desactivar timestamps TCP de Windows:
    - Netsh int tcp set global timestamps=disabled
    - http://www.microsoft.com/en-us/download/details.aspx?id=9152

  - Activar Windows Update per a futures actualitzacions de seguretat importants.

- **Equip simulat Servidor (Windows 2012 Server R2)**

  - Instal·lació actualització KB3042553 de https://support.microsoft.com/kb/3042553

  - Instal·lació actualització KB4013078 de https://support.microsoft.com/en-in/kb/4013078

  - Desinstal·lar compatibilitat pre-Windows 2000:
    - Inici > cmd.exe
    - Net localgroup 'Pre-Windows 2000 Compatible Access' everyone /delete

- Reiniciar l'equip

- Protegir els següents ports de connexions externes mitjançant regles al Firewall:
  - Rang de ports TCP: 49152 – 49159. Corresponen als ports que normalment s'assignen de forma dinàmica a les aplicacions del client. També s'utilitzen en connexions peer to peer (P2P).

- Activar Windows Update per a futures actualitzacions de seguretat importants.

6. Gestió remota

L'objectiu d'aquest dispositiu que incorpora el detector de vulnerabilitats OpenVas i el IDS Snort és la seva instal·lació en les organitzacions que ho requereixin. Per aquest motiu, al estar en ubicacions externes, requerirem d'un sistema de connexió remota per tal de dur a terme les tasques de gestió i manteniment necessàries.

Utilitzarem TeamViewer per tal de portar aquesta gestió remota. La seva instal·lació i configuració en en sistema Kali Linux és el següent:

```
apt-get install teamviewer
```

Una vegada instal·lat podem veure el codi (ID) necessari per a la connexió remota mitjançant:

```
Teamviewer -info
```

```
root@kali:~# teamviewer -info

 TeamViewer                          13.0.3057  (DEB)

 TeamViewer ID:                       291999389

 teamviewerd status                  ● teamviewerd.service - TeamViewer remote control daemon
   Loaded: loaded (/etc/systemd/system/teamviewerd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2017-11-25 15:02:38 UTC; 1 day 12h ago
  Process: 602 ExecStart=/opt/teamviewer/tv_bin/teamviewerd -d (code=exited, status=0/SUCCESS)
Main PID: 610 (teamviewerd)
   CGroup: /system.slice/teamviewerd.service
           └─610 /opt/teamviewer/tv_bin/teamviewerd -d
```

Figura 31. Teamviewer -info

Per últim configurem la contrasenya per accedir a l'equip:

```
sudo teamviewer passwd <password>
```

Amb aquestes dades ja podrem accedir a l'equip des de qualsevol dispositiu amb connexió a internet.

7. Conclusions

Finalitzat el projecte, i amb tot el treball realitzat, crec que s'han aconseguit els objectius plantejats inicialment. També s'han adquirit els coneixements necessaris per poder dissenyar un entorn simulat així com la configuració i posada en marxa d'un dispositiu que integri les eines proposades.

Inicialment es preveia la realització de les proves de vulnerabilitats amb un únic equip simulat, però es va considerar ampliar la mostra d'equips simulats per obtenir un resultat de les deteccions de vulnerabilitats que sigui el més pròxim a un entorn real. Aquesta ampliació ha permès analitzar un equip típic que realitza les funcions de servidor d'una petita organització amb un directori actiu d'usuaris i unitats de xarxa compartides així com un altra equip punt de treball que depèn del servidor on l'usuari realitza tasques ofimàtiques i de gestió.

Observant les vulnerabilitats detectades podem concloure que el servidor a presentat moltes més vulnerabilitats crítiques. Aquest fet es correspon a que es tracta d'un sistema operatiu orientat a oferir múltiples serveis de xarxa i cadascun d'ells requereix l'obertura d'una possible via d'entrada. Per aquest motiu és rellevant la correcta configuració dels serveis que s'utilitzaran així com tancar els serveis que no siguin necessaris.

També, a partir dels resultats de l'anàlisi extraiem la gran importància de mantenir els sistemes actualitzats amb les últimes correccions dels fabricants. Una correcta configuració d'aquestes actualitzacions evitarà exposar els sistemes a vulnerabilitats crítiques ja conegudes per la comunitat i corregides per els desenvolupadors del sistema operatiu.

Per una altra banda, es va modificar la implementació del IDS Snort amb la incorporació d'una interfície d'accés mitjançant web. D'aquesta manera millorem la visualització de les dades recollides per Snort per poder afrontar més ràpidament qualsevol alerta d'intrusió detectada.

Resumint, aquest projecte ens ha permès comprovar les múltiples deficiències de seguretat que poden incorporar els nostres equips informàtics. L'aparició constant de noves vulnerabilitats ens obliga a disposar d'eines que ens ajudin a comprovar la seguretat dels nostres equips. Amb els procediments inclosos en aquest projecte disposarem d'un dispositiu que proporcionarà les eines de seguretat per detectar vulnerabilitats i intrusions a qualsevol organització que ho requereixi. D'aquesta forma millorarem la seguretat davant possibles atacs informàtics i ajudarem al desenvolupament de xarxes de computadors més segures.

8. Annexos

8.1 Preparació Raspberry Pi amb Kali Linux

1.- Preparació SD:

- Formatació targeta SD amb SD Formatter a FAT.
- Descarrega imatge Kali Linux for ARM RaspberryPi (https://www.offensive-security.com/kali-linux-arm-images/)
- Càrrega imatge a SD (ApplePi-Baker):

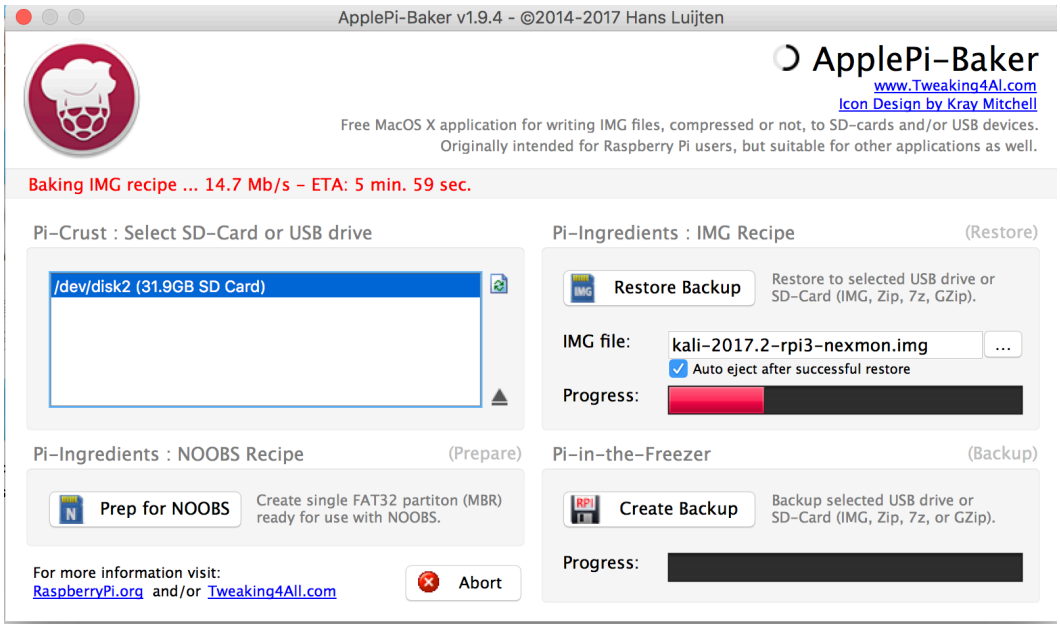


Figura 32. ApplePi-Baker Càrrega imatge a SD

2.- Instalació SO:

- Una vegada iniciem la Raspberry Pi amb la SD s'inicia la instal·lació del sistema operatiu.
- Una primera mesura de seguretat és canviar el password del usuari root:
  - Comanda "passwd"

Una vegada tenim el sistema operatiu instal·lat ja podrem accedir mitjançant la seva IP amb una connexió SSH.

- Expansió de partició a la totalitat de la targeta SD (32GB). La imatge del Kali Linux ve per defecte en 8Gb:

    - Utilitzem eina Parted:

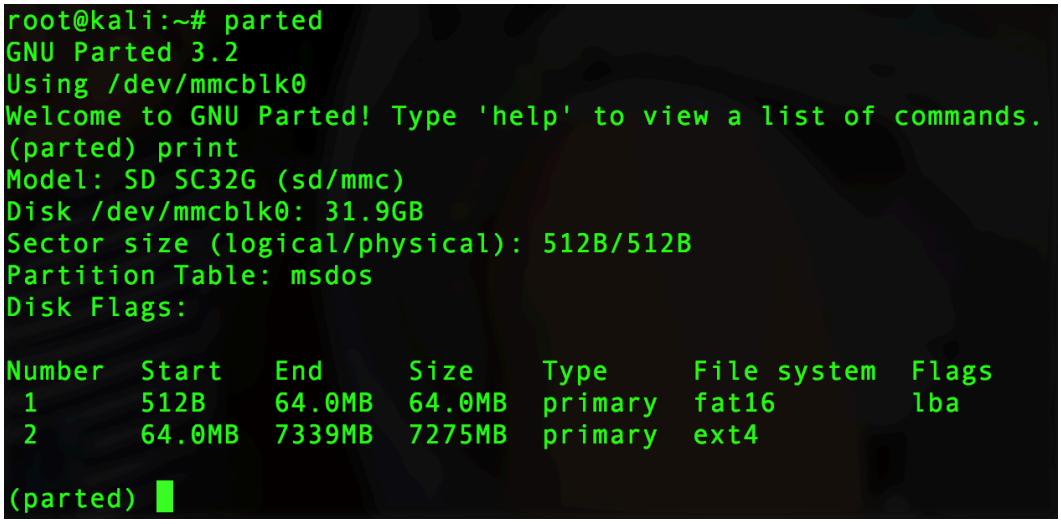


Figura 33. Parted mostra particions de la targeta SD.

Comandes:

- ***print***: mostra particions
- ***resizepart*** > ***partition***: 2 > ***end***: 31900 > ***quit***

Per a canviar la mida de les particions també es poden utilitzar altres eines amb interfície gràfica com gparted, però per tal de minimitzar recursos optem per fer ús de les eines incloses a Kali Linux.

Per últim actualitzem el llistat de paquets i repositoris amb: ***sudo apt-get update.***

8.2 Instal·lació OpenVas

Mitjançant una connexió SSH amb la Raspberry Pi realitzarem la instal·lació del sistema detector de vulnerabilitats OpenVas.

- Instalació Openvas: ***sudo apt-get install openvas***
  sudo openvas-setup

- Comprovació instal·lació correcta: ***openvas-check-setup***

- Creació usuari: ***sudo openvasmd –create-user=dani –role=Admin***
  sudo openvasmd –user=dani –new-password=’*’***

- Inici i aturada servei: ***sudo openvas-start*** | ***sudo openvas-stop***

Amb ***netstat –antp*** podem veure que el servei tcp de OpenVas ja es troba disponible a la ip 192.168.1.108:9390:

```
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 27 08:20:46 2017 from 192.168.1.104
root@kali:~# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.1.108:9390      0.0.0.0:*               LISTEN      1503/openvasmd
tcp        0      0 192.168.1.108:80        0.0.0.0:*               LISTEN      1504/gsad
tcp        0      0 192.168.1.108:9392      0.0.0.0:*               LISTEN      1488/gsad
tcp        0      0 127.0.0.1:5941          0.0.0.0:*               LISTEN      420/teamviewerd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      395/sshd
tcp        0      0 127.0.0.1:5941          127.0.0.1:60994         ESTABLISHED 420/teamviewerd
tcp        0      0 127.0.0.1:60994         127.0.0.1:5941          ESTABLISHED 600/TeamViewer
tcp        0    196 192.168.1.108:22        192.168.1.104:64458     ESTABLISHED 15147/sshd: root@pt
tcp6       0      0 :::22                   :::*                    LISTEN      395/sshd
```

Figura 34. Serveis tcp

Una vegada finalitzat i creat un usuari ja podrem accedir mitjançant el client Greenbone Security Assistant. Per accedir utilitzarem un navegador web amb la ip de l'equip i mitjançant el port 9392.

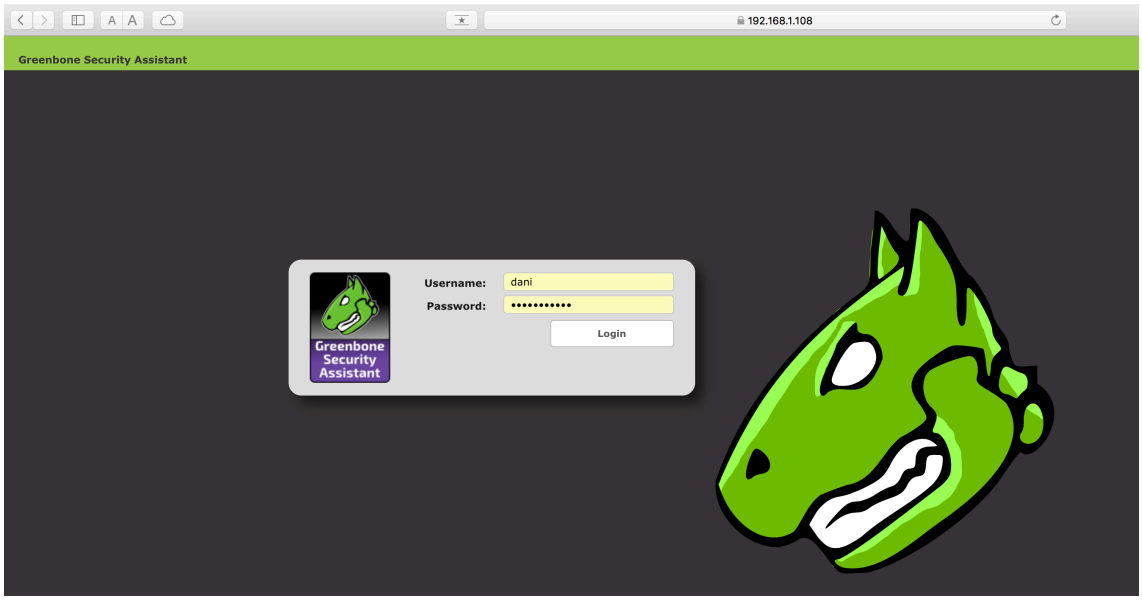


Figura 35. Accés al client Greenbone de OpenVas

Per accedir al portal web amb una IP fixa des de qualsevol equip dins de la mateixa xarxa seguirem els passos següents:

- Modificació dels serveis ***greenbone-security-assitant.service, openvas-manager.service, openvas-scanner.service*** amb l'ordre ***sed*** per reemplaçar la IP local 127.0.0.1 amb la IP que desitgem, en aquest cas 192.168.1.108:
    - ***sed –e 's/127.0.0.1/192.168.1.108/g' greenbone-security-assistant.service openvas-manager.service openvas-scanner.service -i***
- Després reiniciarem els serveis i daemons:

- ***systemctl daemon-reload***
- ***systemctl restart greenbone-security-assistant.service openvas-manager.service openvas-scanner.service***

Actualització de NVT (net vulnerability test) disponibles per al scanner:

- ***openvasmd –update && openvasmd –rebuild***
- ***service openvas-scanner restart***

Una vegada finalitzat totes les configuracions comprovem l'accés al client Greenbone:

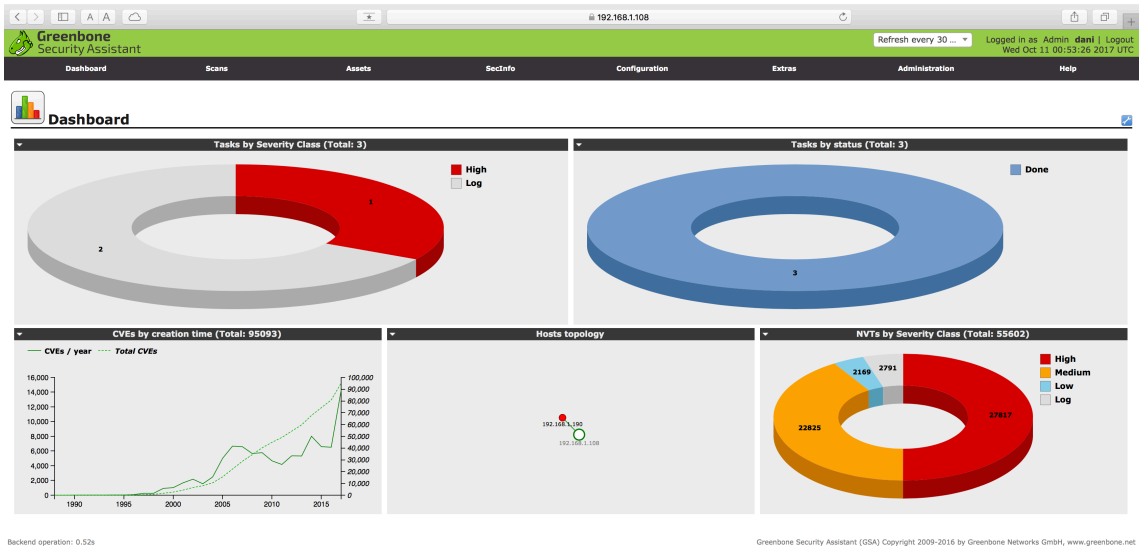


Figura 36. Dashboard de Greenbone Security Assitant

8.3 Instal·lació Snort

Amb ***sudo apt-get install snort*** s'inicia la instal·lació. Ens demana el rang de IP de la nostra xarxa:

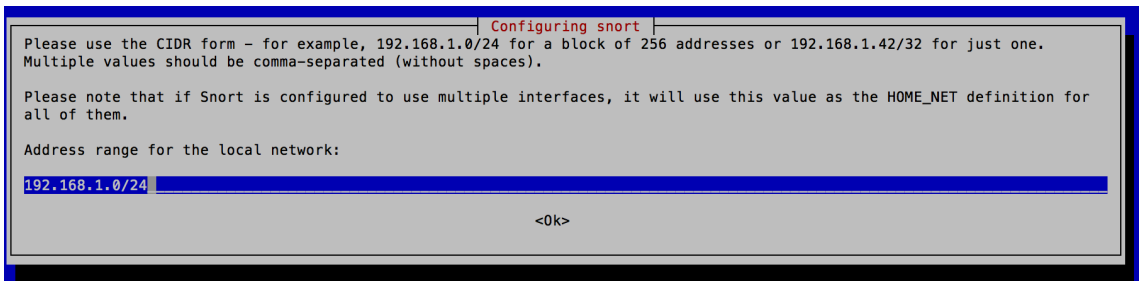


Figura 37. Configuració inicial Snort

```
root@kali:~# sudo apt-get install snort
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libblas-common libgfortran3 libjim0.76 libpcre16-3 libpoppler64 libqgsttools-p1 libqt5multimedia5-plugins
  libqt5multimediaquick-p5 libqt5multimediawidgets5 libwireshark8 libwiretap6 libwsutil7 rename sgml-base xml-core
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort-common snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort snort-common snort-common-libraries snort-rules-default
0 upgraded, 7 newly installed, 0 to remove and 304 not upgraded.
Need to get 2102 kB of archives.
After this operation, 5999 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive-4.kali.org/kali kali-rolling/main armhf libdaq2 armhf 2.0.4-3+b1 [60.6 kB]
Get:2 http://archive-4.kali.org/kali kali-rolling/main armhf libdumbnet1 armhf 1.12-7+b1 [23.2 kB]
Get:3 http://archive-4.kali.org/kali kali-rolling/main armhf snort-common-libraries armhf 2.9.7.0-5 [594 kB]
Get:4 http://archive-4.kali.org/kali kali-rolling/main armhf snort-rules-default all 2.9.7.0-5 [341 kB]
Get:5 http://archive-4.kali.org/kali kali-rolling/main armhf snort-common all 2.9.7.0-5 [243 kB]
Get:6 http://archive-4.kali.org/kali kali-rolling/main armhf snort armhf 2.9.7.0-5 [750 kB]
Get:7 http://archive-4.kali.org/kali kali-rolling/main armhf oinkmaster all 2.0-4 [90.6 kB]
Fetched 2102 kB in 1s (1610 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libdaq2.
(Reading database ... 157728 files and directories currently installed.)
Preparing to unpack .../0-libdaq2_2.0.4-3+b1_armhf.deb ...
Unpacking libdaq2 (2.0.4-3+b1) ...
Selecting previously unselected package libdumbnet1:armhf.
Preparing to unpack .../1-libdumbnet1_1.12-7+b1_armhf.deb ...
Unpacking libdumbnet1:armhf (1.12-7+b1) ...
Selecting previously unselected package snort-common-libraries.
Preparing to unpack .../2-snort-common-libraries_2.9.7.0-5_armhf.deb ...
Unpacking snort-common-libraries (2.9.7.0-5) ...
Selecting previously unselected package snort-rules-default.
Preparing to unpack .../3-snort-rules-default_2.9.7.0-5_all.deb ...
Unpacking snort-rules-default (2.9.7.0-5) ...
Selecting previously unselected package snort-common.
Preparing to unpack .../4-snort-common_2.9.7.0-5_all.deb ...
Unpacking snort-common (2.9.7.0-5) ...
Selecting previously unselected package snort.
Preparing to unpack .../5-snort_2.9.7.0-5_armhf.deb ...
Unpacking snort (2.9.7.0-5) ...
Selecting previously unselected package oinkmaster.
Preparing to unpack .../6-oinkmaster_2.0-4_all.deb ...
Unpacking oinkmaster (2.0-4) ...
Setting up oinkmaster (2.0-4) ...
Setting up snort-common (2.9.7.0-5) ...
Setting up snort-rules-default (2.9.7.0-5) ...
Setting up libdaq2 (2.0.4-3+b1) ...
Processing triggers for libc-bin (2.24-17) ...
Processing triggers for systemd (234-3) ...
Setting up libdumbnet1:armhf (1.12-7+b1) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up snort-common-libraries (2.9.7.0-5) ...
Setting up snort (2.9.7.0-5) ...
update-rc.d: We have no instructions for the snort init script.
update-rc.d: It looks like a network service, we disable it.
Processing triggers for libc-bin (2.24-17) ...
Processing triggers for systemd (234-3) ...
```

Figura 38. Procés instal·lació Snort

Una vegada instal·lat, podem configurar les seves opcions i regles a l'arxiu:
sudo nano /etc/snort/snort.conf

Les regles creades per nosaltres o les descarregades aniran ubicades a:
/etc/snort/rules

Les alertes que produeix Snort es registren a l'arxiu "alert" dins de ***/var/log/snort***

Per executar Snort: ***snort –c /etc/snort/snort.conf –l /var/log/snort*** on ***–c*** indica l'arxiu de regles i ***–l*** el directori on ubiquem el log.

9. Referències

[1] Raspberry Pi 3 Model B [En línia] [Data: 30 / octubre / 2017] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[2] Kali Linux [En línia] [Data: 30 / octubre / 2017] https://www.kali.org/downloads/

[3] Snort [En línia] [Data: 30 / octubre / 2017] https://www.snort.org

[4] Nessus [En línia] [Data: 22 / octubre / 2017] https://www.tenable.com/products/nessus-vulnerability-scanner

[5] OpenVas [En línia] [Data: 20 / octubre / 2017]: http://www.openvas.org

[6] Detecting malware through DNS queries: a Kali Pi / Snort Project [En línia] [Data: 2 / gener / 2015] https://www.securityforrealpeople.com/2015/01/detecting-malware-through-dns-queries.html

[7] Turn your Raspberry Pi into a Snort IDS [En línia] [Data: 25 / octubre / 2016] https://blog.holdenkilbride.com/index.php/2016/10/25/turn-your-raspberry-pi-into-a-snort-ids/

[8] Kali Linux Custom ARM Images [En línia] [Data: 3 / novembre / 2017] https://www.offensive-security.com/kali-linux-arm-images/

[9] IDS Sistema de detección de intrusos [En línia] [Data: 30 / novembre / 2017] https://es.wikipedia.org/wiki/Sistema_de_detección_de_intrusos

[10] OpenVas en Linux [En línia] [Data: 30 / maig / 2014] https://openwebinars.net/blog/openvas-en-linux-explorando-nuestros-sistemas/

[11] Steps to Install and configure Snort on Kali Linux [En línia] [Data: 2 / desembre / 2014] https://techcyberz.wordpress.com/2014/12/02/steps-to-install-and-configure-snort-on-kali-linux/

[12] How to use OpenVas to audit [En línia] [Data: 22 / novembre / 2016] https://komunity.komand.com/learn/article/how-to-use-openvas-to-audit-the-security-of-your-network-22/

[13] OpenVas self-fingerprint [En línia] [Data: 27 / agost / 2014]
http://kinomakino.blogspot.com.es/2014/08/openvas-self-fingerprint-maquillando-al.html

[14] OpenVas vulnerability scanner on the Raspberry Pi [En línia] [Data: 27 / novembre / 2017]
https://cromwell-intl.com/linux/raspberry-pi/openvas.html

[15] How to install Snorby in Kali [En línia] [Data: 21 / novembre / 2014]
http://www.ylabs.co.kr/index.php?document_srl=37892&mid=board_net_forensics&listStyle=viewer

[16] Administración de sistemes operatives Snort-BASE [En línia] [Data: gener / 2015]
http://adminso.es/index.php/Snort-BASE

[17] Aumentando el rendimiento de Snort con Barnyard2 [En línia] [Data: 30/ març / 2011]
http://www.hackplayers.com/2011/03/aumentando-el-rendimiento-de-snort-con.html

[18] Actualización automática de reglas Snort con PulledPork [En línia] [Data: 27/ gener / 2017]
http://www.securityartwork.es/2017/01/27/actualizacion-automatica-reglas-snort-pulledpork/

10. Glossari

DNS: Servidor que s'encarrega de la traducció dels noms de les web a direccions IP.

Firewall: Programa o hardware informàtic configurable per al bloqueig i filtratge del trànsit d'una xarxa.

GUI: Interfície gràfica d'usuari. Representa la informació mitjançant l'ús de gràfics i imatges per tal de presentar la informació a l'usuari.

IDS: Sistema de detecció d'intrusions. Aquests sistemes es basen en l'anàlisi del trànsit de dades i la detecció de patrons coneguts per activar les alarmes corresponents.

LDAP: Protocol lleuger d'accés a directoris. És un protocol a nivell d'aplicació que permet l'accés a un servei de directori ordenat i distribuït.

Linux: És un dels termes utilitzats per englobar la combinació d'un kernel (nucli) amb el sistema GNU. El seu codi font pot ser utilitzat, modificat i redistribuït lliurement per qualsevol sota els termes de la llicència pública general de GNU (GPL) i altres tipus de llicències lliures.

MySQL: Sistema de gestió de bases de dades relacionals desenvolupat sota llicència dual (GPL i llicència comercial) per Oracle Corporation. És considerada com la base de dades open source més popular del mon.

Open source: Expressió amb la que es coneix el programari distribuït i desenvolupat sota el paradigma del programari lliure. Habitualment dóna referència a l'accés total al codi font d'un programari.

RAM: Memòria utilitzada com a memòria de treball per al programari instal·lat en un ordinador. Les seves sigles provenen de l'anglès Randorm-Access Memory.

Root: Compte d'usuari amb privilegis d'administrador.

Sniffer: Programa informàtic que registra tota la informació que envien els equips en una xarxa.

SSH: Secure Shell o intèrpret d'ordres segura, és el nom del protocol i del programa que l'implementa. La seva funcionalitat radica en proporcionar accés a màquines remotes mitjançant una xarxa. Permet gestionar completament l'ordinador mitjançant l'intèrpret d'ordres.

VirtualBox: Programari de virtualització que permet la creació de màquines virtuals operatives. Desenvolupat per Oracle Corporation.

VM: Inicials de Virtual Machine. Referent a la capacitat d'instal·lar un sistema operatiu dins d'un altre.