

UNIVERSIDAD CATÓLICA DE SALTA

PROYECTO DE GRADO



FACULTAD DE INGENIERÍA

**MODELO DE DETECCIÓN DE INTRUSOS BASADO EN LAS TECNICAS DE
DATA MINING EN UNA RED DE DATOS.**



INTEGRANTES:

- AVILA, ALFREDO CARLOS
- BARBOZA, DANIEL FELIX

CARRERA: INGENIERÍA EN INFORMÁTICA

AÑO 2017



Título: Ingeniero en Informática

Profesor Guía: Lic. Carolina Cardoso

Miembros del Tribunal Evaluador:

Nombre:.....

Firma:

Nombre:.....

Firma:

Nombre:.....

Firma:

Fecha de Exposición del Trabajo Final:.....



DEDICATORIA:

Dedico este trabajo a Dios, por haberme dado la sabiduría y las fuerzas para lograr mis objetivos y guiarme en cada momento transcurrido en mi formación profesional.

A mis padres, por su apoyo y amor incondicional en todo momento, por sus consejos y por ser los pilares de mi vida, que me formaron como una persona de bien.

A mis amigos y compañeros, por el apoyo y los buenos momentos compartidos en cada momento de mi carrera y mi vida.

A mis profesores, por las enseñanzas y la motivación recibida durante todos estos años de mi formación académica. Y a nuestra Profesora Guía, la Lic. Carolina Cardoso, por su ayuda y orientación en la elaboración del presente Proyecto de Grado.

¡Gracias a Todos!

Daniel Felix Barboza

Dedico este trabajo primeramente a Dios en agradecimiento por darme la vida y la posibilidad de conocerlo y vivir para El, por cada bendición recibida a lo largo de mi vida y por cada situación difícil en la cuál me ayudó a salir para adelante.

A mi Madre por brindarme su amor incondicional y su esfuerzo para concretar este sueño, por creer en mí y apoyarme en cada etapa de mi vida, pero principalmente por su ejemplo de constancia, perseverancia y pasión por los sueños en la vida.

A familiares, compañeros y amigos por el apoyo continuo en cada momento.

A cada profesor que me tocó tener en mi formación académica, por su dedicación y motivación brindada en todos estos años de aprendizaje. Y a nuestra Profesora Guía, la Lic. Carolina Cardoso, por su apoyo, tiempo y dedicación para poder concretar el presente Proyecto de Grado.

Alfredo Carlos Avila



Contenido

Abstract.....	12
Capítulo 1 – Introducción al Proyecto	13
1.1 Introducción	13
1.2 Problemática y Sistema Objeto	13
1.2.1 Principales áreas del Laboratorio de Informática.....	14
1.2.2 Visión	14
1.2.3 Misión.....	15
1.2.4 Objetivos	15
1.3 Motivación	15
1.4 Público objetivo	16
1.5 Estado de la cuestión.....	16
Capítulo 2 – Objetivos y Alcance.....	19
2.1 Introducción	19
2.2 Antecedentes	19
2.3 Objetivos	20
2.4 Objetivos Específicos:	20
2.5 Alcance	21
2.6 Criterios de Éxito	22
Capítulo 3 – Marco Teórico.....	24
3.1 Introducción	24
3.2 Marco Teórico.....	24
3.3 Seguridad de Redes de Datos.....	24
3.3.1 Intromisión Informática.....	25
3.3.2 Definición de Ataques informáticos.....	25
3.3.3 Tipos de Ataques Informáticos	25
3.3.4 Monitoreo de las redes de datos	30
3.3.5 Herramientas disponibles para el monitoreo de red	30
3.3.6 Detección de Intrusos	32
3.3.7 Clasificación de los IDS.....	32
3.3.8 Tecnologías disponibles para la Detección de intrusos.....	32
3.4 Metodología para administración de redes	33



3.4.1 Administración de la configuración	34
3.4.2 Administración del rendimiento.....	34
3.4.3 Administración de fallas	35
3.4.4 Administración de la contabilidad.....	35
3.4.5 Administración de la seguridad.....	35
3.5 Minería de datos.....	36
3.5.1 Definición de Minería de Datos	36
3.5.2 Técnicas de Minería de Datos	36
3.5.3 Aprendizaje automático.....	37
3.5.4 Árbol de Decisión.....	39
3.5.5 Poda de Árboles de Decisión	40
3.5.6 Validación cruzada.....	41
3.5.7 Metodologías aplicadas al proceso de minería de datos	43
3.5.8 Tecnologías y herramientas disponibles para la minería de datos	45
3.6 GureKDD Data	47
3.7 NSL-KDD Data	49
3.7.1 Tipos de ataques	49
Capítulo 4 – Definición de la Solución	54
4.1 Introducción	54
4.2 Metodología de desarrollo de la solución	54
4.2.1 Comprensión del negocio.....	54
4.2.2 Administración de la configuración	54
4.2.3 Comprensión de datos (Administración del rendimiento)	55
4.2.4 Preparación de datos.....	56
4.2.5 Modelado.....	56
4.2.6 Evaluación.....	56
4.2.7 Administración de la Seguridad	57
4.2.8 Implementación.....	57
Capítulo 5 – Solución Propuesta	58
5.1 Introducción	58
5.2 Herramientas utilizadas para la Solución.....	58
5.2.1 Herramientas utilizadas para el monitoreo de red.....	58



5.2.2 Herramientas utilizadas para la minería de datos.....	58
5.2.3 Herramientas utilizadas para la detección de intrusos	58
5.3 Definición de la Solución	59
5.4 Estructura de desglose de trabajo (WBS)	63
5.5 Planificación del Modelo propuesto (Diagrama de Gantt)	68
5.6 Compresión del negocio	71
5.6.1 Requerimientos del proyecto.....	71
5.7 Administración de la configuración.....	71
5.7.1 Instalación y Configuración del entorno virtual.....	71
5.8 Comprensión de los datos	74
5.8.1 Monitoreo de la red y Captura de datos	74
5.9 Preparación de los datos	86
5.10 Modelado	97
5.10.1 Creación y selección los modelos de clasificación	97
5.10.2 Aplicación de los modelos sobre los datos capturados	108
5.11 Evaluación	112
5.11.1 Generación de reglas para el Servidor IDS	112
5.12 Implementación del IDS	121
Capítulo 6 – Análisis y Factibilidad del Proyecto	127
6.1 Introducción	127
6.2 Análisis FODA	127
6.2.1 Fortalezas	127
6.2.2 Oportunidades	128
6.2.3 Debilidades.....	128
6.2.4 Amenazas	128
6.2.5 Estrategias de Oportunidades - Fortalezas:	128
6.2.6 Estrategias de Oportunidades - Debilidades:	129
6.2.7 Estrategias Riesgos - Fortalezas:.....	129
6.2.8 Estrategias de Amenazas - Debilidades:	129
6.3 Analisis de riesgos	130
6.4 Factibilidad económica	133
6.5 Factibilidad técnica	136



6.6 Factibilidad operativa.....	137
6.7 Factibilidad legal.....	137
6.8 Factibilidad ambiental.....	137
6.9 Análisis Económico-Financiero.....	138
6.9.1 Inversión Inicial del Proyecto	138
6.9.2 Inversión en RRHH.....	139
6.9.3 Valor Actual Neto del Proyecto (VAN).....	142
6.9.4 Tasa Interna de Retorno (TIR)	145
6.9.5 Rentabilidad	145
Capítulo 7 – Resultados obtenidos y Conclusiones.....	146
7.1 Introducción	146
7.2 Resultados o verificación experimental.....	146
7.3 Análisis de resultados (Análisis de rendimiento del IDS).....	149
7.3.1 Comparación de resultados	150
7.4 Proceso Estandarizado para la Creacion de Reglas de Deteccion de Intrusos aplicando Minería de Datos.....	154
7.5 Conclusión	158
7.6 Futura líneas de investigación.....	159
Anexos.....	160
Anexo I - Instalación Kali Linux 2.0	161
Anexo II - Instalación CentOS 6.....	170
Anexo III - Instalación IDS Suricata	178
Anexo IV – Instalación BRO Network Security Monitor	191
Glosario	194
Bibliografía.....	196



Índice de Cuadros

Cuadro 1 - Clasificación de técnicas minería de datos.....	37
Cuadro 2 – Comparación de uso de metodologías de Minería de datos.....	43
Cuadro 3 – Comparación de uso de herramientas de Minería de datos	46
Cuadro 4 – Atributos de la GureKDD Cup data base 2008.....	49
Cuadro 5 - Clasificación de tipos de ataques.....	51
Cuadro 6 – Etapas por metodologías	54
Cuadro 7 – Estructura de desglose de trabajo.....	67
Cuadro 8 – Referencia de campos principales.....	90
Cuadro 9 – Referencia de todos los campos.....	93
Cuadro 10 – Comparación de performance de algoritmos – Gure KDD	103
Cuadro 11 – Comparación de performance de algoritmos – NSL KDD.....	103
Cuadro 12 – Análisis de riesgos	131
Cuadro 13 - Calculo de días trabajados por RRHH.....	133
Cuadro 14 – Inversión en RRHH	134
Cuadro 15- Honorarios RRHH.....	134
Cuadro 16 – Inversión en hardware.....	135
Cuadro 17 – Inversión en software.....	135
Cuadro 18 – Inversión en información	136
Cuadro 19 – Resumen de inversión.....	136
Cuadro 20 – Inversión Inicial en Hardware.....	138
Cuadro 21 – Inversión Inicial en Software.....	138
Cuadro 22 – Plan de inversión en RRHH.....	139
Cuadro 23 – Flujo de Efectivo Proyectado a 12 meses	141
Cuadro 24 – VAN del Proyecto.....	143



Índice de Imágenes

Imagen 1 – Tipos de ataques informáticos	26
Imagen 2 – Ataque HYDRA	27
Imagen 3 – Ataque DOS y DDOS	28
Imagen 4 – Árbol de Decisión	39
Imagen 5 – Validación cruzada de K iteraciones	41
Imagen 6 – Validación cruzada aleatoria	42
Imagen 7 – Validación cruzada dejando uno afuera	42
Imagen 8 – Ciclo de vida de CRISP – DM	44
Imagen 9 – Categoría de Ataque DOS	51
Imagen 10 – Categoría de Ataque R2L	52
Imagen 11 – Categoría de Ataque UR2	52
Imagen 12 – Categoría de Ataque Probing	53
Imagen 13 – Esquema básico del modelo propuesto	60
Imagen 14 – Proceso de desarrollo de la solución de propuesta	62
Imagen 15 – Estructura de desglose de trabajo	63
Imagen 16 – Planificación – Parte 1	68
Imagen 17 – Planificación – Parte 2	69
Imagen 18 – Planificación – Parte 3	70
Imagen 19 – Configuración de red modo NAT	73
Imagen 20 – Diferencia entre Modo NAT y Bridge	73
Imagen 21 – Listado de interface de red Kali Linux	74
Imagen 22 – Interface de red eth0 en modo promiscuo	75
Imagen 23 – Interface de Wireshark	75
Imagen 24 – Opciones de captura de Wireshark	76
Imagen 25 - TCPdump	76
Imagen 26 – Inicio de Wireshark	77
Imagen 27 – Wireshark: Captura de paquetes en tiempo real	77
Imagen 28 – Panel de listado de paquetes	78
Imagen 29 – Panel de información de cada paquete	78
Imagen 30 – Panel de información de formato hexadecimal de cada paquete	78
Imagen 31 – Activación Servicio FTP en Cortafuegos de CentOS 6.5	80
Imagen 32 – Acceso servidor ftp mediante consola CentOS 6.5	80
Imagen 33 – Comando Crunch	81
Imagen 34 – Ataque Hydra-Fuerza Bruta a Servidor FTP	82
Imagen 35 – Hping3 ejecutándose	83
Imagen 36 – Slowloris.pl ejecutándose	83
Imagen 37 – Escaneo de Red con nmap	84
Imagen 38 – Escaneo de puertos con nmap	85
Imagen 39 - Fragmentos del algoritmo Darpa bro	87
Imagen 40 – Fragmentos del algoritmo Darpa bro	88
Imagen 41 – Algoritmo trafAld	90



Imagen 42 – Archivo final obtenido trafAld.list	94
Imagen 43 – Archivo original Gure KDD Cup Database 2008 extensión .arff	95
Imagen 44 – Archivo original Gure KDD Cup Database 2008 abierto con MS Excel 2010...	95
Imagen 45 – Archivo original Gure KDD Cup Database 2008.....	96
Imagen 46 – Campo de clasificación (clase) de los registros del.....	96
Imagen 47 – Expansión del operador Validation	98
Imagen 48 – Proceso de validación	100
Imagen 49 – Panel de carga de datos.....	101
Imagen 50 – Panel de selección de atributos	102
Imagen 51 – Ejemplo de matriz de confusión	103
Imagen 52 – Proceso de clasificación de la captura	108
Imagen 53 – Clasificación de Anormal_dict_ftp.csv con la GUREKDD 2008	109
Imagen 54 – Clasificación de AnormalDoS_slowloris.csv con la GUREKDD 2008.....	110
Imagen 55 - Clasificación de Anormal_probing.csv con la NSLKDD 2012	111
Imagen 56 - Clasificación de Anormal_warez_ftp.csv con la GUREKDD 2012	112
Imagen 57 – Cuadro de relación entre opciones y parámetros de las reglas de IDS.....	113
Imagen 58 - Valores de atributos significativos para las reglas de detección del Ataque DoS	116
Imagen 59 – Valores de atributos significativos para las reglas de detección del Ataque Hydra-Fuerza Bruta	118
Imagen 60 - Valores de atributos significativos para las reglas de detección del Ataque Warezmaster/Warezclient.....	119
Imagen 61 - Valores de atributos significativos para las reglas de detección del Ataque Escaneo de red y puertos	120
Imagen 62 – Mecanismo IDS	122
Imagen 63 - Snorby	123
Imagen 64 – Pagina de Login de Snorby.....	124
Imagen 65 – Pagina Dashboard Snorby	124
Imagen 66 – Barnyard2	125
Imagen 67 – Suricata IDS en funcionamiento.....	126
Imagen 68 - Matriz FODA	127
Imagen 69 – Formula del VAN	142
Imagen 70 – Análisis de Valor Actual Neto con diferentes Tasas de Descuento	144
Imagen 71 – Formula de la TIR.....	145
Imagen 72 – Tabla de resultados de Snorby.....	146
Imagen 73 – Grafico de Tortas de resultados de Snorby.....	147
Imagen 74 – Niveles de severidad de alertas de Snorby	148
Imagen 75 – Tabla de las 10 principales direcciones IP de origen.....	149
Imagen 76 – Tabla de las 10 principales direcciones IP de destino	149
Imagen 77 – Resultados obtenidos con un IDS tradicional.....	151
Imagen 78 – Tabla de resultados de Snorby IDS tradicional	151
Imagen 79 – Niveles de severidad de alertas de Snorby IDS tradicional.....	152



Imagen 80 – Resultados obtenidos con un IDS basado en minería de datos	152
--	-----



Abstract

En este proyecto se propone desarrollar un modelo de detección de intrusos basado en técnicas de minería de datos. Mediante la simulación de una red de datos por donde circule tráfico de red “normal” y “anormal”, y haciendo uso de los algoritmos y técnicas de minería de datos para clasificar el tráfico de la red, se crearán reglas de detección de intrusos que vuelvan al modelo más preciso y confiable, con el fin de brindar la información necesaria sobre anomalías, ataques e intromisiones informáticas que puede llegar a sufrir una red de datos.



Capítulo 1 – Introducción al Proyecto

1.1 Introducción

En la actualidad la información es el recurso más valioso en las organizaciones de todo el mundo, no solamente por su almacenamiento masivo sino más bien por su gestión, de acuerdo a las distintas políticas de seguridad y reglas de negocio de cada organización, que proporcionan el valor agregado a la información. El flujo de información tanto interna, dentro de las diferentes áreas de la organización, como externa, interconectando la misma con el ambiente externo (otras organizaciones, usuarios y/o clientes), hace a la importancia de las redes de datos, ya que el propósito de las redes informáticas es compartir recursos, siendo el recurso principal la información. Por lo cual resulta importante resguardar y proteger la información de una organización de una gran variedad de ataques contra su disponibilidad, integridad y confidencialidad, disponiendo de diferentes protocolos y políticas de seguridad, tanto a nivel técnico como de gestión, que permitan brindar la protección necesaria. Sin embargo, debido a los avances de la tecnología informática, estos ataques evolucionan constantemente presentando cada vez más riesgos, haciendo necesario que las organizaciones tomen conciencia de esta situación y avancen de la misma forma en la protección de la información.

El acceso indebido a un sistema y a los datos que maneja puede resultar en un problema potencial para las organizaciones, que puede llevar a graves pérdidas para la misma.

Es sabido que, hoy por hoy, la información ha adquirido un valor altísimo desde el punto de vista económico, constituyéndose en un bien sustrato del tráfico jurídico, con relevancia jurídico-penal por ser posible objeto de conductas delictivas (acceso ilegítimo, sabotaje o daño informático, espionaje informático, etc.) y por ser instrumento de comisión, facilitación, aseguramiento y calificación de los ilícitos tradicionales. (Arocena, 2015)

Por tal motivo las herramientas informáticas que ayuden a mejorar la seguridad de las redes de datos son necesarias para minimizar los efectos negativos producidos por dichos ataques y apoyar en la toma de decisiones que conduzcan a elevar la seguridad de una determinada organización.

1.2 Problemática y Sistema Objeto

Es sabido que el nivel de riesgo que atenta contra la seguridad de la información de una organización o entidad está dado por la función, rol, negocio o posesiones que tenga la misma, es decir, mientras más alta sea la función, más importante el rol, más grande el negocio o más posesiones de información valiosa tenga una organización o entidad, mayor será el riesgo de sufrir ataques o intromisiones, por lo cual, es necesario tener en cuenta esta realidad para aplicar la seguridad necesaria y contrarrestar dichos ataques o intromisiones. Esta realidad puede resultar positiva o negativa en materia de seguridad informática, de acuerdo a la organización o entidad en cuestión, sin embargo, ninguna red de datos que esté comunicada con el mundo está libre de sufrir ataques, y hoy en día las organizaciones deben poder conectar sus redes de



datos con el mundo (a través de internet) para lograr tener éxito en su rubro o negocio, por lo que la mayoría de las organizaciones tienen acceso a internet. Por tal motivo, del beneficio de estar conectado (en tiempo parcial o total) con la red mundial internet, se presenta al mismo tiempo el problema de estar expuesto como víctima ante atacantes que quieran, ya sea por motivos delictivos o desinteresados, entrar en las redes de datos organizacionales.

Hoy en día no todas las organizaciones que disponen de una red de datos están al tanto de ésta realidad, ya sea por ignorancia o desinterés (por lo mencionado anteriormente sobre los niveles de riesgo), y así mismo no hacen uso de herramientas que ayuden a la seguridad de la red de datos o tienen desactualizadas sus herramientas siendo éstas vulnerables de ser burladas y sobrepasadas sin problemas.

Teniendo en cuenta ésta realidad, se propone en el presente proyecto desarrollar un modelo de una herramienta de seguridad informática basada en minería de datos para la protección de las redes de datos, que ayude a minimizar los riesgos presentados por el hecho de estar conectados con el mundo permitiendo la detección oportuna de ataques e intromisiones de manera automática, específica y directa, pudiéndose adaptar a cualquier infraestructura de red de cualquier tipo de organización sin la necesidad de reestructurar o reconfigurar la red de datos. Para ello se escogió como sistema objeto el Laboratorio de Informática de la Facultad de Ingeniería de la Universidad Católica de Salta (Ucasal), en el cuál se realizará la simulación de una red de datos estándar y un tráfico de red que modele un contexto organizacional estándar, con la ejecución de diversos ataques a la misma.

1.2.1 Principales áreas del Laboratorio de Informática

El Laboratorio de Informática cuenta con distintas subredes que representan distintas aulas y áreas:

- Aula Taller 1: con 16 equipos.
- Aula Taller 2: con 12 equipos.
- Aula Taller 3: con 15 equipos:
- Aula Taller 4: con 20 equipos.
- Oficina del Operador del Laboratorio

Dentro de cada aula taller, cada equipo posee una dirección IP dentro de un rango definido, tenido conexión entre cada uno de ellos, entre equipos de distintas aulas y conexión a internet.

1.2.2 Visión

La visión del Laboratorio de Informática de la Facultad de Ingeniería de la UCASAL consiste en que los alumnos y docentes puedan generar nuevos conocimientos que favorezcan al desarrollo profesional de los mismos y que, a través del uso de las tecnologías de la información y las comunicaciones, ayuden a solventar las necesidades de información de los estudiantes de la Universidad Católica de Salta.



1.2.3 Misión

El Laboratorio de Informática tiene como misión garantizar que tanto alumnos como docentes puedan desempeñar sus funciones académicas de una manera óptima, brindando las herramientas e información necesarias para el cumplimiento de las exigencias de las cátedras y el desarrollo profesional de los estudiantes.

1.2.4 Objetivos

- Brindar el espacio necesario para que los alumnos y docentes puedan desarrollar sus actividades académicas.
- Proporcionar las tecnologías de la información y las comunicaciones necesarias para el desarrollo académico y profesional de las personas que concurren a la Universidad Católica de Salta.
- Permitir que los alumnos desarrollen prácticas y pruebas de índole tecnológica, mediante la utilización de las herramientas y aplicaciones informáticas disponibles en las máquinas del laboratorio.

1.3 Motivación

La motivación del presente proyecto viene dada por la posibilidad de combinar dos ramas de la informática de gran importancia y relevancia en la actualidad, la minería de datos y la seguridad de las redes de datos, y de esta manera desarrollar una herramienta de seguridad diferente a las tradicionales que permita mejorar la seguridad de la red de datos de una determinada organización.

Por un lado, la minería de datos hoy en día es un campo de la informática que está siendo desarrollado cada vez más, teniendo un campo de estudio y aplicación muy amplio que todavía no se ha llegado a explotar al máximo, sin embargo, del estudio y aplicación realizados hasta la actualidad se han obtenido resultados muy favorables para las organizaciones, ya sea optimizando sus procesos, mejorando sus servicios y ventas, optimizando y re direccionando esfuerzos, entre otros. Por otro lado, en lo que respecta a la seguridad de las redes de datos y más específicamente a la ejecución y detección de ataques e intromisiones a las redes informáticas, es un área que avanza constantemente dando lugar a nuevas metodologías, protocolos y herramientas que se proponen a resguardar información, como así también robar información. Para las organizaciones este tema resulta crítico ya que se pone en juego su recurso más valioso (la información que poseen), el cual deben proteger y resguardar para prevenir caos y/o crisis ante situaciones que provoquen la eliminación o alteración de la información.

Otros motivos que impulsaron este proyecto es la voluntad de trabajar en conjunto y abordar un tema que reúna estos dos aspectos mencionados de la informática, debido a lo interesante y apasionante que resulta investigar, estudiar y comprender las características y ámbitos en el cual se desenvuelven, y así poder brindar un resultado que pueda ser de gran utilidad para la sociedad en que vivimos.



1.4 Público objetivo

En la actualidad la aplicación de tecnología se está globalizando mundialmente, llegando a todo tipo de organización, desde organizaciones privadas, públicas, gubernamentales, con fines y sin fines de lucro, afectando todos los niveles jerárquicos organizacionales y a cada área o sector de la misma. Así mismo, los departamentos o áreas de sistemas de información (o tecnología) en las organizaciones cumplen un rol cada vez más protagonista en el funcionamiento y éxito de las organizaciones, siendo un pilar fundamental para las comunicaciones y el flujo de información interno y externo. Por tal motivo el presente proyecto está orientado principalmente al personal de tecnología o de sistemas de información de las organizaciones, abordando las áreas de administración de redes y seguridad informática, desde el nivel gerencial o de gestión hasta el nivel técnico, proporcionando una herramienta que permite minimizar los riesgos y vulnerabilidades en la seguridad informática y ayuda a tomar decisiones que contribuyan a una mejor gestión de la información.

1.5 Estado de la cuestión

Hoy en día la interconexión de las personas dentro de las organizaciones o empresas mediante las computadoras es indispensable para llevar a cabo diversos tipos de actividades y trabajos que permitan expandir sus negocios y productos a casi todo el mundo. Por tal motivo esta complejidad dio origen a nuevos retos y aplicaciones que brinden soporte y sobre todo seguridad a la hora de la transferencia de datos sensibles e información confidencial.

Dentro de este amplio grupo de aplicaciones y mecanismos de defensa para reducir los riesgos de las compañías ante ataques dirigidos hacia los bienes informáticos, los más usados son los Sistemas de Detección de Intrusos (IDS). Una definición de IDS es la siguiente:

Un IDS es un elemento que escucha y analiza toda la información que circula por una red de datos e identifica posibles ataques. Cuando aparece un ataque, el sistema reaccionara informando al administrador y cierra las puertas al posible intruso reconfigurando elementos de la red como firewalls y routers. (Miranda, 2015)

En la actualidad podemos encontrar varios tipos de IDS, clasificados según el tipo de situación física, del tipo de detección que posee o de su naturaleza y reacción cuando detecta un posible ataque. Según la función del software IDS, estos pueden ser:

- NIDS (Network Intrusion Detection System)
- HIDS (Host Intrusion Detection System)

El contexto de nuestro proyecto se centra en una nueva clase de IDS que funciona en base a reglas de minería de datos sobre las redes de datos. Algunos trabajos relacionados con la detección de intrusiones informáticas son:

- ***Arquitectura multi-agente adaptativa para la detección de ataques en entornos dinámicos y distribuidos.*** (Trejos, 2010):



Este trabajo tiene como objetivo final diseñar una arquitectura multi-agente que permita construir sistemas multi-agentes, basados en el modelo de agentes BDI (creencias, deseos e intenciones), Sistemas BDR (Razonamiento basado en casos) y técnicas del aprendizaje automático, para detectar y bloquear intrusiones a nivel de red, tales como ataques de Inyección SQL y Ataques XDoS, en aplicaciones distribuidas. Se entiende por “agente” a una entidad que interactúa con su entorno y con otros agentes permitiendo así que el sistema multi-agentes sea adecuado para ser aplicado en entornos altamente dinámicos.

Los resultados obtenidos en los experimentos de este trabajo permiten demostrar que abordar el problema de la detección de ataques de inyección SQL e inyección XDoS de forma distribuida permite evitar los problemas tradicionales de los sistemas centralizados e incrementar la eficiencia en el proceso de detección por otro lado presenta un avance significativo en el campo de la detección de intrusiones, mediante la aplicación de un conjunto de tecnologías y técnicas del campo de la Inteligencia Artificial.

- ***Detección de Intrusiones en redes de datos con captura distribuida y procesamiento estadístico.*** (Britos, 2010)

El objetivo de este trabajo de investigación consistió en el análisis y desarrollo de tecnologías basadas en la investigación estadística, las redes neuronales y los sistemas autónomos aplicados a los problemas de detección de intrusiones en redes de datos. Aportando valiosos conceptos teóricos sobre diferentes tipos de ataques informáticos y vulnerabilidades, Sistemas de Detección de Intrusos y Herramientas de prueba para estos últimos.

Básicamente se hizo un énfasis principal en la mejora del aprendizaje para la detección de intrusos mediante dos herramientas novedosas, las redes neuronales y las colonias de hormigas (ACCM) obteniendo como logros de la investigación lo siguiente: se comprobó que la red neuronal puede entrenarse fácilmente para distinguir entre condiciones normales y en condiciones de ataque de la red no tan complejos obteniendo un buen rendimiento utilizando una configuración sencilla de la red neuronal (Configuración “back propagation”), por otro lado para ataques más complejos se optó por el método de ACCM que mejora la calidad de las agrupaciones y precisión en la clasificación de anomalías de la red.

- ***Propuesta teórica de Webmining para Detección de Intrusos mediante Lógica Borrosa.*** (Campoy, 2011)

El objetivo de esta investigación consistió en elaborar un modelo de Sistema Experto Borroso (SBRB) aplicado a la Detección de Intrusos WEB, para ello la propuesta consistió en profundizar en los dos aspectos siguientes: el diseño de un SBRB y aplicar la adaptación de estos modelos en “minería de datos” relativos a la detección de intrusos. La investigación muestra las reglas para identificar tráfico malicioso en web-logs donde se han encontrado patrones de asociación descubiertos en la red, tales como: exploración web; ataques específicos a una máquina; alto número de conexiones TCP anómalas



sobre el puerto 80; Lo que se buscó en esta investigación es sentar las bases de aplicación de lógica difusa y que mediante la minería de datos sobre los datos de los ficheros web-logs lograr obtener una clasificación del nivel de peligrosidad de los ataques informáticos y aumentar así la precisión de las reglas de detección de intrusos mediante el autoaprendizaje.

- ***Minería de datos aplicada en detección de intrusos.*** (Vallejo & Germán, 2012)

En este trabajo se utilizó la minería de datos a través de la herramienta “IBM SPSS Modeler” para buscar información no trivial que se encuentre oculta o dispersa en ellos, es decir, se exploran los datos del acceso a los sistemas, en redes locales o internet, para descubrir la interconexión e interrelación y poder obtener la información oculta sobre ataques o intromisiones informáticas. Los datos con los que se trabaja se toman de los utilizados en The Third International Knowledge Discovery and Data Mining Tools Competition y fueron preparados y dirigidos por los laboratorios de MIT Lincoln.

De este trabajo se buscó descubrir la utilidad de la minería de datos en la exploración, a fin de encontrar y predecir ataques que no son detectados por antivirus, cortafuegos o sistemas de detección de intrusos, obteniendo modelos que permiten obtener la información y patrones necesarios para crear reglas de detección de intrusos. En este caso el modelo que mejor comportamiento tuvo en evaluar la variable “Ataque” vs las otras variables de entrada fue el modelo CHAID, que es un tipo de modelo que genera árboles de decisión utilizando estadístico de chi-cuadrado para identificar las divisiones óptimas.

Básicamente un IDS con minería de datos consiste en la detección de intrusos en las redes de datos clasificándolos por tipo de ataques o intromisión que se realiza. Tras la aplicación de algoritmos de minería de datos sobre un conjunto de datos capturados en la red se pretende elaborar todo un conjunto de reglas para la detección de intrusos que permitan mejorar el rendimiento y la eficiencia de un IDS para brindar resultados a corto plazo y lleven a la constante incorporación de reglas para nuevos ataques que fortalezcan y brinden apoyo con la seguridad en las redes.



Capítulo 2 – Objetivos y Alcance

2.1 Introducción

Este capítulo abarca los puntos iniciales del proyecto, describiendo el trabajo a realizar, como así también los límites del mismo, por medio de la definición de los objetivos y alcance.

2.2 Antecedentes

Hoy en día un IDS basado en las técnicas de Data Mining proporciona a la seguridad de redes de datos de una organización la posibilidad de monitorear el tráfico que fluye a través de las redes informáticas y distinguir entre tráfico de datos normal y un tráfico con anomalías permitiendo detectar con anticipación lo que puede resultar en un posible o futuro ataque o intromisión informática, y de este modo tomar las precauciones o medidas de seguridad pertinentes a la situación. Por otro lado, un IDS de este tipo puede ir aprendiendo nuevas reglas de seguridad para la detección de intrusos que son proporcionadas por la aplicación de la minería de datos sobre los datos capturados del monitoreo de la red.

El modelo que se propone llevar a cabo en este proyecto está basado en la Investigación: *“A data mining framework for building intrusion detection models”* (Wenjun, 2010) y en la Tesis de Grado realizada en la Facultad de Informática y Electrónica (FIE) de la Escuela Superior Politécnica de Chimborazo – Riobamba, Ecuador, *“Modelo basado en las técnicas de minería de datos aplicada a la detección de ataques en las redes de datos de la facultad de informática y electrónica”*, cuyos autores fueron Ing. Martha Yolanda Carranza Suica e Ing. Rosley Amparo Naranjo Barragán. (Suica & Naranjo Barragán, 2014)

En *“A data mining framework for building intrusion detection models”* de Lui Wenjun se describe que se utiliza las técnicas de minería de datos para crear reglas para la detección que contribuyan a hacer más automática y mejorar la eficacia y la eficiencia de los sistemas de detección de intrusos, por otro lado el autor indica que para optimizar el rendimiento en minería de datos se lo realiza sobre los datos que se consideren sospechosos ya que según DARPA (Defense Advanced Research Projects Agency), agencia del Departamento de Defensa de los EE.UU. responsable del desarrollo de las tecnologías emergentes para la utilización por los militares, el 95 % del tráfico es normal. Y se concluye que en el futuro será importante combinar la tecnología de minería de datos, tecnología de análisis de protocolos y tecnologías de análisis de la relatividad con el fin de disminuir la tasa de distorsión y mejorar la eficiencia de detección de intrusos desconocidos.

En la Tesis de Grado citada anteriormente se utilizó como base el trabajo de Liu Wenjun, se desarrolló un modelo de detección de intrusos que se basa en las técnicas de minería de datos, y se estableció que la utilidad de un sistema de detección, como mecanismo previo de alarma, radica en facilitar la distinción entre un acceso normal y habitual al sistema, que puede salir de la puesta en marcha de servicios ofrecidos al ambiente externo (entendiendo como ambiente externo cualquier otro sistema ajeno al que ofrece los servicios), de un intento de violar de algún



modo dichos servicios, e incluso de aquellos que no debieran ser públicos, como parte del ataque a dicho sistema.

Para la realización de este trabajo las autoras llevaron a cabo la monitorización y captura de los datos de la red de datos de la FIE utilizando la herramienta TCPDUMP (disponible para el sistema operativo UNIX), que a diferencia del presente proyecto de grado este tráfico de red no fue simulado, sino el tráfico real de la red. Posteriormente llevaron a cabo la preparación de los datos capturados de la red y los integraron junto con la base de datos del concurso KDD de 1999, la cual se explicará más adelante en el presente proyecto. Una vez unificados los datos capturados y los de la KDD 1999 llevaron a cabo la selección, limpieza y transformación de estos para obtener un archivo en formato “. arff” que posee 31333 registros, y posee la estructura de datos que acepta la herramienta de minería de datos denominada WEKA (Waikato Environment for Knowledge Analysis, en español entorno para análisis del conocimiento de la Universidad de Waikato), la cual las autoras se valen para realizar el proceso de minería de datos. Los algoritmos de minería utilizados en el trabajo son:

Algoritmos basados en reglas – PART

Algoritmos basados en arboles – Random Forest

Luego se pasó a la instalación y configuración del Sistema de detección de intrusos denominado Suricata, y posteriormente a la incorporación de las reglas de detección de intrusiones al motor del IDS, obtenidas a través del análisis de los patrones obtenidos del modelo de minería de datos.

Como resultado obtuvieron una optimización del sistema de detección de intrusos con ayuda de las técnicas de minería de datos para mejorar la seguridad en la red de información de la facultad FIE de la ESPOCH.

2.3 Objetivos

Analizar, diseñar y desarrollar un modelo de detección de intrusos basado en técnicas de minería de datos, que permita la detección oportuna de los ataques realizados en la red de datos simulada en el laboratorio de informática de la UCASAL, proporcionando consistencia e integridad a la información que fluye a través de la misma.

2.4 Objetivos Específicos:

- Identificar y documentar información relativa a la seguridad de una red de datos estándar, tal como definiciones de políticas de seguridad, protocolos, tipos de ataques a la red de datos, intromisión informática; necesaria para establecer un marco teórico del proyecto.
- Diseñar y desarrollar un modelo de arquitectura de red de datos simulada por donde circule un tráfico de red “normal” (de acuerdo a reglas, políticas y movimientos organizacionales simulados) y “anormal” (diversos ataques a la red de datos).
- Monitorear y almacenar la información que fluye a través la red de datos simulada.



- Analizar los diferentes algoritmos de clasificación de minería de datos para seleccionar el/los más eficiente/s para detectar los diversos ataques e intromisiones.
- Obtener, a partir del análisis anterior, uno o más modelos de minería de datos (desarrollados con el/los algoritmo/s seleccionado/s) que clasifiquen el tráfico de red simulado en “normal” y “anormal” (etiquetando los datos con algún tipo de ataque).
- Desarrollar un modelo de sistema de detección de ataques a la red datos basado en las técnicas de minería de datos.
- Evaluar los resultados obtenidos a través del modelo implementado por medio de reportes estadísticos, gráficos, y comparando con un Sistema de detección de intrusos tradicional (no basado en minería de datos)

2.5 Alcance

En este proyecto se pretende dar conocimiento de las vulnerabilidades y anomalías que pueden existir en una determinada red de datos, mediante la simulación de una red de datos de 4 equipos (virtuales), en el laboratorio de informática de la Facultad de Ingeniería de la Ucasal, y la detección de ataques e intromisiones a la misma. Se realizarán los siguientes ataques: Hydra-Fuerza Bruta, DOS, Escaneo de red y puertos, y Warezclient/Warezmaster para generar el tráfico “anormal”, y se llevará a cabo la captura de dicho tráfico mediante el uso de herramientas específicas para el monitoreo de red. Así también se realizarán movimientos y operaciones en la red simulando un tráfico “normal”, como ser: acceso a sitios web, envío de mails, acceso a los archivos compartidos, etc., y dicho tráfico también será capturado y almacenado. Como base de información para la detección de intrusos, se hará uso de dos bases de datos: GureKDD data base 2008 y NSL KDD data base 2012, donde ambas contienen un tráfico de red capturado y clasificado como tráfico “normal” o “anormal” (algún tipo de ataque), las cuales servirán para dar formato al tráfico a capturar en la red de datos simulada y para generar el/los modelo/s de minería de datos que permitan detectar ataques e intromisiones. Posteriormente se llevará a cabo el diseño y desarrollo de un modelo de detección de intrusos basado en técnicas de minería de datos, incorporando reglas de detección ante situaciones de riesgo e inseguridad.

El proyecto abarcará los siguientes aspectos:

- 1 Identificación y documentación de la información relativa a la seguridad de la red de datos, tal como definiciones de políticas de seguridad, protocolos, tipos de ataques a la red de datos, intromisión informática, entre otras; necesarias para establecer un marco teórico del contexto en el que se desarrollará el proyecto.
- 2 Identificación de las tecnologías de hardware necesarias para construir la arquitectura de red requerida y llevar a cabo el monitoreo del tráfico de la red de datos, como así también el diseño, instalación y configuración de todo el equipamiento de hardware en cuestión.
- 3 Identificación de las tecnologías de software necesarias para llevar a cabo tanto la simulación y el monitoreo de la red, como así también las técnicas de minería de datos



- y el Sistema de Detección de Intrusos (IDS), y posteriormente realizar la instalación y configuración de las tecnologías de software en cuestión.
- 4 Generación y captura del tráfico de red “normal” y “anormal” en la red de datos simulada.
 - 5 La creación de uno o más modelos de minería de datos con las bases de datos GureKKD data base y NSL KDD data base, que permitan detectar ataques e intromisiones, los cuáles serán aplicados sobre los datos capturados en la red de datos simulada con el fin de detectar y clasificar los ataques realizados a la misma.
 - 6 Generación de reglas de detección de intrusos a partir de los resultados obtenidos del/los modelo/s de minería de datos aplicados al conjunto de datos capturados (clasificados como “normal” o como algún tipo de ataque).
 - 7 La puesta en marcha del IDS en la red de datos, poniendo a prueba las reglas de detección de intrusos obtenidas con la ejecución, nuevamente, de los ataques realizados.
 - 8 La realización de un análisis de los resultados obtenidos en la aplicación del modelo de IDS propuesto, permitiendo visualizar información mediante tablas estadísticas y gráficas necesarias para tomar las decisiones correspondientes en materia de seguridad.

2.6 Criterios de Éxito

Con el presente proyecto se pretende:

- Concientizar a las personas que se encuentran al frente de las empresas sobre la importancia de la seguridad de la información y el adecuado tratamiento de los datos para evitar pérdidas de gran valor.
- Lograr que las organizaciones tomen medidas de seguridad para resguardar la información que fluye a través de los diferentes sectores de la empresa y hacia el exterior, entendiendo sobre la severidad y los riesgos que conllevan los ataques o intromisiones informáticas.
- Realizar un modelo de detección de intrusos basado en técnicas de minería de datos, mediante la utilización de herramientas de software libre capaz de monitorear y detectar de una manera precisa y eficaz las diferentes amenazas que una red de datos puede llegar a sufrir.
- Generar, en base a los resultados obtenidos de las técnicas de minería de datos aplicadas al conjunto de datos capturados, una serie de reglas de detección de intrusos que permitan al IDS detectar y alertar de forma precisa sobre ataques y/o intromisiones informáticas.
- En base a los resultados obtenidos del modelo de IDS propuesto, realizar informes que ayuden a definir nuevas medidas y políticas de seguridad para minimizar el riesgo de intrusión.
- Obtener los conocimientos necesarios sobre minería de datos, ataques informáticos y sistemas de detección de intrusos para poder concluir con el proyecto de grado de una forma óptima y elaborar un resultado de calidad.



- Elaborar un modelo de detección de intrusos basado en técnicas de minería de datos que establezca bases fundamentales para futuros proyectos similares que quieran profundizar en el uso de la minería de datos como una herramienta fundamental para la detección de intrusiones informáticas y así plantear nuevas herramientas de seguridad más eficientes.



Capítulo 3 – Marco Teórico

3.1 Introducción

En este capítulo se desarrollará todo el aspecto teórico que hace al proyecto en cuestión, juntamente con el análisis y selección de las metodologías para el desarrollo de la solución propuesta.

3.2 Marco Teórico

En esta sección daremos a conocer los distintos aspectos teóricos necesarios a tener en cuenta para el desarrollo del presente proyecto, que abarcará la descripción de las diferentes tecnologías utilizadas, tanto en el contexto de la seguridad de las redes de datos, como por ejemplo definiciones o conceptos de detección de intrusos en redes de datos, tipos y clasificación de ataques, entre otros; como la de minería de datos, ya sea conceptos o definiciones de las diferentes metodologías, herramientas y tecnologías utilizadas, con el fin de permitir al lector la comprensión e interpretación teórica del trabajo en cuestión.

El marco teórico se dividirá en dos secciones, las cuales se desarrollan a continuación:

3.3 Seguridad de Redes de Datos

En la actualidad resulta indispensable para las organizaciones resguardar y proteger su información, que viaja a través de las redes de datos, debido a que los grandes avances en materia de tecnologías de la información, telecomunicaciones e internet llevan a las redes de datos a ser cada vez más grandes y abiertas, para permitir a las organizaciones ampliar cada vez más sus horizontes comerciales y mantenerse actualizadas, pero al mismo tiempo esto conlleva un aumento de los riesgos en la seguridad de la información.

Seguridad en redes es mantener bajo protección los recursos y la información con que se cuenta en la red. A través de procedimientos, basados en una política de seguridad, tales que permitan el control de lo actuado. El incremento del comercio móvil y las redes inalámbricas, aumenta la importancia de encontrar soluciones integrales de seguridad, que sean transparentes y flexibles. (Fundacion Carlos Slim, 2009)

Una de las medidas principales a tomar por las organizaciones en materia de seguridad de redes de datos, consiste en establecer políticas de seguridad que proporcionen una serie de pasos a seguir para gestionar adecuadamente los procesos de seguridad de la organización.

Las políticas de seguridad son los lineamientos que las empresas deben seguir para asegurar la confiabilidad de sus sistemas, ya que:

- *Comprenden una descripción de los recursos de la empresa que se van a proteger y justifican el porqué de ello.*



- *Establecen la forma de uso y limitaciones de los recursos y servicios de información críticos de la empresa.*
- *Establecen las medidas y acciones que se deben llevar a cabo en caso de que se presente alguna contingencia relacionada con los mismos.* (Fundacion Carlos Slim, 2009)

Las políticas de seguridad deben actualizarse a menudo para así poder estar adecuadas a los cambios constantes del entorno en materia de seguridad de redes de datos.

3.3.1 Intromisión Informática

La intromisión informática está relacionada con la capacidad que posee un usuario, un proceso, o un sistema de vulnerar las barreras de seguridad impuestas por una organización con el objetivo de ingresar indebidamente a otro sistema de información y por ende a los datos que maneja para así provocar inconsistencia en el sistema o atentar contra su integridad.

3.3.2 Definición de Ataques informáticos

Otros de los conceptos a tener en cuenta son los ataques informáticos. Hoy en día se conocen una gran variedad de ataques informáticos que están destinados a atentar contra la integridad de un sistema, red, o usuario. Se define como ataque informático:

Intento organizado y deliberado de una o más personas para causar daño o problemas a un sistema informático o red. Los ataques en grupo suelen ser hechos por bandas de piratas informáticos por diversión, para causar daño, buenas (relativamente buenas) intenciones, espionaje, obtención de ganancias, etc. (Alegsa, 2015)

3.3.3 Tipos de Ataques Informáticos

Cada día existen más usuarios malintencionados que están destinados a quebrantar la seguridad e integridad de un sistema informático, ya sea por hobbies o para adquirir algún beneficio económico, y pueden producirse desde la organización o de una fuente externa. Entre los tipos de ataques que podemos encontrar tenemos los siguientes:

- Software malicioso o malware en Internet. (Virus, gusanos y troyanos)
- Atacar servidores y la infraestructura de la red. (Denegación de servicio - DOS).
- Analizar los paquetes que fluyen por la red (Sniffers).
- Suplantación de identidad.
- Modificación y borrado de datos enviados.
- Inyección CRLF (CRLF Injection / HTTP Response Splitting)
- Inclusión de ficheros remotos (RFI Remote File Inclusion)
- Dominios Fantasmas



Imagen 1 – Tipos de ataques informáticos

(Fundacion Carlos Slim, 2009)

A continuación, se detallará los tipos de ataques utilizados para la simulación del tráfico de red del proyecto en cuestión:

Ataque Hydra-Fuerza Bruta con diccionarios a Servidor FTP

El ataque Hydra consiste en acceder de forma remota a un servidor FTP (Servidor de Transferencia de Archivos), para poder así obtener acceso a los archivos que aloja el mismo, con el objetivo de descargar información valiosa de una empresa que se encuentra alojada en esta ubicación de red o enviar archivos maliciosos. Se denomina fuerza bruta debido a que se utiliza un diccionario de datos (archivos de texto plano “.txt”) que aloja una cantidad determinada de posibles nombres de usuario y contraseñas que utiliza la Hydra para obtener mediante diferentes combinaciones los datos correctos del login que permiten el acceso al servidor ftp.

Hydra es una poderosa herramienta la cual permite realizar ataques de fuerza bruta a servicios online como ejemplo, FTP, SSH, Mysql, POP3, Telnet, etc. Por lo general estos servicios permiten la comunicación entre Cliente/Servidor, donde es posible interactuar de manera cómoda y de esta manera, enviar o recibir información.

El acrónimo de FTP es protocolo de transferencia de ficheros (File Transfer Protocol) y es un software cliente/servidor que permite a usuarios transferir ficheros entre ordenadores en una red TCP/IP. (Esteban, 2016)

Un ataque de fuerza bruta con diccionarios es la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso. (Ramos, 2014)



```
Archivo Editar Ver Buscar Terminal Ayuda
root@kali:~# hydra
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[-l LOGIN|-L FILE] [-p PASS|-P FILE]] [-C FILE] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME]
RT] [-x MIN:MAX:CHARSET] [-SuvVd46] [service://server[:PORT][[:OPT]]]

Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-C FILE colon separated "login:pass" format, instead of -L/-P options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel (per host, default: 16)
-U service module usage details
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)

Supported services: asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get} http[s]-{get|post}-form http-proxy ht
ap[s] irc ldap2[s] ldap3[-{cram|digest}|md5][s] mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres rd
sh s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.

Example: hydra -l user -P passlist.txt ftp://192.168.0.1
```



Imagen 2 – Ataque HYDRA

Ataque de Denegación de Servicio - (DOS/Slowloris)

Un ataque de Denegación de Servicio (DOS – siglas en inglés) consiste en enviar una masiva cantidad de paquetes a un determinado host con el fin de poder colapsarlo e imposibilitar que brinde los servicios que requieren los usuarios de una organización o usuarios finales.

Un "ataque por denegación de servicio" (DoS, Denial of service) tiene como objetivo imposibilitar el acceso a los servicios y recursos de una organización durante un período indefinido de tiempo. Por lo general, este tipo de ataques está dirigido a los servidores de una compañía, para que no puedan utilizarse ni consultarse.

La denegación de servicio es una complicación que puede afectar a cualquier servidor de la compañía o individuo conectado a Internet. Su objetivo no reside en recuperar ni alterar datos, sino en dañar la reputación de las compañías con presencia en Internet y potencialmente impedir el desarrollo normal de sus actividades en caso de que éstas se basen en un sistema informático.

Generalmente, estos ataques se dividen en dos clases:

- *Las denegaciones de servicio por saturación, que saturan un equipo con solicitudes para que no pueda responder a las solicitudes reales.*
- *Las denegaciones de servicio por explotación de vulnerabilidades, que aprovechan una vulnerabilidad en el sistema para volverlo inestable.*

Los ataques por denegación de servicio envían paquetes IP o datos de tamaños o formatos atípicos que saturan los equipos de destino o los vuelven inestables y, por lo tanto, impiden el funcionamiento normal de los servicios de red que brindan.



Cuando varios equipos activan una denegación de servicio, el proceso se conoce como "sistema distribuido de denegación de servicio" (DDOS, Distributed Denial of Service). Los más conocidos son Tribal Flood Network (TFN) y Trinoo. (ccm, 2016)



Imagen 3 – Ataque DOS y DDOS

(Vezzoni, 2013)

Escaneo de red y puertos con Herramienta NMAP de Kali Linux 2.0

Este tipo de ataque es usado para llevar a cabo un mapeo de la red e identificar los puertos disponibles en los hosts, con el objetivo de detectar vulnerabilidades en la red y que puedan ser utilizadas para fines ilícitos.

Nmap ("Network Mapper") es una herramienta de código abierto para la exploración de la red y auditoría de seguridad. Fue diseñado para escanear rápidamente grandes redes. Nmap utiliza paquetes IP en bruto en formas novedosas para determinar qué hosts están disponibles en la red, qué servicios (nombre de la aplicación y versión) estos equipos ofrecen, qué sistemas operativos (y versiones del sistema operativo) que se están ejecutando, qué tipo de filtros de paquetes / cortafuegos están en uso, y docenas de otras características. Mientras Nmap es comúnmente utilizado para las auditorías de seguridad, muchos sistemas y administradores de red lo utilizan para tareas de rutina, tales como inventario de red, gestión de horarios de servicios de actualización y supervisión de host o tiempo de servicio. (Foro Kali Linux, 2013)

Algunos de los parámetros y opciones más usadas en NMAP se listan a continuación:

Uso: *nmap [Tipo(s) de Análisis] [Opciones] {especificación de objetivos}*

ESPECIFICACIÓN DE OBJETIVO:

Se pueden indicar nombres de sistemas, direcciones IP, redes, etc.

Ej.: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254



- iL <archivo_entrada>: Lee una lista de sistemas/redes del archivo.
- iR <número de sistemas>: Selecciona objetivos al azar
- exclude <sist1[,sist2][,sist3],...>: Excluye ciertos sistemas o redes
- excludefile <fichero_exclusión>: Excluye los sistemas indicados en el fichero

DESCUBRIMIENTO DE HOSTS:

- sL: Sondeo de lista - Simplemente lista los objetivos a analizar
- sP: Sondeo Ping - Sólo determina si el objetivo está vivo
- P0: Asume que todos los objetivos están vivos
- PS/PA/PU [listadepuertos]: Análisis TCP SYN, ACK o UDP de los puertos indicados
- PE/PP/PM: Solicita un análisis ICMP del tipo echo, marca de fecha y máscara de red
- n/-R: No hacer resolución DNS / Siempre resolver [por omisión: a veces]
- dns-servers <serv1[,serv2],...>: Especificar servidores DNS específicos
- system-dns: Utilizar la resolución del sistema operativo

TÉCNICAS DE ANÁLISIS:

- sS/sT/sA/sW/sM: Análisis TCP SYN/Connect()/ACK/Window/Maimon
- sN/sF/sX: Análisis TCP Null, FIN, y Xmas
- scanflags <indicador>: Personalizar los indicadores TCP a utilizar
- sI <sistema zombi[:puerto_sonda]>: Análisis pasivo («Idle», N. del T.)
- sO: Análisis de protocolo IP
- b <servidor ftp rebote>: Análisis por rebote FTP

ESPECIFICACIÓN DE PUERTOS Y ORDEN DE ANÁLISIS:

- p <rango de puertos>: Sólo sondear los puertos indicados
Ej.: -p22; -p1-65535; -p U: 53, 111, 137, T: 21-25, 80, 139,8080
- F: Rápido - Analizar sólo los puertos listados en el archivo nmap-services
- r: Analizar los puertos secuencialmente, no al azar.

DETECCIÓN DE SISTEMA OPERATIVO

- O: Activar la detección de sistema operativo (SO)



--osscan-limit: Limitar la detección de SO a objetivos prometedores

--osscan-guess: Adivinar el SO de la forma más agresiva

nmap -v -iR 10000 -P0 -p 80 (nmap.org, 2016)

Ataque Warez (client/master)

Estos tipos de ataques están principalmente relacionados al software ilegal. Por un lado, tenemos el denominado warezclient, que hace referencia a la descarga de software ilegal de forma anónima desde algún servidor web, y por otro lado el warezmaster que es relacionado con la subida de software ilegal a la web.

Warez se refiere principalmente al material bajo copyright, lo cual está distribuido con infracción al derecho de autor. De esta manera, Warez es referido generalmente a distribuciones ilegales por grupos o personas en contraposición a la acción de compartir archivos “cliente a cliente” entre amigos o grandes grupos de personas con intereses comunes. Normalmente, Warez no se refiere a la guerra contra el software comercial, ya que este término fue creado inicialmente por miembros de varios círculos informáticos de manera que deriva del término software, pero desde entonces se ha convertido en un término común entre los usuarios de internet.

Así mismo, el término Warez (falsificación) se usa para referirse al uso no autorizado de la propiedad intelectual, donde “no autorizado” hace referencia a la falta de autoridad proporcionada por el poseedor de los derechos de autor y de uso bajo la jurisdicción de la autoridad legal en la que recae dicha jurisdicción. (Ortiz, 2001)

3.3.4 Monitoreo de las redes de datos

Debido a que la complejidad de las redes de cómputos de las diferentes organizaciones aumenta cada vez más, es decir las redes soportan nuevas aplicaciones y servicios estratégicos que son de un valor muy importante, es necesario llevar a cabo un análisis y monitoreo de estas redes de datos para evitar cualquier tipo de problemas que puedan atentar contra la seguridad (integridad) de las mismas.

El monitoreo define un proceso continuo de recolección y análisis de datos con el fin de anticipar problemas en la red. Un sistema de gestión y monitoreo de redes está diseñado para ver la red entera como una arquitectura unificada con direcciones y etiquetas asignadas a cada punto y con atributos específicos en cada elemento y enlace del sistema conocidos. (Herrera, 2009)

3.3.5 Herramientas disponibles para el monitoreo de red

A la hora de escoger una herramienta para el monitoreo de red se debe tener en cuenta varios factores tales como: la eficiencia, el costo, la calidad, entre otros. Entre las diferentes herramientas de monitoreo de red que existen podemos identificar las siguientes:



- *OpManager: ManageEngine OpManager es una solución de monitorización de redes, servidores y aplicaciones. Ofrece gestión avanzada del rendimiento y disponibilidad de recursos críticos TI tales como routers, switches, firewalls, aceleradores WAN, puntos de acceso, llamadas VoIP, servidores físicos y virtuales, controladores de dominio, aplicaciones, etc. (Ireo, 2016)*
- *Nagios: Nagios XI proporciona una monitorización de todos los componentes de la infraestructura de misión crítica, incluyendo aplicaciones, servicios, sistemas operativos, protocolos de red, métricos de sistemas y la infraestructura de red. (Nagios - The Industry Standard In IT Infrastructure Monitoring, 2016)*
- *Zabbix: Zabbix es el último software de nivel empresarial diseñado para la monitorización en tiempo real de millones de métricas recogidas de decenas de miles de servidores, máquinas virtuales y dispositivos de red. Zabbix es Open Source y viene sin costo alguno. (Zabbix - The Enterprise-class Monitoring Solution for Everyone, 2016)*
- *Cacti: Cacti es una solución completa de gráficos de red diseñado para aprovechar el poder de almacenamiento de datos de RRDtool y la funcionalidad de gráficos. Cacti proporciona plantillas gráficas avanzadas, múltiples métodos de adquisición de datos y gestión de usuarios. Todo esto está envuelto en una interfaz intuitiva y fácil de usar, para instalaciones de redes LAN básicas o redes complejas con cientos de dispositivos. (Cacti, 2016)*
- *Zenoss: Zenoss es el proveedor líder de monitoreo de TI unificada y software de gestión de infraestructuras de TI físicos, virtuales y basados en la nube. (Zenoss - Own IT, 2016)*
- *Munin: Munin es una herramienta de monitorización de recursos en red que pueden ayudar a analizar las tendencias de recursos y problemas de rendimiento. Está diseñado para ser muy plug and play. Una instalación predeterminada proporciona una gran cantidad de gráficos con casi ningún trabajo. (Munin, 2016)*
- *Wireshark: Wireshark es un analizador de protocolos de red para Unix y Windows utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica. (Wireshark, 2016)*
- *TCPDUMP: un analizador de paquetes de gran alcance de línea de comandos; y libpcap, un portátil de C/C++ biblioteca para la captura de tráfico de red. (Tcpdump, 2016)*
- *WinDump: WinDump es la versión para Windows de tcpdump, el analizador de redes de línea de comandos para UNIX. WinDump es totalmente compatible con tcpdump y se puede utilizar para ver, diagnosticar y guardar en el tráfico de red de disco de acuerdo con diversas reglas complejas. Se puede ejecutar en Windows 95, 98, ME, NT, 2000, XP, 2003 y Vista. (Windump, 2016)*



3.3.6 Detección de Intrusos

Un sistema de detección de intrusos es un mecanismo utilizado para monitorear el tráfico de datos de una red, para descubrir compartimientos anormales en el flujo de datos que pueden derivar en posibles ataques o intromisiones a un sistema informático de una organización o empresa.

*El término **IDS** (Sistema de detección de intrusiones) hace referencia a un mecanismo que, sigilosamente, escucha el tráfico en la red para detectar actividades anormales o sospechosas, y de este modo, reducir el riesgo de intrusión. (ccm, 2003)*

3.3.7 Clasificación de los IDS

Existen dos claras familias importantes de IDS:

- *El grupo **N-IDS** (Sistema de detección de intrusiones de red), que garantiza la seguridad dentro de la red.*
- *El grupo **H-IDS** (Sistema de detección de intrusiones en el host), que garantiza la seguridad en el host. (ccm, 2003)*

3.3.8 Tecnologías disponibles para la Detección de intrusos

Entre las tecnologías de detección de intrusos disponibles que se pueden utilizar tenemos las siguientes:

- *Snort: Se trata de un sistema de prevención de intrusiones de código abierto capaz de análisis de tráfico en tiempo real y registro de paquetes. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos aprovechar alguna vulnerabilidad, análisis de protocolos, etc. conocidos. Todo esto en tiempo real. (Snort, 2016) (Maestros del Web, 2003)*
- *Suricata: Suricata es un IDS de red de alto rendimiento, IPS y motor de supervisión de la seguridad de la red. Es Open Source y propiedad de una fundación sin fines de lucro organizado por la comunidad, la Fundación Seguridad de la Información abierta (OISF). Suricata es desarrollado por el OISF y sus proveedores de apoyo. (Suricata IDS, 2016)*
- *Bro: Bro es un sistema de detección de intrusiones para UNIX/Linux Open Source que analiza el tráfico de red en busca de actividad sospechosa. su característica principal es que sus reglas de detección están basadas en scribd en lenguaje nativo que suponen políticas (policies) que son las encargadas de detectar, generar logs o eventos y acciones a nivel de sistema operativo. (Seguridad y Redes, 2011)*
- *Kismet: Kismet es un sniffer, un husmeador de paquetes, y un sistema de detección de intrusiones para redes inalámbricas 802.11. Para usar Kismet necesitamos de un sistema basado en Unix (Linux, FreeBSD u OSX) y una tarjeta WiFi que permita modo monitoreo o promiscuo. (Highsec, 2014)*



- OSSEC: OSSEC es una plataforma de monitorización y control de sistemas. Se trata de un sistema HIDS (Host Intrusion Detection System), es decir, un sistema de detección de intrusos. También se considera SIM (Security Incident Management) A pesar de eso tiene más funciones, ya que tiene un servicio de log centralizado, un sistema de alertas y avisos, entre otras muchas funciones. (Ossec, 2016)
- Security Onion: es una distribución de Linux para la detección de intrusos, control de seguridad de la red y de gestión de registros. Está basado en Ubuntu y contiene Snort, Suricata, Bro, OSSEC, sguil, Squert, ELSA, Xplico, NetworkMiner, y muchas otras herramientas de seguridad. (Security Onion, 2016)

3.4 Metodología para administración de redes

Para llevar a cabo una correcta aplicación en el manejo de las redes de datos utilizaremos para el proyecto en cuestión una metodología de administración de redes basada en modelos funcionales estándar de la ITU (Organismo especializado de las Naciones Unidas para las Tecnologías de la Información y la Comunicación – TIC) y de la ISO (Organización Internacional de Normalización).

Estos modelos detallan las tareas y funciones que deben ser ejecutadas en el proceso de administración de redes.

El término administración de redes es definido como la suma total de todas las políticas, procedimientos que intervienen en la planeación, configuración, control, monitoreo de los elementos que conforman a una red con el fin de asegurar el eficiente y efectivo empleo de sus recursos. Lo cual se verá reflejado en la calidad de los servicios ofrecidos.

Tres dimensiones de la administración de redes.

a) Dimensión Funcional. Se refiere a la asignación de tareas de administración por medio de áreas funcionales.

b) Dimensión Temporal. Se refiere a dividir el proceso de administración en diferentes fases cíclicas, incluyendo las fases de planeación, implementación y operación.

c) Dimensión del escenario. Se refiere al resto de los escenarios adicionales al de administración de redes, como son administración de sistemas, administración de aplicaciones, etc.

Dimensión Funcional

Existen diversos modelos sobre arquitecturas de administración de redes. Tanto el modelo TMN (Serie de recomendaciones M.3000 de la ITU-T), como el modelo OSI-NM (ISO/IEC 7498-4:1989) son modelos funcionales que dividen la administración de una red en áreas funcionales (configuración, fallas, desempeño, contabilidad y seguridad), definiendo de ésta



forma una estructura organizacional con funciones bien definidas. De esto se deriva el nombre de modelos funcionales. (Untiveros, 2004)

La metodología en cuestión sugiere la creación de diferentes áreas funcionales para llevar a cabo la administración de redes, las cuales serán detalladas a continuación:

3.4.1 Administración de la configuración

Las actividades a desarrollar en esta etapa son:

- Planeación y diseño de la red: La meta de esta actividad es satisfacer los requerimientos inmediatos y futuros de la red, reflejarlos en su diseño hasta llegar a su implementación.
- Instalaciones de hardware: Las tareas de la instalación de hardware contemplan, tanto la agregación como la sustitución de equipamiento, y abarcan un dispositivo completo, como un switch o un ruteador; o solo una parte de los mismos, como una tarjeta de red, tarjeta procesadora, un módulo, etc.
- Instalación y administración del software: El objetivo de estas actividades es conseguir un manejo adecuado de los recursos de hardware y software dentro de la red.
- Aprovisionamiento: Esta tarea tiene la función de asegurar la redundancia de los elementos de software y hardware más importantes de la red. Puede llevarse a cabo en diferentes niveles, a nivel de la red global o de un elemento particular de la red. Es la responsable de abastecer los recursos necesarios para que la red funcione, elementos físicos como conectores, cables, multiplexores, tarjetas, módulos, elementos de software como versiones de sistema operativo, parches y aplicaciones.

3.4.2 Administración del rendimiento

Tiene como objetivo recolectar y analizar el tráfico que circula por la red para determinar su comportamiento en diversos aspectos, ya sea en un momento en particular (tiempo real) o en un intervalo de tiempo. Esto permitirá tomar las decisiones pertinentes de acuerdo al comportamiento encontrado.

La administración del rendimiento se divide en 2 etapas:

- Monitoreo: El monitoreo consiste en observar y recolectar la información referente al comportamiento de la red.
- Análisis: Una vez recolectada la información mediante la actividad de monitoreo, es necesario interpretarla para determinar el comportamiento de la red y tomar decisiones adecuadas que ayuden a mejorar su desempeño.



3.4.3 Administración de fallas

Tiene como objetivo la detección y resolución oportuna de situaciones anormales en la red. El proceso de la administración de fallas consiste de distintas fases:

- _ *Monitoreo de alarmas.* Se realiza la notificación de la existencia de una falla y del lugar donde se ha generado. Esto se puede realizar con el auxilio de las herramientas basadas en el protocolo SNMP.
- _ *Localización de fallas.* Determinar el origen de una falla.
- _ *Pruebas de diagnóstico.* Diseñar y realizar pruebas que apoyen la localización de una falla.
- _ *Corrección de fallas.* Tomar las medidas necesarias para corregir el problema, una vez que el origen de la misma ha sido identificado.
- _ *Administración de reportes.* Registrar y dar seguimiento a todos los reportes generados por los usuarios o por el mismo administrador de la red.

3.4.4 Administración de la contabilidad

Es el proceso de recolección de información acerca de los recursos utilizados por los elementos de la red, desde equipos de interconexión hasta usuarios finales. Esto se realiza con el objetivo de realizar los cobros correspondientes a los clientes del servicio mediante tarifas establecidas. Este proceso, también llamado tarificación, es muy común en los proveedores de servicio de Internet o ISP.

3.4.5 Administración de la seguridad

Su objetivo es ofrecer servicios de seguridad a cada uno de los elementos de la red así como a la red en su conjunto, creando estrategias para la prevención y detección de ataques, así como para la respuesta ante incidentes de seguridad.

Prevención de ataques

El objetivo es mantener los recursos de red fuera del alcance de potenciales usuarios maliciosos. Una acción puede ser la implementación de alguna estrategia de control de acceso. Obviamente, los ataques solamente se reducen, pero nunca se eliminan del todo.

Detección de intrusos

El objetivo de la detección de intrusos se puede lograr mediante un sistema de detección de intrusos que vigile y registre el tráfico que circula por la red apoyado en un esquema de notificaciones o alarmes que indiquen el momento en que se detecte una situación anormal en la red.



3.5 Minería de datos

La minería de datos actualmente es una herramienta con muchos beneficios para las organizaciones, principalmente por que se encarga de trabajar con el principal recurso que estas tienen, que es “La información”. Hoy en día existe una tendencia de almacenar gran cantidad de información con el motivo de tener registro de cada movimiento, evento o suceso que ocurre dentro y fuera de una organización, y cada vez más el método de almacenamiento tiende hacia la digitalización dejando de lado el papel físico (ya que cada vez cuestan menos los medios de almacenamientos digitales). Así mismo, con gran cantidad de información, no se sabe cómo hacer uso de la misma para obtener beneficios, teniendo únicamente una noción superficial de la información (mirando la información almacenada como partes independientes sin relación) y no pudiendo observar el conocimiento oculto que se puede descubrir al relacionar y hacer un uso inteligente y automatizado de toda la información almacenada. Por ello la minería de datos propone utilizar toda la información disponible con el fin de obtener un conocimiento exclusivo, mientras más información se tenga más alto será el nivel de precisión del conocimiento obtenido. Mediante la ejecución de subprocesos y la aplicación de técnicas y algoritmos, la minería de datos proporciona, no solamente un análisis de los eventos del pasado, sino también una predicción hacia el futuro basada en la detección de patrones y reglas. De esta manera el fin último de la minería de datos puede verse reflejado al brindar apoyo en la toma de decisiones de las organizaciones proporcionando de manera cuantitativa predicciones sobre las acciones a realizar.

3.5.1 Definición de Minería de Datos

Podemos definir a la minería de datos como el proceso de analizar los datos de grandes bases de datos o almacenes de datos, mediante el uso de herramientas, técnicas y algoritmos para la detección de patrones que representan el conocimiento exclusivo potencialmente útil para el usuario. Esta detección de patrones se logra por la aplicación de modelos matemáticos y estadísticos que son la base de los algoritmos usados en el proceso de minería de datos.

3.5.2 Técnicas de Minería de Datos

A diferencia de los métodos tradicionales de análisis de datos, la minería de datos está basada en un análisis orientado al descubrimiento, específicamente al descubrimiento del conocimiento, en contraste con los tradicionales análisis que requerían de una hipótesis que verificar para cumplir con sus objetivos, la minería de datos no requiere de conocimiento previo sobre lo que se quiere descubrir sino que detecta patrones que anteriormente no se podían observar, haciendo uso de toda la información almacenada del sector o área de interés que se desea para obtener nuevo conocimiento. Lo único que se requiere tener o saber previamente es establecer acerca de qué se quiere tener conocimiento.

Para realizar el análisis antes mencionado se hace uso de técnicas y/o algoritmos con distintos grados de complejidad basados en modelos matemáticos y estadísticos. Las técnicas y/o algoritmos los podemos clasificar en dos grupos: supervisados o predictivos, y no supervisados o de descubrimiento del conocimiento.



Los algoritmos supervisados son aquellos que predicen el valor de un atributo (etiqueta) de un conjunto de datos, conociendo previamente el valor de otros atributos (atributos descriptivos). A partir del uso de un conjunto de datos donde se conocen los valores de los atributos descriptivos y la etiqueta, se induce una relación entre estos para predecir el valor de la etiqueta de otro conjunto de datos con los mismos atributos descriptivos. Esta manera de trabajar se conoce como aprendizaje supervisado y realiza en dos fases: Entrenamiento (construcción de un modelo usando un subconjunto de datos con etiqueta conocida) y Prueba (prueba del modelo sobre el resto de los datos).

Cuando en una aplicación de la minería de datos no se posee los requisitos necesarios para la realización de la predicción, se recurre a los algoritmos no supervisados o de descubrimiento del conocimiento donde se descubren patrones y tendencias en los datos actuales, sin la necesidad de usar datos históricos.

En la siguiente tabla se muestran las técnicas más usadas de ambas categorías:

SUPERVISADAS	NO SUPERVISADAS
Arboles de Decisión	Detección de desviaciones
Redes neuronal	Segmentación
Regresión	Agrupamiento
Series temporales	Reglas de Asociación
	Patrones secuenciales

Cuadro 1 - Clasificación de técnicas minería de datos

3.5.3 Aprendizaje automático

El aprendizaje automático (AA) es una rama de la inteligencia artificial que permite a las computadoras aprender, se basa en algoritmos capaces de generalizar comportamiento y reconocer patrones a partir de información suministrada en forma de ejemplos. En otras palabras, es un proceso que permite obtener afirmaciones generales a partir de afirmaciones particulares por medio de la generalización.

Uno de los campos de aplicación donde actúa el aprendizaje automático es en la minería de datos, ya que ambas disciplinas están enfocadas en el análisis de datos, en donde el aprendizaje automático se centra más en estudiar la complejidad computacional de los problemas con la intención de hacerlos factibles desde el punto de vista práctico. Una de las tareas fundamentales del AA es extraer conocimiento sobre algunas propiedades no observadas de un objeto basándose en propiedades que si han sido observadas (o incluso propiedades observadas en otros objetos similares), es decir, predecir comportamiento futuro a partir un comportamiento del pasado.



El concepto a tener bien en claro antes de trabajar con el aprendizaje automático es “Aprendizaje”, ya que a partir de entender este concepto es que vamos a poder aplicar el AA para dar soluciones prácticas a problemas puntuales. Una definición general de aprendizaje dentro del contexto humano puede ser:

“Proceso a través del cual se adquieren o modifican habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio, la experiencia, la instrucción, el razonamiento y la observación.” (Caparrini, 2015)

Aplicando este concepto al aprendizaje automático (AA), vamos a considerar todo aquel conocimiento que una maquina puede aprender a partir de la experiencia, por lo tanto, para que el AA permanezca constante debemos alimentar la experiencia por medio de objetos de entrenamiento para que de esta forma la maquina reconozca patrones y pueda aplicarlos sobre otros objetos distintos.

Dentro del aprendizaje automatizado podemos reconocer 4 tipos de aprendizaje:

- *Aprendizaje inductivo: En donde se buscan patrones comunes para describir los ejemplos disponibles.*
- *Aprendizaje analítico o deductivo: Se aplica la deducción para obtener descripciones generales a partir de un ejemplo y su explicación.*
- *Aprendizaje genético: Aplica algoritmos inspirados en la teoría de la evolución para encontrar descripciones generales a conjuntos de ejemplos.*
- *Aprendizaje conexionista: Busca descripciones generales mediante el uso de capacidad de adaptación de redes de neuronas artificiales.*

(Universidad Politécnica de Cataluña, 2014)

El tipo de aprendizaje más utilizado en aplicaciones prácticas es el Aprendizaje Inductivo Supervisado. Algunos de los métodos o técnicas más utilizados en el Aprendizaje Inductivo Supervisado son: Arboles de decisión/regla de inducción, modelos probabilísticos y redes de neuronas artificiales, todas estas técnicas son técnicas de clasificación que intentan predecir la clasificación de objetos sobre un conjunto de clases prefijadas.

Como se mencionó la clasificación posee técnicas que sirven para el aprendizaje inductivo supervisado, por lo cual podemos hablar de la clasificación supervisada, la cual podemos definir de la siguiente manera:

La clasificación supervisada cuenta con un conocimiento a priori, es decir para la tarea de clasificar un objeto dentro de una categoría o clase contamos con modelos ya clasificados (objetos agrupados que tienen características comunes). Podemos diferenciar dos fases dentro de este tipo de clasificación:

En la primera fase tenemos un conjunto de entrenamiento o de aprendizaje (para el diseño del clasificador) y otro llamado de test o de validación (para clasificación), estos nos servirán para construir un modelo o regla general para la clasificación.



En la segunda fase es el proceso en sí de clasificar los objetos o muestras de las que se desconoce la clase a las que pertenecen. (ATGC, 2008)

Así mismo podemos distinguir la clasificación no supervisada definiéndola de la siguiente manera:

A diferencia de la supervisada no contamos con conocimiento a priori, por lo que tendremos un área de entrenamiento disponible para la tarea de clasificación. A la clasificación no supervisada se la suele llamar también clustering.

En este tipo de clasificación contamos con “objetos” o muestras que tiene un conjunto de características, de las que no sabemos a qué clase o categoría pertenece, entonces la finalidad es el descubrimiento de grupos de “objetos” cuyas características afines nos permitan separar las diferentes clases. (ATGC, 2008)

3.5.4 Árbol de Decisión

Dentro del Aprendizaje Inductivo, los Árboles de Decisión son unos de los métodos más usados, en donde cada árbol de decisión representa un conjunto de reglas que permiten la clasificación de objetos, por ejemplo:

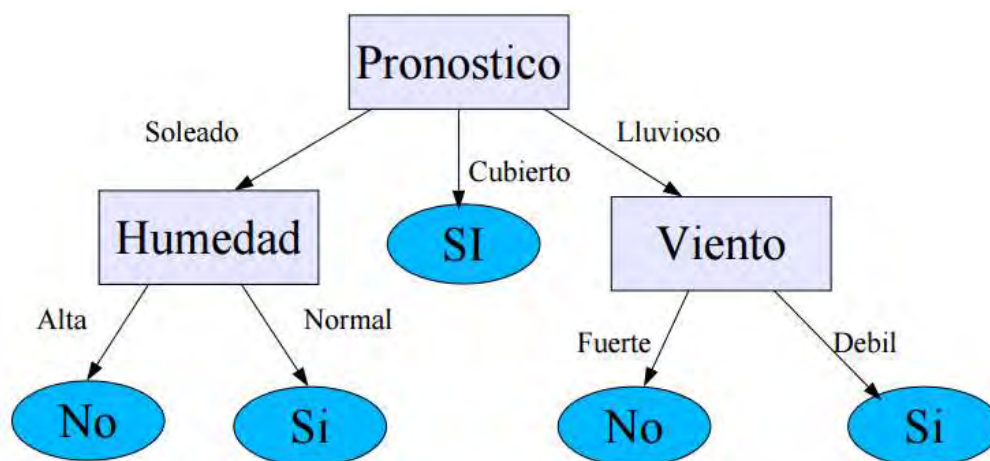


Imagen 4 – Árbol de Decisión

(Nebrija, 2016)

En la imagen podemos ver un árbol de decisión modelado para tomar la decisión de realizar alguna actividad en base al pronóstico del tiempo. En un árbol de decisión cada nodo representa un atributo del objeto, y cada rama representa un posible valor de ese atributo, finalizando en el último nodo hijo de la rama elegida, que determina la decisión a tomar.

Formalmente podemos definir un árbol de decisión como:



Una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento. Nos ayudan a tomar la decisión “más acertada”, desde un punto de vista probabilístico, ante un abanico de posibles decisiones. (Cruiz, 2014)

Dentro de un árbol de decisión podemos encontrar los siguientes componentes:

Nodo padre: Es el nodo que no tiene un nodo/s padre/s pero tiene uno o varios nodos hijos.

Nodo hijo intermedio: Posee uno o más nodos padres y tiene uno o varios nodos hijos.

Nodo hijo final: Posee uno o más nodos padres, pero no posee nodos hijos.

A la vez existen muchos algoritmos para construir un árbol de decisión, pero el más usado y el más simple de aprender es el ID3. Este algoritmo realiza una búsqueda por ascenso en el árbol, para cada nuevo nodo de decisión un atributo es elegido y a las ramas son distribuidas según sus valores, este proceso es repetido recursivamente hasta que cada rama pertenece a un nodo. La selección de cada atributo es decidida mediante una función de heurística.

3.5.5 Poda de Árboles de Decisión

Podar significa cambiar el modelo mediante la supresión de los nodos hijo, los único que pueden ser podados son los nodos hijos intermedios, que son tomados como nodos hijos finales, y los nodos hijos finales no pueden ser podados. Existen 2 tipos de poda de árboles de decisión:

- Pre-poda: En donde se aplica un criterio según van creciendo. Su objetivo es detener el crecimiento del árbol antes que se produzca.
- Pos-poda: Una vez creado el árbol se decide cuales nodos no serán tomados en cuenta de acuerdo a un criterio determinado. Elimina los subárboles que no contribuyen significativamente a la precisión de la clasificación.

A la vez existen 3 métodos para podar arboles de decisión:

- *Poda por estimación del error*
Un nodo se poda si el error de resustitución del nodo considerado como hoja es menor que error de resustitución del subárbol cuya raíz es el nodo.
- *Poda por coste-complejidad*
Se puede realizar utilizando un conjunto de prueba independiente del conjunto de entrenamiento o validación cruzada.
- *Poda pesimista*
Esta técnica utiliza solo el conjunto de casos de entrenamiento con los que se construye el árbol, con lo que nos ahorramos tener que reservar casos para realizar la simplificación del árbol.
(Alexa, 2008)



3.5.6 Validación cruzada

A la vez para determinar la precisión de la clasificación/predicción, se utiliza el método de la validación cruzada (usado en las técnicas supervisadas). Este método consiste en, a partir de una base de datos ya clasificada, obtener dos subconjuntos complementarios de datos, uno denominado subconjunto de entrenamiento y el otro denominado subconjunto de prueba. Seguidamente se realiza un análisis del subconjunto de entrenamiento donde se crea una función de aproximación (un árbol de decisión o reglas de clasificación) que se ajusta únicamente con los datos de entrenamiento, es decir, que refleja la manera en que los datos de entrenamiento están clasificados; y se valida dicho análisis con el subconjunto de prueba al aplicar la función de aproximación en los datos de prueba generando valores de clasificación, adicionales a los ya existentes (valores reales), para posteriormente corroborar la exactitud de esta clasificación con respecto a los valores reales de clasificación del subconjunto de prueba.

Existen 3 tipos de validaciones cruzadas:

Validación cruzada de K iteraciones

Consiste en dividir los datos de muestra (clasificados) en K subconjuntos donde uno es utilizado como subconjunto de prueba y el resto (K-1 conjuntos) conforman el subconjunto de entrenamiento. El proceso de validación cruzada se repite durante K iteraciones variando el subconjunto de prueba entre los k subconjuntos establecidos, finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método resulta muy preciso ya que se evalúa a partir de K combinaciones de datos de entrenamiento y de prueba. La elección del número de iteraciones depende de la medida del conjunto de datos, siendo lo más común utilizar la validación cruzada con 10 iteraciones.

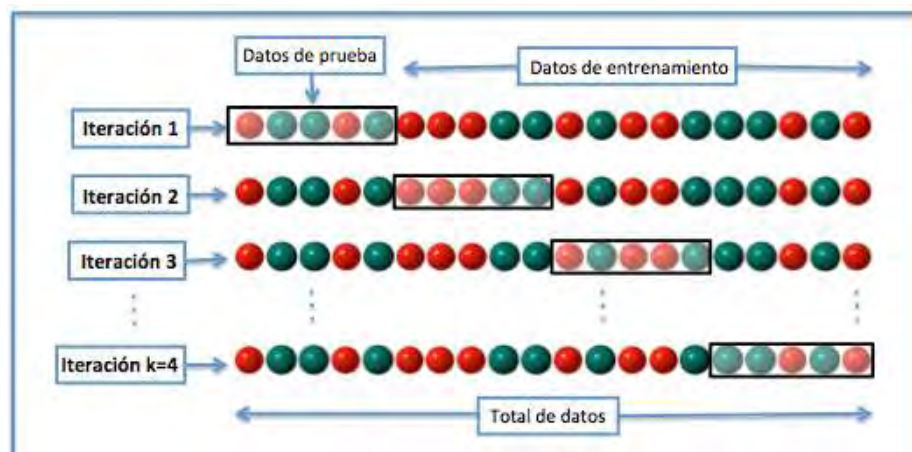


Imagen 5 – Validación cruzada de K iteraciones

(Fuentes, 2014)



Validación cruzada aleatoria

Consiste en dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final se corresponde a la media aritmética de los valores obtenidos para las diferentes divisiones.

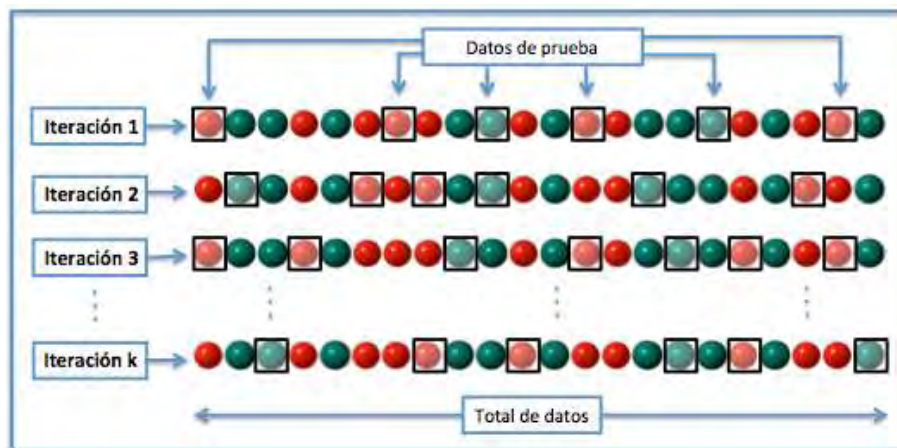


Imagen 6 – Validación cruzada aleatoria

(Fuentes, 2014)

Validación cruzada dejando uno fuera

Consiste en separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conforma los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

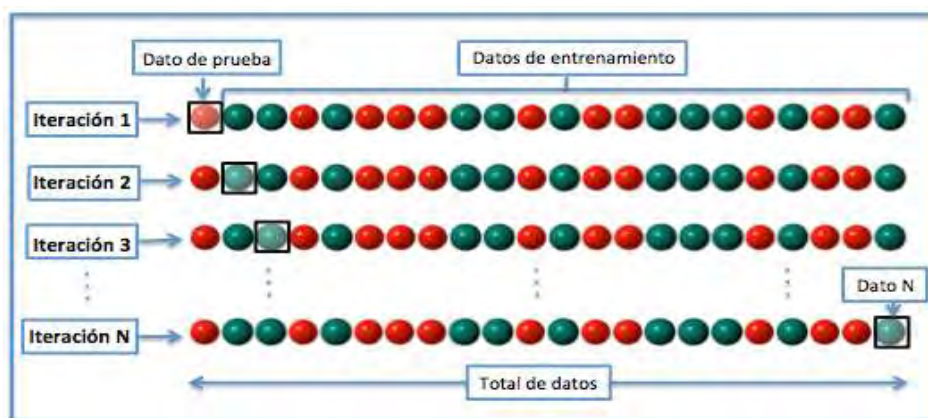


Imagen 7 – Validación cruzada dejando uno fuera

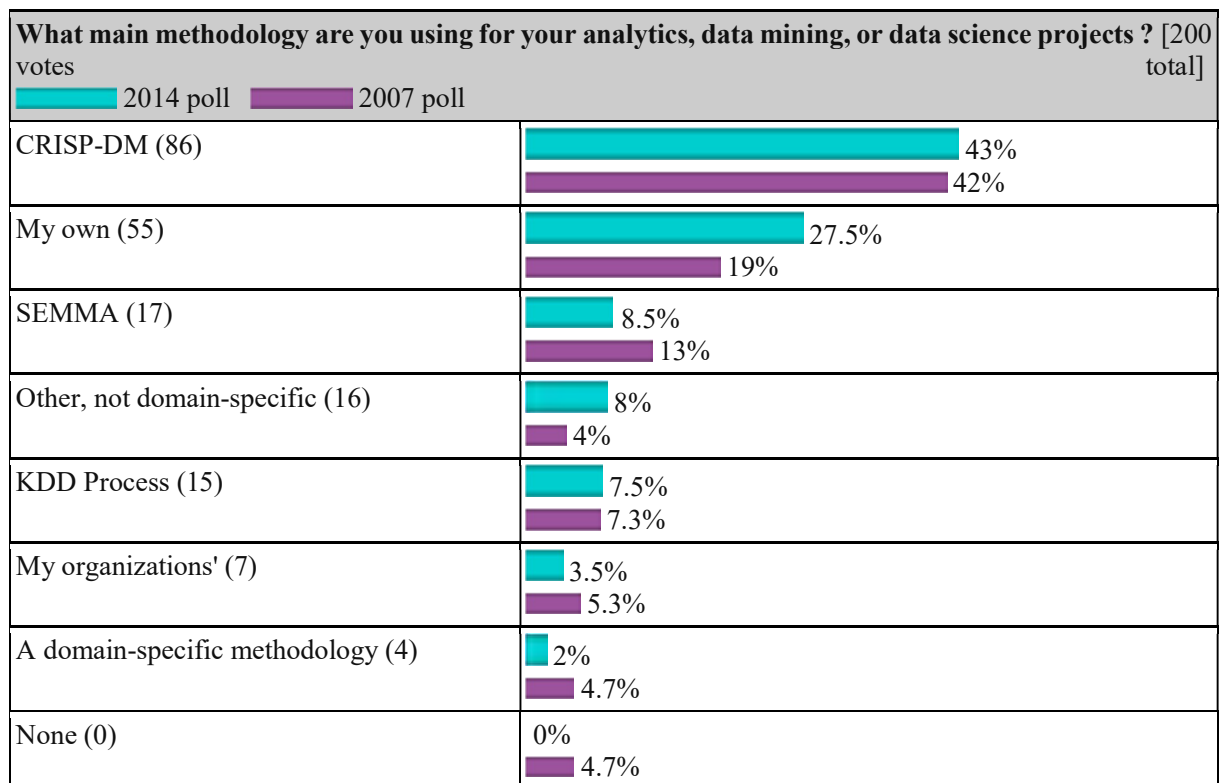
(Fuentes, 2014)



3.5.7 Metodologías aplicadas al proceso de minería de datos

Para la aplicación de la minería de datos existen distintas metodologías que establecen las etapas y los pasos a seguir con fin de obtener los resultados esperados, las más conocidas con: CRISP-DM, SEMMA Y KDD Process.

En el siguiente cuadro se pueden ver los nombres de las distintas metodologías existente y su porcentaje de uso, que se según una encuesta realizada por la kdnuggets a través de su página [www. Kdnuggets.com](http://www.Kdnuggets.com) los usuarios han determinado.



Cuadro 2 – Comparación de uso de metodologías de Minería de datos

(Piatetsky, kdnuggets, 2014)

Como se puede ver en la imagen, la metodología más usada por los usuarios hasta el año 2014 es CRISP-DM, por lo cual se hará la descripción de dicha metodología, ya que también, el presente proyecto no será la excepción en el uso de esta metodología.

CRISP-DM

CRISP-DM (CRoss Industry Standard Process for Data Mining, por sus siglas en ingles), es una metodología creada en el año 1999 por las empresas SPSS, Daimler Chrysler y NCR. Es una metodología construida en base a la experiencia de sus creadores, con enfoque práctico.



La metodología está estructurada en un proceso jerárquico de niveles de abstracción, específicamente 4 niveles de abstracción, que van desde lo general a lo específico.

CRISP-DM, propone en el nivel más alto de abstracción seis etapas para el proceso de minería de datos: entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación e implementación.

Cada etapa se descompone en un conjunto de tareas generales de segundo nivel. Estas tareas son generales ya que tratan de abarcar la mayoría de las situaciones posibles en minería de datos. En el tercer nivel de abstracción, se realiza una relación de las tareas generales definidas en el modelo con situaciones específicas de la aplicación. De esta forma, las tareas generales se traducen en tareas específicas para casos y proyectos concretos. Y por último en el cuarto nivel, encontramos las instancias de proceso, donde se describen las acciones, decisiones y resultados de un proyecto particular de minería de datos.

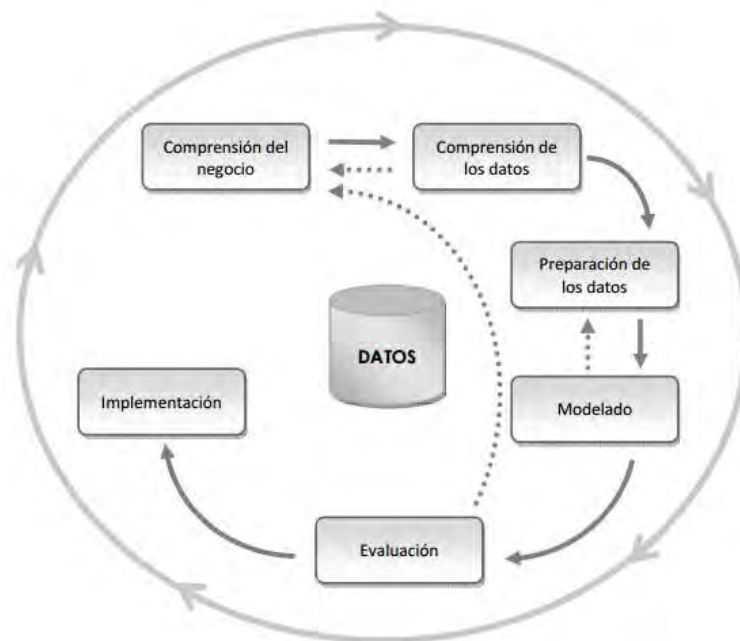


Imagen 8 – Ciclo de vida de CRISP – DM

(Moine, 2013)

Como se puede ver en la imagen, la metodología CRISP - DM comprende un ciclo de vida que termina en la etapa de implementación cuando la evaluación del modelo es aceptada. Además, se puede ver que no existe una rigidez en la ejecución de la secuencia de etapas, sino se permite volver a la etapa anterior en caso de ser necesario. A continuación, se describen cada una de las etapas de la metodología:

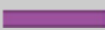


- **Comprensión del negocio:** En esta etapa se determinan los objetivos y requerimientos del proyecto de Minería de Datos desde una perspectiva del negocio, definiendo el problema de minería y el plan de trabajo.
- **Comprensión de los datos:** En esta etapa se realiza la recolección de datos que se utilizarán en el proyecto y la familiarización con los mismos.
- **Preparación de los datos:** Etapa donde se realizan aquellas actividades de tratamiento de los datos para construir la vista minable o conjunto de datos final sobre el cual se aplicarán las técnicas de minería.
- **Modelado:** En esta etapa se aplican las diversas técnicas y/o algoritmos de minería sobre el conjunto de datos para obtener la información oculta y los patrones implícitos en ellos.
- **Evaluación:** Etapa en la que se analizan los patrones obtenidos en función de los objetivos organizacionales. En esta etapa se debería determinar si se ha omitido algún objetivo importante del negocio y si el nuevo conocimiento será implementado, es decir, si se pasará a la próxima etapa.
- **Implementación:** Finalmente esta etapa consiste en la comunicación e implementación del nuevo conocimiento, el cual debe ser representado de forma entendible para el usuario.

3.5.8 Tecnologías y herramientas disponibles para la minería de datos

Para dar soporte a las tareas a realizar, establecidas por la metodología elegida en un proceso de Minería de Datos, existen distintas herramientas (software) que facilitan la ejecución de dichas tareas e incorporan las técnicas y algoritmos antes mencionados.

En el siguiente cuadro se pueden ver los nombres de las distintas herramientas y los porcentajes de uso, según una encuesta realizada por la kdnuggets a través de su página www.kdnuggets.com.

What Analytics, Big Data, Data mining, Data Science software you used in the past 12 months for a real project? [2759 voters]	
Legend: Red: Free/Open Source tools Green: Commercial tools Fuchsia: Hadoop/Big Data tools	 % users in 2015  % users in 2014  % users in 2013
R (1293), 3.6% alone	 46.9%  38.5%  37.4%
RapidMiner (870), 13.7% alone	 31.5%  44.2%  39.2%
SQL (853), 0% alone	 30.9%  25.3%  na



Python (837), 0% alone	
Excel (631), 0% alone	
KNIME (553), 6.7% alone	
Hadoop (507), 0% alone	
Tableau (341), 0% alone	
SAS base (313), 0.6% alone	
Spark (311), 0% alone	
Weka (310), 0% alone	
SAS Enterprise Miner (302), 3.6% alone	
Microsoft SQL Server (268), 0% alone	
MATLAB (243), 0% alone	
scikit-learn (229), 0% alone	
Unix shell/awk/gawk (221), 0% alone	

Cuadro 3 – Comparación de uso de herramientas de Minería de datos

(Piatetsky, Kdnuggets, 2015)

Como se puede ver en la imagen una de las elecciones más frecuentes para el proceso de minería de datos es la herramienta Rapid Miner, siendo esta una herramienta Free/Open Source. En el presente proyecto también se hará uso de esta herramienta debido a las características de uso que brinda: facilidad de aprendizaje, entendibilidad, interfaz amigable y definición clara de parámetros de entrada y salida.



Rapid Miner es una herramienta de Inteligencia de Negocios que proporciona un entorno integrado de aprendizaje automático para la Minería de Datos, se utiliza para aplicaciones de negocio, investigación, educación, formación, creación de modelos y desarrollo de aplicaciones, es compatible con todos los pasos de minería de datos. (Tecnología, Ingeniería en sistemas, 2015)

Dentro de la herramienta Rapid Miner podemos encontrar, entre otros, los siguientes algoritmos de árbol de decisión:

- **C4.5:** Este algoritmo es una mejora del algoritmo ID3, el árbol se crea a partir de la recursividad y utiliza una técnica heurística denominada proporción de ganancia que es una medida de información que considera diferentes números y probabilidades de los resultados de las pruebas.

El algoritmo considera todas las pruebas posibles que pueden dividir al conjunto de datos y selecciona la prueba que le haya generado mayor ganancia de información. Para cada atributo discreto se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo. Para cada atributo continuo se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos. (Gerardo, 2010)

- **Random Forest:** Es una combinación de árboles de clasificación, que mejora la clasificación mediante la incorporación de aleatoriedad en la construcción de cada árbol de decisión.

El algoritmo consiste en crear muchos árboles independientes a partir de datos de entrada. Se selecciona aleatoriamente un porcentaje de la muestra total, seguidamente se selecciona nuevamente de manera aleatoria un conjunto de atributos (nodos) en cada árbol. Cada árbol se evalúa de forma independiente y la predicción del bosque será la media de los árboles seleccionados. La proporción de árboles que toman una misma respuesta se interpreta como la probabilidad de la misma. (Quandare, 2015)

3.6 GureKDD Data

El archivo GureKDD Cup database 2008 es la adaptación de la KDDcup database 1999 (University of California, Irvine, 1999), utilizado en la tercera competencia internacional del descubrimiento de conocimiento y herramientas de minería de datos, la cual consistía en construir un detector de intrusión en la red, un modelo predictivo capaz de distinguir entre “malas” conexiones, llamadas intrusiones o ataques, y “buenas” conexiones (normales). Esta base de datos contiene un conjunto estándar de datos para ser auditado, que incluye una amplia variedad de ataques simulados en un entorno de red. La misma puede ser obtenida desde el siguiente link: http://www.aldapa.eus/res/gureKddcup/galdetegia_jaso.php



La GureKDD Contiene conexiones de la kddcup99, pero añade su carga útil (contenido de los paquetes de red) para cada una de las conexiones. Esto permitirá extraer información directamente de la carga útil de cada conexión para ser utilizado en el proceso de minería de datos. Los atributos y tipos de datos que posee la GureKDD son los siguientes:

ID	Nombre	Tipo de dato
1	connection_number	Integer
2	start_time	Real
3	orig_port	Integer
4	resp_port	Integer
5	Duration	Real
6	protocol_type	Nominal
7	Service	Nominal
8	Flag	Nominal
9	src_bytes	Integer
10	dst_bytes	Integer
11	Land	Binary
12	wrong_frangment	Integer
13	Urgent	Integer
14	Hot	Integer
15	num_failed_logins	Integer
16	logged_in	Binary
17	num_compromised	Integer
18	root_shell	Binary
19	su_attempted	Binary
20	num_root	Integer
21	num_file_creations	Integer
22	num_shell	Integer
23	num_access_files	Integer
24	num_outbound_cmds	Integer
25	is_hot_login	Binay
26	is_guest_login	Binary
27	Count	Integer
28	srv_count	Integer
29	serror_rate	Real
30	srv_serror_rate	Real
31	error_rate	Real
32	srv_error_rate	Real
33	same_srv_rate	Real
34	diff_srv_rate	Real
35	srv_diff_host_rate	Real
36	dst_host_count	Integer



37	dst_host_srv_count	Integer
38	dst_host_same_srv_rate	Real
39	dst_host_diff_srv_rate	Real
40	dst_host_same_src_port_rate	Real
41	dst_host_srv_diff_host_rate	Real
42	dst_host_serror_rate	Real
43	dst_host_srv_serror_rate	Real
44	dst_host_rerror_rate	Real
45	dst_host_srv_error_rate	Real

Cuadro 4 – Atributos de la GureKDD Cup data base 2008

La base de datos completa posee un tamaño de 9.3GB, en el presente proyecto se utilizará el archivo “gurekdd6percent” que contiene el 6% (tamaño: 30MB) de toda la captura del tráfico de red ocurrido en el evento, y así también cada conexión de red capturada está clasificada de acuerdo a los valores de los atributos que contiene el archivo. El mismo posee 178.810 registros (conexiones), cada uno de los cuales está clasificado como “normal” o como algún tipo de ataque.

3.7 NSL-KDD Data

La base de datos NSL-KDD (2012) es otra adaptación de la KDDcup database 1999, tal como la GureKDD posee la misma estructura y tipo de datos. La NSL-KDD fue obtenida gracias al aporte del Dr. Arash Habibi Lashkari, Director de Proyectos del Instituto Canadiense de Cyberseguridad (CIC), perteneciente a la Facultad de Ciencias de la Computación – Universidad de New Brunswick, el cual nos envió por correo un link de descarga temporal de la NSL-KDD. Contacto: A.Habibi.L@unb.ca (el contacto del Dr. Arash Habibi Lashkari se lo obtuvo de la siguiente pagina web: <http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html>)

Esta base puede ser aplicada como un conjunto de datos de referencia eficaz para ayudar a los investigadores a comparar los métodos de detección de intrusiones diferentes. Por otra parte, el número de registros de entrenamiento y de prueba NSL-KDD son razonables. Esta ventaja hace que sea asequible para realizar los experimentos sobre el conjunto completo sin la necesidad de seleccionar al azar una pequeña porción. En consecuencia, los resultados de la evaluación de diferentes trabajos de investigación serán consistentes y comparables. (Information Security Center of eXcellence, 2015)

El archivo utilizado para este proyecto es “KDDTrain-20Percent”, el cual posee un 20% de la base de datos total, con una cantidad de 25.193 registros en total.

3.7.1 Tipos de ataques

Los tipos de ataques identificados en ambos sets de datos (gurekdd6percent y KDDTrain-20Percent) se dividen en cuatro categorías principales:

Denegación de servicio o Denial of Service (DoS): El objetivo de este tipo de ataques es conseguir que un servidor no pueda ofrecer su servicio a un usuario legítimo. La técnica



habitual de este tipo de ataques se basa en el envío masivo de paquetes o técnica de inundación (flood) aunque en algunos casos basta con un único paquete que explote una vulnerabilidad conocida. Aunque este tipo de ataques puede generar grandes molestias su peligrosidad es moderada ya que no son peligrosos para la integridad de los sistemas informáticos.

Escaneo o Probing: Los ataques de este tipo se dedican a rastrear la red para obtener la mayor cantidad de información posible. Estos ataques suelen preceder a otros más dañinos. Es habitual que este tipo de ataques generen una gran cantidad de paquetes en poco tiempo, por lo que se consideran del tipo inundación, aunque hay versiones “pacientes” que realizan escaneos espaciados en el tiempo. La peligrosidad de estos ataques es baja, pero es interesante detectarlos porque pueden servir como aviso de ataques más peligrosos.

De remoto a local o Remote to Local (R2L): Este grupo de ataques engloba a todos aquellos que tienen como objetivo la obtención de acceso a un sistema para el que el atacante no tiene permisos de acceso. En general, a diferencia de los anteriores, estos ataques no generan un tráfico elevado. Es habitual que estos ataques precedan a los ataques del siguiente grupo (U2R). La peligrosidad de estos ataques es alta ya que conlleva que usuarios ilegítimos accedan al sistema.

Usuario a superusuario o User to Root (U2R): El objetivo de estos ataques es el de obtener permisos de superusuario, root o administrador a partir de un usuario “normal”. El acceso al usuario normal puede ser legítimo o no. Al igual que los ataques R2L, éstos no requieren un envío masivo de paquetes. Su peligrosidad es extrema ya que si el ataque tiene éxito el intruso obtiene el control total del sistema. (Donostia, 2010)

El siguiente cuadro muestra los tipos de ataques identificados y capturados en los archivos gurekdd6percent y KDDTrain-20Percent, agrupados en las 4 categorías descriptas anteriormente, en total son 38 tipos de ataques:

Clase de Ataques	DoS	R2L	U2R	Probing
Tipos de Ataques	Teardrop	Dict	Rootkit	Nmap
	Syslog	Guess-paswd	Loadmodule	Ipsweep
	Land	Imap	Eject	Mscan
	Anomaly	Phf	Format	PortswEEP
			Format-clear	Satan
	Back	Multihop	Ffb	
	Neptune	Warezcclient	Ffb-clear	
	Pod	Warezmaster	Perl	
	Smurf	Imap	Format-fail	
		Ftp-write	Perl-clear	
		Spy	Eject-fail	
		Warez		



		Dict_simple Load_clear	Buffer- overflow	
--	--	---------------------------	---------------------	--

Cuadro 5 - Clasificación de tipos de ataques

A continuación, se describen los tipos de ataques más relevantes para ambos sets de datos:

Ataque	Descripción	Tipo
back	Ataque contra el servidor web Apache cuando un cliente pide una URL que contiene muchas barras.	DoS
land	Envío de TCP/SYN falso con la dirección de la víctima como origen y destino, causando que se responda a sí mismo continuamente.	DoS
neptune	Inundación por envíos de TCP/SYN en uno o más puertos.	DoS
pod	Ping de la muerte: manda muchos paquetes ICMP muy pesados.	DoS
smurf	El atacante envía un ping, que parece proceder de la víctima, en broadcast a una tercera parte de la red, donde todos los host responderán a la víctima.	DoS

Imagen 9 – Categoría de Ataque DOS

(Ayuso & Barcenilla Mancha , www.it.uc3m.es, 2008)



ftp_write	Usuario FTP remoto crea un archivo .rhost y obtiene un login local.	R2L
guess_passwd	Trata de adivinar la contraseña con telnet para la cuenta de visitante	R2L
imap	Desbordamiento remoto del búfer utilizando el puerto imap.	R2L
multihop	Escenario de varios días donde el atacante primero accede a una máquina que luego usa como trampolín para atacar a otras máquinas.	R2L
phf	Script CGI que permite ejecutar comandos en una máquina con un servidor web mal configurado.	R2L
spy	Analizador de protocolos LAN por la interfaz de red.	R2L
warezclient	Los usuarios descargan software ilegal publicado a través de FTP anónimo por el warezmaster.	R2L
warezmaster	Subida FTP anónima de Warez (copias ilegales de software).	R2L

Imagen 10 – Categoría de Ataque R2L

(Ayuso & Barcenilla Mancha , www.it.uc3m.es, 2008)

buffer_overflow	Desbordamiento de la pila del búfer.	UR2
loadmodule	Ataque furtivo que reinicia la IFS para un usuario normal y crea un shell de root.	UR2
perl	Establece el id de usuario como root en un script de perl y crea un shell de root.	UR2
rootkit	Escenario de varios días donde un usuario instala componentes de un rootkit.	UR2

Imagen 11 – Categoría de Ataque UR2

(Ayuso & Barcenilla Mancha , www.it.uc3m.es, 2008)



ipsweep	Sondeo con barrido de puertos o mandando pings a múltiples direcciones de host.	Probing
nmap	Escaneo de redes mediante la herramienta nmap.	Probing
portsweep	Barrido de puertos para determinar qué servicios se apoyan en un único host.	Probing
satan	Herramienta de sondeo de redes que busca debilidades conocidas.	Probing

Imagen 12 – Categoría de Ataque Probing

(Ayuso & Barcenilla Mancha , www.it.uc3m.es, 2008)

Es importante mencionar que la principal diferencia entre el archivo gurekdd6percent y KDDTrain-20Percent está en los tipos de ataques que poseen cada uno, si bien la mayoría de los tipos de ataques que poseen ambos archivos son los mismos, existen tipos de ataques que están presentes en un archivo y en el otro no, por ejemplo, el tipo de ataque “nmap”, el cual no está contemplado en el archivo “gurekdd6percent”, y es por ello que en el presente proyecto hacemos uso de los 2 archivos, para poder abarcar la mayor cantidad de tipos de ataques a la hora de detectar los ataques a realizar en la simulación de la red de datos.



Capítulo 4 – Definición de la Solución

4.1 Introducción

Este capítulo abarcará la descripción de la metodología elegida para el desarrollo de la solución propuesta y los pasos específicos a seguir para la consecución de los objetivos del proyecto.

4.2 Metodología de desarrollo de la solución

Para el desarrollo de la solución propuesta en el presente proyecto se hará uso de dos metodologías distintas, la metodología de minería de datos CRISP-DM y la metodología de administración de redes basada en modelos funcionales estándar de la ITU y de la ISO. Se propone una integración de ambas metodologías, adaptando las etapas de cada una para el desarrollo del modelo propuesto.

Etapas	Metodología CRISP-DM	Metodología de Administración de Redes
1	Comprensión del Negocio	
2		Administración de la Configuración
3	Comprensión de datos	Administración del Rendimiento
3.1		Monitoreo
3.2		Análisis
4	Preparación de datos	
5	Modelado	
6	Evaluación	
7		Administración de la Seguridad
8	Implementación	

Cuadro 6 – Etapas por metodologías

A continuación, se describen las etapas de la metodología propuesta:

4.2.1 Comprensión del negocio

En esta etapa se llevará a cabo el estudio del contexto donde se pondrá en ejecución la solución propuesta en el presente proyecto. Si bien el contexto organizacional será simulado, se realizará un análisis de las fortalezas y vulnerabilidades en materia de seguridad de la red de datos tomando como referencia la red de datos del Laboratorio de Informática de la Ucasal, tales como protocolos y políticas de seguridad existentes, procesos y clasificación de la información, reglas de negocio y otros conceptos que sean de interés para el desarrollo de la solución propuesta.

4.2.2 Administración de la configuración

En esta etapa se realizará un análisis y relevamiento de las herramientas necesarias tanto de hardware como de software para la aplicación del modelo propuesto, que abarcará las herramientas para la elaboración de la red de datos simulada y el desarrollo de la solución



propuesta, llevando a cabo las respectivas instalaciones y configuraciones necesaria para asegurar su correcto funcionamiento. Las tareas a realizar serán:

- Diseño y elaboración de una red de datos.
Se llevará a cabo la elaboración de una red de datos para la implementación del modelo propuesto, donde se simularán los ataques e intromisiones a detectar, realizando previamente un estudio y relevamiento de los requisitos necesarios de herramientas de hardware y software necesarios para desarrollar dicha red de datos.
- Instalación y administración de hardware
Se llevará a cabo la instalación de los equipos ya adquiridos en la red de datos, teniendo en cuenta la seguridad física y ambiental, como así también la configuración de los mismos para conectarse a la red.
- Instalación y administración del software
Se llevará a cabo la instalación de Sistemas Operativos, software utilitario y las herramientas necesarias para realizar el monitoreo y la captura de los datos de la red, como así también se realizará la configuración necesaria y adecuada para que el software funcione de una manera correcta y eficiente.

4.2.3 Comprensión de datos (Administración del rendimiento)

En esta etapa se realizará la captura del tráfico de datos que viaja a través de la red de datos simulada en intervalos de tiempo específicos, con la realización de diversos movimientos, operaciones y ataques para simular el tráfico de red a capturar, permitiendo de esta manera obtener la información necesaria para empezar a trabajar en el/los modelos de minería de datos para la detección de intrusos. Los pasos a llevar a cabo serán los siguientes:

Monitoreo

Se llevará a cabo la captura y almacenamiento del tráfico en la red de datos simulada para su posterior análisis, teniendo en cuenta los siguientes aspectos:

- Ancho de banda utilizado.
- Determinación de tipos de paquetes que circulan en la red, de acuerdo a los servicios de red utilizados (http, ftp, etc.).
- Porcentajes de transmisión y recepción de información.
- Utilización del servidor.

Análisis

Se llevará a cabo un análisis de la información obtenida en el monitoreo para determinar si los datos obtenidos son útiles y oportunos para el proceso posterior a desarrollar. Teniendo en cuenta los siguientes aspectos:

- Utilización de enlaces y ancho de banda.
- Tráfico normal y anormal.
- Determinación de elementos críticos de la red (más utilizados en el tráfico).
- Calidad de servicio.



4.2.4 Preparación de datos

En esta etapa se procederá a realizar una preparación de los datos de manera tal que queden refinados y dispuestos para ser utilizados en la próxima etapa, entre las actividades que se realizarán están:

- Eliminar datos redundantes y datos basura.
- Establecer estructura y tipo de datos.
- Crear dominios de valores de atributos.
- Completar datos faltantes.

Se hará uso del formato de las bases de datos GureKDD y NSL KDD, estableciendo los mismos atributos y tipos de datos.

4.2.5 Modelado

Para la creación del/los modelos de minería de datos, primeramente se realizará un análisis de diversos algoritmos de clasificación de minería de datos para determinar cuál/es son más eficientes a la hora clasificar; se hará uso de la técnica de validación cruzada para determinar la precisión de clasificación de cada algoritmo, con el uso de los datos ya clasificados (GureKDD data base 2008y NSL KDD data base 2012), y se seleccionará el/los algoritmos más destacados en su desempeño (precisión). Seguidamente, del resultado del análisis de los algoritmos, se obtendrá el/los modelos de minería de datos (modelos de clasificación), los cuales serán guardados en distintas memorias (stores) y serán usados para clasificar los datos capturados en la red de datos simulada.

4.2.6 Evaluación

En la etapa de evaluación se llevará a cabo la aplicación del/los modelos de minería de datos obtenidos sobre los datos no clasificados, es decir, los datos capturados en la simulación de la red de datos. Seguidamente se realizará un análisis de los resultados obtenidos en la clasificación, las actividades a realizar serán:

- Descubrimiento del conocimiento
Aplicación del/los modelos de clasificación sobre los datos capturados en la simulación de la red de datos, con el fin de descubrir los patrones de comportamiento en los mismos, donde no solamente se podrá conocer los ataques e intromisiones, sino también patrones del tráfico “normal” reflejados en movimiento y operaciones internos y hacia el exterior de la red de datos.
- Determinación de patrones “anormales”
Con el descubrimiento de patrones en los datos clasificados se determinarán y establecerán aquellos patrones que sirvan para la detección de intrusos, determinando aquellos parámetros y atributos significativos, y sus valores, para la detección de intromisiones y ataques.
- Establecimiento de reglas
Se llevará a cabo la creación de reglas de detección de intrusos de acuerdo a la herramienta IDS a implementar (es decir, tomando en cuenta el formato de regla



requerido por la herramienta), haciendo uso de los valores de los parámetros y atributos significativos en los patrones “anormales”.

4.2.7 Administración de la Seguridad

En esta etapa se procederá a la instalación y configuración del IDS para la puesta en marcha del modelo de detección de intrusos. Los pasos a realizar serán:

- Instalación del software IDS
Se trabajará detalladamente en todos los aspectos referidos a la instalación del software IDS, tales como la herramienta software, los requerimientos de instalación y el proceso de instalación.
- Incorporación de las reglas de detección de intrusos
Una vez instalada la herramienta de software IDS en el equipo correspondiente, se procederá a la configuración de la herramienta (parámetros y variables) e incorporación de las reglas establecidas en la etapa anterior al motor de base de datos del IDS.
- Configuración del IDS en la red
Se adaptará el modelo IDS con las configuraciones necesarias de hardware y software tanto del modelo, como de los equipos y herramientas software del lugar de trabajo (red de datos y organización simulada).

4.2.8 Implementación

En esta etapa se pondrá en funcionamiento el modelo IDS en la red de datos simulada, llevando a cabo nuevamente diversos ataques e intromisiones para ponerlo a prueba. Posteriormente se llevará a cabo el proceso de presentación y análisis de resultados mediante informes, gráficos y tablas de comparación obtenidos, con el fin de apoyar a la toma de decisiones.



Capítulo 5 – Solución Propuesta

5.1 Introducción

El presente capítulo hace referencia a la aplicación del modelo propuesto en el lugar de trabajo definido con todas las herramientas de software y hardware definidas para el funcionamiento eficiente del mismo.

5.2 Herramientas utilizadas para la Solución

Para el desarrollo de la solución propuesta se seleccionaron las siguientes herramientas, en base a su disponibilidad y facilidad de uso:

5.2.1 Herramientas utilizadas para el monitoreo de red

Entre las herramientas disponibles para el monitoreo descriptas en el Capítulo 3 – Marco Teórico, seleccionamos las siguientes:

- Wireshark: *Wireshark es un analizador de protocolos de red para Unix y Windows utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica.* (Wireshark, 2016)
- TCPDUMP: *un analizador de paquetes de gran alcance de línea de comandos; y libpcap, un portátil de C / C ++ biblioteca para la captura de tráfico de red.* (Tcpdump, 2016)

5.2.2 Herramientas utilizadas para la minería de datos

Entre las herramientas disponibles para la minería de datos descriptas en el Capítulo 3 – Marco Teórico, seleccionamos las siguientes:

- Rapid Miner: *Rapid Miner es una herramienta de Inteligencia de Negocios que proporciona un entorno integrado de aprendizaje automático para la Minería de Datos, se utiliza para aplicaciones de negocio, investigación, educación, formación, creación de modelos y desarrollo de aplicaciones, es compatible con todos los pasos de minería de datos.* (Tecnología, Ingeniería en sistemas, 2015)
- Hoja de calculo: *Se denomina hoja de cálculo a una herramienta informática que sirve como medio para realizar cálculos, operaciones lógicas y manejo de datos. Al usuario se le presenta como un conjunto de columnas y filas identificables por letras y números respectivamente que van formando celdas. Su uso se orienta especialmente a actividades que requieren un manejo de muchos cálculos en paralelo.* (definicion.mx, 2017)

5.2.3 Herramientas utilizadas para la detección de intrusos

Entre las herramientas disponibles para la detección de intrusos descriptas en el Capítulo 3 – Marco Teórico, seleccionamos las siguientes:



- *Suricata: Suricata es un IDS de red de alto rendimiento, IPS y motor de supervisión de la seguridad de la red. Es Open Source y propiedad de una fundación sin fines de lucro organizado por la comunidad, la Fundación Seguridad de la Información abierta (OISF). Suricata es desarrollado por el OISF y sus proveedores de apoyo. (Suricata IDS, 2016)*
- *Bro: Bro es un sistema de detección de intrusiones para UNIX/Linux Open Source que analiza el tráfico de red en busca de actividad sospechosa. su característica principal es que sus reglas de detección están basadas en scribd en lenguaje nativo que suponen políticas (policies) que son las encargadas de detectar, generar logs o eventos y acciones a nivel de sistema operativo. (Seguridad y Redes, 2011)*
- *Snorby Dash Board: Snorby es un front-end para la gestión de alertas de detaccion de intrusos basado en sensores. Poseé una interfaz gráfica muy sencilla con una visión amplia e intuitiva para la visualización de alertas. (Seguridad y Redes, 2010)*

5.3 Definición de la Solución

Se comenzará con la simulación de una red de datos LAN en el entorno VMWare Workstation 10 haciendo uso de los equipos del Laboratorio de Informática de la Facultad de Ingeniería de la UCASAL (se utilizará un entorno virtual para no afectar las configuraciones de la red real del laboratorio) y equipos externos al laboratorio (2 notebooks para realizar el monitoreo de red, los ataques y el proceso de minería de datos) con las características suficientes para la puesta en marcha del modelo propuesto, así también se hará uso del servicio de internet de la Facultad de Ingeniería de la Ucasal. La red contará con un equipo Switch virtual propio de Vmware que interconecta las siguientes máquinas virtuales a través de los propios adaptadores de red virtuales de cada una configurados en modo NAT:

- Máquina Virtual 1: Sistema Operativo Kali Linux 2.0 (Atacante)
- Máquina Virtual 2: Sistema Operativo CentOS 6.7 (IDS y Servidor)
- Máquina Virtual 3: Sistema Operativo Windows 7 (Cliente)
- Máquina Virtual 4: Sistema Operativo Windows XP (Cliente)

Esta funcionalidad de red en una infraestructura virtual permite administrar la comunicación entre dichos equipos y proporciona la conexión a internet requerida; donde se simulará los ataques o intromisiones a red de datos mediante el sistema atacante Kali Linux.

Posteriormente se hará uso de los conceptos y técnicas en seguridad de redes de datos para realizar un monitoreo y captura de flujo de información simulada en la red de datos, descubriendo anomalías y situaciones inusuales en el tráfico de datos.

Los cuáles serán adaptados en la red de datos mediante herramientas software destinado al monitoreo y captura:

- TCPDUMP (bajo plataforma Linux)
- WIRESHARK (bajo plataforma Windows y Linux)



El siguiente grafico muestra el esquema básico para poder implementar el modelo propuesto.

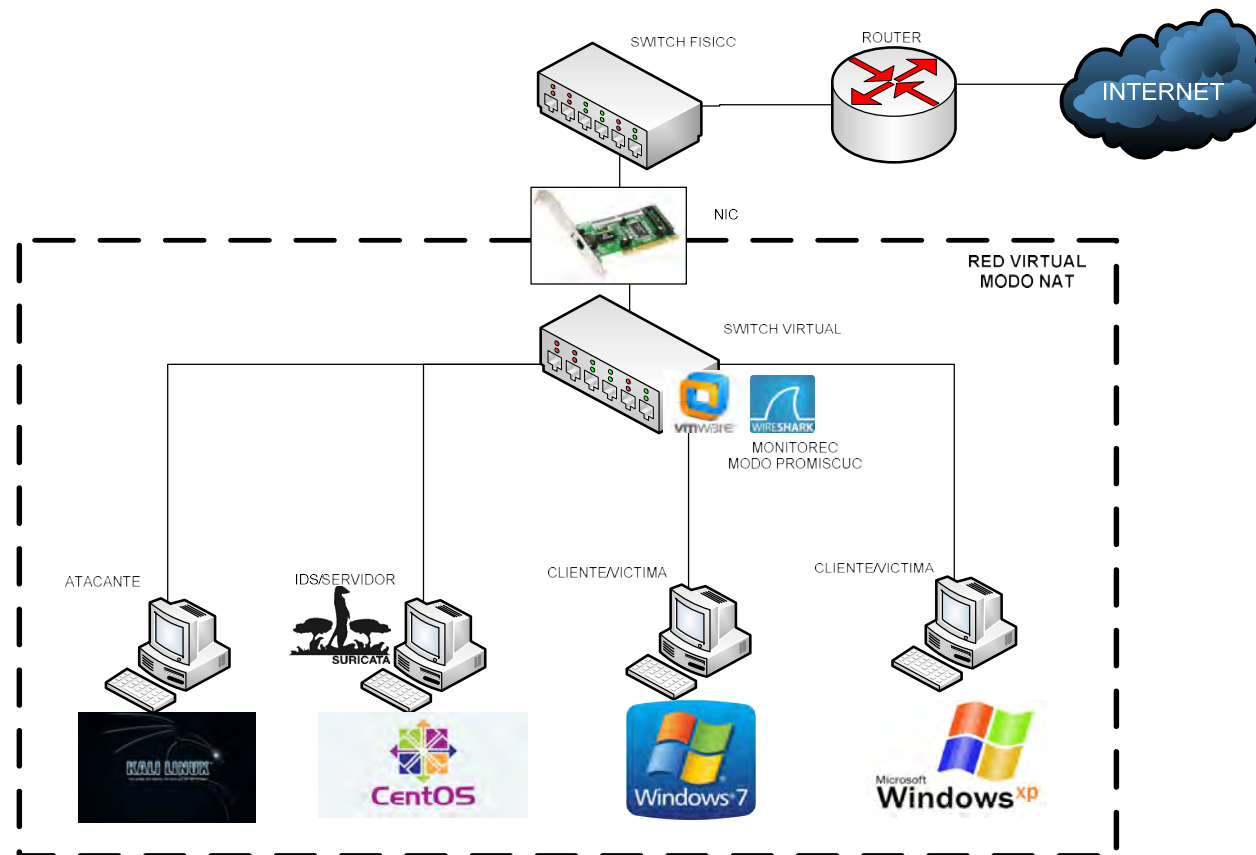


Imagen 13 – Esquema básico del modelo propuesto



Paso a seguir, toda la información obtenida en el monitoreo de la red se almacenará en nuestro servidor y seguidamente se realizará la preparación y refinamiento de todos los datos almacenados conjuntamente con los Sets de datos GureKDD Cup database 2008 (gurekdd6percent) y NSL KDD 2012 (KDDTrain-20Percent), llevando a cabo: limpieza de datos basura, normalización de datos y creación de nuevos campos y estructuras de datos, mediante la herramienta Rapid Miner Studio 5 y Microsoft Excel 2010.

Posteriormente, se hará uso de los algoritmos de clasificación de minería de datos incorporados en Rapid Miner, para la creación del/los modelos de minería de datos con los cuales se clasificarán los datos capturados en la simulación de la red y se obtendrán las reglas de detección de intrusos que serán integradas al Sistema de Detección de Intrusos.

Seguidamente se pasará a la instalación y configuración del modelo IDS, Suricata, en nuestro servidor adaptado a la red de datos, con la incorporación de las reglas creadas anteriormente al motor de base de datos del IDS.

Finalmente se pondrá en ejecución el IDS en la red de datos, según lo estipulado en la metodología de la solución propuesta y se pasará al análisis y evaluación de los resultados obtenidos, juntamente con la elaboración de informes y graficas estadísticas que demuestren la eficiencia del modelo.

A continuación, se mencionan pasos prácticos que resumen lo anteriormente descripto:

1. Elaboración de la red de datos mediante el entorno de virtualización Vmware Workstation 10.
2. Instalación y configuración de las herramientas de monitoreo.
3. Simulación de Ataques mediante Kali Linux 2.0.
4. Captura y almacenamiento de tráfico de datos. (TCPDUMP y WireShark)
5. Preparación y refinamiento de los datos. (Rapid Miner)
6. Creación del/los modelos de minería de datos a partir de las bases de datos GUREKDD 2008 y NSL KDD 2012 (conjuntos de datos clasificados) (Rapid Miner)
7. Clasificación de datos capturados aplicando el/los modelos de clasificación obtenidos. (Rapid Miner)
8. Generación de las reglas para la detección de intrusos.
9. Instalación y configuración del IDS Suricata.
10. Integración de reglas al motor de base de datos del IDS Suricata

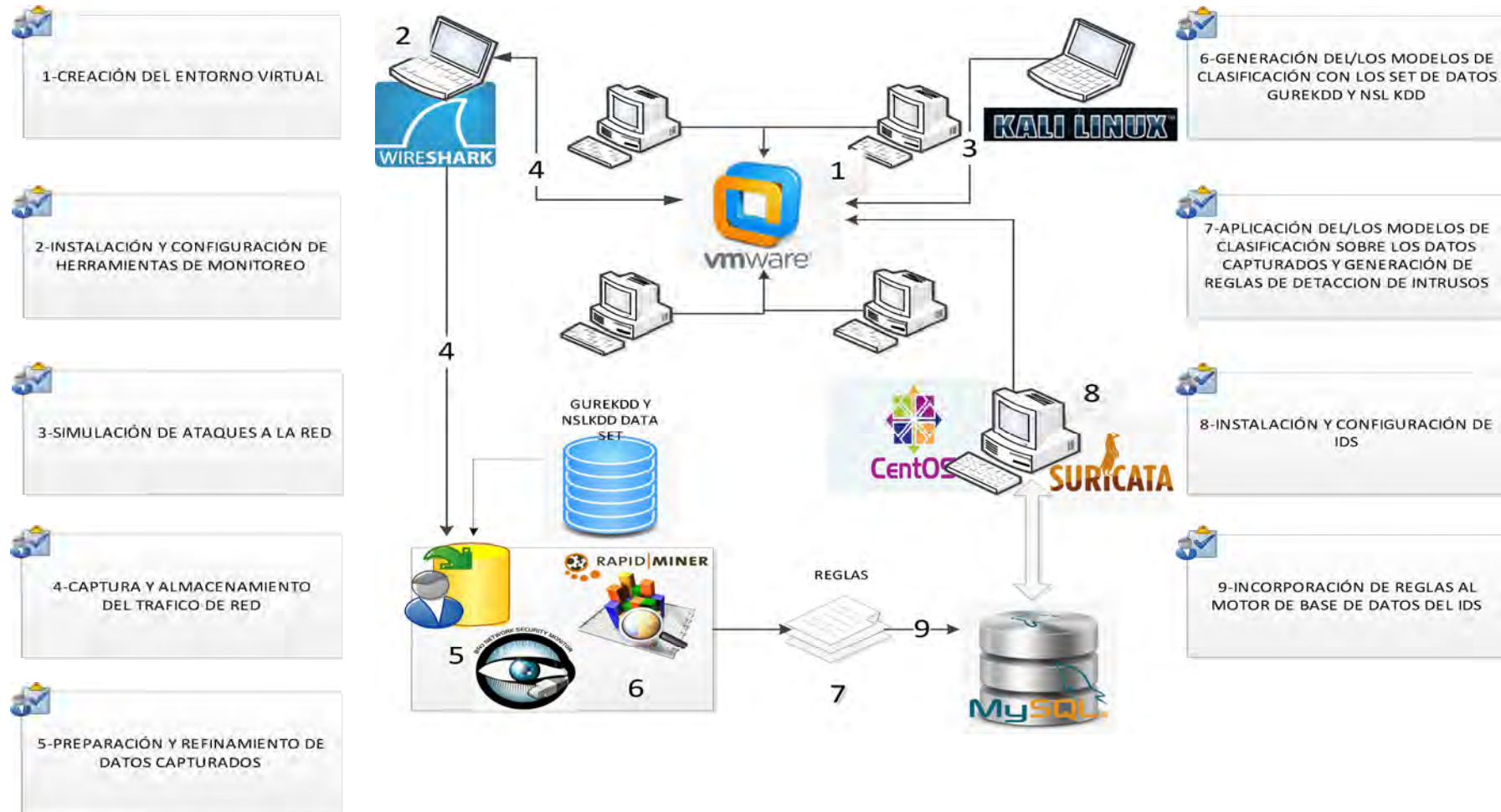


Imagen 14 – Proceso de desarrollo de la solución de propuesta



5.4 Estructura de desglose de trabajo (WBS)

Para iniciar con el desarrollo del modelo propuesto se establecieron las siguientes tareas:

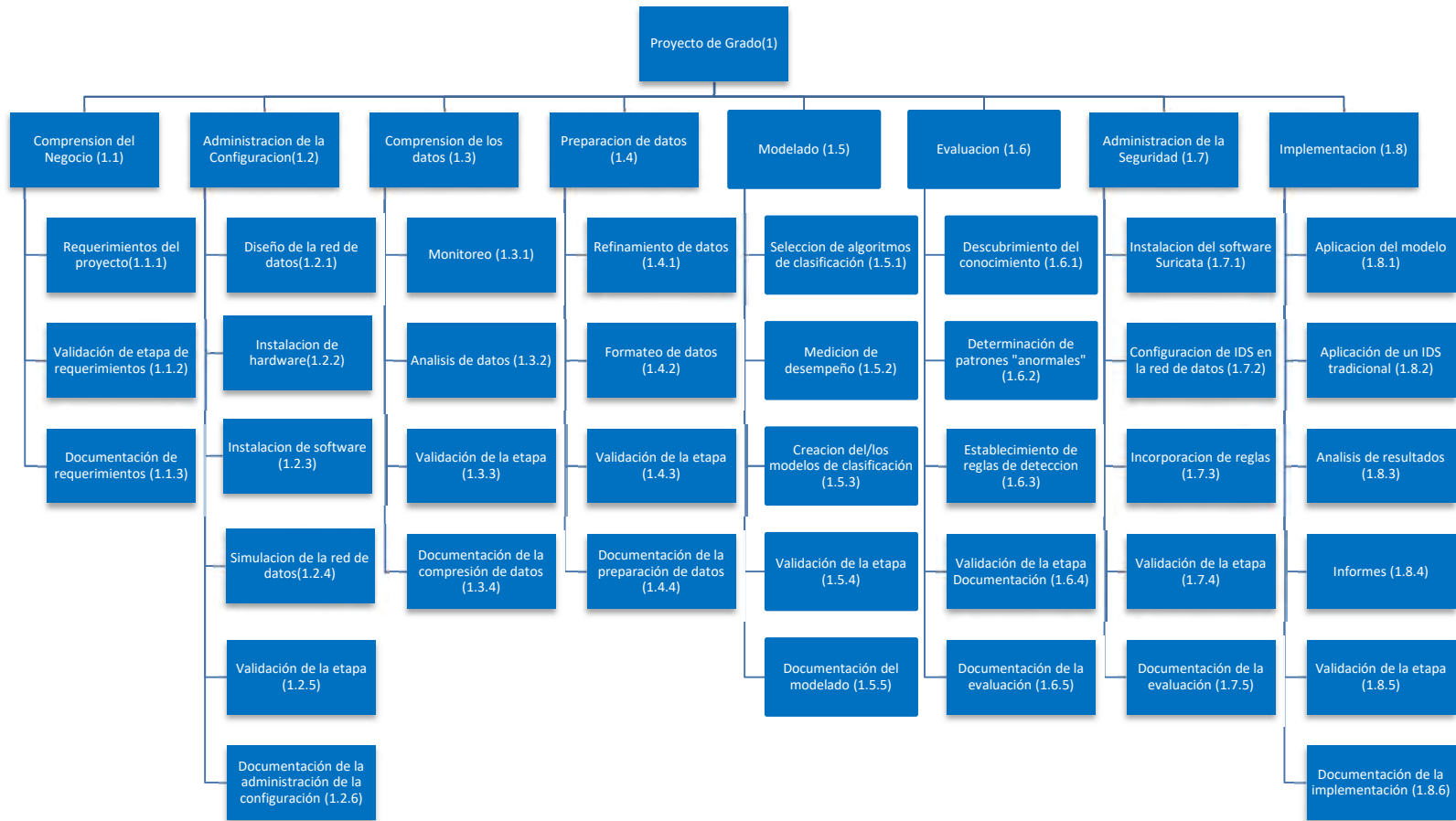


Imagen 15 – Estructura de desglose de trabajo



Código	Función	Descripción
(1)	Proyecto de Grado	Modelo de Detección de Intrusos basado en las Técnicas de Data Mining en una red de datos.
(1.1)	Comprensión del Negocio	
(1.1.1)	Requerimientos del Proyecto	Estudio de las características técnicas de seguridad y operativas de la red de datos del Laboratorio de Informática de la UCASAL necesarias para la simulación de la infraestructura de la red del presente proyecto.
(1.1.2)	Validación de etapa de requerimientos	Verificación del cumplimiento de los requerimientos del proyecto para llevar a cabo una correcta implementación del modelo.
(1.1.3)	Documentación de requerimientos	Documentación de las tareas realizadas durante la etapa en cuestión y requerimientos necesarios para el proyecto.
(1.2)	Administración de la Configuración	
(1.2.1)	Diseño de la red de datos	Elaboración del diseño de la infraestructura de red de datos a simular necesaria para el desarrollo del modelo propuesto.
(1.2.2)	Instalación de hardware	Creación de las máquinas virtuales necesarias para el funcionamiento de la red de datos simulada en el entorno VMWare Workstation 10.
(1.2.3)	Instalación de Software	Instalación y configuración del software de las máquinas virtuales (S.O, programas utilitarios, etc.) para su correcto funcionamiento en la red de datos simulada.
(1.2.4)	Simulación de la red de datos	Simulación de la red de datos mediante el entorno VMWare Workstation 10, realizando pruebas de funcionamiento y ataques informáticos a la red.
(1.2.5)	Validación de la etapa	Verificación del correcto funcionamiento de la red de datos simulada en el entorno virtual VMWare Workstation 10.
(1.2.6)	Documentación de la administración de la configuración	Documentación de las tareas realizadas durante la etapa en cuestión.



(1.3)	Comprensión de los datos – Administración del Rendimiento	
(1.3.1)	Monitoreo	Captura y almacenamiento del tráfico de la red de datos simulada en VMWare mediante la herramienta de monitoreo de redes Wireshark 1.12.6.
(1.3.2)	Análisis de datos	Análisis de la información obtenida en el monitoreo de la red simulada, comprobando la veracidad de los datos obtenidos respecto de las acciones realizadas (ataques y flujo de datos).
(1.3.3)	Validación de la etapa	Verificación de la calidad del conjunto de datos obtenidos.
(1.3.4)	Documentación de la comprensión de datos	Documentación de las tareas realizadas durante la etapa en cuestión.
(1.4)	Preparación de los datos	
(1.4.1)	Refinamiento de datos	Preparación y refinamiento del conjunto de datos obtenidos de la simulación, eliminando datos basura y redundantes.
(1.4.2)	Formateo de datos	Establecer el mismo formato para los datos capturados y las bases de datos GURE kdd 2008 y NSL kdd 2012, estableciendo tipos de datos y dominio de valores.
(1.4.3)	Validación de la etapa	Verificación del correcto proceso de normalización de los conjuntos de datos (datos capturados, GURE kdd 2008 y NSL kdd 2012).
(1.4.4)	Documentación de la preparación de datos	Documentación de las tareas realizadas durante la etapa en cuestión.
(1.5)	Modelado	
(1.5.1)	Selección de algoritmos de clasificación	Selección de algoritmos para la creación del/los modelos de clasificación.
(1.5.2)	Medición de desempeño.	Medición del nivel de precisión de clasificación de los algoritmos en los datos de entrenamiento (GURE kdd 2008 y NSL kdd 2012).
(1.5.3)	Creación del/los modelos de clasificación	Creación del/los modelos de clasificación en base a los algoritmos con mayor precisión obtenidos en la etapa anterior, mediante la herramienta Rapid Miner 5.



(1.5.3)	Validación de la etapa	Verificación de la correcta creación y funcionamiento del/los modelos de clasificación de minería de datos obtenidos.
(1.5.4)	Documentación del modelado	Documentación de las tareas realizadas durante la etapa en cuestión.
(1.6)	Evaluación	
(1.6.1)	Descubrimiento del conocimiento	Aplicación del/los modelos de clasificación sobre los datos capturados en la simulación de la red para descubrir patrones de comportamiento.
(1.6.2)	Determinación de patrones “anormales”	Determinación y establecimiento de patrones y características de atributos dentro del conjunto de datos capturado, que permitan detectar intromisiones y/o ataques informáticos.
(1.6.3)	Establecimiento de reglas de detección.	Creación de reglas de detección de intrusos a partir de la información obtenida en la tarea anterior, a ser incorporadas en el motor de BD del IDS.
(1.6.4)	Validación de la etapa Documentación	Verificación de la correcta creación de reglas de detección de intrusos con su correspondiente sintaxis y parámetros.
(1.6.5)	Documentación de la evaluación	Documentación de las tareas realizadas durante la etapa en cuestión.
(1.7)	Administración de la Seguridad	
(1.7.1)	Instalación del software Suricata	Instalación y configuración del IDS “SURICATA” en nuestro servidor Centos 6.7 (máquina virtual).
(1.7.2)	Configuración del IDS en la red de datos	Configuración de los parámetros de red del Servidor IDS necesarios para el correcto funcionamiento del mismo en la red de datos simulada.
(1.7.3)	Incorporación de reglas	Incorporación de las reglas establecidas en la etapa de evaluación al motor de base de datos del Servidor IDS.
(1.7.4)	Validación de la etapa	Verificación del correcto funcionamiento del Servidor IDS en la red datos simulada.



(1.7.5)	Documentación de la administración de la seguridad	Documentación de las tareas realizadas durante la etapa en cuestión.
(1.8)	Implementación	
(1.8.1)	Aplicación del modelo	Simulación de los ataques e intromisiones a la red de datos con el modelo de detección de intrusos basado en técnicas de data minig (Servidor IDS) en ejecución.
(1.8.2)	Aplicación de un IDS tradicional	Simulación de los ataques e intromisiones a la red de datos con un IDS tradicional (sin minería de datos, con reglas estándares) en ejecución.
(1.8.3)	Análisis de resultados	Estudio y evaluación de los resultados obtenidos tras la puesta en marcha del modelo de IDS propuesto en comparación a un IDS tradicional.
(1.8.4)	Informes	Informes, gráficos y tablas del rendimiento del modelo de IDS propuesto para concluir el trabajo realizado.
(1.8.5)	Validación de la etapa	Verificación de los resultados obtenidos y del correcto funcionamiento de las reglas de detección de intrusos del modelo de IDS propuesto.
(1.8.6)	Documentación de la implementación	Documentación de las tareas realizadas durante la etapa en cuestión.

Cuadro 7 – Estructura de desglose de trabajo



5.5 Planificación del Modelo propuesto (Diagrama de Gantt)

El desarrollo del modelo propuesto tiene como fecha de inicio el 2 de Noviembre de 2015 y como fecha de finalización el 16 de Febrero de 2016. El trabajo realizado en el tiempo establecido se puede ver en el siguiente diagrama:

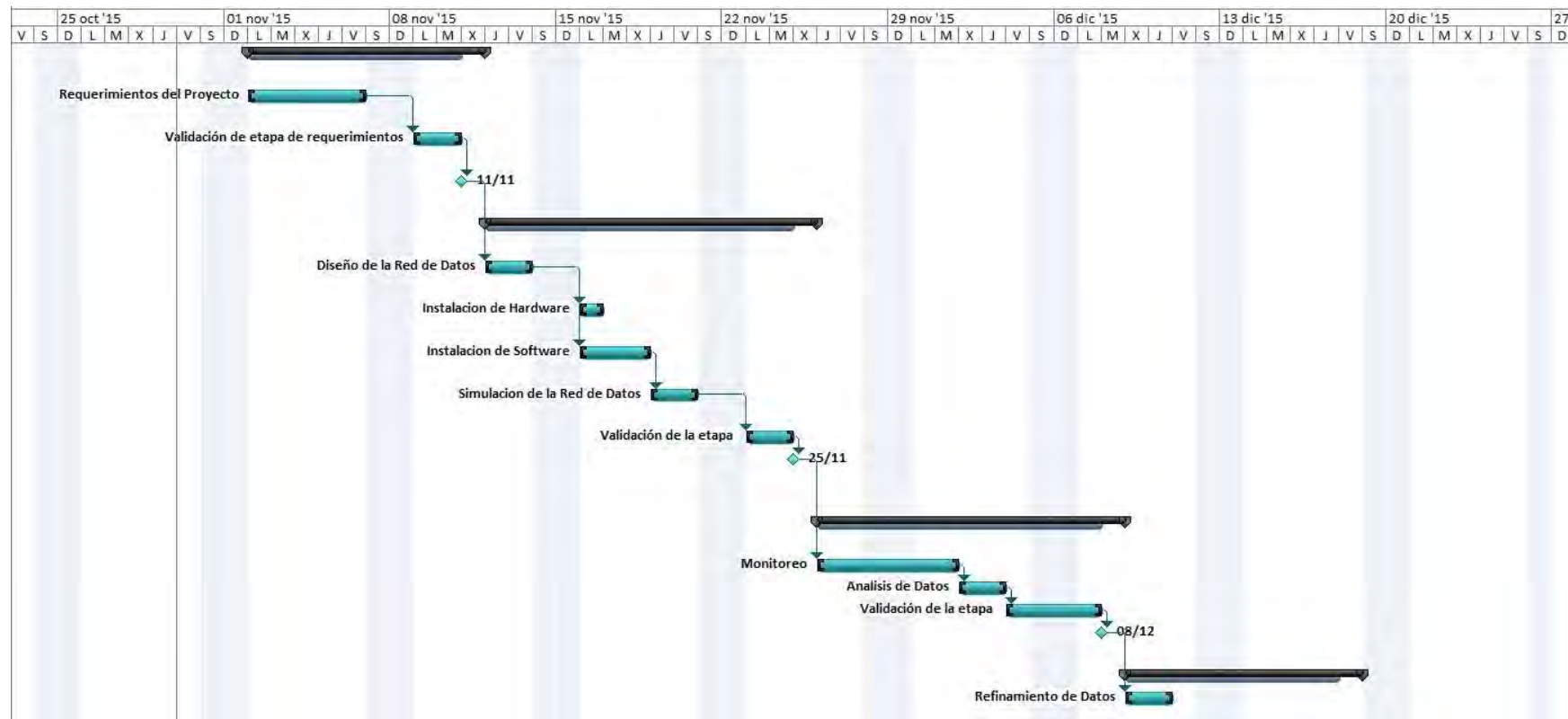


Imagen 16 – Planificación – Parte 1

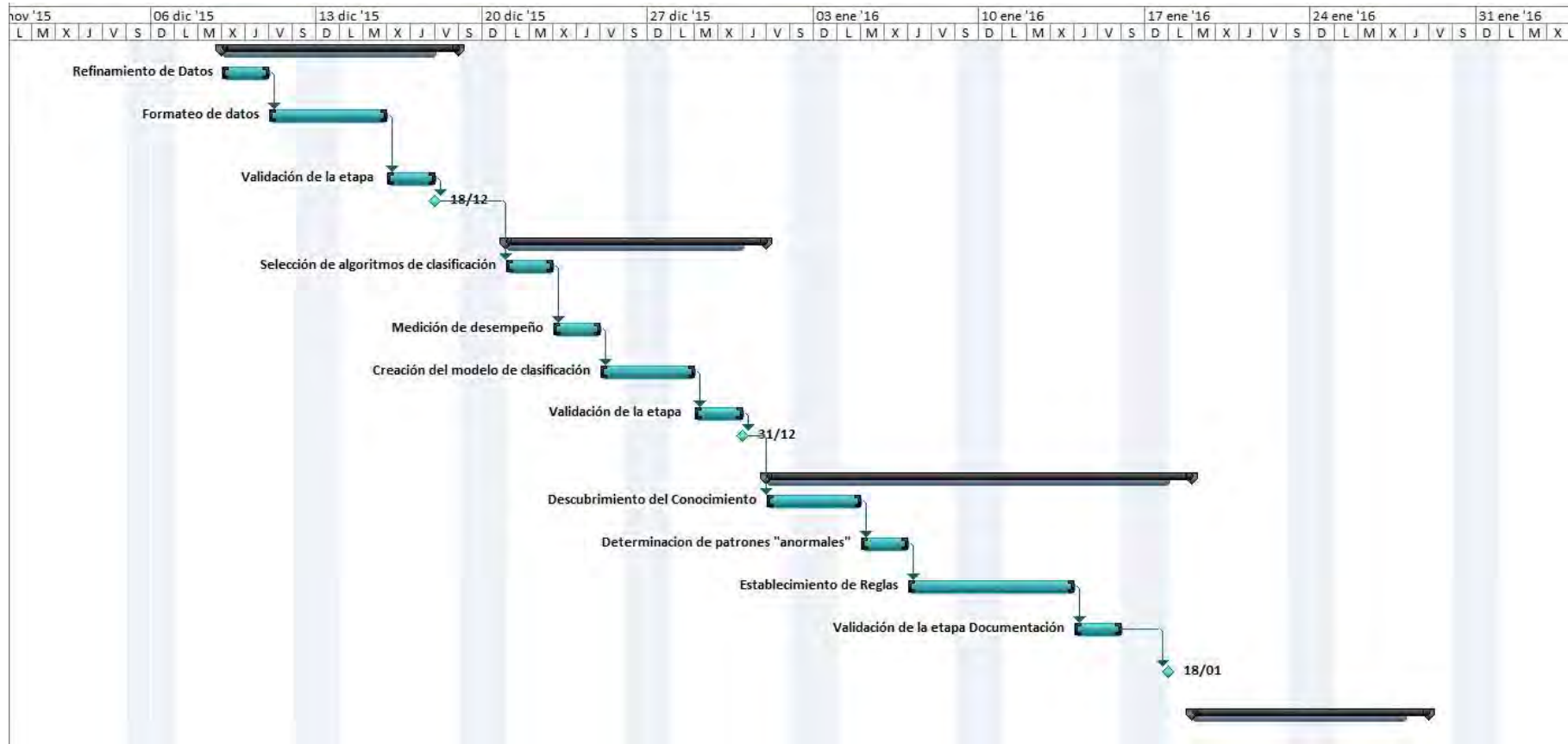


Imagen 17 – Planificación – Parte 2

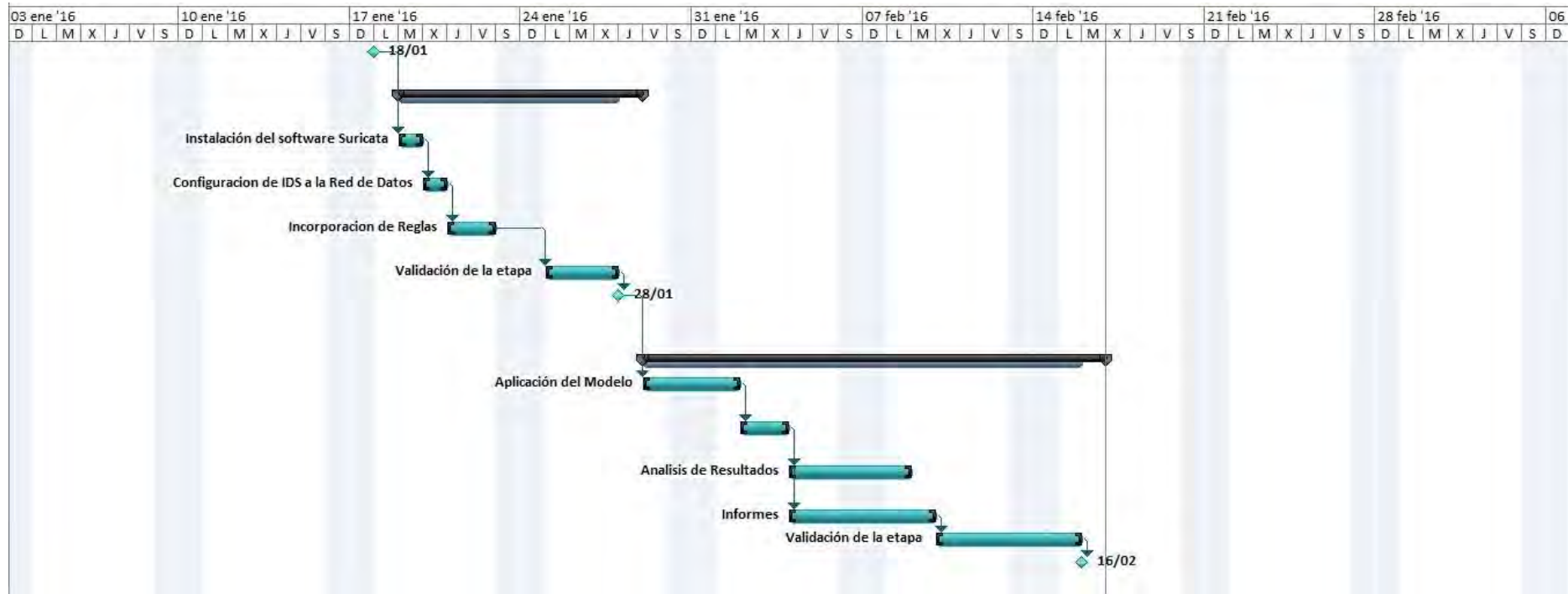


Imagen 18 – Planificación – Parte 3



5.6 Compresión del negocio

5.6.1 Requerimientos del proyecto

Para comenzar con la metodología de desarrollo se realizó una compresión del negocio donde se pondrá en ejecución la solución propuesta, para ello se simuló un entorno organizacional basado en el Laboratorio de Informática de la Ucasal y la red de datos del mismo, en donde se encuentran las siguientes características: existencia de una páginas webs informativas y transaccionales (página web de la universidad y plataforma Moodle), uso de cuentas personales en sistemas webs (SAG), envío y recepción de e-mails y acceso a redes sociales, como así también el uso de herramientas ofimáticas (procesadores de texto, planillas de cálculo, etc.) , servidor (físico) de archivos y carpetas compartidas.

Las políticas de seguridad de la información en materia de tecnología son las siguientes:

- Firewall propio del sistema operativo, en cada PC y configurado de una manera predeterminada.
- Control de acceso de usuarios establecida por el sistema operativo de cada PC.
- Todos los equipos conectados a una red LAN con una asignación de dirección IP fija a través de un equipo switch.
- La información de la red LAN administrada por un equipo router con servidor DHCP para la conexión de equipos inalámbricos y salida a Internet.
- Acceso a servidores y a carpetas compartidas de archivos por contraseñas.
- Redes WLAN (wireless) protegidas con contraseñas.

Las políticas de seguridad de la información de acuerdo a las reglas de negocio de la organización simulada son las siguientes:

- Backup de la información del servidor 1 vez por semana en dispositivos de almacenamientos externos.
- Compartimiento de carpetas de archivos de los servidores a grupo de usuarios.
- Carpetas de archivos del servidor protegidas por contraseñas.
- Distribución de contraseñas a usuarios de acuerdo a la información que puede ver cada uno.

5.7 Administración de la configuración

5.7.1 Instalación y Configuración del entorno virtual

Para desarrollar el ambiente virtual en el cual se desarrolla el presente proyecto se utilizó la herramienta software de virtualización VmWare Workstation 10.0.3, tanto para recrear el entorno de red virtual, como así también las máquinas virtuales nombradas anteriormente. Las características de cada una de estas son las siguientes:



Máquina Virtual 1: Kali Linux 2.0 (Sistema Atacante)

- Memoria: 1 GB
- Procesadores: 1
- Disco duro: 15 GB
- Adaptador de red: NAT
- Controladora USB: Presente

Máquina Virtual 2: CentOS 6.7 (Servidor/IDS)

- Memoria: 1 GB
- Procesadores: 1
- Disco duro: 20 GB
- Adaptador de red: NAT
- Controladora USB: Presente

Máquina Virtual 3: Windows 7 Professional (Cliente/Sistema Víctima)

- Memoria: 1 GB
- Procesadores: 1
- Disco duro: 15 GB
- Adaptador de red: NAT
- Controladora USB: Presente

Máquina Virtual 4: Windows XP Professional (Cliente/Sistema Víctima)

- Memoria: 512 MB
- Procesadores: 1
- Disco duro: 10 GB
- Adaptador de red: NAT
- Controladora USB: Presente

Nótese que cada una de las máquinas virtuales tiene configurado su adaptador de red en modo NAT. Este modo de configurar la placa de red de las máquinas virtuales permite transportar el entorno virtual de un sistema físico a otro sin sufrir cambios en la configuración de red del entorno virtual, lo que resulta de mucha utilidad en el desarrollo de nuestro proyecto. El modo NAT consiste en obtener una salida a internet emulando un DHCP (Dynamic Host Configuration Protocol) y utilizando la misma IP del equipo real. A diferencia del modo Bridge el cual permite tener a cada máquina virtual una IP real, perteneciente al mismo rango de IPs del Switch físico al cual se encuentra conectado el host real, es decir es como disponer de otra máquina real en la red.

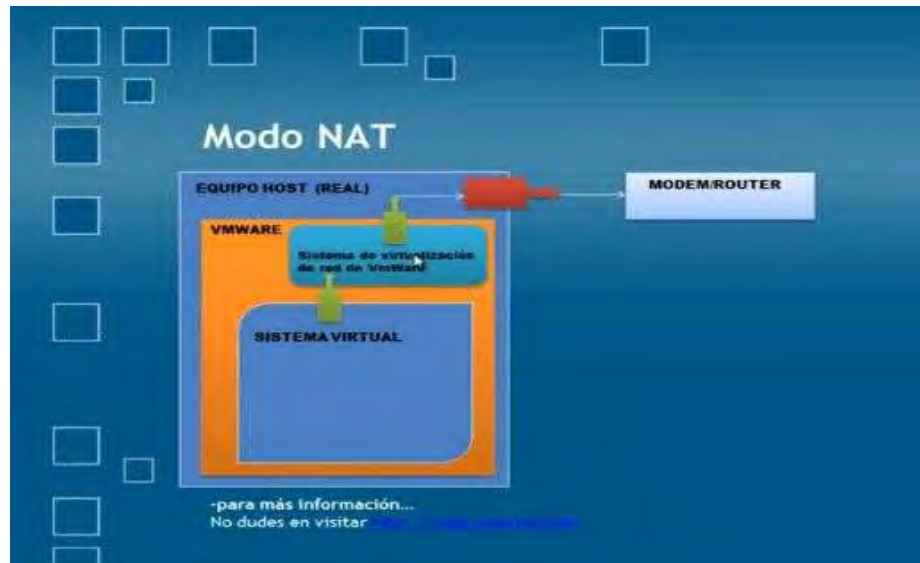


Imagen 19 – Configuración de red modo NAT

(Soporte TI - Canal YouTube, 2011)

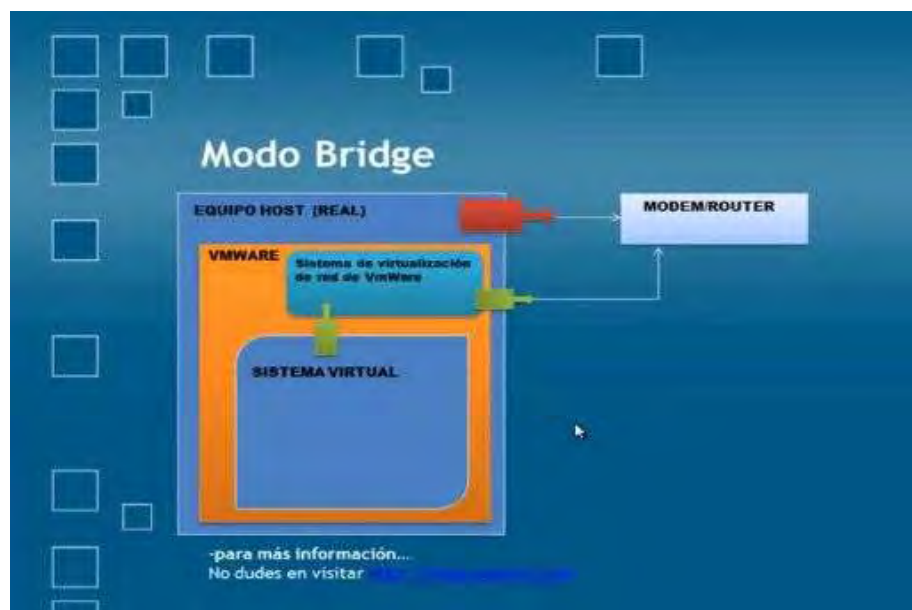


Imagen 20 – Diferencia entre Modo NAT y Bridge

(Soporte TI - Canal YouTube, 2011)

Una vez instalado y configurado el entorno virtual del proyecto, se procede a instalar los programas necesarios para poder llevar a cabo cada una de las diferentes etapas de la solución, tales como:

- Sistemas Operativos de las Máquinas Virtuales
- Software de monitoreo de red:
 - Wireshark 1.12.6



- TCPdump 4.6.2
- Bro Network Security Monitor 2.4.1

Cabe mencionar que las dos primeras herramientas (Wireshark y TCPdump) ya se encuentran disponibles tras la instalación del S.O Kali Linux 2.0 cuyo nuevo kernel 4.0 está basado en una distribución Debían “Jessie”. Mientras que BRO NSM se lo instaló manualmente en la máquina virtual de CentOS 6.7.

5.8 Comprensión de los datos

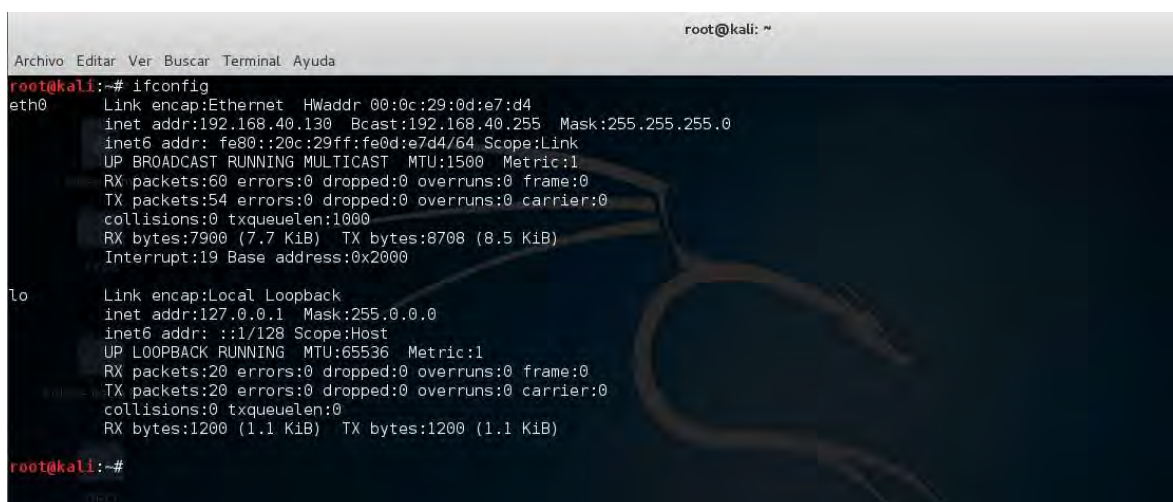
5.8.1 Monitoreo de la red y Captura de datos

En esta etapa se valió de las herramientas de monitoreo y captura instaladas anteriormente para las cuales se debe configurar la interfaz de red de la máquina virtual que vaya a realizar el monitoreo (Kali Linux 2.0) en modo promiscuo, este modo permite a la maquina en cuestión capturar no solo su propio tráfico de red sino también el de todas las maquinas conectadas a la misma red ya sea alámbrica o inalámbricas.

El modo promiscuo es un modo de operación en el que una computadora conectada a una red compartida captura todos los paquetes, incluyendo los paquetes destinados a otras computadoras. Es muy útil para supervisar la red, pero presenta un riesgo de seguridad dentro de una red de producción. (ConocimientosWeb, 2013)

Para configurar el modo promiscuo en una máquina primero hay que comprobar si la tarjeta de red con que se dispone soporta este modo, partiendo de eso para configurar nuestra máquina virtual de Kali Linux en modo promiscuo, se procede de la siguiente manera:

Abrimos una terminal en Kali Linux y ejecutamos el comando: **ifconfig**, para así poder obtener un listado de nuestras interfaces de red disponibles.



```
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:0d:e7:d4
          inet addr:192.168.40.130  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe0d:e7d4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7900 (7.7 KiB)  TX bytes:8708 (8.5 KiB)
          Interrupt:19 Base address:0x2000

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
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)

root@kali:~#
```

Imagen 21 – Listado de interface de red Kali Linux



Podemos ver que se dispone de la interfaz **eth0** y **lo**. La que nos interesa a nosotros es la primera por lo que debemos configurar o establecerla en modo promiscuo para que esté preparada para empezar el monitoreo de red, para ello en la terminal tipeamos el siguiente comando: **ifconfig eth0 promisc**

```
root@kali:~# ifconfig eth0 promisc
root@kali:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:0d:e7:d4
          inet addr:192.168.40.130  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe0d:e7d4/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:123 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13696 (13.3 KiB)  TX bytes:8708 (8.5 KiB)
          Interrupt:19 Base address:0x2000

root@kali:~#
```

Imagen 22 – Interface de red eth0 en modo promiscuo

Con esto especificamos que la interfaz eth0 se establezca en modo promiscuo, después comprobamos que dicha interfaz esta efectivamente configurada en ese modo con el comando **ifconfig eth0**.

Posteriormente se configura también la herramienta de monitoreo Wireshark para que capture los datos en modo promiscuo. Para ello iniciamos Wireshark desde Kali Linux 2.0, nos encontraremos con la siguiente ventana:

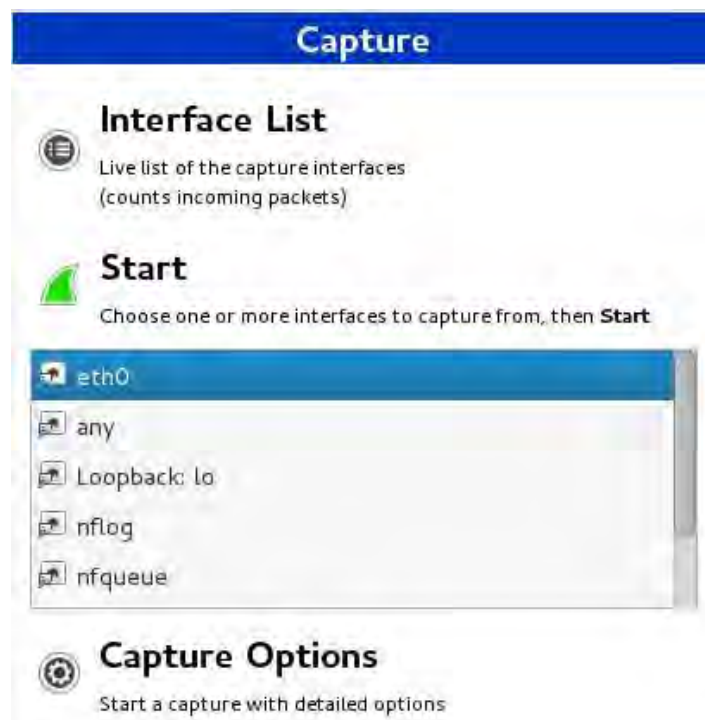


Imagen 23 – Interface de Wireshark



Donde seleccionaremos eth0 y nos dirigiremos a las opciones de captura

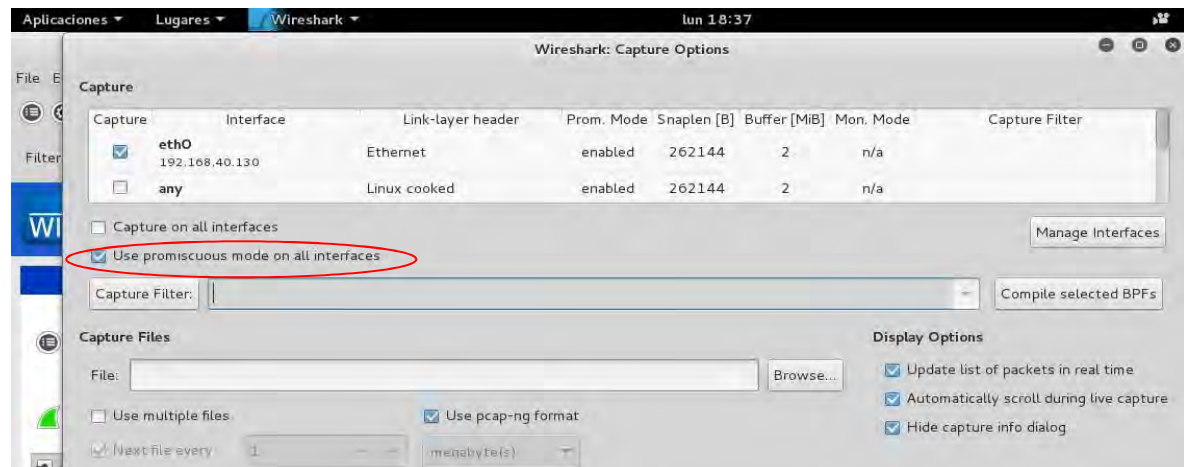


Imagen 24 – Opciones de captura de Wireshark

En donde tildaremos la opción “**use promiscuous mode on all interfaces**” de esta manera nos aseguramos que el monitoreo de la red virtual se lleve a cabo de una manera correcta.

En lo que respecta a TCPdump no es necesario configurarlo en modo promiscuo ya que el mismo viene configurado en este modo por defecto.



Imagen 25 - TCPdump

Una vez configurada la interfaz de red se procede al proceso de monitoreo y captura de los datos que se transmiten a través de la red virtual. Para ello hacemos uso de la herramienta Wireshark mencionada anteriormente. Para empezar con el monitoreo encendemos todas las máquinas virtuales para simular el tráfico de datos entre ellas, una vez ingresado al programa seleccionamos la interfaz **eth0** y pulsamos en **start**:



Imagen 26 – Inicio de Wireshark

Al iniciar con este proceso podemos ver el tráfico de la red, es decir los paquetes que se transmiten o se reciben desde cada máquina virtual, ya sea entre estas, o a través de las páginas de internet a las que ingresa cada una respectivamente.

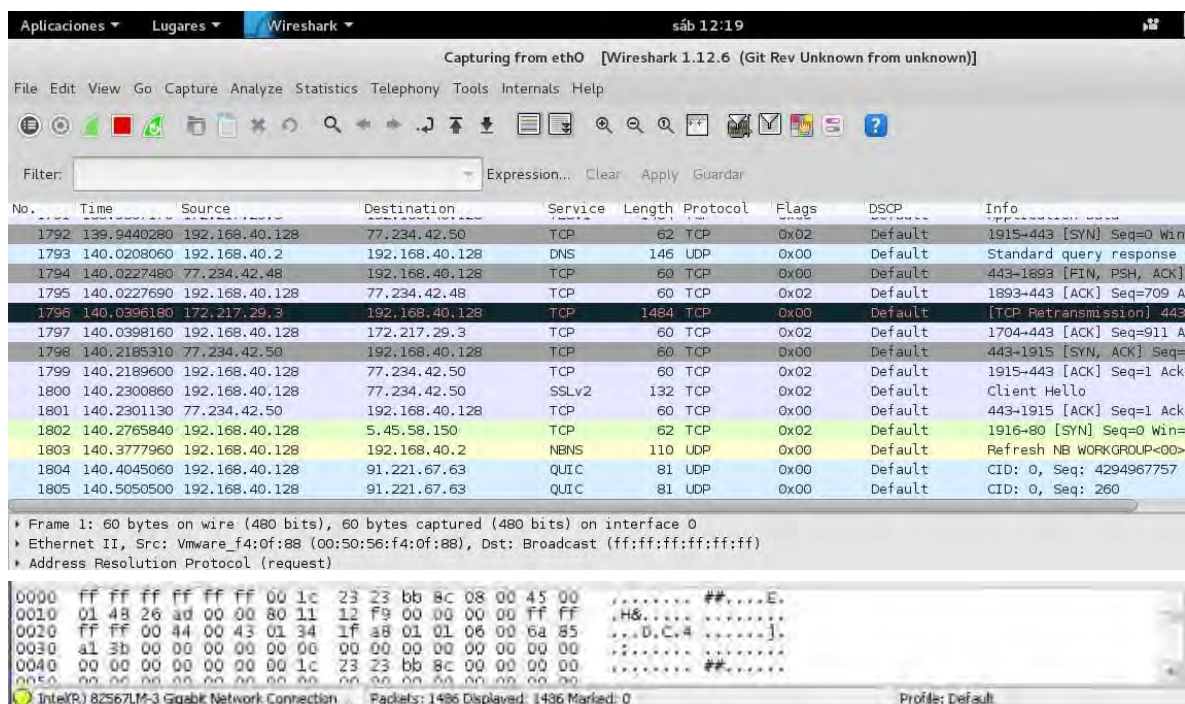


Imagen 27 – Wireshark: Captura de paquetes en tiempo real

Como podemos observar en la imagen, Wireshark cuenta con tres paneles de visualización. En el primero se puede observar los diferentes paquetes que circulan en la red de



datos, en el segundo se puede obtener la información de cada uno de estos paquetes específicamente en diferentes niveles de red, y por último se muestra la información del paquete seleccionado en formato hexadecimal. Cada uno de estos paquetes posee un nombre respectivo:

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-06-09	fe80::3978:143f:95c:ff02::c		SSDP	208	M-SEARCH * HTTP/1.1
2	2013-06-09	192.168.1.1	190.248.35.76	TCP	66	50431 > http [SYN] Seq=0 win=81
3	2013-06-09	190.248.35.76	192.168.1.1	TCP	66	http > 50431 [SYN, ACK] Seq=0 A
4	2013-06-09	192.168.1.1	190.248.35.76	TCP	54	50431 > http [ACK] Seq=1 Ack=1
5	2013-06-09	192.168.1.1	190.248.35.76	HTTP	2531	GET /videoplayback?algorithm=thi
6	2013-06-09	190.248.35.76	192.168.1.1	TCP	60	http > 50431 [ACK] Seq=1 Ack=14
7	2013-06-09	190.248.35.76	192.168.1.1	TCP	60	http > 50431 [ACK] Seq=1 Ack=24
8	2013-06-09	190.248.35.76	192.168.1.1	TCP	1250	[TCP segment of a reassembled P
9	2013-06-09	190.248.35.76	192.168.1.1	HTTP	60	HTTP/1.1 302 Found
10	2013-06-09	192.168.1.1	190.248.35.76	TCP	54	50431 > http [ACK] Seq=2478 Ack
11	2013-06-09	192.168.1.1	190.248.35.76	TCP	54	50431 > http [FIN, ACK] Seq=247
12	2013-06-09	190.248.35.76	192.168.1.1	TCP	60	http > 50431 [ACK] Seq=1198 Ack
13	2013-06-09	69.171.246.16	192.168.1.1	TLSv1.1	1035	Application Data
14	2013-06-09	192.168.1.1	69.171.246.16	TCP	54	50187 > https [ACK] Seq=1 Ack=9
15	2013-06-09	fe80::3978:143f:95c:ff02::c		SSDP	208	M-SEARCH * HTTP/1.1

Imagen 28 – Panel de listado de paquetes

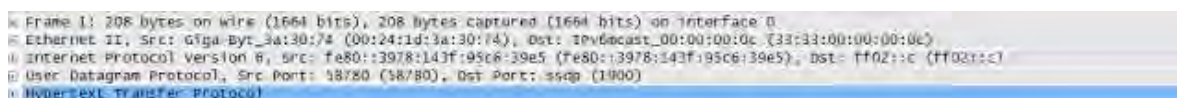


Imagen 29 – Panel de información de cada paquete



Imagen 30 – Panel de información de formato hexadecimal de cada paquete

Wireshark dispone de una serie de filtros que pueden ser aplicados durante la captura (en tiempo real) o a un archivo guardado anteriormente. En ambos casos estos filtros permiten obtener más o menos detalle y/o precisión a la hora de buscar ciertos criterios o aspectos que uno considere necesarios para realizar el trabajo de análisis del tráfico que fluye a través de la red. En lo que concierne a nuestro proyecto no haremos uso de tales filtros ya que la información será procesada y refinada posteriormente con otra herramienta que mencionaremos más adelante

El proceso de captura fue planificado para llevarse a cabo en una serie de etapas, las cuales se describen a continuación:

5.8.1.1 Captura de Tráfico Normal

Se simuló un tráfico de datos normal ingresando a páginas Web de forma manual desde las máquinas virtuales clientes respectivas y también estableciendo comunicación entre estas y con el servidor. El formato del archivo de captura de Wireshark es de extensión **.pcapng**. Del monitoreo y captura del tráfico normal se obtuvo el archivo: **TraficoNormal.pcapng**

5.8.1.2 Captura de Tráfico Normal y Anormal

Para simular el tráfico anormal se hizo uso de las herramientas para pruebas de penetración disponibles en el sistema operativo de Kali Linux 2.0, cuya distribución basada en



Debian, es utilizado principalmente para auditoria y seguridad informática. Algunas de las herramientas que podemos encontrar son:

- Herramientas de Recopilación de Información (nmap,dmitry,sparta, etc)
- Herramientas de Análisis de vulnerabilidades (golismo, nikto)
- Herramientas de Ataques de contraseñas(medusa,ncrack)
- Herramientas de Explotación (armitage, metasploit, set)
- Herramientas de Monitoreo/Envenenamiento (ettercap, Wireshark, netsniff-ng)

Las simulaciones realizadas para el tráfico anormal consistieron en el monitoreo de la red mientras se llevaban a cabo los siguientes ataques:

Ataque Hydra-Fuerza Bruta a servidor FTP

Para llevar a cabo la simulación del ataque de fuerza bruta con diccionarios a través de la herramienta Hydra de Kali Linux 2.0 en primer lugar se creó y configuro un servidor FTP en la máquina Virtual de CentOS, para esto se siguieron los siguientes pasos:

1. Instalar FTP, se ejecuta los siguientes comandos:
 - a. **Yum install vsftpd**
 - b. **Yum install ftp**
2. Editar el archivo de configuración vsftpd.conf:
 - a. **Nano /etc/vsftpd/vsftpd.conf**
 - b. Los siguientes parámetros del archivo de configuración deben seguir el siguiente formato:

```
Anonymous_enable=NO  
Local_enable=YES  
Write_enable=YES  
Chroot_local_user=YES
```

3. Reiniciar el servicio, con el siguiente comando:

Service vsftpd restart

4. Habilitar Puertos 20 y 21 para así poder acceder al servidor desde afuera, lo hacemos con los siguientes comandos:

Iptables -P INPUT ACCEPT

Iptables -A INPUT -i eth0 -p tcp - -dport 20 -j ACCEPT

Iptables -A INPUT -i eth0 -p tcp - -dport 21 -j ACCEPT

Service Iptables save

5. Activar en el Cortafuegos de CentOS el servicio FTP:

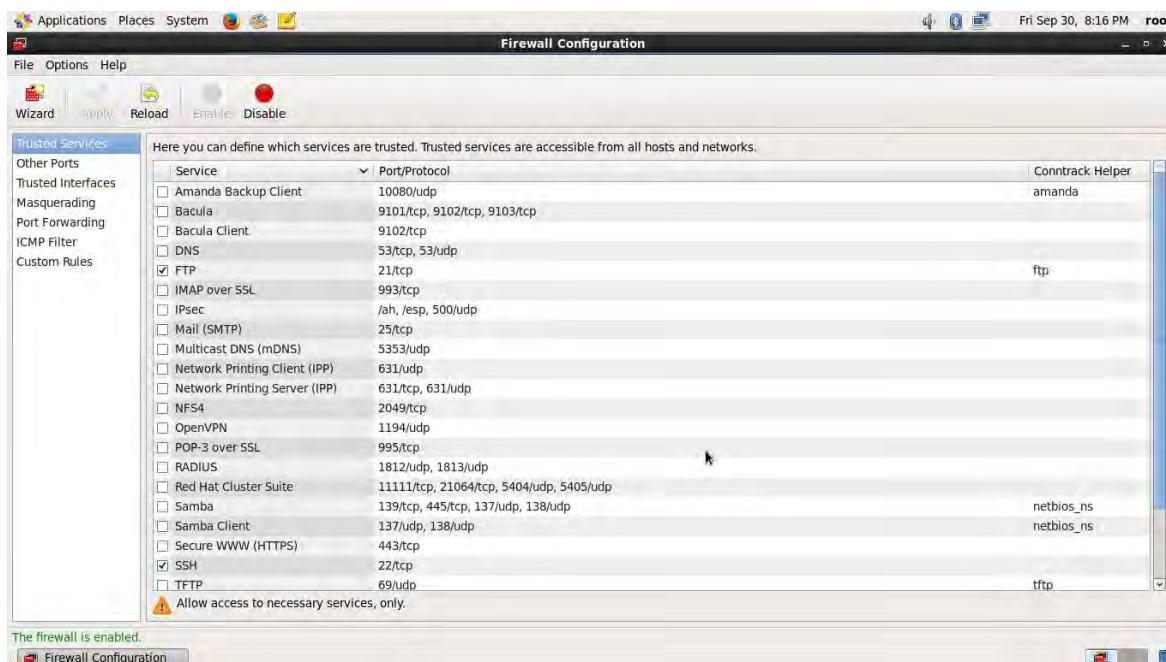


Imagen 31 – Activación Servicio FTP en Cortafuegos de CentOS 6.5

6. Ingresar el siguiente comando para que los usuarios solo puedan acceder a la carpeta home que tengan en el servidor:

Setsebool –P ftp_home_dir on

7. Creación de usuarios para acceso al ftp, ingresamos las siguientes líneas de comando:

Useradd Daniel

Passwd Daniel

8. Posteriormente reiniciamos el servicio Iptables y nuestro servidor ftp:

Service Iptables restart

Service vsftpd restart

9. Por ultimo probamos el ingreso a nuestro servidor FTP, como se muestra en la imagen:



Imagen 32 – Acceso servidor ftp mediante consola CentOS 6.5



Podemos apreciar en la imagen que para acceder al servidor ftp se requiere de un Login de usuario y contraseña. Este servidor de ejemplo aloja un archivo denominado prueba.txt

Una vez creado y configurado nuestro servidor FTP, proseguimos a realizar el ataque de fuerza bruta con diccionarios mediante la herramienta HYDRA, para ello los pasos a seguir son:

1. Crear el diccionario de nombres de usuarios posibles en un archivo denominado users.txt, el cual en cada fila tendrá un nombre de usuario posible.
2. Crear el diccionario de contraseñas de usuarios posibles en un archivo denominado pass.txt, el cual en cada fila tendrá una contraseña de usuario posible.
3. También es posible generar un diccionario aleatorio de usuarios y contraseñas en Kali Linux mediante el comando **crunch**, como podemos observar en la siguiente imagen:

```
root@kali: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@kali:~# crunch 1 5 admin>users.txt  
Crunch will now generate the following amount of data: 22460 bytes  
0 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 3905  
root@kali:~# crunch 1 5 qweaz>pass.txt  
Crunch will now generate the following amount of data: 22460 bytes  
0 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 3905  
root@kali:~#
```

Imagen 33 – Comando Crunch.

El comando crunch <min> <max> [options] especifica:

- Min: el mínimo de combinaciones con caracteres
- Max: el máximo de combinaciones con caracteres
- Options: los caracteres que deseemos colocar para realizar combinaciones entre ellos

En la imagen observamos que colocamos un mínimo de 1 y un máximo de 5 combinaciones acompañados de letras al azar para realizar las combinaciones posibles (3905), al ejecutarse el comando guarda estos valores generados en un archivo de texto, uno para los usuarios (users.txt) y otro para las contraseñas (pass.txt) los cuales serán utilizados posteriormente para el comando Hydra.

4. Una vez obtenidos los diccionarios, procedemos a ejecutar el ataque en cuestión mediante el siguiente comando:

Hydra -L /root/Escritorio/FTP/users.txt -P /root/Escritorio/FTP/pass.txt [dirección_IP] ftp



Dónde:

- L: permite definir un archivo conteniendo una lista de palabras de posibles usuarios del sistema.
- P: permite definir un archivo conteniendo una lista de palabras de posibles contraseñas del sistema.



Imagen 34 – Ataque Hydra-Fuerza Bruta a Servidor FTP

En la imagen se puede observar que el ataque se realiza sobre el puerto 21 del servicio FTP, y se vale de los archivos txt para poder así averiguar el usuario y contraseña requeridos para poder logearse en el servidor ftp que se encuentra alojado en la dirección IP proporcionada, una vez terminada la ejecución Hydra devolverá los datos del Login correctos encontrados dentro de las combinaciones posibles de los archivos users.txt y pass.txt, que permitirán el acceso remoto al servidor FTP.

Toda la actividad realizada fue capturada mediante Wireshark obteniendo así el siguiente archivo: **Anormal_dict_ftp.pcapng**

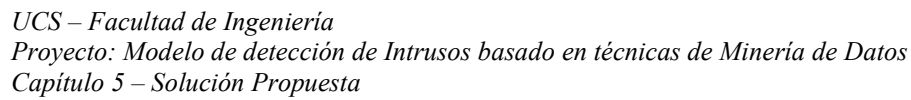
Ataque de Denegación de Servicio - (DOS/Slowloris)

En nuestro proyecto, para la simulación de un ataque de DOS utilizamos dos tipos de técnicas, la primera consiste en el uso de la herramienta **hping3** de Kali Linux y la segunda mediante un script especialmente diseñado para realizar pruebas de ataques a sitios web denominado **Slowloris.pl**

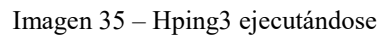
Para llevar a cabo el ataque de DOS con hping3 ejecutamos el siguiente comando en una terminal de Kali Linux:

Hping3 -p 80 -flood IP_victima

Donde:



--flood (envía paquetes tan rápido como sea posible a la IP de la víctima)



Una vez obtenido el archivo se procede a ejecutar en una terminal de Kali Linux el siguiente comando, mediante la utilización del lenguaje de programación perl:

D'onde:

-dns (se especifica la dirección web o IP victima)



Toda la actividad realizada fue capturada mediante Wireshark obteniendo así el siguiente archivo: **AnormalDoS_slowloris.pcapng**



Escaneo de red y puertos con NMAP

Para llevar el cabo el mapeo de redes y puertos con la herramienta nmap Versión 6.49 de Kali Linux 2.0, se ejecutaron los siguientes comandos en terminal:

Nmap -sP 192.168.40.0/24

Donde:

-sP: Realiza un sondeo de ping, para determinar si el objetivo está disponible.

```
root@kali:~# nmap -sP 192.168.40.0/24
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-10-01 10:05 ART
Nmap scan report for 192.168.40.1
Host is up (0.00061s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.40.2
Host is up (0.00052s latency).
MAC Address: 00:50:56:ED:B7:42 (VMware)
Nmap scan report for 192.168.40.128
Host is up (0.00036s latency).
MAC Address: 00:0C:29:2A:02:07 (VMware)
Nmap scan report for 192.168.40.131
Host is up (0.00043s latency).
MAC Address: 00:0C:29:4D:D4:B3 (VMware)
Nmap scan report for 192.168.40.254
Host is up (0.00012s latency).
MAC Address: 00:50:56:E0:5A:E5 (VMware)
Nmap scan report for 192.168.40.130
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.15 seconds
```

Imagen 37 – Escaneo de Red con nmap

Una vez identificados los hosts disponibles en la red, procedemos a determinar que puertos están abiertos o disponibles en cada uno de ellos, para esto ejecutamos los siguientes comandos:

Nmap -top-ports 10 192.168.40.128 (Windows XP)

Nmap -top-ports 10 192.168.40.131 (CentOS 6.5)



```
Aplicaciones Lugares Terminal sáb 10:08
root@kali: ~
Archivo Editar Ver Buscar Terminal Ayuda
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-10-01 10:08 ART
Nmap scan report for 192.168.40.128
Host is up (0.00049s latency).
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    closed ssh
23/tcp    closed telnet
25/tcp    closed smtp
80/tcp    closed http
110/tcp   closed pop3
139/tcp   open  netbios-ssn
443/tcp   closed https
445/tcp   open  microsoft-ds
3389/tcp  closed ms-wbt-server
MAC Address: 08:0C:29:2A:02:07 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
root@kali:~# nmap -top-ports 10 192.168.40.131

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-10-01 10:08 ART
Nmap scan report for 192.168.40.131
Host is up (0.0022s latency).
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    filtered telnet
25/tcp    filtered smtp
80/tcp    filtered http
110/tcp   filtered pop3
139/tcp   filtered netbios-ssn
443/tcp   filtered https
445/tcp   filtered microsoft-ds
3389/tcp  filtered ms-wbt-server
MAC Address: 08:0C:29:4D:04:B3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
root@kali:~#
```

Imagen 38 – Escaneo de puertos con nmap

Toda la actividad realizada fue capturada mediante Wireshark obteniendo así el siguiente archivo: **Anormal_nmap.pcapng**

Ataque warezclient/warezmaster

Para simular un ataque warezclient, como también warezmaster, se valió del servidor ftp ya configurado anteriormente para el ataque de fuerza bruta con diccionarios. Básicamente los pasos para realizar los ataques mencionados son:

1. Login al servidor FTP, desde una maquina remota:

ftp [Dirección_IP]

2. Una vez que se accede al servidor FTP, se procede a realizar la subida de software y archivos (warezmaster) con el comando PUT:

Put [ubicacion/nombre_archivo]

3. Posteriormente se realizó la descarga de este “software ilegal” desde otra máquina remota (warezclient), con el comando GET:

Get [ubicación/nombre_archivo]

*cabe mencionar el comando “ls” o “dir”, que muestra el árbol de directorios del servidor ftp.



Toda la actividad realizada fue capturada mediante Wireshark obteniendo así el siguiente archivo: **Anormal_warez_ftp.pcapng**

Una vez terminado el proceso de monitoreo y captura de datos en la red, los archivos obtenidos mediante Wireshark son:

- TraficoNormal.pcapng
- Anormal_dict_ftp.pcapng
- Anormal_DOSslowloris.pcapng
- Anormal_nmap.pcapng
- Anormal_warez_ftp.pcapng

Estos archivos de extensión .pcapng serán utilizados en el siguiente proceso.

5.9 Preparación de los datos

Los archivos obtenidos del monitoreo mediante Wireshark todavía no se encuentran listos para iniciar el proceso de minería de datos, es necesario que pasen por un proceso de refinación y preparación para eliminar datos basura y redundantes.

Para preparar los datos en el formato adecuado para realizar el proceso de minería nos valimos de la herramienta BRO Network Security Monitor v2.4.1, la cual mediante un algoritmo determinado nos permite obtener los datos en el formato de la GureKDD cup data base (2008). A este algoritmo lo denominamos darpa2gurekddcup.bro o darpa.bro (para simplificar) y tiene el siguiente formato:



```
##### UDP #####
redef udp_content_deliver_all_orig = I;
redef udp_content_deliver_all_resp = I;
event udp_contents(u: connection, is_orig: bool, contents: string)
{
    local startTime : string = fmt ("%f", u$start_time-6*60min);
    local duration : string = fmt ("%f", u$duration);
    local orig_h : addr = u$id$orig_h;
    local resp_h : addr = u$id$resp_h;
    local orig_p : port = u$id$orig_p;
    local resp_p : port = u$id$resp_p;
    if ([startTime, duration, orig_h, orig_p, resp_h, resp_p] !in konexioak)
    {
        local empty_record: konexioa;
        konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p] = empty_record;

        ++num_conn;
        konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p]$num_conn = num_conn;
    }

    if ([startTime, orig_h, orig_p, resp_h, resp_p] !in byteak){
        local empty_byte: byte;
        byteak[startTime, orig_h, orig_p, resp_h, resp_p] = empty_byte;
    }

    konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p]$protocoloa = "udp";
    konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p]$flag = conn_state(u, "udp");

    if (is_orig){
        konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p]$isOrig = I;

        konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p]$src_bytes =
            fmt ("%d", u$orig$size-byteak[startTime, orig_h, orig_p, resp_h, resp_p]$src_bytes);
        konexioak[startTime, duration, orig_h, orig_p, resp_h, resp_p]$dst_bytes = "0";
        byteak[startTime, orig_h, orig_p, resp_h, resp_p]$src_bytes = u$orig$size;

        #Payload
    }
```

Imagen 39 - Fragmentos del algoritmo Darpa bro

(Perona, 2015)



```
##### TCP #####
function record_connectionTCP(c: connection){
    local startTime : string = fmt("%.6f", c$start_time-6*60min);
    local orig_h : addr = c$id$orig_h;
    local resp_h : addr = c$id$resp_h;
    local orig_p : port = c$id$orig_p;
    local resp_p : port = c$id$resp_p;
    if (startTime, orig_h, orig_p, resp_h, resp_p] !in konexioaktcp)
    {
        local empty_record: konexica;
        konexioaktcp[startTime, orig_h, orig_p, resp_h, resp_p] = empty_record;

        ++num_conn;
        konexioaktcp[startTime, orig_h, orig_p, resp_h, resp_p]$num_conn = num_conn;
    }
    konexioaktcp[startTime, orig_h, orig_p, resp_h, resp_p]$duration = fmt("%.6f", c$duration);
    konexioaktcp[startTime, orig_h, orig_p, resp_h, resp_p]$flag = conn_state(c, "tcp");
    konexioaktcp[startTime, orig_h, orig_p, resp_h, resp_p]$src_bytes = fmt("%d", c$orig$size);
    konexioaktcp[startTime, orig_h, orig_p, resp_h, resp_p]$dst_bytes = fmt("%d", c$resp$size);
}

event new_connection(c: connection){
    local trans : string = fmt("%s", get_port_transport_proto(c$id$resp_p));
    if(trans=="tcp"){
        record_connectionTCP(c);
    }
}

event connection_state_remove(c: connection){
    local trans : string = fmt("%s", get_port_transport_proto(c$id$resp_p));
    if (trans == "tcp"){
        record_connectionTCP(c);
    }
}

event tcp_packet (c: connection, is_orig: bool, flags: string, seq: count, ack: count, len: count, payload: string){
    local startTime : string = fmt("%.6f", c$start_time-6*60min);
}
```

Imagen 40 – Fragmentos del algoritmo Darpa bro

(Perona, 2015)

Cada uno de los archivos .pcapng son procesados utilizando el algoritmo darpa.bro para obtener los primeros 26 atributos intrínsecos que son atributos extraídos desde el área de cabecera de cada uno de los paquetes de red capturados, para esto inicializamos BRO desde una terminal ejecutando el siguiente comando:

export PATH=/usr/local/bro/bin:\$PATH

-esta línea de comando indica el camino o dirección donde se ejecuta la herramienta bro.

Posteriormente para iniciar la herramienta ejecutamos:

Broctl (inicia la consola de la herramienta BRO)

Start (inicia BRO)

Stop (detiene BRO)

Una vez iniciada la instancia de BRO ejecutamos el siguiente comando en una terminal en CentOS:



bro -r archivo.pcapng darpa2gurekddcup.bro > conn.list

dónde:

-r: indica el archivo a ser procesado por el algoritmo darpa.bro

El resultado obtenido por cada archivo .pcapng es un archivo denominado conn.list, donde para cada conexión los atributos de conn.list son:

ID	Nombre	Tipo de dato	Descripción
1	connection_number	Integer	ID de la conexión o paquete
2	start_time	Real	Tiempo en que comenzó la conexión
3	orig_port	Integer	Número del puerto de origen de la conexión
4	resp_port	Integer	Número del puerto que responde la conexión
5	Duration	Real	Duración de la conexión
6	protocol_type	Nominal	Tipo de protocolo de la conexión
7	Service	Nominal	http, ftp, smtp, telnet... y otros
8	Flag	Nominal	Estado de la conexión (SF,S0,S1, OTH, REJ, SH, etc)
9	src_bytes	Integer	Bytes enviados en una conexión
10	dst_bytes	Integer	Bytes recibidos en una conexión
11	Land	Binary	Si el emisor, el receptor y los números de los puertos son iguales entonces esta variable es 1 sino 0
12	wrong_frangment	Integer	Suma de los paquetes malos de la suma de comprobación en una conexión
13	Urgent	Integer	Suma de los paquetes urgentes en una conexión. Los paquetes urgentes son los paquetes con el bit urgente activado
14	Hot	Integer	Suma de las acciones riesgosas en una conexión tales como: accesos a un directorio del sistema, creación de programas y ejecución de programas
15	num_failed_logins	Integer	Numero de logueos incorrectos
16	logged_in	Binary	Si el logueo es correcto entonces esta variable es 1 sino 0
17	num_compromised	Integer	Suma de las veces que aparece el error "not found" en una conexión
18	root_shell	Binary	Si la raíz es con cobertura entonces es 1 sino 0
19	su_attempted	Binary	Si el comando SU se ha utilizado es 1 sino 0
20	num_root	Integer	Suma de la operaciones como administrador en una conexión
21	num_file_creations	Integer	Suma de la creación de archivos en una conexión
22	num_shell	Integer	Numero de logueos de usuarios normales



23	num_access_files	Integer	Suma de operaciones en los archivos de control en una conexión
24	num_outbound_cmds	Integer	Suma de comandos salientes en una sesión ftp
25	is_hot_login	Binay	Si el usuario accede como administrador es 1 sino 0
26	is_guest_login	Binary	Si el usuario accede como huésped, anónimo o visitante es 1 sino 0

Cuadro 8 – Referencia de campos principales

Seguidamente, compilamos y ejecutamos un segundo algoritmo (escrito en C), denominado **trafAld.c**, para crear los restantes 19 atributos de tráfico de red (desde el 27 al 45), calculados a partir de los primeros 26 atributos. Este programa en C tiene el siguiente formato:

```
#include <stdio.h>
#include <string.h>

#define MaxCon 800000
#define MaxLine 800
#define MaxToken 20

char inigo[MaxCon][MaxLine];
int inigoKop;

int main(int argc, char **argv)
{
    FILE *finigo, *ftrafAld;
    int i,j;
    char galdera[MaxLine];
    int noraino;
    int nondik2, nondik100;
    char line[MaxLine];

    //Uneko konexioaren datuak
    char line1[MaxLine];
    int konZenb1;
    char hasUnea1s[MaxToken];
    int hasUnea1, hasUnea12;
    int orig_p1,resp_p1;
    char orig_h1[MaxToken], resp_h1[MaxToken];
    char duration1[MaxToken], protokoloa1[MaxToken], service1[MaxToken], flag1[MaxToken];

    //Aurreko konexioen datuentzat
    char line2[MaxLine];
    int konZenb2;
    char hasUnea2s[MaxToken];
    int hasUnea2, hasUnea22;
    int orig_p2,resp_p2;
    char orig_h2[MaxToken], resp_h2[MaxToken];
    char duration2[MaxToken], protokoloa2[MaxToken], service2[MaxToken], flag2[MaxToken];
    int sartuDa2;

    //trafiko aldagaiak
    int count,svr_count,error,rrror,same_svr,diff_svr,svr_serror,svr_error,svr_diff_host;
    int same_src_port;
    float serror_rate,svr_serror_rate,error_rate,svr_error_rate,same_svr_rate,diff_svr_rate,svr_diff_host_rate;
    float svr_rerror_rate,same_src_port_rate;
```

Imagen 41 – Algoritmo trafAld

(Perona, 2015)

Compilamos el archivo trafAld.c mediante:



gcc trafAld.c -o trafAld.out

Ejecutamos el archivo trafAld.out mediante:

./trafAld.out conn.list

-crea el archivo trafAld.list que contiene los atributos que incluye la gureKDDCup 2008.

Finalmente se obtiene un listado de 45 atributos, donde los primeros 26 son atributos de la cabecera de los paquetes, y los restantes 19 son atributos de tráfico de red, que a la vez se dividen en atributos de tiempo de tráfico (desde el atributo 27 hasta el 35) y atributos de tráfico de maquina (desde el atributo 36 hasta el 45)

Para cada conexión los atributos del archivo final denominado trafAld.list son:

ID	Nombre	Tipo de dato	Descripción
1	connection_number	Integer	ID de la conexión o paquete
2	start_time	Real	Tiempo en que comenzó la conexión
3	orig_port	Integer	Número del puerto de origen de la conexión
4	resp_port	Integer	Número del puerto que responde la conexión
5	Duration	Real	Duración de la conexión
6	protocol_type	Nominal	Tipo de protocolo de la conexión
7	Service	nominal	http, ftp, smtp, telnet... y otros
8	Flag	Nominal	Estado de la conexión (SF,S0,S1, OTH, REJ, SH, etc)
9	src_bytes	Integer	Bytes enviados en una conexión
10	dst_bytes	Integer	Bytes recibidos en una conexión
11	Land	Binary	Si el emisor, el receptor y los números de los puertos son iguales entonces esta variable es 1 sino 0
12	wrong_frangment	Integer	Suma de los paquetes malos de la suma de comprobación en una conexión
13	Urgent	Integer	Suma de los paquetes urgentes en una conexión. Los paquetes urgentes son los paquetes con el bit urgente activado
14	Hot	Integer	Suma de las acciones riesgosas en una conexión tales como: accesos a un directorio del sistema, creación de programas y ejecución de programas
15	num_failed_logins	Integer	Numero de logueos incorrectos

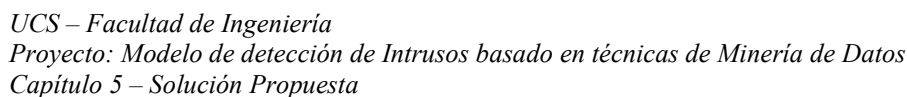


16	logged_in	Binary	Si el logueo es correcto entonces esta variable es 1 sino 0
17	num_compromised	Integer	Suma de las veces que aparece el error "not found" en una conexión
18	root_shell	Binary	Si la raíz es con cobertura entonces es 1 sino 0
19	su_attempted	Binary	Si el comando SU se ha utilizado es 1 sino 0
20	num_root	Integer	Suma de la operaciones como administrador en una conexión
21	num_file_creations	Integer	Suma de la creación de archivos en una conexión
22	num_shell	Integer	Numero de logueos de usuarios normales
23	num_access_files	Integer	Suma de operaciones en los archivos de control en una conexión
24	num_outbound_cmds	Integer	Suma de comandos salientes en una sesión ftp
25	is_hot_login	Binary	Si el usuario accede como administrador es 1 sino 0
26	is_guest_login	Binary	Si el usuario accede como huésped, anónimo o visitante es 1 sino 0
27	Count	Integer	Suma de conexiones a la misma dirección IP de destino
28	srv_count	Integer	Suma de conexiones al mismo número de puerto de destino
29	error_rate	Real	El porcentaje de conexiones que tienen activado el flag (4): S0, S1, S2 o S3, entre las conexiones agregadas en count (23)
30	srv_error_rate	Real	El porcentaje de conexiones que tienen activado el flag (4): S0, S1, S2 o S3, entre las conexiones agregadas en srv_count (24)
31	reror_rate	Real	El porcentaje de conexiones que tienen activado el flag (4): REJ, entre las conexiones agregadas en count (23)
32	srv_error_rate	Real	El porcentaje de conexiones que tienen activados el flag (4): REJ, entre las conexiones agregadas en srv_count (4)
33	same_srv_rate	Real	El porcentaje de conexiones que fueron con el mismo servicio, entre las conexiones agregadas en count (23)
34	diff_srv_rate	Real	El porcentaje de conexiones que fueron con diferentes servicios, entre las conexiones agregadas en count (23)



35	srv_diff_host_rate	Real	El porcentaje de conexiones que fueron a diferentes destinatarios, entre las conexiones agregadas en srv_count (24)
36	dst_host_count	Integer	Suma de conexiones con la misma dirección IP de destino
37	dst_host_srv_count	Integer	Suma de conexiones con el mismo número de puerto de destino
38	dst_host_same_srv_rate	Real	El porcentaje de conexiones que fueron con el mismo servicio, entre las conexiones agregadas en dst_host_count (32)
39	dst_host_diff_srv_rate	Real	El porcentaje de conexiones que fueron con diferentes servicios, entre las conexiones agregadas en dst_host_count (32)
40	dst_host_same_src_port_rate	Real	El porcentaje de conexiones que salieron del mismo puerto origen, entre las conexiones agregadas en dst_host_srv_count (33)
41	dst_host_srv_diff_host_rate	Real	El porcentaje de conexiones que fueron a diferentes equipos destinatarios, entre las conexiones agregadas en dst_host_srv_count (33)
42	dst_host_serror_rate	Real	El porcentaje de conexiones que tienen activado el flag (4): S0,S1,S2 o S3, entre las conexiones agregadas en dst_host_count (32)
43	dst_host_srv_serror_rate	Real	El porcentaje de conexiones que tienen activado el flag (4): S0,S1,S2 o S3, entre las conexiones agregadas en dst_host_srv_count (33)
44	dst_host_rerror_rate	Real	El porcentaje de conexiones que tienen activado el flag (4): REJ, entre las conexiones agregadas en dst_host_count (32)
45	dst_host_srv_rerror_rate	Real	El porcentaje de conexiones que tienen activado el flag (4) REJ, entre las conexiones agregadas en dst_host_srv_count (33)

Cuadro 9 – Referencia de todos los campos

[illegible]

Cabe recordar que para cada archivo .pcapng obtenido del monitoreo de wireshark, hay un archivo trafAld.list que lo muestra con el formato adecuado de la GureKDD Cup database 2008.

Alumnos: Ávila Alfredo Carlos – Barboza Daniel Félix



UCS – Facultad de Ingeniería
Proyecto: Modelo de detección de Intrusos basado en técnicas de Minería de Datos
Capítulo 5 – Solución Propuesta



Imagen 43 – Archivo original Gure KDD Cup Database 2008 extensión .arff

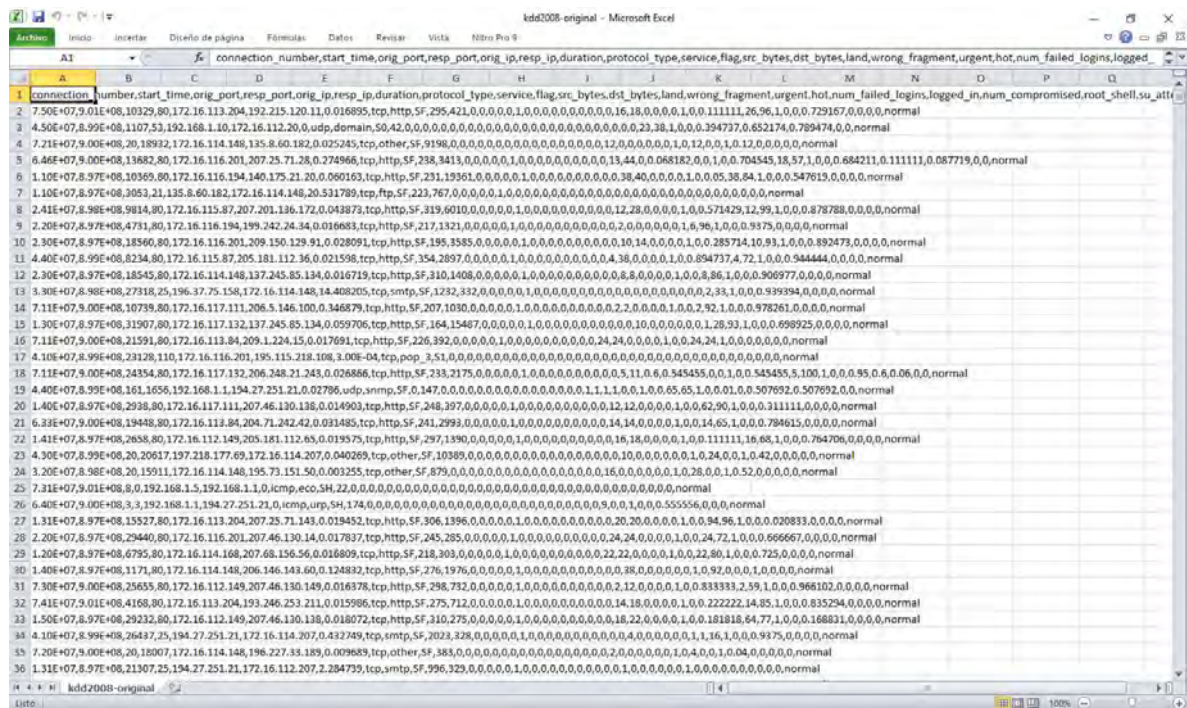


Imagen 44 – Archivo original Gure KDD Cup Database 2008 abierto con MS Excel 2010

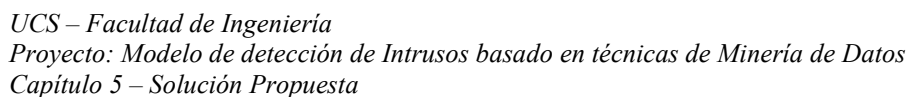


Imagen 45 – Archivo original Gure KDD Cup Database 2008

[illegible]

Imagen 46 – Campo de clasificación (clase) de los registros del
archivo Gure KDD Cup Database 2008



El proceso de preparación de datos siguió con normalización de los valores de los atributos de los archivos: GureKDD Cup 2008 (gurekdd6percent), NSL KDD 2012 (KDDTrain-20Percent) y los archivos capturados al momento de realizar cada uno de los ataques. Los archivos capturados fueron nombrados de la siguiente manera:

- **TraficoNormal.csv:** 205 registros
- **Anormal_dict_ftp.csv:** 160 registros
- **Anormal_probing.csv:** 98 registros
- **AnormalDoS_slowloris.csv:** 3357 registros
- **Anormal_warez_ftp.csv:** 217 registros

Para la normalización de los datos se trabajó con los atributos de tipo nominal creando distintos dominios de valores que se apliquen en todos los archivos, por ejemplo, para el atributo “service” se estableció que todos los archivos tengan los valores de dicho atributo dentro del dominio creado para ese atributo. Los dominios de valores están basados en los valores de los atributos de las bases de datos previamente clasificadas (GUREKDD 2008 y NSL KDD 2012). Para crear los dominios de valores se tuvieron que renombrar algunos valores buscando relaciones entre los valores distintos, como por ejemplo, los servicios capturados como “dns” renombrarlos como “domain”.

En el caso de los campos de tipo numérico (integer, real, binario) se los dejó como estaban, ya que los valores numéricos pueden variar amplia e indistintamente entre los paquetes (adquiriendo valores propios de una conexión de red), pudiendo ser estos valores discretos (por ejemplo, número de bytes enviados) o continuos (duración de la conexión).

5.10 Modelado

Para poder detectar los ataques realizados, se procedió con la creación modelos de minería de datos (modelos de clasificación) que determinen las características principales de las conexiones de red y permitan clasificar los datos capturados, para ello se utilizaron los archivos gureKddcup6percent y KDDTrain-20Percent (GUREKDD 2008 y NSL KDD 2012 respectivamente), y los algoritmos de clasificación: Random forest, J48 y Part (los cuales generan reglas y árboles de decisión). Se creó un modelo de clasificación por cada combinación de un archivo y un algoritmo, para luego seleccionar el/los modelos con mayor precisión de clasificación.

5.10.1 Creación y selección los modelos de clasificación

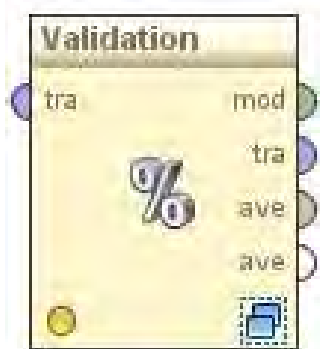
Para crear y seleccionar los modelos de minería de clasificación más eficientes se utilizaron los siguientes operadores de Rapid Miner, con los cuales se cargaron los datos de los archivos gureKddcup6percent y KDDTrain-20Percent, se seleccionaron los atributos más significativos y se aplicaron cada uno de los algoritmos mencionados anteriormente, como así también se evaluó la performance de cada uno:



Read CSV: Este operador se utiliza para leer archivos CSV.



Select Attributes: Este operador permite seleccionar los atributos del archivo cargado, para crear el modelo de minería de datos.



XValidation: Este operador realiza una validación cruzada con el fin de estimar la capacidad predictiva de un operador de aprendizaje.

Este operador se compone de dos entornos, uno de entrenamiento y otro de prueba, que se pueden visualizar de la siguiente manera:

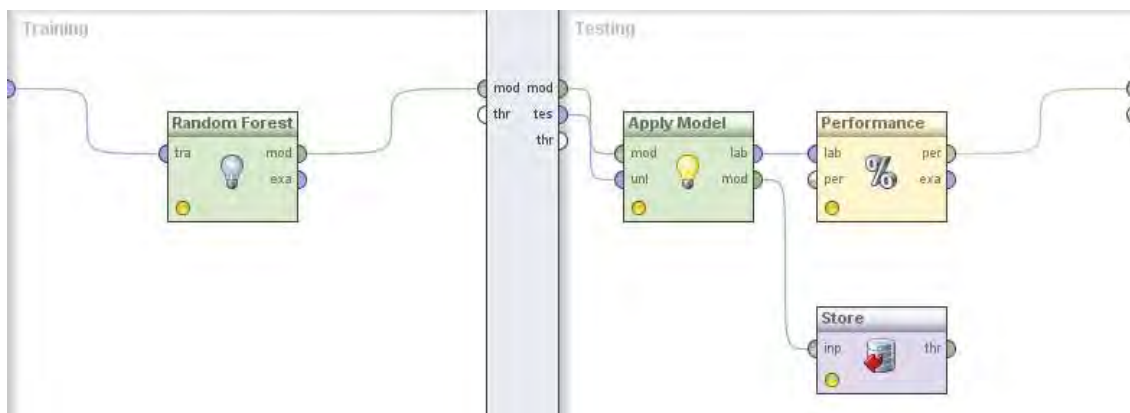


Imagen 47 – Expansión del operador Validation



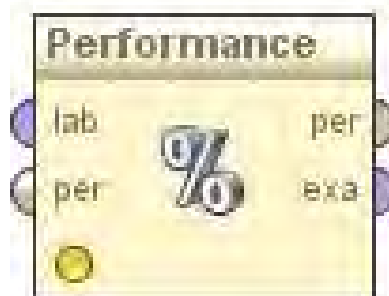
Dentro de Rapid Miner encontramos operadores que modelan los algoritmos antes mencionados, estos operadores realizan su función de clasificación dentro del entorno de entrenamiento o training (en imagen 48). Por ejemplo, en la imagen 48 se muestra el operador Random Forest que opera de acuerdo al algoritmo Random Forest.

La elección de estos algoritmos fue realizada al considerar que para este modelo de clasificación (modelo predictivo), las técnicas de árboles de decisión y generación de reglas se adecuan mejor para la detección de ataques en los flujos de datos, ya que establecen todos los caminos posibles para detectar un posible ataque.

Dentro del entorno de prueba o testing (en imagen 47), se usan los siguientes operadores:



Apply Model: Este operador aplica el modelo obtenido al conjunto no etiquetado (no clasificados).



Performance: Este operador se utiliza para la evaluación del modelo. Evalúa la cantidad de aciertos y errores en la clasificación del modelo.



Store: El operador Store almacena el modelo de minería de datos creado por el algoritmos usado sobre los datos cargado.



Proceso de validación completo:

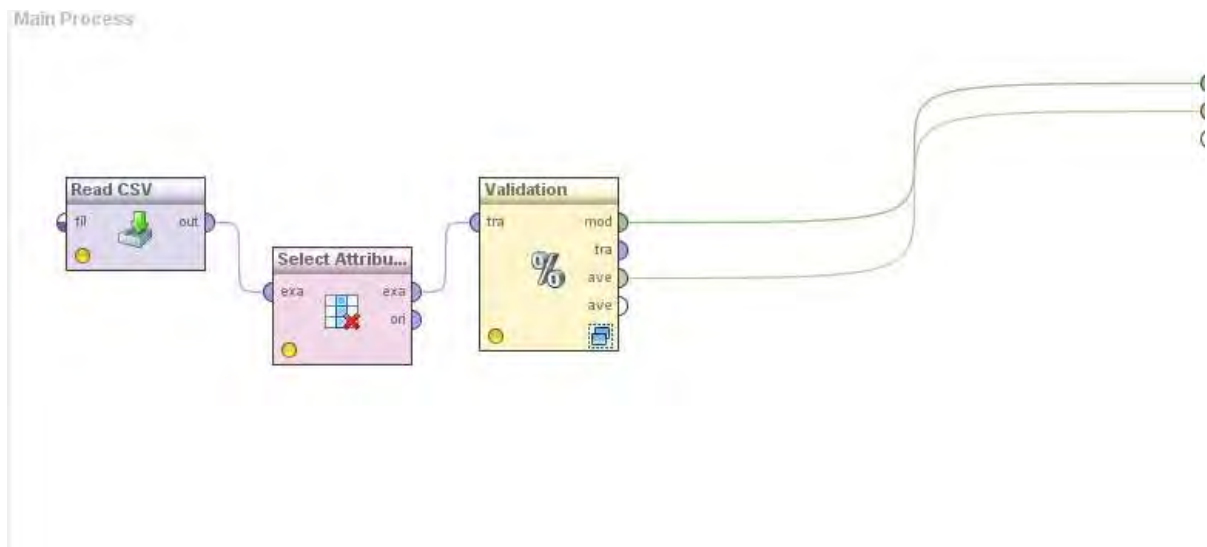


Imagen 48 – Proceso de validación

Como puede verse en la imagen, solamente se carga un archivo CSV, siendo este el archivo gureKddcup6percent o el archivo KDDTrain-20Percent, luego con el operador Select Attributes se seleccionan los atributos más significativos para crear el modelo clasificación, dichos atributos son:

- Protocol_type: El tipo de protocolo es bastante importante ya que ciertos tipos de protocolos son más vulnerables que otros.
- Service: De igual manera que los tipos de protocolo, los servicios también juegan un rol importante en la vulnerabilidad de las conexiones.
- Duration: La duración de una conexión entre una máquina y otra puede determinar la posibilidad de un ataque.
- Src_bytes: El número de bytes de datos de fuente a destino es significativo puesto que normalmente los ataques se realizan con paquetes pequeños.
- Dst_bytes: De la misma manera que con los bytes de fuente a destino, los bytes de datos de destino a fuente también juegan un rol importante en los ataques.
- Flag: Representa el estado de conexión, donde muchos ataques se suelen realizar cuando se está en un estado conocido (SF, S1, REJ, etc).
- Land: El valor es 1 si la conexión es siempre hacia o desde un mismo puesto o host, o el valor 0 si es de otro modo.
- Wrong_fragment: Cuando el número de fragmentos erróneos es elevado, puede deberse a que se está produciendo un ataque.
- Urgent: Representa el número de paquetes urgentes, puede ocurrir lo mismo que con wrong_fragment.
- Count: Indica el número de conexiones hacia una misma máquina, y superado cierto límite puede alertar que se está queriendo entrar o atacar dicha máquina.



- Orig_port: Indica el puerto desde donde se realizan las conexiones.
- Resp_port: Indica el puerto hacia donde se realiza la conexión, en conjunto con Orig_port se pueden determinar anomalías en las conexiones de red sobre configuraciones previas (establecimiento de puertos de comunicación). A diferencia de las direcciones IP, los puertos conexión siempre son los mismos en cada equipo, pudiéndose realizar diversas configuraciones.

Estos atributos fueron seleccionados como los mas significativos, ya que son los principales de los atributos obtenidos de la cabecera de los paquetes de red, que contienen características propias de los ataques que sirven para detectar los ataques realizados (más adelante servirán como parámetros de las reglas de detección de intrusos), como ser información del origen de los paquetes y del destino (puertos), medios por el cual son enviados y recibidos (protocolos y servicios), características del contenido (bytes enviados y recibidos, estado de las conexiones, sentido de las conexiones, duración, fragmentos y paquetes urgentes), y además, (no formando parte de la información de la cabecera de los paquetes, pero determina características de ciertos ataques) el registro de la cantidad de conexiones a un mismo destino (host) en un tiempo determinado. El resto de los atributos no fueron utilizados en los modelos de minería de datos ya que no contienen información relevante para la detección de ataques y no tienen relación con los parámetros de las reglas de detección de intrusos, ya que son obtenidos a partir de cálculos realizados, tomando como base los atributos principales.

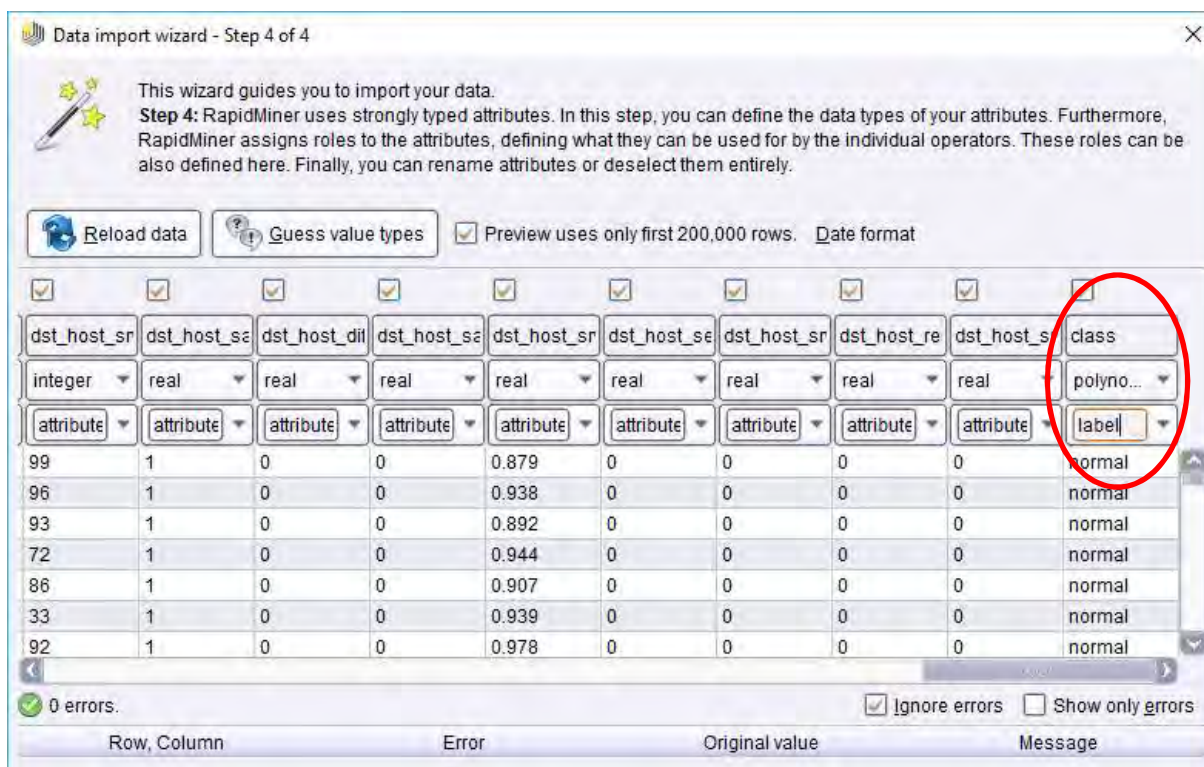


Imagen 49 – Panel de carga de datos



En la imagen 49 se puede ver el panel donde se cargan los datos (con el operador Read CSV), en el mismo se establecen los nombres de los campos, tipos de datos y tipos de campos, por ejemplo, se puede ver remarcado con un círculo rojo el campo que establece el valor de clasificación de cada registro, donde el nombre del campo es “Class”, el tipo de dato es Polinomial y el tipo de campo es Label (la etiqueta de clasificación), el resto de los campos tienen como tipo de dato Integer, Real o Polinomial, y como tipo de campo Atributo (attribute).

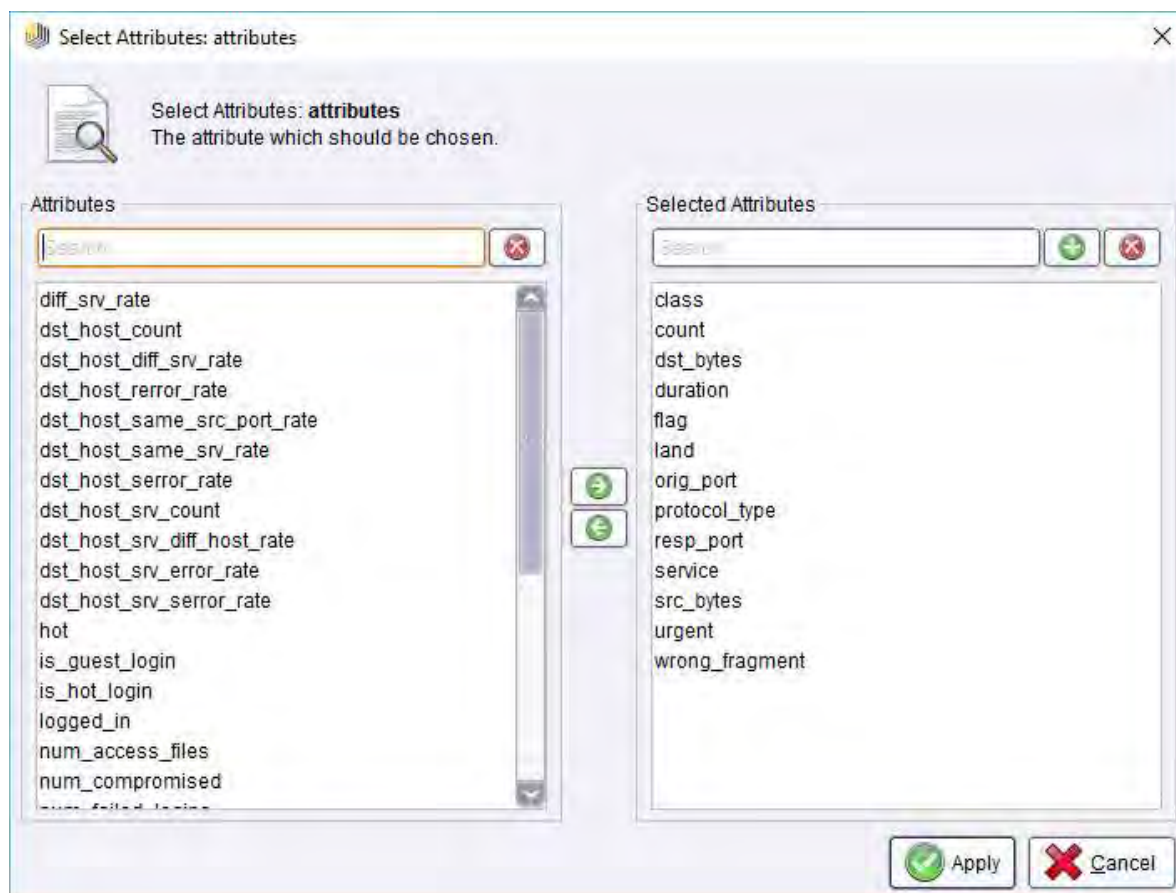


Imagen 50 – Panel de selección de atributos

Una vez cargados los datos, se seleccionan los atributos más significativos con el operador Select Attributes, con los cuales se crea el modelo de clasificación por cada archivo cargado.

Seguidamente con el operador Xvalidation se realiza la validación cruzada con K iteraciones (que de modo predeterminado K=10), creando el modelo de clasificación a partir del algoritmo cargado. Una vez determinado el modelo de clasificación, este es evaluado con el operador Performace que determina la precisión de clasificación del modelo creado mediante la devolución de la “Matriz de confusión” o “Matriz de clasificación” que muestra el desempeño del algoritmo utilizado. El operador Store almacena el modelo de minería de datos creado para posteriormente ser usado con los datos no clasificados.



accuracy: 90.28% +/- 2.61% (mikro: 90.28%)

	true normal	true warezci	true dict	true warezm	true teardrc	true syslog	true land	true guest	true imap	true phf	true multih	true rootkit	true ftp-write	true loadmo	true eject	true fo
red. norma	158057	9	4	0	0	0	2	2	2	2	4	3	2	0	1	0
red. warez	1847	1278	0	0	6	0	0	0	0	0	0	4	0	0	0	0
red. dict	0	0	885	0	0	0	0	0	0	0	0	0	0	0	0	0
red. warezm	19	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0
red. teardrc	0	0	0	0	1079	0	0	0	0	0	0	0	0	0	0	0
red. syslog	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
red. land	2	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0
red. guest	0	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0
red. imap	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
red. phf	7271	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
red. multih	1006	1	0	2	0	0	0	0	1	0	1	0	2	0	0	0
red. rootkit	940	7	0	0	0	0	0	0	0	0	0	14	0	1	1	1
red. ftp-write	400	10	0	0	0	0	0	0	0	0	0	0	3	0	0	0
red. loadmo	416	16	9	0	0	0	0	0	0	0	2	0	0	5	2	0
red. eject	72	0	1	0	0	0	0	1	0	0	1	0	0	1	4	0
red. format	859	268	0	0	0	1	0	0	0	0	0	5	0	1	0	4
red. format	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
red. ffb	3972	160	0	0	0	0	0	0	1	0	1	3	1	0	0	1
red. ffb_cle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
red. anoma	12	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
red. perfme	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Imagen 51 – Ejemplo de matriz de confusión

En la matriz de confusión se puede ver en la parte superior izquierda el porcentaje de precisión de la clasificación y el porcentaje del error, y en el contenido de la matriz, en la diagonal principal, la cantidad de aciertos de los distintos valores de clasificación, y en el resto de las celdas la cantidad de desaciertos en la clasificación.

Una vez realizado este proceso de Validación con todos los algoritmos y las bases de datos GUREKDD y NSLKDD, se volcaron los resultados sobre dos tablas (una por cada archivo) para comparar los resultados y realizar la selección definitiva de los modelos de clasificación para clasificar los datos capturados. Los resultados obtenidos se resumen de la siguiente manera:

GURE KDD	Precisión	Error	Tiempo	Aciertos	Desaciertos
Part	99.92%	+/- 0.02%	01.13.30	178669	141
J48	99.92%	+/- 0.01%	00.56.38	178664	146
Random	99.93%	+/- 0.02%	01.23.17	178677	133

Cuadro 10 – Comparación de performance de algoritmos – Gure KDD

NSL KDD	Precision	Error	Tiempo	Aciertos	Desaciertos
Part	98.18%	+/- 0.09%	00.59.46	123684	2289
J48	98.13%	+/- 0.08%	00.11.3	123612	2361
Random	97.98%	+/- 0.08%	00.25.54	123431	2542

Cuadro 11 – Comparación de performance de algoritmos – NSL KDD

Como puede verse en los cuadros, el algoritmo que más precisión tiene a la hora de predecir los valores de clasificación en el archivo gurekdd6percent (GUREKDD) es el algoritmo Random Forest, y para el archivo KDDTrain-20Percent (NSLKDD) es el algoritmo



Part (tomando en cuenta la precisión de la clasificación y la cantidad de aciertos y desaciertos). De este modo se seleccionaron los modelos de clasificación creados por los algoritmos Random Forest y Part para clasificar los archivos generados con la captura del tráfico de red simulado.

Como se describió anteriormente, los ataques realizados en la simulación del tráfico de red fueron: Hydra-Fuerza Bruta, Denegación de Servicio, Escaneo de red y puertos, y warezclient/warezmaster, todos estos ataques fueron capturados en archivos diferentes. Para la detección de cada uno de estos ataques se utilizaron los modelos de clasificación seleccionados (empleando cada modelo para detectar los ataques que contienen en su clasificación):

Modelo generado con el archivo **gurekdd6percent** (GURE KDD database 2008) para detectar los ataques: Hydra-Fuerza Bruta, Denegación de Servicio y warezclient/warezmaster.

Class= dict

dst_bytes <= 195 AND dst_bytes > 138 AND flag = SF AND src_bytes > 116: dict

service = telnet AND dst_bytes <= 213 AND duration > 0.390726 AND flag = S3 AND dst_bytes > 184: dict

service = telnet AND dst_bytes <= 213 AND src_bytes > 43 AND src_bytes <= 124: dict

service = telnet AND dst_bytes <= 213 AND src_bytes <= 62 AND orig_port <= 17721 AND orig_port > 12905: dict

service = telnet AND orig_port <= 11339: dict

Class= teardrop

wrong_fragment > 0 AND orig_port > 161: teardrop

Class= multihop

orig_port <= 2068 AND src_bytes <= 28 AND orig_port > 8: multihop

service = ftp: multihop

Class= warezclient

src_bytes > 245 AND dst_bytes <= 2472 AND wrong_fragment > 0: warezclient

src_bytes > 245 AND dst_bytes <= 2472 AND resp_port > 20 AND service = ftp: warezclient

dst_bytes <= 4 AND src_bytes > 245 AND orig_port <= 514 AND duration <= 3.63992 AND src_bytes <= 353 AND duration <= 0.008517 AND src_bytes > 305: warezclient

dst_bytes <= 4 AND duration > 3.63992 AND count > 0 AND resp_port > 520: warezclient



dst_bytes <= 5 AND src_bytes > 245 AND src_bytes <= 249: warezclient

dst_bytes <= 254 AND service = ftp AND flag = S0: warezclient

resp_port > 79 AND service = other AND protocol_type = tcp AND orig_port > 8 AND dst_bytes <= 229971: warezclient

Class= warezmaster

orig_port <= 1031 AND dst_bytes > 397 AND resp_port <= 3551 AND orig_port <= 514: warezmaster

Modelo generado con el archivo **KDDTrain-20Percent** (NSL KDD database 2012) para detectar el ataque Escaneo de red y puertos.

Class= satan

count > 20 AND count > 300 AND flag = S0 AND count > 329: satan

count > 20 AND count > 300 AND flag = REJ: satan

count > 2 AND src_bytes > 0 AND src_bytes <= 6 AND flag = SF AND dst_bytes <= 130: satan

src_bytes > 0 AND protocol_type = icmp AND service = eco_i: satan

count > 14 AND duration > 1: satan

flag = SF AND protocol_type = icmp AND src_bytes <= 42: satan

flag = REJ AND service = private: satan

service = other: satan

service = pm_dump: satan

service = auth: satan

service = nntp: satan

service = uucp: satan

src_bytes > 0 AND protocol_type = udp AND src_bytes <= 50: satan

flag = REJ AND service = remote_job: satan

flag = REJ AND service = gopher AND count > 1: satan



flag = REJ AND service = systat AND count <= 99: satan

dst_bytes <= 399 AND service = finger AND count > 1: satan

service = smtp: satan

dst_bytes <= 351 AND duration <= 6: satan

Class= ipsweep

protocol_type = icmp AND src_bytes <= 19: ipsweep

count <= 1 AND src_bytes <= 0 AND flag = REJ AND service = private: ipsweep

flag = RSTO AND service = ftp AND src_bytes <= 38: ipsweep

flag = RSTO AND dst_bytes <= 197: ipsweep

flag = REJ AND service = ftp_data AND count <= 1: ipsweep

flag = REJ AND service = link: ipsweep

flag = REJ AND service = ctf AND count <= 1: ipsweep

flag = REJ AND service = mtp AND count <= 1: ipsweep

flag = REJ AND service = rje AND count <= 17: ipsweep

service = ssh: ipsweep

dst_bytes <= 128 AND count <= 1 AND src_bytes <= 1 AND service = finger: ipsweep

flag = REJ AND service = remote_job AND count <= 1: ipsweep

flag = REJ AND service = name AND count <= 4: ipsweep

flag = REJ AND service = whois: ipsweep

flag = REJ AND service = domain: ipsweep

dst_bytes <= 399 AND src_bytes <= 1 AND service = smtp: ipsweep

dst_bytes <= 399 AND src_bytes <= 1 AND service = gopher: ipsweep

dst_bytes <= 152 AND count <= 1 AND src_bytes <= 1 AND duration <= 6: ipsweep

Class= portsweep

count > 20 AND src_bytes <= 0 AND duration > 0 AND flag = RSTR: portsweep



count > 24 AND src_bytes <= 0 AND flag = RSTOS0: portsweep

count > 14 AND flag = RSTO AND dst_bytes <= 14: portsweep

flag = RSTR AND src_bytes <= 1: portsweep

flag = OTH AND service = private: portsweep

flag = RSTOS0: portsweep

flag = S0 AND land <= 0 AND service = supdup AND count > 9: portsweep

flag = RSTO AND dst_bytes <= 1: portsweep

flag = REJ AND count <= 2 AND service = private: portsweep

service = mtp: portsweep

service = echo: portsweep

service = hostnames: portsweep

service = netstat: portsweep

service = supdup: portsweep

service = uucp_path: portsweep

count > 1 AND flag = REJ AND service = whois: portsweep

flag = REJ AND service = name: portsweep

flag = REJ: portsweep



5.10.2 Aplicación de los modelos sobre los datos capturados

Para la aplicación de los modelos seleccionados sobre los datos capturados se realizó el siguiente proceso en Rapid Miner:

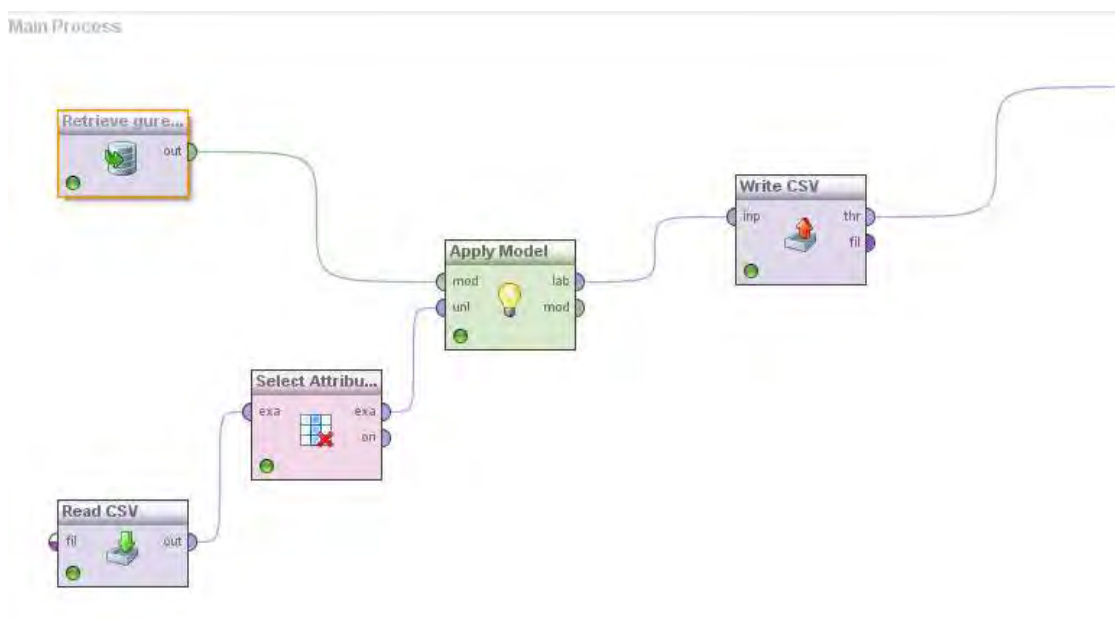
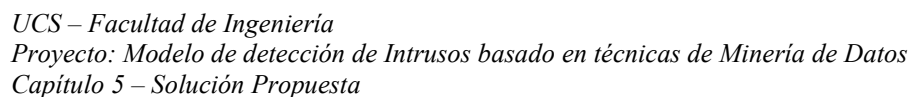


Imagen 52 – Proceso de clasificación de la captura

En donde con el operador Read CSV cargamos cada archivo generado con la captura del tráfico de red simulado, con el operador Select Attributes seleccionamos los mismos atributos seleccionados para crear los modelos de clasificación, con el operador Retrieve cargamos cada uno de los modelos de clasificación obtenidos previamente, con operador Apply model aplicamos el modelo de clasificación a los datos no etiquetados (capturados), este operador agrega una columna por cada valor de tipo de ataque con el título de **Confidence** (“tipo de ataque”) (por ej. Confidence (“portsweep”)), donde para cada registro determina un porcentaje de probabilidad de ocurrencia para cada tipo de ataque, y además, agrega una columna que indica el valor de tipo de ataque predicho para cada registro, con el título de **Prediction (class)**. Por ultimo con el operador Write CSV exportamos cada registro clasificado (con la clase asignada) y con el valor de confianza.

Para el ataque de Hydra-Fuerza Bruta a servidor FTP se utilizó el archivo capturado “Anormal_dict_ftp.csv”, el cual se compone de 160 registros. A la hora de clasificarlo con el modelo generado con el archivo de la GUREKDD database 2008, se obtuvieron los siguientes valores: “Normal” y “Dict”.



BN	BO	BP	BQ	BR	BS	BT
confidence(perl_clea	confidence(sp	confidence(eject-fa	confidence(ware	confidence(dict_simpl	confidence(load_clea	prediction(clas
	0	0	0	0	A ↓	Ordenar de A a Z
	0	0	0	0	Z ↓	Ordenar de Z a A
	0	0	0	0		Ordenar por color
	0	0	0	0	X	Borrar filtro de prediction(class)
	0	0	0	0		Filtrar por color
	0	0	0	0		Filtros de texto
	0	0	0	0.25		Buscar
	0	0	0	0	0.12	☒ (Seleccionar todo)
	0	0	0	0	0.12	☒ dict
	0	0	0	0	0.12	☒ normal
	0	0	0	0	0.12	
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	0	0.25		
	0	0	6.37E-04	8.77E-04	0	0 normal
	0	0	0.008155236	8.77E-04	0	0 normal
	0	0	0.008155236	8.77E-04	0	0 normal
	0	0	0.008155236	8.77E-04	0	0 normal

Donde 6 de los 160 registros son “Dict” y el resto “Normal”, con esta clasificación se pudo detectar el ataque en cuestión ya que el tipo de ataque “Dict”, que se encuentra en la categoría R2L, representa el uso de un diccionario de datos para el acceso forzado a una cuenta, obteniendo el nombre de usuario y la contraseña. Los resultados obtenidos concuerdan con la forma en que se puede llevar a cabo ese tipo de ataque, ya que cada registro del archivo clasificado como “Dict” hace referencia a una conexión realizada por el atacante para obtener los datos de ingreso.

Para el ataque de Denegación de Servicio se utilizó el archivo capturado “**AnormalDoS_slowloris.csv**” el cual se compone de 3357 registros. Dicho archivo se clasificó con el modelo creado con la GUREKDD database 2008, con el cual se obtuvieron los siguientes valores: “Teardrop” y “Multihop”.

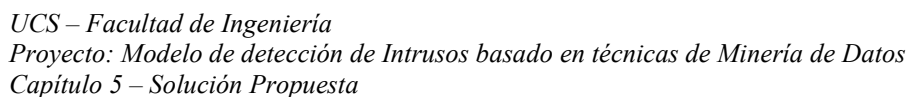
[illegible]

Imagen 54 – Clasificación de AnormalDoS slowloris.csv con la GUREKDD 2008

En esta clasificación, 3338 registros se clasificaron como “Teardrop” y 19 registros como “Multihop”, de esta manera se pudo detectar el ataque en cuestión ya que el tipo de ataque Teardrop (que pertenece a la categoría de DOS) envía paquetes corruptos hacia una víctima (ver descripción del tipo de ataque en el marco teórico), y el tipo de ataque “Multihop”, si bien no pertenece a la categoría de DOS (pertenece a la categoría R2L), este ataque permite detectar un envío de paquetes anormales en una red de datos.

Para detectar el ataque de Escaneo de red y puertos se utilizó el archivo capturado “**Anormal_probing.csv**”, el cual se compone de 98 registros, y se empleó el modelo de clasificación obtenido con la NSL KDD database 2012, ya que el archivo de la NSL KDD contiene tipos de ataques relacionados con el ataque mencionado, y no así el archivo de GURE KDD. Al realizar la clasificación se obtuvieron los siguientes valores: “Neptune”, “Ipsweep”, “Satan”, “Rootkit”, “Portssweep” y “Normal”.

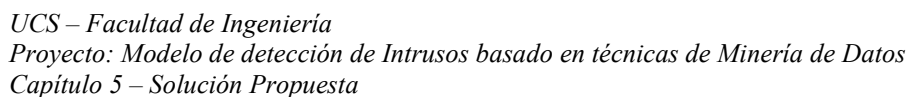


Imagen 55 - Clasificación de Anormal probing.csv con la NSLKDD 2012

Para la detección del ataque warezclient/warezmaster se utilizó el archivo capturado “**Anormal_warez_ftp.csv**”, el cual cuenta con 217 registros, y se empleó el modelo de clasificación creado a partir de la GUREKDD database 2008. A la hora de clasificar dicho archivo se obtuvieron los siguientes valores de tipos de ataques: “Warezclient”, “Warezmaster” y “Normal”.

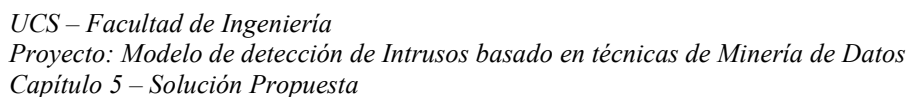
[illegible]

Imagen 56 - Clasificación de Anormal warez ftp.csv con la GUREKDD 2012

De los 217 registros, 214 fueron clasificados como “Normal”, 2 como “Warezclicnt” y 1 como “Warezmaste”, para la deteccion del ataque en cuestion se utilizaron los registros obtenidos clasificados como “Warezclicnt” y “Warezmaste”, ya que tales tipos de ataques pertenecen a la catagoria R2L que consiste en el acceso no autorizado a una maquina remota. Cada registro utilizado para la deteccion del ataque pertenece a una conexi3n realizada por parte del atacante, en este caso “Warezclicnt” representa una conexi3n para la descargar de software malicioso (2 descargas), mientras que “Warezmaste” representa una conexi3n para la carga de software malicioso (1 carga).

5.11 Evaluación

Una vez clasificados los archivos de la captura del tráfico de red simulado, se procedió con la creación de las reglas de detección de intrusos, para ello es importante aclarar que dichas reglas no son las mismas que las reglas generadas en los modelos de clasificación (generadas por los algoritmos Part y Random Forest), sino que las reglas de los modelos de clasificación se utilizaron únicamente para clasificar los archivos capturados, y recién a partir de dichos archivos, con cada uno de sus registros clasificados, utilizar los valores de los atributos significativos que tienen relación con los parámetros y el formatos de las reglas aceptadas por el Servidor IDS. Para ello, se describirá dicho formato y sus parametros para posteriormente crear las reglas de detección de intrusos.

5.11.1 Generación de reglas para el Servidor IDS

Para poder generar las reglas de detección de intrusos e incorporarlas en el motor de base de datos del Servidor IDS, se siguió el formato de regla definido en el Servidor IDS, específicamente en la herramienta Suricata. El formato que deben tener las reglas del IDS es:

- Deben ser escritas en una sola línea



- Se las divide en dos secciones:
 - La **cabecera** que contiene: Acción, Protocolos involucrados, direcciones IP de origen y destino, números de puertos y dirección de la operación.
 - Las **opciones** que contienen: Mensajes y opciones de decisión.

Las opciones de una regla contienen la información necesaria para tomar una decisión. Existen cuatro tipos de opciones:

- Metadata: Nos proporciona información adicional sobre la regla.
- Payload: Busca patrones dentro de la carga útil del paquete, como por ejemplo los campos de datos de un formulario web.
- Non-Payload: Busca patrones dentro de los demás campos del paquete, como por ejemplo la cabecera.
- Post-detection: Permite activar reglas específicas después de la ejecución de otra.

A la vez cada una de estas opciones contiene distintos parámetros que conforman e identifican cada regla del servidor:

Tabla 3-2. Opciones de las reglas de Snort	
Categoría	Opciones
Metadatos	msg, reference, sid, rev, classtype y priority
Payload	content, nocase, rawbytes, depth, Offset, distance, within, uricontent, isdataat, pcre, byte_test, byte_jump, ftpbounce, regex y content-list.
Non-payload	fragoffset, ttl, tos, id, ipopts, fragbits, dsize, flags, flow, flowbits, seq, ack, windows, iftype, icode, icmp_id, icmp_seq, rpc, ip_proto y sameip
Post-Detection	logto, session, resp, react y tag

Imagen 57 – Cuadro de relación entre opciones y parámetros de las reglas de IDS

(Universidad de Almería, 2016)

La información mostrada en la imagen anterior hace referencia al servidor IDS Snort, ya que Snort y Suricata (el IDS empleado en la solución propuesta) utilizan una misma configuración y formato de regla.

Definición de parámetros

Metadatos:

- *msg*: muestra el mensaje de alerta cuando se activa la regla.
- *reference*: permite referenciar la regla a través de un formato definido o personalizado, como por ejemplo *reference: url,www.securityfocus.com;*



- *sid*: permite identificar la regla con un identificador único.
- *rev*: indica el número de revisión de la regla.
- *priority*: indica la prioridad de la regla.
- *classtype*: permite indicar la categoría de la regla, por ejemplo exploits o virus.

Payload, los más importantes son:

- *content*: especifica el contenido (string) que debe detectar la regla de la parte útil de un paquete, por ejemplo: *content*: "facebook"; (en ascii), *content*="|00 23 71 88|"; (en hexadecimal).
- *nocase*: se utiliza para no distinguir entre mayúsculas y minúsculas en el string, por ejemplo: *content*: "Facebook"; *nocase*.
- *depth*: indica que el contenido se debe encontrar en los primeros *X* bytes, por ejemplo: *content*: "GET"; *depth* 10;
- *offset*: Indica que el contenido debe estar a partir de los *X* primeros bytes, por ejemplo: *content*: "attack code"; *offset* 50;
- *within*: Indica que la regla tiene dos contenidos que se encuentran separados a una cierta distancia: por ejemplo: *content*: "hack"; *content*: "Windows"; *within* 20;.
- *distance*: Indica que el contenido debe estar a partir de los *X* bytes desde la última coincidencia, por ejemplo: *content*: "exploit"; "root"; *distance*: 10;
- *pcre*: indica el contenido que se debe buscar dentro de un paquete. A diferencia de *content*, *pcre* busca los datos en una línea y *content* afecta todo el paquete.

Non-payload:

- *fragoffset*: Indica la posición que ocupa el paquete actual dentro del datagrama original de forma que el destino pueda reconstruirlo.
- *ttl*: Hace referencia al tiempo de vida de un paquete
- *tos*: Comprueba el valor del campo TOS de la cabecera IP que corresponde al tipo de servicio.
- *id*: comprueba el valor del datagrama IP.
- *ipopts*: Se utiliza para comprobar el campo opción de la cabecera del datagrama IP.
- *fragbits*: Permite corroborar el bit de fragmentación del datagrama IP.
- *dsize*: Se utiliza para corroborar el tamaño del payload.
- *frag*: Se utiliza para comprobar si se encuentra activa alguna opción del frag TCP.
- *flow*: permite indicar la dirección del flujo del tráfico, por ejemplo: *flow*:to_server, established;
- *flowbits*: Se utiliza para seguir los estados de las sesiones del protocolo de transporte.
- *seq*: Se utiliza para identificar el número de secuencia TCP.



- *ack*: Se utiliza para identificar el valor ACK de un paquete TCP.
- *windows*: Se utiliza para identificar un determinado tamaño de la ventana TCP.
- *itype*: Se utiliza para identificar el tipo de mensaje ICMP.
- *icode*: Se utiliza para identificar el valor del código ICMP.
- *icmp*: Se utiliza para identificar el identificador ICMP.
- *icmp_seq*: Se utiliza para identificar una determinada secuencia de paquetes ICMP.
- *rpc*: Se utiliza para identificar una determinada aplicación RCP.
- *ip_proto*: Permite identificar el protocolo de la cabecera del mensaje IP.
- *same_ip*: Permite indicar que utiliza la misma dirección de y de destino.

Post-detection

- *logto*: Permite guardar el paquete en un determinado archivo.
- *sesión*: Permite extraer datos de usuario de las sesiones TCP
- *resp*: Permite realizar una respuesta activa aun paquete.
- *react*: Permite reaccionar ante un determinado paquete, por ejemplo, bloquear un paquete o generar un aviso.
- *tag*: Permite registrar un determinado número de paquetes después de que se active una regla.

(Universidad de Almería, 2016)

A la vez existe otros parámetros que no forman partes de las opciones mencionadas, sino que representan otras características no relacionadas con el contenido de las conexiones, que son calculadas en el momento de la detección, como ser: count (cantidad de conexiones de una dirección IP a otra) y second (duración de la conexión).

Para poder entender mejor el formato de regla y utilización de los parámetros, a continuación, se muestra como ejemplo una regla ya existente en el servidor IDS Suricata:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 143 (msg:"GPL IMAP Overflow Attempt";  
flow:to_server, established; content:"|E8 C0 FF FF FF|/bin/sh"; classtype: attempted-admin;  
sid:2100293; rev:8;)
```

Cada regla empieza con una palabra reservada que indica la acción a tomar al momento de activarse, para todas las reglas a implementar en el Servidor IDS la palabra reservada será “alert” para producir una alerta de un posible ataque, seguidamente en el encabezado se especifican: el tipo de protocolo (en el ejemplo TCP), la dirección IP de origen (en el ejemplo \$EXTERNAL_NET, cualquier dirección IP que no pertenecen a la red), el puerto de origen (en el ejemplo any, cualquier puerto), la dirección IP de destino (en el ejemplo \$HOME_NET, cualquier dirección IP de la red) y el puerto de destino (en el ejemplo puerto 143); a continuación se establecen los parámetros antes mencionados, en donde no es necesario utilizar todos en una sola regla sino solamente aquellos que son necesarios y esenciales para el fin de



la detección. En el ejemplo se utilizan msg, flow, content, classtype, sid y rev (el orden de los parámetros no afecta el funcionamiento de la regla).

5.11.1.1 Creación de reglas

Una vez determinado el formato de las reglas del Servidor IDS, se procedió con la creación de las mismas a partir de la información obtenida en la clasificación de los archivos de la captura del tráfico de red simulado, para ello se utilizaron los registros filtrados con los valores de tipos de ataques detectados (ver en imágenes 54, 55, 56 y 57), tomando los valores de los atributos significativos (seleccionados anteriormente) para introducirlos como parámetros de las reglas de detección de intrusos.

Para la detección del Ataque de Denegación de Servicio de utilizo en archivo clasificado “AnormalDoS_slowloris.csv”, donde se utilizaron los registros filtrados por el valor de tipos de ataque “Teardrop”.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	orig_port	resp_port	duration	protocol_ty	service	flag	src_bytes	dst_bytes	land	wrong_frag	urgent	count	prediction(Iss)
3	53716	80	0	tcp	http	RSTRH	0	0	1	3	0	0	teardrop
5	37536	80	144.558871	tcp	http	SF	247	0	1	3	0	0	teardrop
8	37537	80	144.559553	tcp	http	SF	247	0	1	3	0	1	teardrop
11	37538	80	144.555	tcp	http	SF	247	0	1	3	0	2	teardrop
14	37539	80	144.550686	tcp	http	SF	247	0	1	3	0	3	teardrop
17	37540	80	144.540315	tcp	http	SF	247	0	1	3	0	4	teardrop
20	37541	80	144.536285	tcp	http	SF	247	0	1	3	0	5	teardrop
23	37542	80	144.531728	tcp	http	SF	247	0	1	3	0	6	teardrop
26	37543	80	144.52611	tcp	http	SF	247	0	1	3	0	7	teardrop
29	37544	80	144.521472	tcp	http	SF	247	0	1	3	0	8	teardrop
32	37545	80	144.516664	tcp	http	SF	247	0	1	3	0	9	teardrop
35	37546	80	144.512297	tcp	http	SF	247	0	1	3	0	10	teardrop
38	37547	80	144.507602	tcp	http	SF	247	0	1	3	0	11	teardrop
41	37548	80	144.503502	tcp	http	SF	247	0	1	3	0	12	teardrop
44	37549	80	144.498886	tcp	http	SF	247	0	1	3	0	13	teardrop
47	37550	80	144.495192	tcp	http	SF	247	0	1	3	0	14	teardrop
50	37551	80	144.490603	tcp	http	SF	247	0	1	3	0	15	teardrop
53	37552	80	144.486294	tcp	http	SF	247	0	1	3	0	16	teardrop
56	37553	80	144.482192	tcp	http	SF	247	0	1	3	0	17	teardrop

Imagen 58 - Valores de atributos significativos para las reglas de detección del Ataque DoS

Los registros clasificados como “Teardrop” fueron 3338, para la creación de las reglas de detección de intrusos se filtró además por el tipo de protocolo (“udp” y “tcp”) donde se obtuvieron 1102 registro con tipo de protocolo “tcp” y 2235 registros con el tipo de protocolo “udp”. Se crearon solamente 2 reglas por cada tipo de protocolo:

#REGLAS Teardrop

```
alert tcp any 38536 -> $LOCAL 80 (msg:"Posible Ataque TCP DoS"; flow: to_server; flags: SF; detection_filter: track by_src, count: 1005, seconds 23; sid:1000003; rev:1; classtype: attempted-dos;)
```



```
alert tcp any 1139-> $LOCAL 80 (msg:"Posible Ataque TCP DoS"; flow: to_server; flags: SF;  
detection_filter: track by_src, count: 995, seconds 1.02; sid:1000004; rev:1; classtype:  
attempted-dos;)
```

```
alert udp any 45428-> $LOCAL 53 (msg:"Posible Ataque TCP DoS"; flow: to_server; flags:  
SF; detection_filter: track by_src, count: 1681, seconds 0.002; sid:1000005; rev:1; classtype:  
attempted-dos;)
```

```
alert udp any 39929-> $LOCAL 53 (msg:"Posible Ataque TCP DoS"; flow: to_server; flags:  
SF; detection_filter: track by_src, count: 1081, seconds 0.003; sid:1000006; rev:1; classtype:  
attempted-dos;)
```

Siendo:

PROTOCOLO_TYPE	ORIG_PORT	RESP_PORT	FLAG	COUNT	DURATION
----------------	-----------	-----------	------	-------	----------

En el campo de opciones de las reglas se utiliza los siguientes parámetros:

msg - para dar un mensaje de alerta.

flow: to_server - se usa en conjunto con los flujos tcp, que indica al IDS la dirección del tráfico que debe detectar, en este caso **to_server**, conexiones dirigidas al servidor en un solo sentido, desde el exterior a la red interna.

flags: SF - tipo de bandera utilizada por la conexión tcp.

detection_filter: track by_src - indica un seguimiento realizado a la dirección IP de origen, haciendo un recuento de la cantidad de conexiones establecidas por la misma IP.

count: 1081 - cantidad límite de conexiones permitidas, después de que se supere ese límite se activa la regla en cuestión, se utiliza en conjunto con **seconds:0.003** para indicar la frecuencia de conexiones por intervalo de tiempo, es decir el límite es de 50 conexiones en menos de 0.003 seg.

sid:1000003 - Identificador unico de la regla de deteccion de DOS

rev:1 - otro tipo de identificador de las reglas de Suricata que, junto con el sid de la regla permite que las firmas y las descripciones pueden ser refinadas y reemplazadas con información actualizada.

classtype: attempted-dos - categoriza una regla de detección de un ataque que es parte de un tipo más general de la clase de ataque (categoría del ataque).

Como puede verse, no todos los atributos significativos fueron colocados como parámetros de las reglas, ya que dichos atributos no forman parte del formato de las reglas de detección de intrusos. Los atributos utilizados fueron: protocol_type, orig_port, resp_port, flag,



count y duration, los demás parámetros fueron definidos a partir de la investigación y prueba de compilación y tomando como ejemplo modelos de reglas generales ya establecidas en motor de base de datos del IDS.

Para la detección del ataque Hydra-Fuerza Bruta con diccionario se utilizó el archivo clasificado “**Anormal_dict_ftp.csv**”, dentro de este archivo se tomaron los registros clasificados como el tipo de ataque “Dict” o “Dictionary” (6 registros). De este modo los valores de los atributos significativos de estos registros son:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	orig_port	resp_port	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	count	prediction(class)
6	50162	21	4.480622	tcp	ftp	RSTO	88	241	0	0	0	0	dict
7	50163	21	7.084944	tcp	ftp	SF	86	188	0	0	0	1	dict
9	50165	21	7.064203	tcp	ftp	SF	84	188	0	0	0	3	dict
10	50166	21	7.101056	tcp	ftp	SF	86	188	0	0	0	4	dict
11	50167	21	7.09823	tcp	ftp	SF	88	188	0	0	0	5	dict
12	50168	21	7.098797	tcp	ftp	SF	85	188	0	0	0	6	dict
162													

Imagen 59 – Valores de atributos significativos para las reglas de detección del Ataque Hydra-Fuerza Bruta

Con estos valores se crearon 6 reglas para la detección del Ataque Hydra-Fuerza Bruta

#REGLAS Fuerza Bruta con Diccionarios

alert **tcp** \$EXTERNAL_NET **50168** -> \$HOME_NET **21** (msg:"FTP Ataque de Fuerza Bruta"; content:"530 Login "; nocase; flags: SF; flow: from_server, established; threshold: type both, track by_dst, **count 6**, **seconds 7.09**; sid:1000016;)

alert **tcp** \$EXTERNAL_NET **50162** -> \$HOME_NET **21** (msg:"FTP Ataque de Fuerza Bruta"; content:"530 Login "; nocase; flags: RSTO; flow: from_server, established; threshold: type both, track by_dst, **count 0**, **seconds 4.48**; sid:1000019;)

alert **tcp** \$EXTERNAL_NET **50163** -> \$HOME_NET **21** (msg:"FTP Ataque de Fuerza Bruta"; content:"530 Login "; nocase; flags: SF; flow: from_server, established; threshold: type both, track by_dst, **count 1**, **seconds 7.08**; sid: 1000021;)

alert **tcp** \$EXTERNAL_NET **50165** -> \$HOME_NET **21** (msg:"FTP Ataque de Fuerza Bruta"; content:"530 Login "; nocase; flags: SF; flow: from_server, established; threshold: type both, track by_dst, **count 3**, **seconds 7.06**; sid: 1000022;)

alert **tcp** \$EXTERNAL_NET **50166** -> \$HOME_NET **21** (msg:"FTP Ataque de Fuerza Bruta"; content:"530 Login "; nocase; flags: SF; flow: from_server, established; threshold: type both, track by_dst, **count 4**, **seconds 7.10**; sid: 1000023;)

alert **tcp** \$EXTERNAL_NET **50167** -> \$HOME_NET **21** (msg:"FTP Ataque de Fuerza Bruta"; content:"530 Login "; nocase; flags: SF; flow: from_server, established; threshold: type both, track by_dst, **count 5**, **seconds 7.09**; sid: 1000024;)



Para la detección del Ataque Warezclient/Warezmaster se utilizó el archivo clasificado “Anormal_warez_ftp.csv”, donde se utilizaron los registros filtrados con los valores de tipos de ataques “Warezmaster” y “Warezclient” (3 registros en total).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	orig_port	resp_port	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	count	prediction(class)
47	20	58067	0.002942	tcp	private	SF	346	0	0	0	0	0	warezclient
78	20	51327	0.020411	tcp	private	SF	0	12358	0	0	0	3	warezclient
134	20	1176	0.132164	tcp	other	SF	0	1078591	0	0	0	27	warezmaster
219													

Imagen 60 - Valores de atributos significativos para las reglas de detección del Ataque Warezmaster/Warezclient

De este modo se crearon 3 reglas para la detección del Ataque Warezclient/Warezmaster

#REGLAS Warezclient

```
alert tcp $EXTERNAL_NET 20 -> $HOME_NET 51327 (msg:"FTP WarezClient";  
content:"RETR "; nocase; content:"file_id.diz"; nocase; flags:Sf; flow:to_server,established;  
threshold: type both, track by_dst, count 3, seconds 0.020; classtype:suspicious-filename-  
detect; sid:1000017; rev:5;)
```

```
alert tcp $EXTERNAL_NET 20 -> $HOME_NET 58067 (msg:"FTP WarezClient";  
content:"RETR "; nocase; content:"file_id.diz"; nocase; flags:Sf; flow:to_server,established;  
threshold: type both, track by_dst, count: 0, seconds 0.002; classtype:suspicious-filename-  
detect; sid:1000025; rev:5;)
```

#REGLA Warezmaster

```
alert tcp $EXTERNAL_NET 20 -> $HOME_NET 1176 (msg:"FTP WarezMaster";  
content:"STOR"; nocase; content:" MB"; distance:1; nocase; flags:Sf;  
flow:to_server,established; threshold: type both, track by_dst, count 27, seconds 0.13;  
classtype:misc-activity; sid:1000018; rev:6;)
```

content - Permite al usuario establecer reglas de búsqueda de contenido específico en la carga útil del paquete. Si los datos que coinciden exactamente con la cadena de datos del argumento están contenidos en cualquier punto de la carga útil del paquete, la prueba es satisfactoria y se llevan a cabo el resto de las pruebas de opción regla. En el caso del ataque Warez Client se busca dentro de la carga útil del paquete la palabra RETR, referente a uno de los comandos del protocolo de transferencia de archivos (FTP), el cual pide al servidor de transferencia de datos una copia del archivo cuya ruta de acceso se da en los parámetros, es decir la descarga de un archivo.

Nocase – Especifica que se debe ignorar mayúsculas o minúsculas.



Por ultimo para la detección del Ataque Escaneo de red y puertos, se utilizó el archivo clasificado “**Anormal_probing.csv**”, de donde se tomaron los valores de los registros filtrados por los valores de tipos de ataques: “Satan” (13 registros), “Ipsweep” (5 registros) y “PortswEEP” (8 registros).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	orig_port	resp_port	duration	protocol	service	flag	src_bytes	dst_bytes	land	wrong_fr	urgent	count	prediction
15	61636	5355	0	udp	other	S0	0	0	0	0	0	0	0 ipsweep
40	44547	80	0	tcp	http	S0	0	0	0	0	0	0	0 ipsweep
41	3	10	0	icmp	urp	SH	0	0	0	0	0	0	0 satan
51	3	10	0.042093	icmp	urp	SH	0	0	0	0	0	0	1 ipsweep
52	3	10	0.042156	icmp	urp	SH	0	0	0	0	0	0	2 ipsweep
53	3	10	0.042377	icmp	urp	SH	0	0	0	0	0	0	3 satan
54	3	10	0.042395	icmp	urp	SH	0	0	0	0	0	0	4 satan
55	3	10	0.042402	icmp	urp	SH	0	0	0	0	0	0	5 satan
57	3	10	1.126598	icmp	urp	SH	0	0	0	0	0	0	6 satan
69	139	2097	0	tcp	other	OTH	0	0	0	0	0	0	1 portswEEP
75	39265	22	6.12E-04	tcp	ssh	REJ	0	0	0	0	0	0	4 ipsweep
76	39265	110	7.16E-04	tcp	pop_3	REJ	0	0	0	0	0	0	5 portswEEP
77	39265	21	6.36E-04	tcp	ftp	REJ	0	0	0	0	0	0	6 portswEEP
78	39265	3389	5.89E-04	tcp	other	REJ	0	0	0	0	0	0	7 portswEEP
79	39265	445	9.16E-04	tcp	other	RST	0	0	0	0	0	0	8 portswEEP
80	39265	80	6.30E-04	tcp	http	REJ	0	0	0	0	0	0	9 portswEEP
81	39265	443	7.48E-04	tcp	http_443	REJ	0	0	0	0	0	0	10 portswEEP
82	39265	23	7.00E-04	tcp	telnet	REJ	0	0	0	0	0	0	11 portswEEP
83	3	1	0	icmp	urp	SH	0	0	0	0	0	0	12 satan
84	3	1	1.494908	icmp	urp	SH	0	0	0	0	0	0	13 satan
85	3	1	3.002844	icmp	urp	SH	0	0	0	0	0	0	14 satan
89	50994	1900	1.001804	udp	snmp	S0	0	0	0	0	0	0	1 satan
90	50994	1900	2.00181	udp	snmp	S0	0	0	0	0	0	0	2 satan
92	50994	1900	3.025207	udp	snmp	S0	0	0	0	0	0	0	3 satan

Imagen 61 - Valores de atributos significativos para las reglas de detección del Ataque Escaneo de red y puertos

Se crearon 2 reglas para el tipo de “Satan”, 1 regla para el tipo de ataque “PortswEEP” y 3 reglas para el tipo de ataque “Ipsweep”.

#REGLAS Satan

alert icmp \$EXTERNAL_NET 3 -> \$HOME_NET 10 (msg:"Posible Satan"; dsizE:0; itype:8; flags:SA; flow:stateless; reference:arachnids,162; classtype:attempted-recon; sid:2100469; rev:4;)

Donde SA=SH

alert udp \$EXTERNAL_NET 50994 -> \$HOME_NET 1900 (msg:"Posible Satan"; dsizE:0; flags: S0; classtype: attempted-recon; sid:1000010; rev:3;)

dsizE - se utiliza para probar el tamaño del paquete de carga útil. Esto se puede utilizar para comprobar si hay paquetes de tamaño anormal que podría causar desbordamientos de búfer.

Itype - se utiliza para comprobar si hay un valor específico tipo ICMP.

Reference - La palabra clave de referencia permite que las reglas incluyan referencias a los sistemas de identificación de ataques externos.



#REGLA PortswEEP

```
alert tcp $EXTERNAL_NET 39265 -> $HOME_NET 445 (msg: "Posible portswEEP"; flags:R;  
threshold: type both,track by_dst count 8, seconds 0.0001; sid:1000009;)
```

Donde R=RST

#REGLA ipsweep

```
alert udp $EXTERNAL_NET 61636 -> $HOME_NET 5355 (flags:S0; msg: "Posible  
ipsweep"; sid:1000013;rev:1; classtype: network-scan;)
```

```
alert tcp $EXTERNAL_NET 44547 -> $HOME_NET 80 (flags:S0; msg: "Posible ipsweep";  
sid:1000012;rev:1; classtype: network-scan;)
```

```
alert icmp $EXTERNAL_NET 3 -> $HOME_NET 10 (msg: "Posible ipsweep"; flags:S0;  
threshold: type both,track by_dst, count 2, seconds 0.0001; sid:1000011;rev:1)
```

Una vez creadas todas las reglas de detección de intrusos, las mismas son insertadas en el motor de base de datos del Servidor IDS.

5.12 Implementación del IDS

Para llevar a cabo la implementación del IDS con las reglas de detección de intrusos creadas, se procedió con la instalación de Suricata IDS en la máquina virtual de CentOS que actúa como servidor y aloja el servidor web de Apache2 (Ver Anexo III).

Una vez instalado y configurado el sistema de detección de intrusos, se pasó a implementarlo en nuestra red de datos simulada en donde las reglas creadas se cargaron en la carpeta **etc/suricata/rules** creando un nuevo archivo con la extensión .rules (por ej. nuevasreglas.rules). A la vez también se debe agregar el nombre del nuevo archivo rules en la sección **rule-files** del archivo suricata.yaml ubicado en **etc/suricata**, como así también los identificadores de cada regla creada (parámetro sid) deben agregarse en el archivo de identificadores sid_msg.map ubicado en **etc/suricata**.

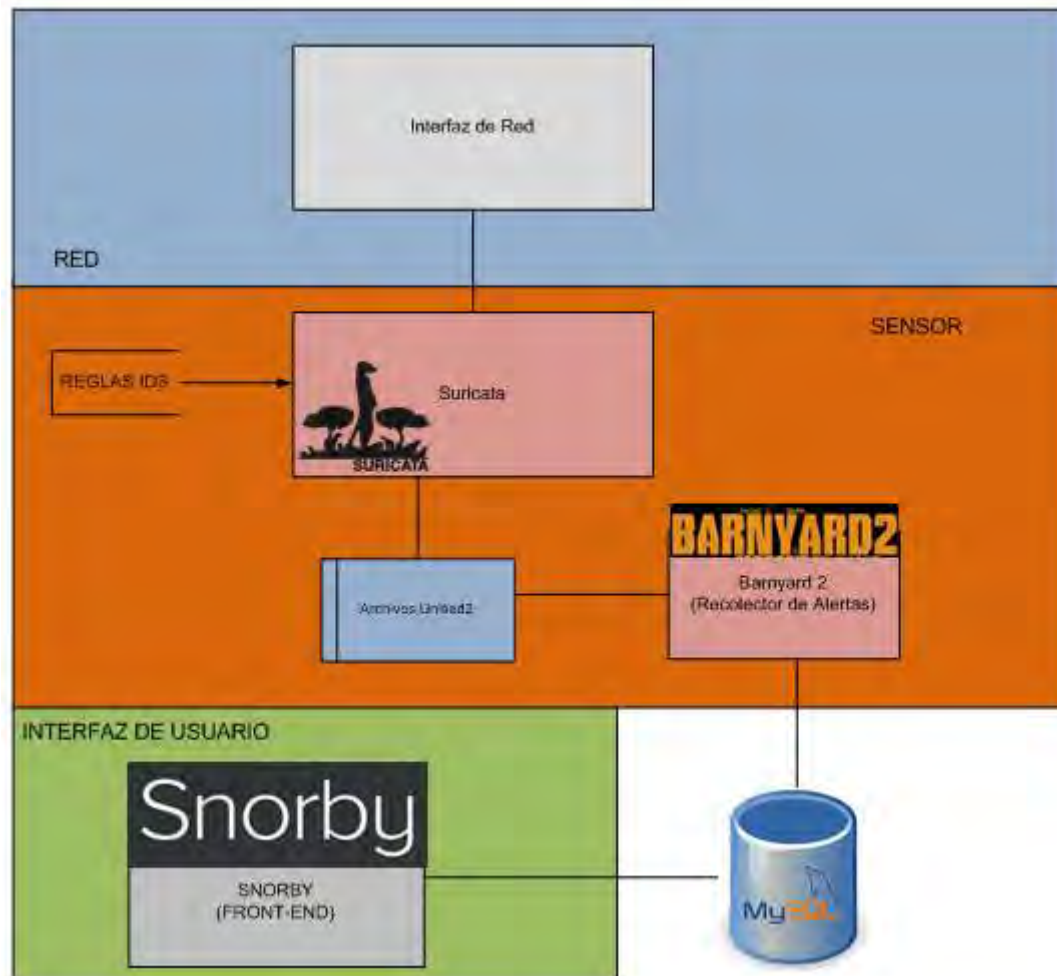


Imagen 62 – Mecanismo IDS

Los componentes del mecanismo del IDS consiste en:

- **Interfaz de Red:** de la cual se vale el IDS Suricata para monitorear y detectar amenazas que puedan producirse en la red local.
- **Suricata:** el IDS que genera las alertas necesarias sobre intrusiones en la red local. Se vale de las reglas generadas para detectar diferentes tipos de ataques. La salida del IDS genera archivos denominados Unifed2 que son utilizados por otro componente del mecanismo en cuestión.
- **Barnyard2:** El recolector de alertas generadas por Suricata, cuya función consiste en interpretar los archivos de salida binaria “Unifed2” y permitir almacenarlos de manera eficiente en diferentes formatos para su posterior análisis.
- **MySQL:** La base de datos que almacena los registros de alertas generados por Barnyard2 para su posterior análisis.
- **Snorby:** Un Front-End para el análisis y gestión de alertas generadas por el IDS Suricata, basado en sensores. Posee una interfaz gráfica sencilla, amplia e intuitiva para la visualización de las alertas y su análisis.



El funcionamiento del mecanismo IDS consiste en los siguientes pasos:

Primeramente, configurar la Interfaz de red del CentOS para que trabaje en modo Promiscuo, es decir que permita escuchar todo el tráfico de la red. Esto se realiza mediante el siguiente comando:

Ifconfig eth0 promisc

También se debe desactivar las opciones GRO (generic-receive-offload) y LRO (large-receive-offload) de la interfaz de red del CentOS para que no haya inconvenientes en el momento de la captura con el IDS Suricata, esto se realiza mediante los siguientes comandos:

Ethool –K eth0 gro off

Ethool –K eth0 lro off

El siguiente paso es activar el Front-End Snorby mediante los siguientes pasos:

cd /var/www/html/snorby (nos posicionamos en el directorio de instalación de snorby)

sudo RAILS_ENV=production script/rails r “Snorby: Worker.start” (inicia los procesos de snorby)

bundle exec rails server –e production (inicia snorby en el host local)

```
root@localhost:~# cd /var/www/html/snorby
root@localhost:~# sudo RAILS_ENV=production script/rails r "Snorby::Worker.start"
syck has been removed, psych is used instead
No time zone specified in snorby_config.yml; detected time_zone: America/Buenos_Aires
syck has been removed, psych is used instead
[root@localhost snorby]# bundle exec rails server -e production
syck has been removed, psych is used instead
No time zone specified in snorby_config.yml; detected time_zone: America/Buenos_Aires
=> 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
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
```

Imagen 63 - Snorby

La página del Dashboard se aloja en la dirección IP del CentOS (host local) en el puerto 3000, es decir en **localhost:3000**. Al entrar nos pedirá ingresar un usuario y contraseña. Con esto se puede visualizar de una forma gráfica y entendible las alertas generadas por el IDS, gestionar y llevar a cabo un análisis posterior de las mismas.

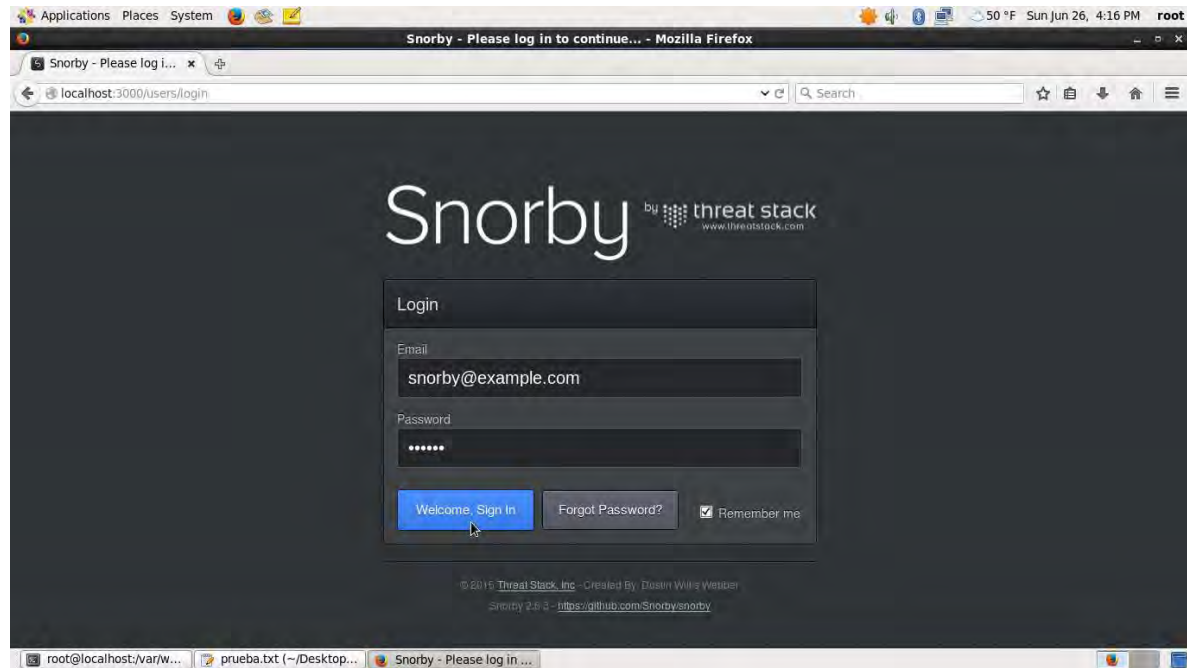


Imagen 64 – Pagina de Login de Snorby

El usuario predeterminado para iniciar Snorby es: snorby@example.com, y la contraseña es: snorby. En la pantalla principal se puede ver el Dashboard, donde se presentan una serie de opciones para analizar la información detectada y generada por el servidor IDS, las cuales se mostrarán en detalle en el análisis de resultados (Capítulo 7):

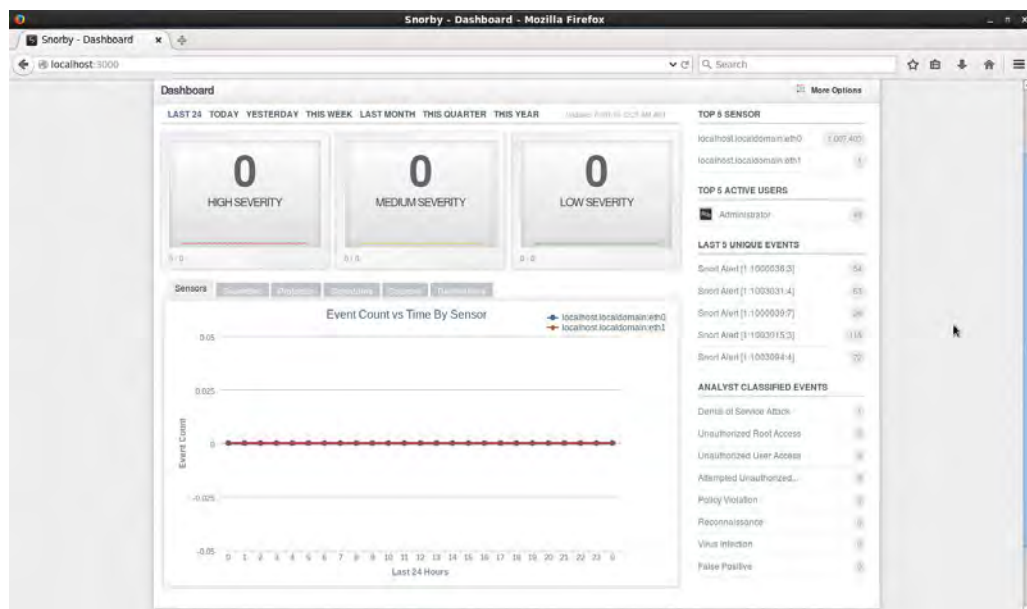


Imagen 65 – Pagina Dashboard Snorby

En el siguiente paso iniciamos el servicio de mysql que permitirá guardar los registros de alertas (Unified2.alert) que generará el IDS Suricata mediante el comando:



Service mysqld start

Después iniciamos Barnyard2 en cuestión, que permitirá recolectar las alertas mencionadas anteriormente, esto lo llevamos a cabo mediante el siguiente comando:

Barnyard2 -c /etc/suricata/barnyard2.conf -d /var/log/suricata -f unified2.alert -w /var/log/suricata/barnyard2.waldo

Donde:

- c indica el uso del archivo de configuración
- d especifica la ubicación del almacenamiento de logs del IDS suricata
- f especifica el nombre del archivo donde se guardarán las alertas generadas por el IDS
- w permite el uso de marcadores, especifica un fichero de chequeo (barnyard2.waldo) de las alertas que se insertan en la base de datos mysql.



Imagen 66 – Barnyard2

Barnyard2 irá guardando todas las alertas que genere Suricata en la base de datos MySQL para el posterior análisis mediante Snorby.

Por último se pone en marcha el IDS mediante terminal, ejecutando el comando:

Suricata -c /etc/suricata/suricata.yaml -i eth0 -v

Donde:

- c indica ruta y archivo de configuración
- i especifica la interfaz de red a utilizar
- v aumenta la información de las alertas detectadas por el IDS suricata



```
Applications Places System root@localhost:~
File Edit View Search Terminal Help

[root@localhost ~]# suricata -c /etc/suricata/suricata.yaml -i eth0 -v
26/6/2016 -- 13:50:57 -- <Notice> - This is Suricata version 2.0.5 RELEASE
26/6/2016 -- 13:50:57 -- <Info> - CPUs/cores online: 1
26/6/2016 -- 13:50:57 -- <Info> - 'default' server has 'request-body-minimal-inspect-size' set to 33882 and 'request-body-inspect-window' set to 4053 after randomization
26/6/2016 -- 13:50:57 -- <Info> - 'default' server has 'response-body-minimal-inspect-size' set to 33695 and 'response-body-inspect-window' set to 4218 after randomization
26/6/2016 -- 13:50:57 -- <Info> - DNS request flood protection level: 500
26/6/2016 -- 13:50:57 -- <Info> - DNS per flow memcap (state-memcap): 524288
26/6/2016 -- 13:50:57 -- <Info> - DNS global memcap: 16777216
26/6/2016 -- 13:50:57 -- <Info> - Found an MTU of 1500 for 'eth0'
26/6/2016 -- 13:50:57 -- <Info> - allocated 2697152 bytes of memory for the defrag hash... 65536 buckets of size 32
26/6/2016 -- 13:50:57 -- <Info> - preallocated 65535 defrag trackers of size 116
26/6/2016 -- 13:50:57 -- <Info> - defrag memory usage: 9699212 bytes, maximum: 33554432
26/6/2016 -- 13:50:57 -- <Info> - AutoPP mode using default 'Active Packets' flow load balancer
26/6/2016 -- 13:50:57 -- <Info> - preallocated 1024 packets, Total memory 2789376
26/6/2016 -- 13:50:57 -- <Info> - allocated 262144 bytes of memory for the host hash... 4096 buckets of size 64
26/6/2016 -- 13:50:57 -- <Info> - preallocated 1000 hosts of size 72
26/6/2016 -- 13:50:57 -- <Info> - host memory usage: 342144 bytes, maximum: 16777216
26/6/2016 -- 13:50:57 -- <Info> - allocated 4194304 bytes of memory for the flow hash... 65536 buckets of size 64
26/6/2016 -- 13:50:57 -- <Info> - preallocated 10000 flows of size 188
26/6/2016 -- 13:50:57 -- <Info> - flow memory usage: 6114304 bytes, maximum: 134217728
26/6/2016 -- 13:50:57 -- <Info> - IP reputation disabled
26/6/2016 -- 13:50:57 -- <Info> - using magic-file /usr/share/file/magic
26/6/2016 -- 13:50:57 -- <Info> - Delayed detect disabled
26/6/2016 -- 13:51:06 -- <Info> - 49 rule files processed, 17784 rules successfully loaded, 0 rules failed
26/6/2016 -- 13:51:06 -- <Info> - 17792 signatures processed, 1041 are IP-only rules, 6813 are inspecting packet payload, 13235 inspect application layer, 72 are decoder
event only
```

Imagen 67 – Suricata IDS en funcionamiento.

Tras iniciarlo se compilan las reglas generadas y se cargan al motor de base de datos del IDS, en caso de que se produjese algún error en la compilación de las reglas, Suricata enviara un mensaje de error indicando el código del mismo, la ubicación de la regla y la línea en que se produjo el inconveniente. Si la compilación de las reglas y la carga de la misma se realizaron de manera exitosa el IDS se pondrá en funcionamiento.

Con esto tenemos el mecanismo de IDS listo para empezar con el monitoreo y detección de ataques que se produzcan en la red de datos.



Capítulo 6 – Análisis y Factibilidad del Proyecto

6.1 Introducción

Este capítulo contiene los distintos análisis requeridos para determinar los riesgos del proyecto y los planes de respuestas ante los mismos, como así también las factibilidades del trabajo elaborado.

6.2 Análisis FODA

	FORTALEZAS	DEBILIDADES
	1. 2. 3. 4.	1. 2. 3. 4.
OPORTUNIDADES	Estrategia de Oportunidades-Fortalezas (OF)	Estrategia Oportunidades-Debilidades (OD)
1.	Usar fortalezas para obtener ventaja de las oportunidades	Superar las debilidades aprovechando las oportunidades
2.		
3.		
4.	1. 2.	1. 2.
AMENAZAS	Estrategia de Riesgos-Fortalezas (RF)	Estrategia de Amenazas-Debilidades (AD)
1.	Utilizar las fortalezas para evitar las amenazas	Minimizar las debilidades y evitar las amenazas
2.		
3.	1.	
4.	2.	1. 2.

Imagen 68 - Matriz FODA

(Renault, 2016)

A continuación, se llevará a cabo un análisis FODA de solución propuesta.

6.2.1 Fortalezas

- El modelo propuesto es independiente del contexto de desarrollo, pudiendo llevarse a cabo e implementarse en cualquier ambiente de una red de datos.
- El modelo propuesto es independiente de la infraestructura de red de datos donde se desee implementar.
- El modelo propuesto se adapta y modela las reglas y políticas de negocio de la organización donde se quiera implementar.



- Disponibilidad de las herramientas necesarias (Software Libre).
- Modelo fácil de configurar y operar una vez implementado.

6.2.2 Oportunidades

- El IDS propuesto trabaja con un proceso de minería de datos, lo cual proporciona un trabajo objetivo y eficiente en la detección de ataques e intrusos.
- El modelo IDS propuesto minimiza la vulnerabilidad ante ataques e intromisión de las redes de datos que lo implementen.
- Reutilización de equipos (hardware) de la organización que lo implemente.
- Ahorro de costos de implementación de IDS.

6.2.3 Debilidades

- No existe una actualización automática de las nuevas amenazas para las reglas de detección.
- Poca información sobre las características y atributos significativos de los diversos ataques o intromisiones informáticas a nivel de red (Paquete de datos).
- Escasa infraestructura informática.
- Poca experiencia en la aplicación de las tecnologías a implementar en el modelo (minería de datos y seguridad de redes).

6.2.4 Amenazas

- Incompatibilidad en la configuración de las herramientas de software y hardware, a la hora de reemplazo por imprevistos.
- Creación de modelos de IDS con licencia paga de uso similar al propuesto.
- Pérdida de los datos monitoreados.
- Hardware (equipos informáticos y de conectividad) obsoleto con el tiempo.

6.2.5 Estrategias de Oportunidades - Fortalezas:

- Debido a la rápida adaptación del modelo a la de red de datos, puede provocar una gran aceptación por parte de diferentes empresas para su implementación.
- Gracias a la disponibilidad de las herramientas de software libre necesarias para desarrollar el modelo propuesto, puede provocar un ahorro en los costos de implementación y una gran escalabilidad en el proyecto de las empresas que deseen poner en marcha el IDS propuesto.
- Debido a que el modelo es sencillo de operar una vez instalado, es posible captar la atención de los clientes que deseen implementar el modelo propuesto y generar conciencia en cuanto a seguridad de los datos sensibles de las empresas y la necesidad de adquirir soluciones que minimicen estos riesgos.



6.2.6 Estrategias de Oportunidades - Debilidades:

- Debido a la posibilidad de reutilización de equipos que posea la organización, no es necesaria la compra de infraestructura informática que impliquen gastos en la empresa.
- Debido a los bajos costos de implementación del modelo de IDS, puede destinarse los fondos para capacitar a personal de la empresa en el uso de tecnologías de minería de datos y seguridad de redes.
- Teniendo en cuenta de que el modelo de IDS proporciona información objetiva y eficiente en la detección de intrusos, entonces se podrá obtener información más precisa sobre el comportamiento o características de los ataques, lo que conlleva a elaborar mejores estrategias y políticas de seguridad para minimizar el riesgo de intrusión en la red de datos de una empresa.

6.2.7 Estrategias Riesgos - Fortalezas:

- Debido a la disponibilidad de diferentes herramientas de software libre necesarias para la implementación del modelo, es posible buscar diferentes alternativas que resulten en un mejor rendimiento, compatibilidad y que optimicen eficientemente los recursos de hardware con los que se cuenta.
- Debido a la gran escalabilidad y adaptación del modelo a cualquier red, puede llegar a competir con modelos similares que se desarrollen.
- Al ser implementado mediante software libre, para las empresas puede llegar a ser más rentable económicamente que modelos de IDS con licencia paga de uso.
- Gracias a la adaptabilidad del modelo a cualquier infraestructura de red de la organización, no resultaría tan complejo buscar alternativas de hardware que sean compatibles con su configuración.

6.2.8 Estrategias de Amenazas - Debilidades:

- Capacitar permanentemente al personal que implemente el modelo propuesto en nuevas tecnologías de minería de datos y detección de intrusos, para poder corregir fallos o realizar mejoras al Sistema de detección de intrusos basado en técnicas de minería de datos que permitan posicionar el modelo en cuestión por sobre los competidores.



- Buscar y tener en cuenta diferentes alternativas de software de monitoreo y detección para poder estar al día ante los cambios que se produzcan en el ámbito de la seguridad de redes y detección de nuevas amenazas.
- Buscar y sugerir continuamente a las empresas que implementen el modelo de detección en la renovación del hardware de red y servidores, que permitan optimizar el rendimiento y eficiencia del IDS.
- Buscar continuamente nuevas definiciones de amenazas o intromisiones de datos para poder generar nuevas reglas de detección que vuelvan más eficiente al modelo de IDS propuesto.

6.3 Análisis de riesgos

Área	Riesgo	Medida a tomar
Organización	Reestructuración de la organización	• Tener toda la documentación al día. Mantener una comunicación fluida con todas las personas relacionadas al proyecto para prever futuros cambios que puedan ocurrir en el lugar de desarrollo e implementación.
	Cambios en las prioridades (requerimientos) de la organización (Procesos, Personas).	• Comunicación fluida con los distintos sectores asociados al proyecto para estar al día con las novedades y sus posibles cambios que puedan afectar al desarrollo del mismo.
Proyecto	Hardware/Software no disponible-	• Disponer de posibles alternativas al hardware/software requerido, que también sean acorde al modelo a desarrollar.
	Fallas en la infraestructura y servicios externos (Tecnología).	• Realizar un análisis anticipado de los posibles desperfectos que puedan existir en el contexto físico (Hardware) donde se desarrollara las pruebas del modelo IDS, para así tenerlos identificados y poder hacer los arreglos necesarios para mejorar las fallas en la infraestructura y como así también los servicios externos a la organización.



	Retrasos por imprevistos	• Tener muy presentes y dar por sentado que se producirán contratiempos, y tenerlos en cuenta en el desarrollo de la solución y en el tiempo estimado de cada tarea relacionada con el proyecto.
	Disponibilidad de personal	• Tener un listado amplio de personal para posibles reemplazos en el equipo.
	Disponibilidad de BD de entrenamiento o aprendizaje.	• Buscar continuamente y disponer de gran variedad de datos de entrenamiento.
	Compatibilidad de las herramientas para el desarrollo del modelo	• Investigar y contar con una amplia variedad de herramientas de monitoreo y detección de intrusos Open Source.
	Entendimiento en el uso de las herramientas necesarias para el desarrollo del modelo propuesto	• Investigar y capacitar continuamente al personal que esta frente del proyecto en el uso de las diferentes herramientas de monitoreo, minería de datos y detección de intrusos

Cuadro 12 – Análisis de riesgos

Referencias para Matriz de Probabilidad e Impacto:

- Importancia: 1-5
- Probabilidad: 0.1 – 1.0
- Ponderado: Importancia * Probabilidad
- Prioridad:
 - Baja: 0 – 2.0
 - Media: 2.0 – 6.0
 - Alta: 6.0 - 10



Riesgo	Importancia	Probabilidad	Ponderado	Prioridad
Reestructuración de la organización	4,00	0,10	0,40	Baja
Cambios en las prioridades (requerimientos) de la organización (Procesos, Personas).	2,00	0,10	0,20	Baja
Hardware/Software no disponible-	4,00	0,50	2,00	Media
Fallas en la infraestructura y servicios externos (Tecnología).	4,50	0,60	2,70	Media
Retrasos por imprevistos	3,00	0,70	2,10	Media
Disponibilidad de personal	4,0	0,4	1,6	Media
Disponibilidad de BD de entrenamiento o aprendizaje	5,0	1,0	5,0	Alta
Compatibilidad de las herramientas para el desarrollo del modelo	4,0	0,8	3,2	Alta
Entendimiento en el uso de las herramientas necesarias para el desarrollo del modelo propuesto	5,0	0,7	4,3	Alta
Total	20,00	2,23	7,59	

Cuadro 12 – Matriz de Probabilidad e Impacto



6.4 Factibilidad económica

Debido a las dimensiones del proyecto y tomando en cuenta que se trabajará con herramientas de software libre, y que el equipamiento de hardware y software pago están a disposición en el laboratorio de informática de la Facultad de Ingeniería de la Universidad, se llegó a la conclusión de que es factible económicamente el desarrollo y la puesta en marcha del proyecto en cuestión.

A continuación, se describen una serie de recursos considerados necesarios para desarrollar la solución propuesta:

Recursos Humanos

- Administrador de red
- Técnico Informático
- Especialista en minería de datos
- Especialista de seguridad de redes

Teniendo en cuenta que el proyecto tiene una duración de 77 días hábiles y que cada jornada laboral costa de 6 horas, a continuación se describen la cantidad de días que trabajará cada uno de los recursos humanos en las distintas etapas del proyecto:

Etapas/RRHH	Técnico	Administrador de Red	Especialista Minería de Datos	Especialista Seguridad
Comprensión del negocio	4	8	0	2
Administración de la Configuración	4	6	0	3
Comprensión de los Datos	0	6	3	2
Preparación de Datos	0	1	8	0
Modelado	0	0	9	0
Evaluación	0	0	6	7
Administración de la seguridad	1	2	1	7
Implementación	1	5	2	10
Total DIAS	10	28	29	31

Cuadro 13 - Calculo de dias trabajados por RRHH



Una vez determinado la cantidad de días que trabajará cada recurso humano, se establecerán los sueldos para cada uno de acuerdo al precio por hora que cobra cada profesional en su area:

Recursos	Precio por hora	Total de horas (6 horas por día)	Subtotal
Administrador de red	\$ 240	168 horas	\$ 40.320,00
Técnico	\$ 120	60 horas	\$ 7.200,00
Especialista en minería de datos	\$ 240	174 horas	\$ 41.760,00
Especialista de seguridad de redes	\$240	186 horas	\$ 44.640,00
Total			\$ 133.920,00

Cuadro 14 – Inversión en RRHH

Referencia de sueldos: <http://www.copaipa.org.ar/informatica/> fecha: 25/07/2016

A continuación se detalla el cobro de los honorarios de cada uno de los profesionales por mes durante la duración del proyecto:

Mes/Recurso	Administrador de Red	Técnico	Especialista en Seguridad de Redes	Especialista en Minería de Datos	Total x Mes
Noviembre	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Diciembre	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Enero	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Febrero	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Total x Recurso	\$ 40.320,00	\$ 7.200,00	\$ 44.640,00	\$ 41.760,00	\$ 133.920,00

Cuadro 15- Honorarios RRHH



Recursos de Hardware

Debido a que se utilizarán los equipos de las instalaciones del laboratorio de Informática de la UCASAL, solamente se requerirá la adquisición de dos notebooks para ser usadas por los profesionales:

Recursos	Cantidad	Descripción	Costo Unitario +IVA	Subtotal
Notebook Hp	2	Hp Intel I7 Quad 6ta 1tb 16gb 17.3 Touch Video 4gb	\$ 17.000,00	\$ 50.000,00
Total				\$ 50.000,00

Cuadro 16 – Inversión en hardware

Referencia de precios: www.mercadolibre.com.ar fecha: 25/07/2016

Recursos de software

Recursos	Cantidad	Descripción	Costo Unitario +IVA	Subtotal
Licencia Windows 7 SP1 Profesional 64 bits	2	Licencia OLP	\$ 2.682,00	\$ 5.364,00
Sistema Operativo CentOS 6.7	1	Licencia GNU	\$0,00	\$ 0,00 *
Rapid Miner 5.3.015	1	Licencia AGPL	\$ 0,00	\$ 0,00 *
Suricata IDS 2.0.5	1	Licencia GPL	\$ 0,00	\$ 0,00 *
Snorby 2.6.3	1	Licencia GPL	\$ 0,00	\$ 0,00 *
Barnyard 2.1.10	1			\$ 0,00 *
Mysql 14.14	1			\$ 0,00 *
TCPdump 4.6.2	1		\$ 0,00	\$ 0,00 *
Wireshark v1.12.6	1	Licencia GPL	\$ 0,00	\$ 0,00 *
Sistema Operativo Kali Linux 2.0	1	Licencia GNU	\$ 0,00	\$ 0,00 *
Google Chrome	1		\$ 0,00	\$ 0,00 *
Microsoft Office 2010	2	Licencia OLP	\$ 1.662,48	\$ 3.324,48
VMWare Workstation 10.0.3	2		\$ 3.346,66	\$ 6.693,32
Bro Network Security Monitor 2.4.1	1		\$ 0,00	\$ 0,00*
Total				\$ 15.381,8

Cuadro 17 – Inversión en software



(*) El costo es \$0 ya que el software es de licencia libre y gratuita.

Referencia de precios:

https://www.amazon.es/Microsoft-Windows-Professional-Sistema-Operativo/dp/B00H09BXI2/ref=sr_1_6?s=software&ie=UTF8&qid=1471403988&sr=1-6

http://store.vmware.com/store/vmware/es_MX/DisplayProductDetailsPage/ThemeID.2485600/productID.323690000

Recursos en información

Recursos	Costo
Base de Datos gureKddcup 2008	\$ 0,00 *
Base de Datos nslKddcup 2008	\$ 0,00 *
Total	\$ 0,00

Cuadro 18 – Inversión en información

(*) El costo es de \$0 ya que el material está destinado para fines científicos y académicos, y se encuentra disponible de manera libre y gratuita en internet.

Link de descarga: http://www.aldapa.eus/res/gureKddcup/galdetegia_jaso.php

Resumen de inversión

Recursos	Subtotal
Humanos	\$ 133.920,00
Hardware	\$ 50.000,00
Software	\$ 15.381,8
Información	\$ 0,00
Total	\$ 199.301,8

Cuadro 19 – Resumen de inversión

6.5 Factibilidad técnica

Debido a que el presente proyecto se llevará a cabo en las instalaciones del laboratorio de informática de la Facultad de Ingeniería de la Universidad, se hará uso de los equipos hardware disponibles en el laboratorio, los cuales abarcan los requisitos de hardware del presente proyecto, disponiendo de 16 computadoras (en el aula taller 1) con sus respectivos monitores, cableado de conectividad UTP cat. 5e, equipos de conectividad (route's y switch's)



y conexión a internet. En cuanto a los requerimientos de software del proyecto, las computadoras del laboratorio ya tienen instalados los sistemas operativos Windows 7 Profesional 64 bits y Linux Ubuntu junto con los drivers correspondientes (audio, video, red, periféricos, etc.), además de software utilitario (Microsoft Office 2010, Google Chrome, Mozilla Firefox, etc) y VMWare 10 para la virtualización de equipos; en cuanto a las herramientas de software específicas para el desarrollo del modelo propuesto: Suricata, Rapid Miner, Wireshark, Kali Linux y las bases de datos GureKDD Cup 2008 y NSLKDD Cup 2012, han de ser adquiridos de internet, ya que son de uso libre y bajo licencias de uso gratuitas. Así también las 2 notebooks externas al laboratorio a utilizar en el desarrollo de la solución propuesta, son propiedad de los integrantes del equipo de desarrollo del presente proyecto, las mismas cuentan los S.O. Windows 7 Home Premium y Windows 8 Profesional (adquiridos con licencia de uso) y las herramientas de software requerido (VMWare 10, Office 2010, etc). Considerando la disponibilidad de los equipos y herramientas para desarrollar el modelo propuesto, podemos decir que el presente proyecto es técnicamente factible.

6.6 Factibilidad operativa

En cuanto a la factibilidad operativa debemos mencionar que el presente proyecto se llevara a cabo bajo la dirección, asesoramiento y control de profesionales especializados en las dos áreas informáticas que conforman el proyecto: Seguridad de redes datos y minería de datos (docentes de la Ucasal), lo cual presenta una factibilidad operativa para el desarrollo del modelo propuesto. Así mismo cabe mencionar que el modelo propuesto será desarrollado genéricamente con la posibilidad de ser adaptado a cualquier ámbito organizacional y red de datos, con lo cual decimos que es factible operativamente para el personal de la organización que implemente dicho modelo, que posea conocimientos básicos en administración de redes de datos.

6.7 Factibilidad legal

Este proyecto no inflige ninguna norma legal o ley, ya que se está trabajando con herramientas de software libre y las que son de uso bajo licencias pagas están adquiridas legalmente por la Facultad de Ingeniería de la Universidad, por lo tanto, es factible legalmente.

6.8 Factibilidad ambiental

En lo que respecta al aspecto físico-ambiental relacionado con el lugar de trabajo del proyecto en cuestión, se puede afirmar que el mismo requerirá de la adaptación o adecuación mínima del ambiente, laboratorio de informática, para poder desarrollar eficientemente las tareas involucradas con el modelo propuesto. Sin embargo, no alterará ni traerá consecuencias negativas para el contexto en sí, por lo que se llega a la conclusión de que es factible desde el punto de vista ambiental llevar a cabo el proyecto.



6.9 Análisis Económico-Financiero

6.9.1 Inversión Inicial del Proyecto

Si se tiene en cuenta que la empresa donde se desarrollará el proyecto en cuestión ya cuenta con una red LAN con un cableado estructurado en funcionamiento y equipos en uso, tales como un servidor y maquinas clientes, entonces los recursos necesarios para iniciar con el proyecto son:

6.9.1.1 Equipos para desarrollar el proyecto:

Equipo	Cantidad	Precio Unitario	Total
Notebook Hp Intel I7 Quad 6ta 1tb 16gb 17.3 Touch Video 4gb	2	\$ 25.000,00	\$ 50.000,00
Referencia de Precios (01/04/2017): http://articulo.mercadolibre.com.ar/MLA-646994926-notebook-hp-intel-i7-quad-6ta-1tb-16gb-173-touch-video-4gb-_JM			

Cuadro 20 – Inversión Inicial en Hardware

6.9.1.2 Software Requerido:

Producto	Cantidad	Precio Unitario	Subtotal
Licencia VMware Workstation Pro 10	2	\$ 3.346,66	\$ 6.693,32
Licencias de Windows 7 Professional	2	\$ 2.682,00	\$ 5.364,00
Licencia Microsoft Office 2010 Estándar	2	\$ 1.662,48	\$ 3.324,48
TOTAL			\$ 15.381,80

Cuadro 21 – Inversión Inicial en Software

Por lo tanto, la inversión inicial para el proyecto en cuestión es de:



Inversión Inicial Total: Equipos + Software Requerido = \$ 65.381,80

6.9.2 Inversión en RRHH

Mes/Recurso	Administrador de Red	Técnico	Especialista en Seguridad de Redes	Especialista en Minería de Datos	Total x Mes
Noviembre	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Diciembre	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Enero	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Febrero	\$ 10.080,00	\$ 1.800,00	\$ 11.160,00	\$ 10.440,00	\$ 33.480,00
Total x Recurso	\$ 40.320,00	\$ 7.200,00	\$ 44.640,00	\$ 41.760,00	\$ 133,920.00

Cuadro 22 – Plan de inversión en RRHH



Total de Ingresos

Los totales de ingresos están asociados a la cantidad de ventas estimadas del modelo de IDS propuesto considerando un incremento por ventas de un 25% mensual. Los ingresos empiezan a ser visibles recién en el mes de Marzo, si se considera que el producto sale al mercado. Esto incluye la instalación, configuración y puesta en marcha del Sistema de Detección de Intrusos basado en técnicas de Data Mining.

Total de Egresos

Los Egresos totales del proyecto consisten en los sueldos a pagar a los profesionales del proyecto durante la ejecución del mismo (Noviembre hasta Febrero) y una vez implementado el IDS, se considera los sueldos a pagar para los recursos humanos responsables de realizar el mantenimiento del mismo.

Costos de Mantenimiento

Para los fines del proyecto se realizará un mantenimiento perfectivo, el cual consiste, una vez implementado el IDS basado en técnicas de minería de datos de forma correcta y con un óptimo funcionamiento, en seguir profundizando en los aspectos de seguridad y minería de datos y optimizando aún más el rendimiento del IDS en la detección de intrusos en la red de datos.



Flujo de caja

		15	15	16	16	16	16	16	16	16	16	16	16	16
		Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oc	Total
Inversión Inicial														
Ingresos														
Ventas en efectivo						50000	62500	78125	97656,25	122070,3	152587,9	190734,9	238418,6	992092,9
Total Ingresos		0	0	0	0	50000	62500	78125	97656,25	122070,3	152587,9	190734,9	238418,6	992092,9
Egresos														
Técnico Informático		1800	1800	1800	1800	0	0	0	0	0	0	0	0	7200
Administrador de Red		10080	10080	10080	10080	3840	3840	3840	3840	3840	3840	3840	3840	71040
Especialista Minería de Datos		10440	10440	10440	10440	4800	4800	4800	4800	4800	4800	4800	4800	80160
Especialista Seguridad de Red		11160	11160	11160	11160	4800	4800	4800	4800	4800	4800	4800	4800	83040
Total Egresos	-65381,8	33480	33480	33480	33480	13440	13440	13440	13440	13440	13440	13440	13440	241440
Flujo de caja económico	-65381,8	-33480	-33480	-33480	-33480	36560	49060	64685	84216,25	108630,3	139147,9	177294,9	224978,6	

Cuadro 23 – Flujo de Efectivo Proyectado a 12 meses



6.9.3 Valor Actual Neto del Proyecto (VAN)

$$VAN = -I_0 + \sum_{t=1}^n \frac{F_t}{(1+k)^t} = -I_0 + \frac{F_1}{(1+k)} + \frac{F_2}{(1+k)^2} + \dots + \frac{F_n}{(1+k)^n}$$

Imagen 69 – Formula del VAN

- F_t son los flujos de dinero en cada periodo t
- I_0 es la inversión realiza en el momento inicial ($t = 0$)
- n es el número de periodos de tiempo
- k es el tipo de descuento o tipo de interés exigido a la inversión

$I_0 = \$ 65381,8$

$N = 12$

$K = 15 \%$



VAN = \$ 75.701,41

Tasa de Descuento	VAN
10%	\$ 250.291,28
20%	\$ 74.711,36
30%	\$ 8.004,29
40%	\$ -18.341,21
50%	\$ -28.517,49
60%	\$ -31.850,83
70%	\$ -32.201,25

Cuadro 24 – VAN del Proyecto

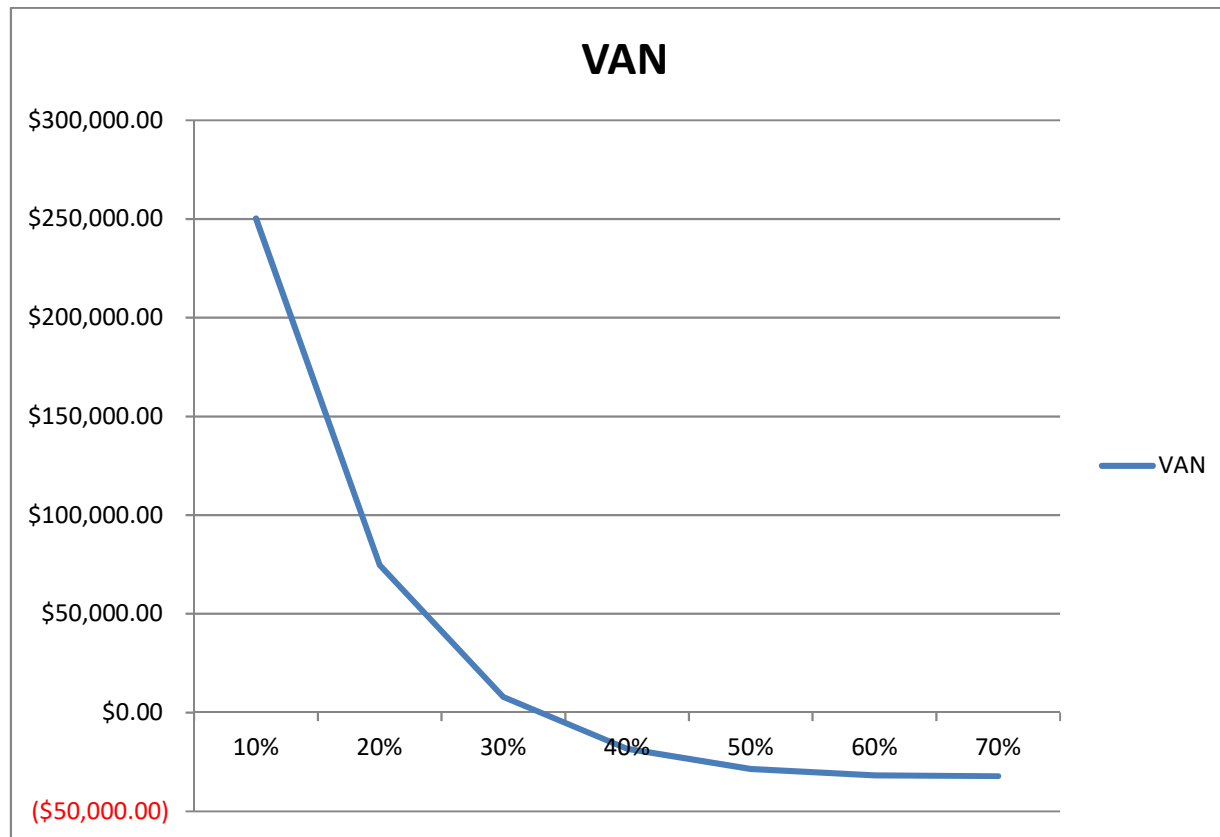


Imagen 70 – Análisis de Valor Actual Neto con diferentes Tasas de Descuento



6.9.4 Tasa Interna de Retorno (TIR)

$$VPN = \frac{\sum R_t}{(1+i)^t} = 0$$

Imagen 71 – Formula de la TIR

- t el tiempo del flujo de caja.
- i la tasa de descuento.
- R_t el flujo neto de efectivo (la cantidad de dinero en efectivo, entradas menos salidas) en el tiempo t.

T= 12

I= 15%

TIR = 32%

6.9.5 Rentabilidad

El VAN y el TIR son dos herramientas financieras que nos permiten evaluar la rentabilidad del proyecto, determinando si es factible invertir dinero o no.

El VAN es un indicador financiero que mide los flujos de los futuros ingresos y egresos que tendrá un proyecto, para determinar, si luego de descontar la inversión inicial, nos quedaría alguna ganancia. Si el resultado es positivo, el proyecto es viable.

- $VAN > 0 \rightarrow$ el proyecto es rentable.
- $VAN = 0 \rightarrow$ el proyecto es rentable también, porque ya está incorporado ganancia de la Tasa de Descuento.
- $VAN < 0 \rightarrow$ el proyecto no es rentable.

La TIR es la tasa de descuento (TD) de un proyecto de inversión que permite que el beneficio neto actualizado (BNA) sea igual a la inversión (VAN igual a 0). La TIR es la máxima TD que puede tener un proyecto para que sea rentable, pues una mayor tasa ocasionaría que el BNA sea menor que la inversión (VAN menor que 0). (Crece Negocios, 2014).

Para nuestro caso obtuvimos un Valor Actual Neto de \$ 75.701,41 el cual es mayor a cero, por lo que podemos concluir que nuestro proyecto es económicamente rentable.

Por otro lado, la Tasa Interna de Retorno de nuestro proyecto de del 32% por lo que es un valor aceptable en términos financieros.



Capítulo 7 – Resultados obtenidos y Conclusiones

7.1 Introducción

En este último capítulo se describirán los resultados obtenidos, juntamente con el análisis de los mismos, y se hará mención de aspectos relacionados al presente proyecto para el desarrollo de futuras líneas de investigación.

7.2 Resultados o verificación experimental

Tras la implementación del Servidor IDS en la red simulada, se realizaron nuevamente los ataques informáticos, pero esta vez fueron detectados por el Servidor IDS, devolviendo los siguientes resultados a través tablas y gráficos estadísticos:

Top 15 Signatures		
Signature Name	Percentage	Event Count
TCP DOS ATTACK	71.19%	4356
SURICATA STREAM Packet with invalid ack	12.88%	788
SURICATA STREAM ESTABLISHED invalid ack	12.47%	763
ET SCAN Potential FTP Brute-Force attempt response	0.77%	47
Snort Alert [1:1000012:1]	0.62%	38
MITM ATTACK - ARP POISONING	0.34%	21
FUERZA BRUTA/DICCIONARIOS	0.31%	19
SURICATA STREAM SHUTDOWN RST invalid ack	0.26%	16
Snort Alert [1:2240001:1]	0.23%	14
SURICATA STREAM FIN out of window	0.2%	12
ET POLICY FTP Login Successful	0.15%	9
ET DNS Non-DNS or Non-Compliant DNS traffic on DNS port Opcode...	0.15%	9
SURICATA STREAM FIN invalid ack	0.15%	9
NMAP IPSWEEP TCP	0.08%	5
Snort Alert [1:2210000:1]	0.05%	3
GPL NETBIOS SMB IPC\$ unicode share access	0.05%	3
Snort Alert [1:2210032:1]	0.05%	3
WAREZMASTER	0.03%	2
WAREZCLIENT	0.02%	1
Snort Alert [1:1445:5]	0.02%	1

Imagen 72 – Tabla de resultados de Snorby



Como podemos observar, Snorby devuelve un listado de las 15 mejores primeras alertas de ataques detectados ordenándolos en base al porcentaje de ocurrencia en todo el tiempo desde que se puso en marcha el IDS, y también en base a la cantidad de veces que cada regla es activada por la detección de un posible ataque. En el nuestro caso el mayor porcentaje es del ataque de DOS debido al comportamiento del mismo (el gran flujo de paquetes que conlleva un ataque de este tipo). También podemos observar los resultados obtenidos mediante un gráfico de tortas:

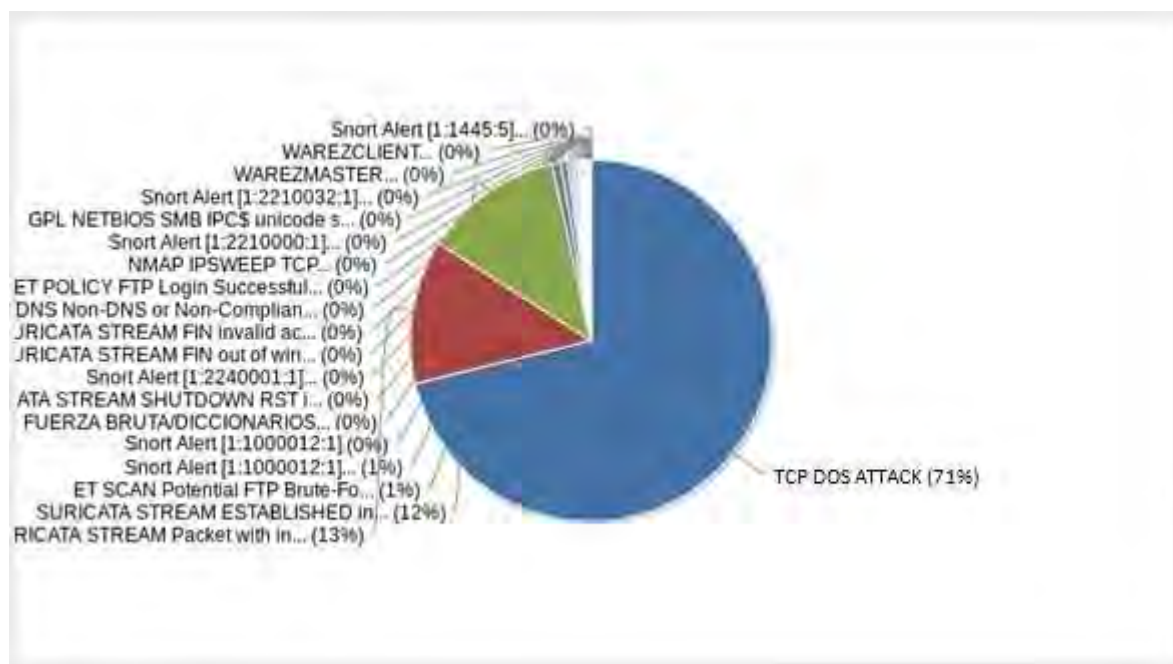


Imagen 73 – Grafico de Tortas de resultados de Snorby

También podemos observar los resultados de la severidad del tráfico producido en la red de datos en el siguiente gráfico:



Imagen 74 – Niveles de severidad de alertas de Snorby

Se observa mediante un gráfico de barras los niveles de severidad producidos en el tráfico de la red de datos, en nuestro caso la clasificación de High Severity (alto riesgo) posee una cantidad de 108 conexiones que implican ataques de alto riesgo, seguidamente la clasificación Medium Severity (riesgo medio) muestra una cantidad de 7956 conexiones que implican ataques de severidad media y por último la clasificación Low Severity que muestra una cantidad de 7732 conexiones de bajo riesgo pero que no deben ser pasadas por alto. También se puede observar mediante un gráfico de área la relación entre los niveles de severidad (severity count) con las horas en un día específico (hour or day) en que se produjeron los distintos tipos de alertas, en la imagen 76 podemos ver que desde las 20 hs del 25/05/2016 hasta las 20 hs del 26/05/2016 la actividad de los ataques realizados y detectados por el IDS ocurrieron entre las 15 y 18 hs del día 26/05/2016.

Por otro lado, podemos identificar las 10 principales direcciones IP de origen y destino de las conexiones ordenadas por el porcentaje de ocurrencia de las mismas en las siguientes tablas:



Top 10 Source Addresses

Source IP Address	Percentage	Event Count
192.168.40.130	74.72%	11814
192.168.40.131	24.53%	3878
192.168.40.128	0.68%	108
192.168.40.2	0.05%	8
95.141.36.6	0.02%	3
172.217.28.142	0.01%	1

Imagen 75 – Tabla de las 10 principales direcciones IP de origen

Top 10 Destination Addresses

Destination IP Address	Percentage	Event Count
192.168.40.131	74.79%	11821
192.168.40.130	24.6%	3889
192.168.40.128	0.07%	11
95.213.193.55	0.05%	8
95.141.36.6	0.04%	6
178.255.43.146	0.04%	6
77.234.42.30	0.04%	6
161.202.108.200	0.04%	6
5.45.62.74	0.04%	6
168.1.69.112	0.04%	6

Imagen 76 – Tabla de las 10 principales direcciones IP de destino

En nuestro caso, en la dirección IP de origen, se encuentra en primer lugar la dirección 192.168.40.130 que hace referencia a la maquina atacante (Kali Linux 2.0), y en la dirección IP de destino tenemos la dirección 192.168.40.131 que hace referencia al servidor (Centos 6.7).

7.3 Análisis de resultados (Análisis de rendimiento del IDS)

En base a los resultados obtenidos se procedió a realizar un análisis de los mismos. Los ataques realizados en la red de datos simulada fueron:

- Fuerza Bruta con Diccionarios
- DOS – Denegacion de servicios
- Warezclient/Warezmaster



- Escaneo de redes con nmap

Estos ataques pertenecen a una serie de categoría de ataques utilizados por la Gure KDD Data Base 2008 y la NSL KDD database 2012, las cuales son las siguientes:

- *DOS: Denegación de Servicios*
- *R2L: Acceso no autorizado - se produce cuando un atacante tiene la capacidad de enviar paquetes a una máquina a través de una red, pero que no tiene una cuenta en esa máquina y explota alguna vulnerabilidad para obtener acceso local como un usuario de esa máquina.*
- *U2R: Acceso no autorizado a los privilegios de súper usuario - es un método para explotar que el atacante utiliza y comienza con el acceso a una cuenta de usuario en el sistema (tal vez adquirida por el robo de contraseñas, ataque por diccionario o ingeniería social) y es capaz de explotar alguna vulnerabilidad para obtener la raíz de acceso al sistema*
- *Ataque de sondeo (probing): Es un intento de reunir información sobre una red de ordenadores con el fin aparente de eludir los controles de seguridad.*

(Mahbod Tavallaee, 2009)

Los ataques simulados en la red corresponden a las categorías DOS, R2L y PROBING, y en base a los resultados obtenidos el mayor porcentaje de ataques detectados por el IDS basado en reglas creadas a partir de la minería de datos son de la categoría DOS y en segundo lugar de la categoría R2L.

7.3.1 Comparación de resultados

A continuación, se realiza una comparación entre dos Sistemas de Detección de Intrusos, uno tradicional y el otro basado en minería de datos (modelo propuesto en el presente proyecto), realizando para ambos casos los mismos ataques mencionados anteriormente (con los mismos parámetros), bajo las mismas condiciones de las máquinas virtuales y la red de datos (configuraciones de S.O., adaptadores de red, direcciones IP, servidores web y ftp, y sus puertos de comunicación).

Los resultados obtenidos con un IDS tradicional y los ataques simulados descriptos anteriormente, antes de la aplicación de las reglas de detección de intrusos obtenidas con la ayuda del proceso de minería de datos, se muestran en la siguiente imagen:

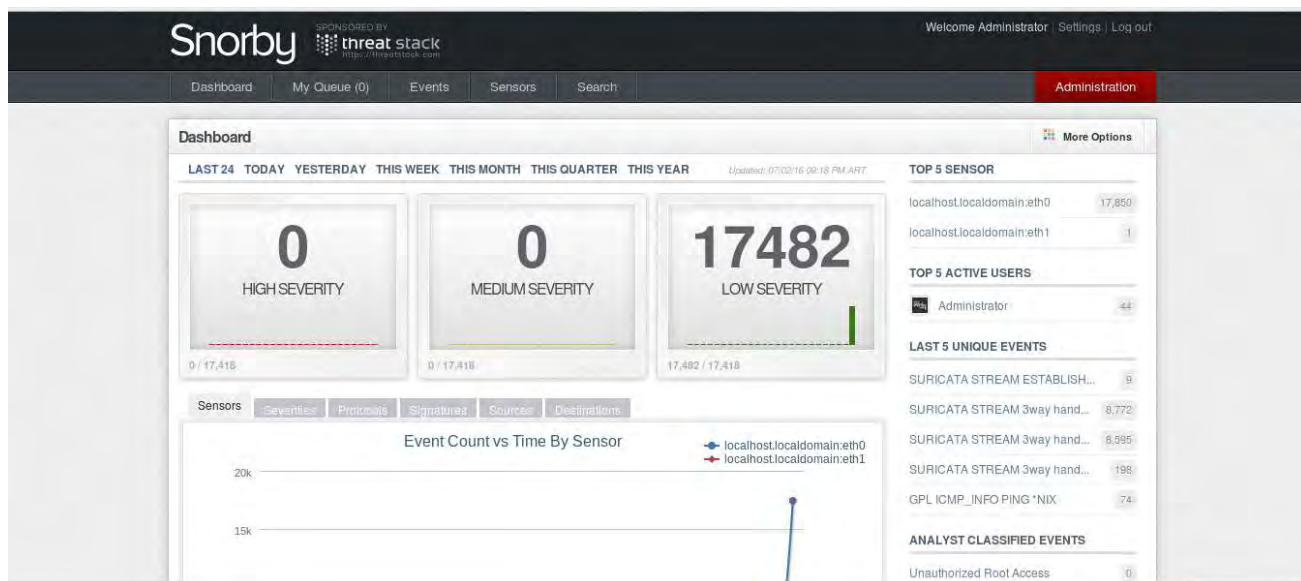


Imagen 77 – Resultados obtenidos con un IDS tradicional

Como puede observar un IDS tradicional, es decir sin la incorporación de las reglas en base a minería de datos, posee un rendimiento inferior en cuestiones tales como: clasificación y precisión de ataques y una menor escalabilidad en lo que respecta a los cambios que puedan surgir en el tráfico de datos. Estos resultados son obtenidos al principio de la etapa de 1.7 (Instalación del IDS) de la Estructura de Desglose del Trabajo, antes de la ejecución de la tarea 1.7.3 (Incorporación de las Reglas), es decir entre los días 21/04/2016 y 22/04/2016.

La siguiente tabla muestra los resultados obtenidos de la clasificación de intrusiones en el IDS tradicional:

Top 15 Signatures

Signature Name	Percentage	Event Count
SURICATA STREAM 3way handshake SYN resend different seq on SYN...	49.94%	8753
SURICATA STREAM 3way handshake SYNACK resend with different ack	48.94%	8578
SURICATA STREAM 3way handshake SYNACK with wrong ack	1.12%	196

Imagen 78 – Tabla de resultados de Snorby IDS tradicional

Puede verse que no detecta ningún tipo de ataque específico los cuales ya fueron mencionados anteriormente. Por otro lado, en el siguiente grafico se puede observar que el IDS tradicional solo clasifica las intrusiones de la red en la categoría de severidad baja (severity low).

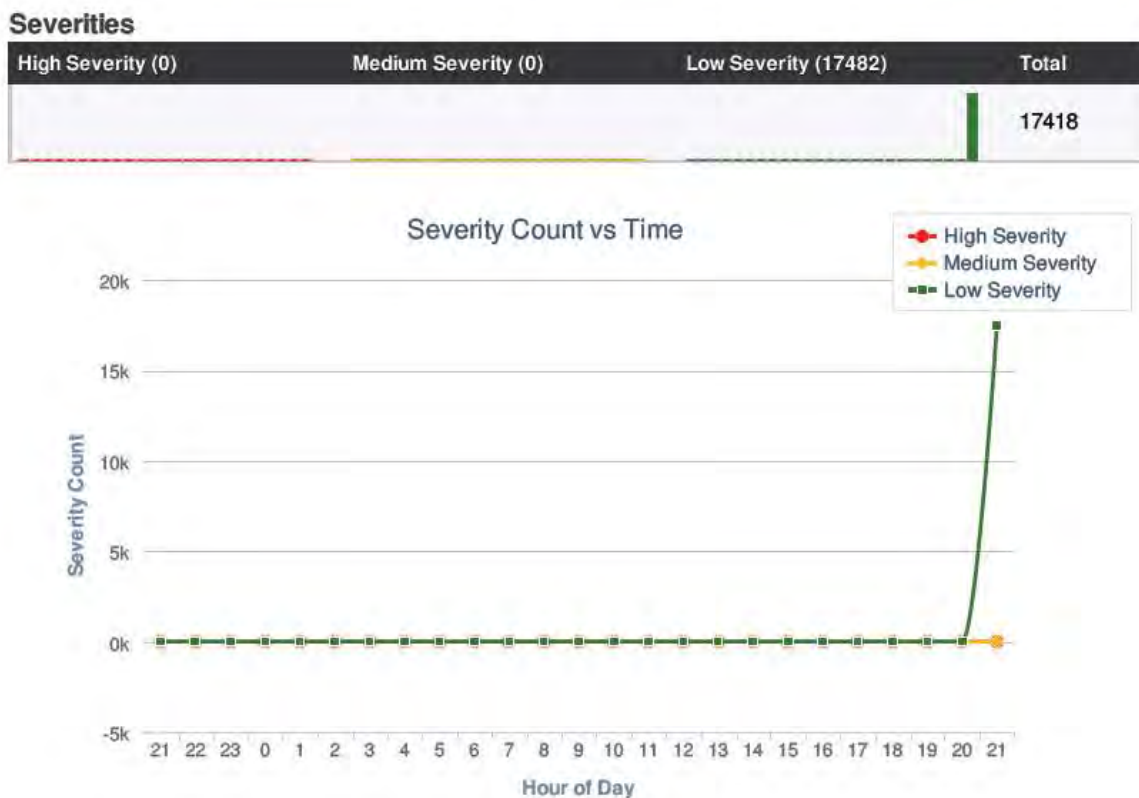


Imagen 79 – Niveles de severidad de alertas de Snorby IDS tradicional

En comparación a un IDS basado en minería de datos, cuyos resultados fueron obtenidos después de la ejecución de la tarea 1.8.1 (Aplicación del modelo) de la etapa 1.8 (Implementación) del WBS y realizando la misma simulación de ataques que para el IDS tradicional, podemos observar que con este último obtenemos un mejor rendimiento y precisión a la hora de detectar las intrusiones a una red de datos como se puede visualizar en la siguiente imagen:

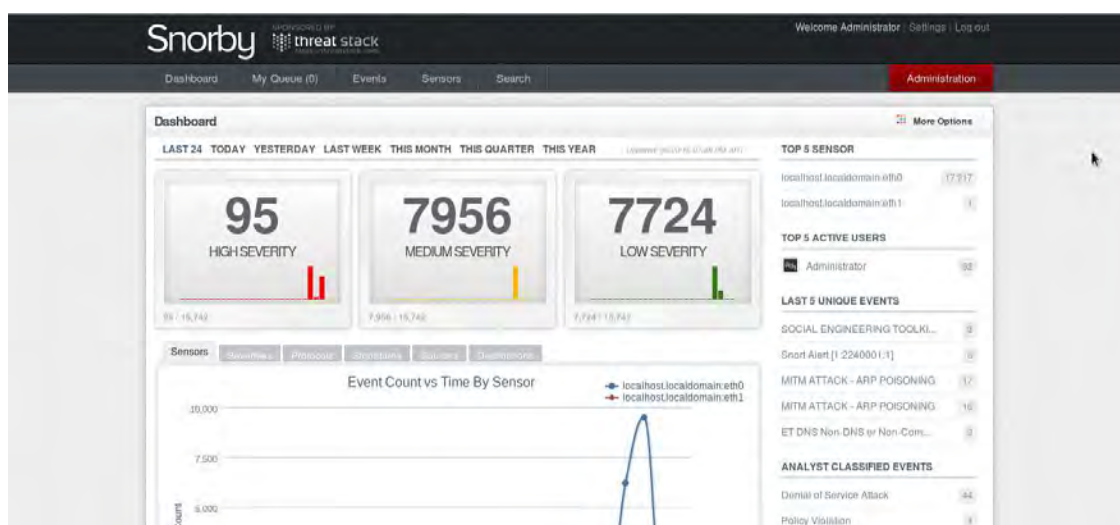


Imagen 80 – Resultados obtenidos con un IDS basado en minería de datos



En cuestión, queda demostrado que con ayuda de la minería de datos podemos conseguir resultados más precisos y específicos de alertas de ataques producidos a la red de datos, que permiten obtener un IDS más personalizado al tráfico de red de cualquier organización, y escalable a los cambios que puedan surgir en dicha red. El modelo propuesto en el presente proyecto contribuye a obtener una mejor seguridad, reduciendo las vulnerabilidades y riesgos de intrusiones que puedan llegar a suceder en una red datos de cualquier organización.



7.4 Proceso Estandarizado para la Creacion de Reglas de Deteccion de Intrusos aplicando Minería de Datos.

 UNIVERSIDAD CATOLICA DE SALTA	Elaboración de Reglas de Detección de Intrusos mediante la aplicación de técnicas de minería de datos sobre el tráfico de red capturado.	Versión: 001 Fecha de Elaboración:
Objetivo: Estandarizar el procedimiento de creación de reglas de detección de intrusos, desde el monitoreo y captura de datos de red, aplicación de técnicas de minería de datos, e incorporación de las reglas al motor de Base de Datos del Sistema de Detección de Intrusos.		
Alcance: El proceso aplica a todo el personal Académico (Estudiantes/Profesores) y Profesionales que tengan como objetivo el estudio de la Minería de Datos aplicada a los Sistemas de Detección de Intrusos.		
Definiciones: • IDS: Sistema de Detección de Intrusos • Ataque Informático: Intento organizado y deliberado de una o más personas para causar daño o problemas a un sistema informático o red. • Algoritmo: conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema. • Compilar: Proceso de traducción de un código fuente (escrito en un lenguaje de programación de alto nivel) a lenguaje de máquina (código objeto) para que pueda ser ejecutado por la computadora.		
Resultado: Con la correcta aplicación de este procedimiento se obtendrá las reglas de detección de intrusos para la detección y alerta eficaz de intromisiones informáticas en la red de datos de una empresa.		
Recursos: Humanos: • Administrador de Red. • Especialista en Minería de Datos. • Especialista en Seguridad Informática. Software: • Wireshark 1.12.6 o superior. • Bro Network Security Monitor 2.4.1 o superior. • Hoja de Calculo. • Rapid Miner v5 o superior. • Suricata IDS 2.0.5 o superior. Base de Datos: • Gure KDD database 2008. • NSL database 2012.		



 UNIVERSIDAD CATOLICA DE SALTA		Elaboración de Reglas de Detección de Intrusos mediante la aplicación de técnicas de minería de datos sobre el tráfico de red capturado		Versión: 001 Fecha de Elaboración:
ETAPA	DESCRIPCIÓN DEL PROCEDIMIENTO	ÁREA RESPONSABLE	CARGO RESPONSIBLE	
1- Monitoreo y captura de datos de la red.	1.1- Monitorear y capturar el flujo de información de la red de datos en un periodo de 3 a 5 días, mediante la herramienta Wireshark. 1.2- Guardar la captura realizada en el archivo de extensión “.pcapng”.	-Área de Administración de redes.	-Administrador de red.	
2- Preparación de datos.	2.1- Aplicar el algoritmo “darpa.bro”, de la herramienta Bro Network Security Monitor sobre el archivo “.pcapng” y obtener los 26 atributos del área de cabecera de cada paquete de red capturado. 2.2- Ordenar los registros obtenidos mediante el comando “bro -r archivo.pcapng darpa2gurekddcup.bro > conn.list”. 2.3- Aplicar el algoritmo TrafAld.c sobre el archivo conn.list y obtener los 19 atributos de tráfico de red. El archivo final se denomina TrafAld.list. 2.4- Abrir el archivo TrafAld.list con una hoja de cálculo y convertirlo en un archivo de valores separados	-Área de Administración de redes. -Área de Administración de base de datos.	-Administrador de red. -Especialista en Minería de Datos.	



 UNIVERSIDAD CATOLICA DE SALTA	Elaboración de Reglas de Detección de Intrusos mediante la aplicación de técnicas de minería de datos sobre el tráfico de red capturado		Versión: 001 Fecha de Elaboración:
	Rapid Miner.		
4- Evaluación.	4.1- Aplicar los modelos de clasificación obtenidos en los datos capturados, obtenidos del paso 2.5, mediante la herramienta Rapid Miner. 4.2- Exportar mediante Rapid Miner los resultados de la clasificación de los datos capturados, a un archivo de hoja de cálculo. 4.3- Filtrar los registros de la hoja de cálculo de acuerdo a los tipos de ataques obtenidos en la clasificación, y determinar el patrón de valores de los atributos principales para cada tipo de ataque. 4.4- Crear las reglas de detección de intrusos utilizando los valores distintivos de los atributos de cada tipo de ataque, siguiendo el formato de regla establecido por el IDS Suricata y guardando las reglas en un archivo con extensión “.rules”.	-Área de Administración de base de datos. -Área de Administración de redes.	-Especialista en Minería de Datos. -Especialista en Seguridad Informática.



	UNIVERSIDAD CATOLICA DE SALTA	Elaboración de Reglas de Detección de Intrusos mediante la aplicación de técnicas de minería de datos sobre el tráfico de red capturado	Versión: 001 Fecha de Elaboración:
5- Implementación.	5.1- Incorporar el archivo con extensión .rules, que contiene las reglas de detección de intrusos, en el directorio del servidor IDS: “etc/suricata/rules”. 5.2- Compilar las reglas del archive “.rules” mediante la consola del IDS Suricata. 5.3- Poner en marcha el Servidor IDS mediante el comando “Suricata -c /etc/suricata/suricata.yaml -i eth0 -v”.	-Área de Administración de redes.	-Especialista en Seguridad Informática. -Administrador de red.
OBSERVACIONES:			

Elaborado por:	Revisado por:	Aprobó:
- Ávila, Carlos Alfredo - Barboza, Daniel Félix		



7.5 Conclusión

Finalmente, para concluir podemos resaltar los siguientes puntos:

- 1- Tanto la minería de datos como la seguridad informática son dos ramas de la informática que en la actualidad están en constante investigación y explotación de uso, cada vez se van ampliando más los conceptos y herramientas para la incursión de la seguridad informática, estando dichas herramientas al alcance de cualquier persona. Así también la cantidad de información digitalizada se está volviendo absoluta dejando de lado el papel, con lo cual, con la capacidad de procesamiento de las computadoras toda la información existente puede ser usada con un fin específico, estudiando y analizando hechos del pasado y previendo futuros sucesos en base a patrones de comportamiento, que muchas veces se encuentran ocultos en la misma información, es aquí donde la minería de datos cumple un rol fundamental brindando un uso inteligente de la información disponible, a través de la estadística y algoritmos matemáticos que modelan el conocimiento que se desee descubrir.
- 2- Si bien existen muchas herramientas de seguridad informática que ayudan a la protección de la información, la mayoría de estas están basadas en reglas y políticas de seguridad estándar o definida con anterioridad, que abarcan una serie de posibles ataques y/o actos riesgosos, para la protección de dicha información. Sin embargo, muchas veces estas herramientas no se adecuan con el flujo de información en las redes de datos de las organizaciones (que está dado por las políticas y reglas de negocio de las mismas), por lo cual quedan expuestas vulnerabilidades que pueden ser descubiertas con el estudio del tráfico de red (que hoy en día con el uso de distintas herramientas, es posible acceder a la red de datos y analizar el tráfico). De esta manera tener políticas y reglas de seguridad que no se adecuen con las políticas y reglas de negocio resulta ser ineficiente para el funcionamiento de la organización y representa un alto riesgo para la información de la misma.
- 3- El uso de la minería de datos en las herramientas de seguridad informática permite establecer una seguridad más personalizada que se adecua a las reglas y políticas de negocio reflejadas en el flujo de información de las organizaciones, sin la necesidad de tener que adecuar la organización a las herramientas de seguridad informática. Así también está la posibilidad de que dicha herramienta de seguridad aprenda constantemente nuevas reglas proporcionando así una seguridad en tiempo real (ver en las futuras líneas de investigación).
- 4- Si bien es imposible llegar a eliminar los riesgos en cuanto la seguridad de la información, el presente proyecto propone brindar una herramienta de seguridad informática “dinámica”, con capacidad de aprendizaje en base al contexto de flujo de información que rodea y compone a una determinada organización, estableciendo reglas que se generan a partir del mismo tráfico de red de la dicha organización. Por lo cual la inversión en la seguridad de la información se reduce al generar la protección a partir de las propias reglas de negocio, detectando aquello que no forma parte de las mismas,



y utilizando recursos que no generan mayores gastos y retribuyen una herramienta propia y a medida de la organización que decida implementarla.

7.6 Futura líneas de investigación

El presente proyecto propuso, como ya mencionó anteriormente, desarrollar un modelo de una herramienta de seguridad informática (IDS) para la detección de intrusos (ataques o cualquier acto que atente contra la integridad, confidencialidad o disponibilidad de la información) basado en técnicas y algoritmos de minería de datos (data mining), para ello todo el desarrollo se basó en la creación de reglas personalizadas para la detección de intrusos, a partir de la captura en un momento determinado del tráfico de datos de una red de datos simulada, permitiendo de esta manera, con los algoritmos y técnicas de minería de datos clasificar el tráfico “normal” y el “anormal” capturado (identificando diversos tipos de ataques e intromisiones). En lo sucesivo se propone a los interesados en la temática del presente proyecto continuar con la investigación haciendo hincapié en el funcionamiento en tiempo real del servidor IDS (Servidor de detección de intrusos), es decir, la posibilidad de automatizar la creación de nuevas reglas a partir del monitoreo del tráfico de red en cualquier momento y ser insertadas en el motor de base de datos del IDS, y de esta manera complementar más la seguridad y adaptarla a cambios en el flujo de información, como así también establecer de manera automática acciones definidas y específicas para la protección de la información (por ej. cerrar ciertos puertos de comunicación), que reduzcan al mínimo los riesgos de intrusiones informáticas.



Anexos

Para una mejor comprensión y debido a que el modelo realizado lleva un tiempo considerable para su ejecución, se procedió a realizar 2 videos demostrativos donde se explicará lo detallado a continuación:

- Video N° 1: Se explicará la estructura y puesta en marcha del mecanismo de IDS propuesto
- Video N° 2: Se detallará como llevar a cabo los ataques y las diferencias entre un IDS Tradicional y un IDS con las reglas obtenidas a partir de los resultados del proceso de minería de datos.

Se adjunta junto con el documento del proyecto de grado un CD que contiene los respectivos videos debidamente identificados.

A continuación, en los anexos I, II, III y IV se detallan la instalación de las aplicaciones utilizadas en este proyecto de grado.



Anexo I - Instalación Kali Linux 2.0

1. Seleccionar Instalación Grafica



2. Seleccionar el lenguaje del sistema operativo.





3. Seleccionar la ubicación.

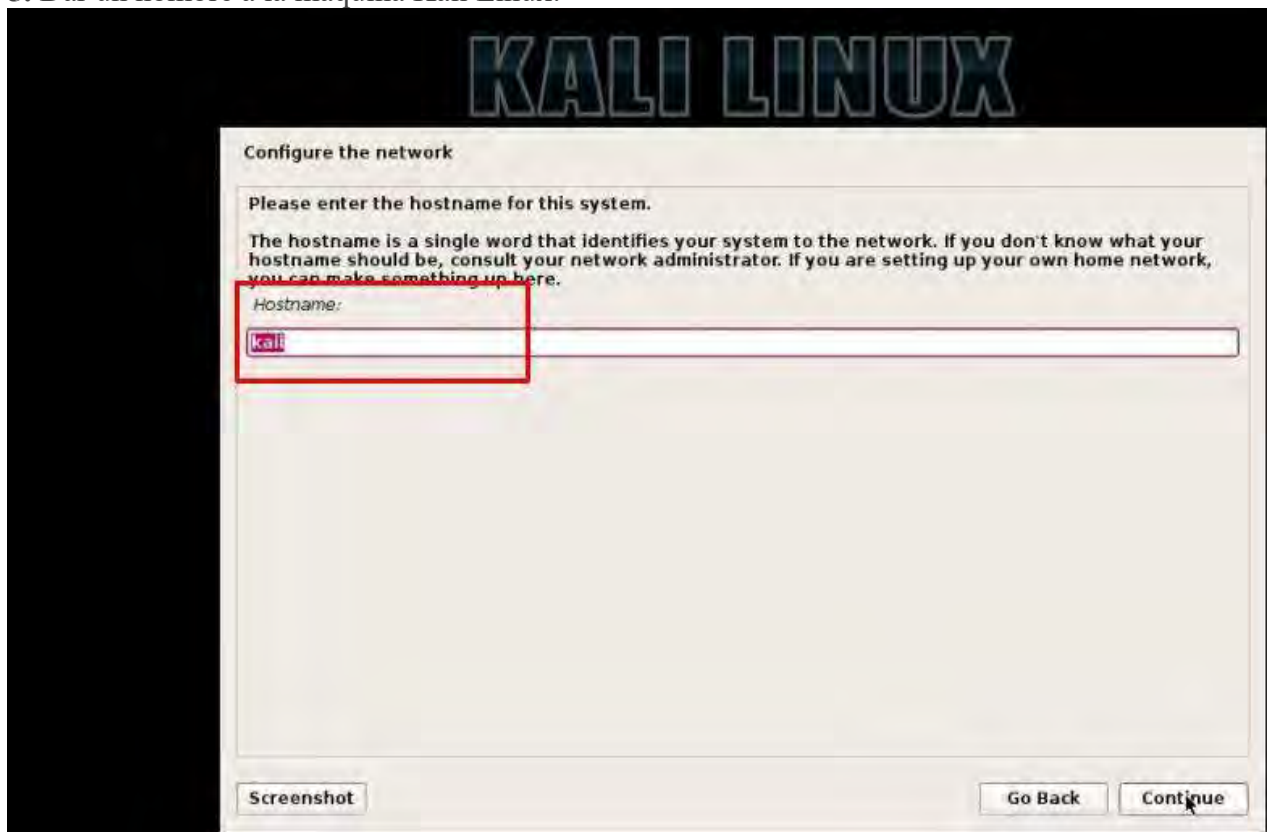


4. Seleccionar la distribución del teclado.





5. Dar un nombre a la maquina Kali Linux.

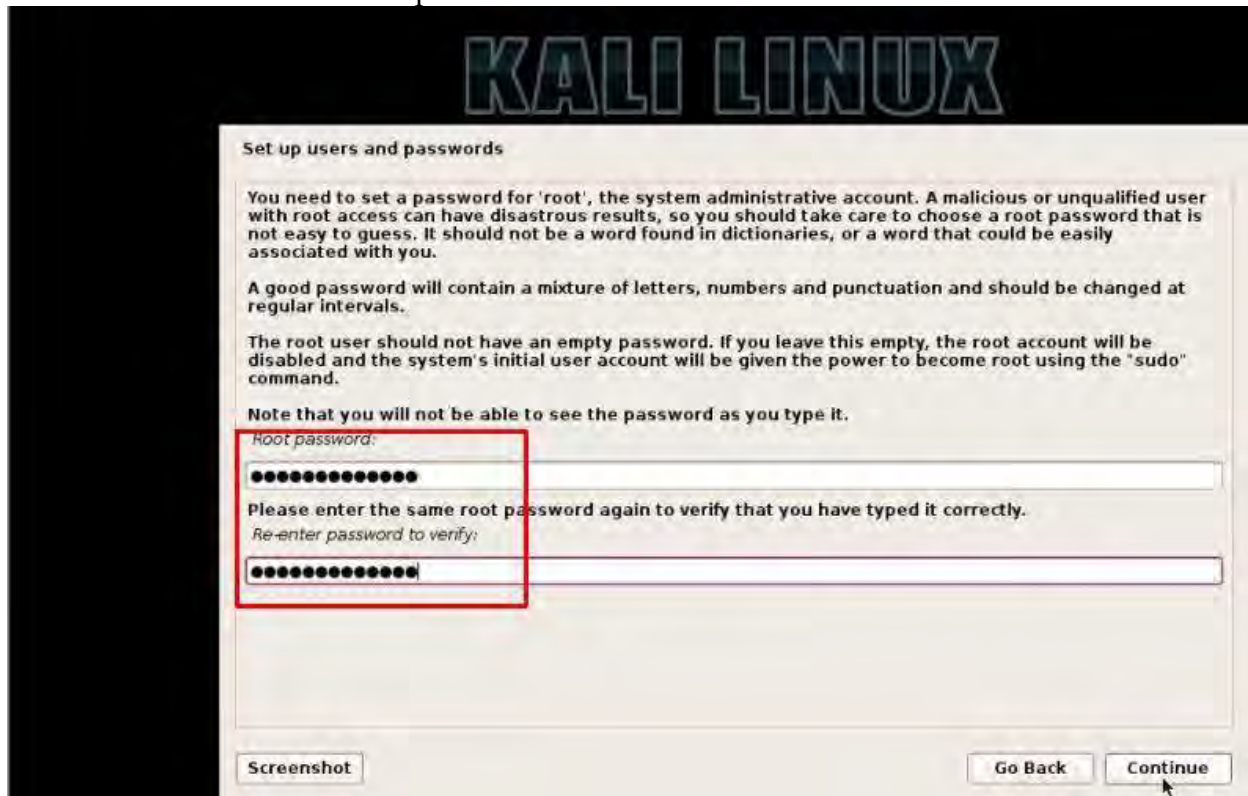


6. Click para continuar a la configuración de la red.





7. Configurar contraseña de usuario root y haga clic en Continuar. Va a ingresar en Kali Linux 2 con esta contraseña después de la instalación hecha.



8. Elija Guiado - utilizar la opción de disco entero y luego click en Continuar. Es seguro usar todo el disco porque es disco virtual.



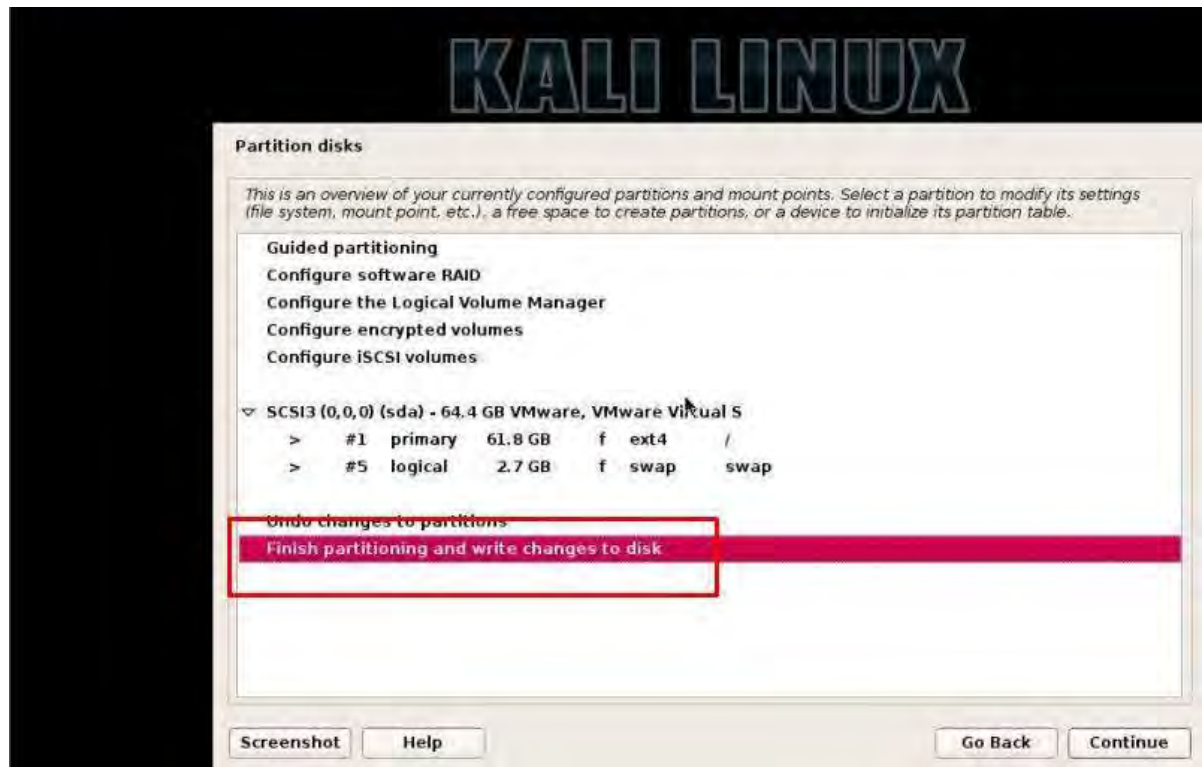
9. Click continuar a la pantalla de partición de discos



10. Elegir todos los archivos en una partición.



11. Seleccionar finalizar partición y escribir cambios en disco



12. Seleccionar si para guardar cambios en el disco





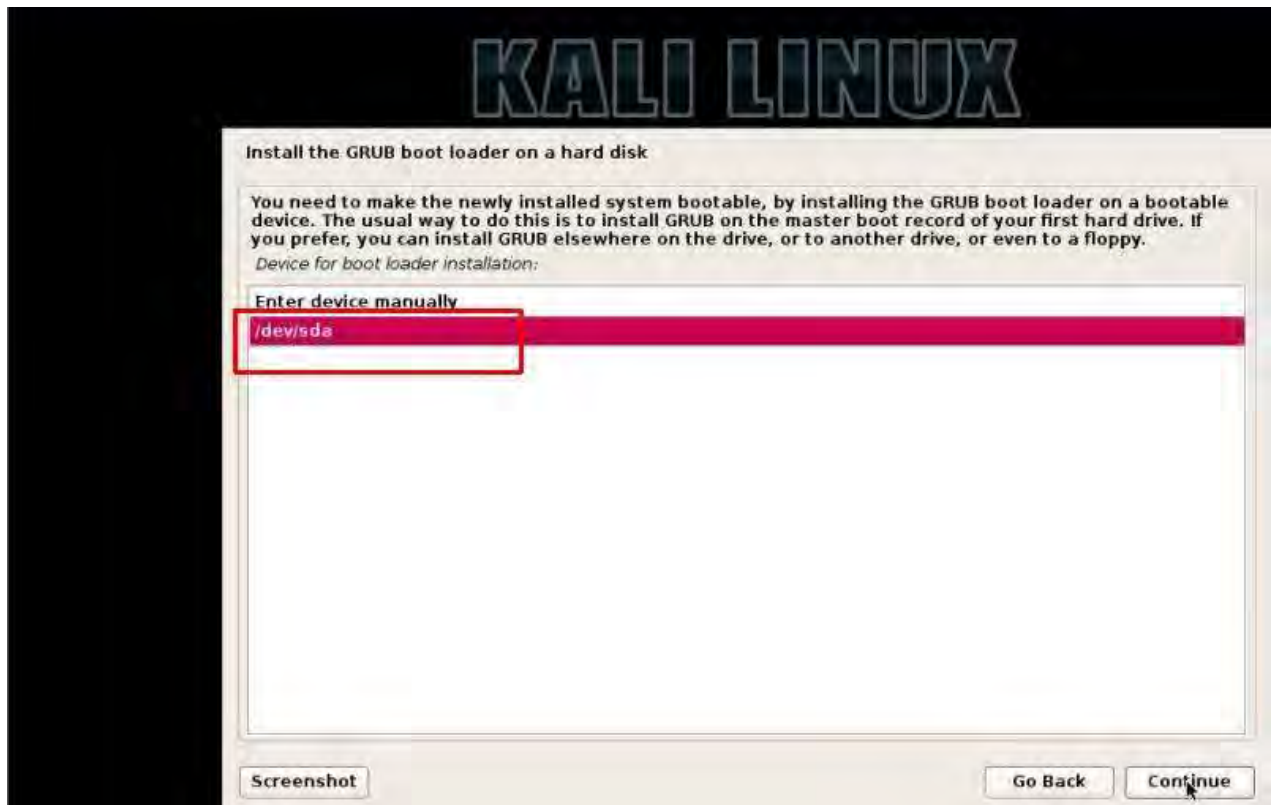
13. Elija Sí para configurar el gestor de paquetes de modo que una réplica de red se puede utilizar para las actualizaciones disponibles.



14. Elija Sí para instalar el gestor de arranque en el disco duro virtual. Este es un paso muy importante, de lo contrario su máquina virtual no se iniciará.



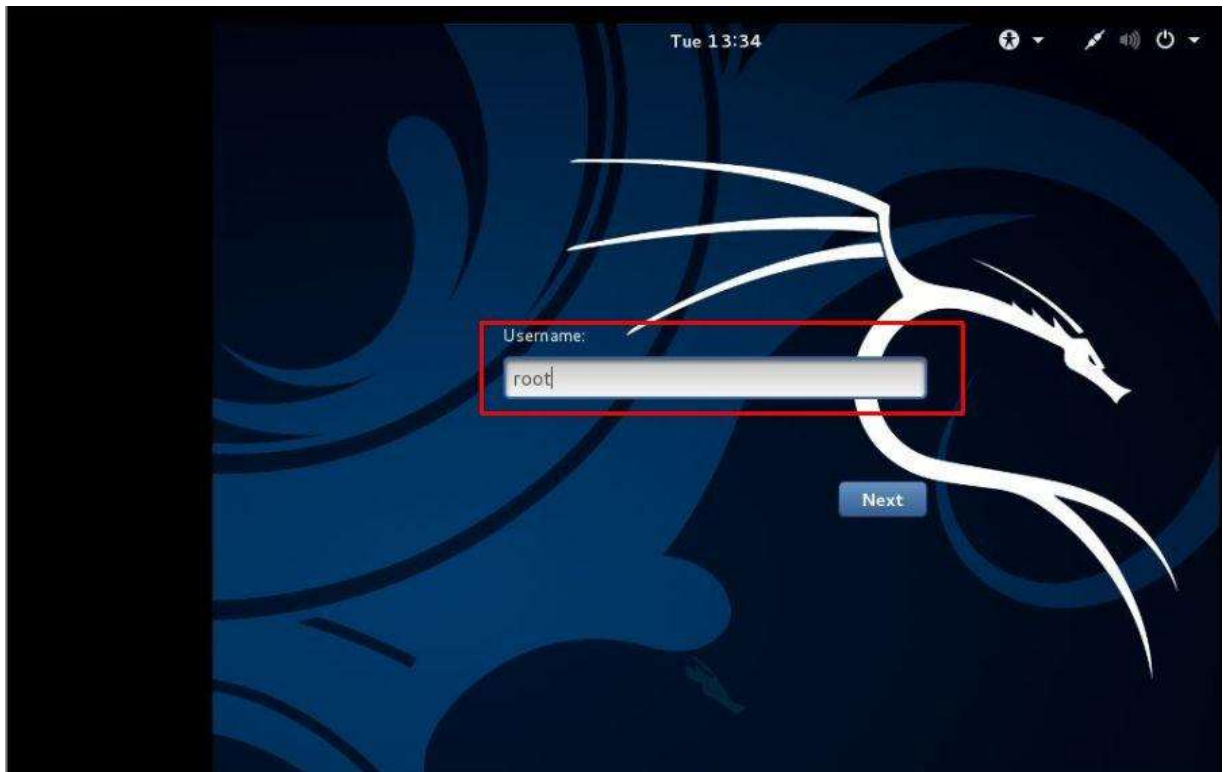
15. Elegir `/dev/sda` donde GRUB sera instalado.



16. Click continuar para finalizar la instalación



17. Ingresa a su máquina virtual Kali Linux 2 con el usuario y la contraseña root.



Referencia de instalación: (System Admins Howto, 2016)



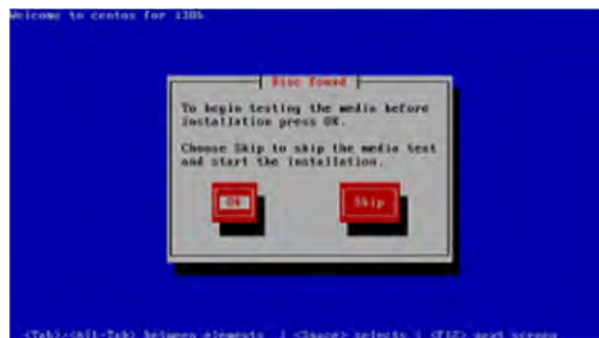
Anexo II - Instalación CentOS 6

La instalación de CentOS 6 se detalla a continuación:

Primero elegimos que tipo de instalación vamos a realizar, en modo gráfico (opción 1) o modo consola (opción 2), en nuestro caso elegiremos la opción 1,



Ahora nos indica si queremos realizar un Test del CD para verificar que esta ok y puedes realizar la instalación, en nuestro caso pulsamos Skip,

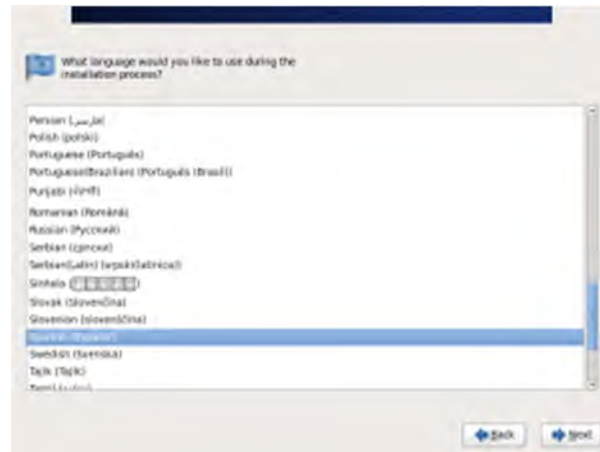


Pulsamos next en la pantalla de Bienvenida,

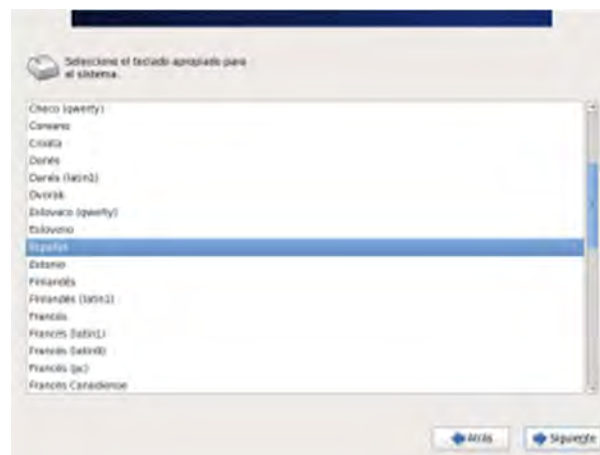




Elegimos el idioma en el que vamos a realizar la instalación, la realizaremos en español,



Ahora el idioma del teclado, en nuestro caso español,



Seleccionamos el tipo de almacenamiento básico, la primera opción indica que vamos a realizar la instalación en un disco local, si elegimos la 2ª opción sería si tuviésemos un almacenamiento externos tipo SAN, en nuestro caso elegimos la 1ª opción,





En esta pantalla nos indica que quiere reinicializar el disco al ser nuevo, como no hay datos se puede hacer sin riesgo de pérdida de información, pulsamos reinicializar todo,



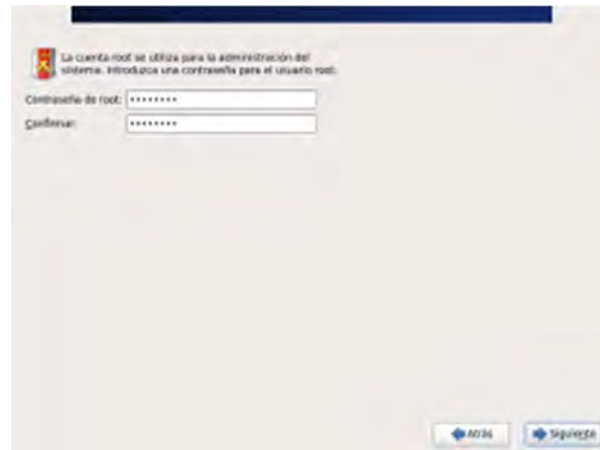
Ahora indicamos el nombre de host que tendrá la computadora en la red,



Seleccionamos la zona horaria,



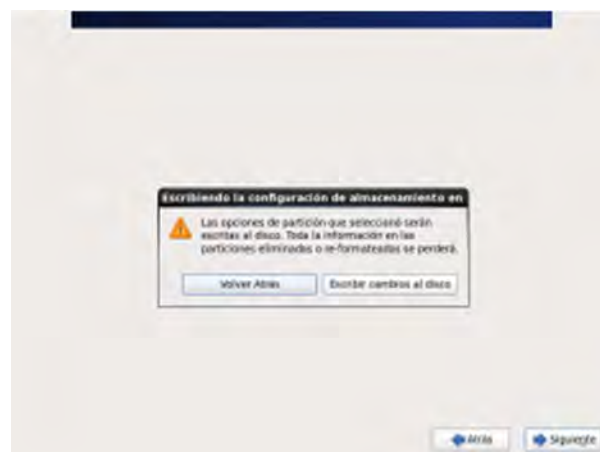
Ahora nos solicita la contraseña de root,



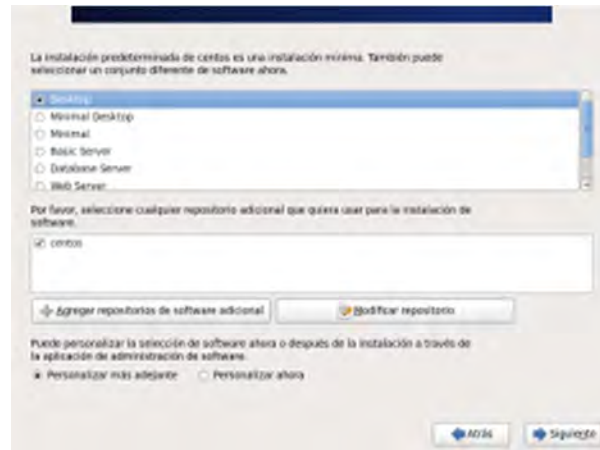
Elegimos el tipo de particionamiento, en nuestro caso vamos a usar todo el espacio,



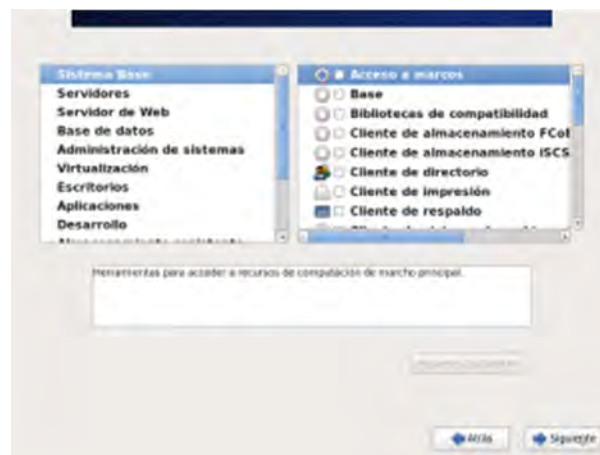
Escribimos los cambios en el disco, al hacer esto formateará la unidad,



Este paso es importante, por defecto centos realiza una instalación mínima, como queremos que la instalación sea con entorno gráfico elegimos la opción Desktop,



A la vez existen dos tipos opciones para instalar: “Personalizar más adelante” y “Personalizar ahora”, en esta última se pueden instalar diversas herramientas complementarias.



Comienza el proceso de instalación, tardará un poco en función de lo que hayamos elegido a instalar,



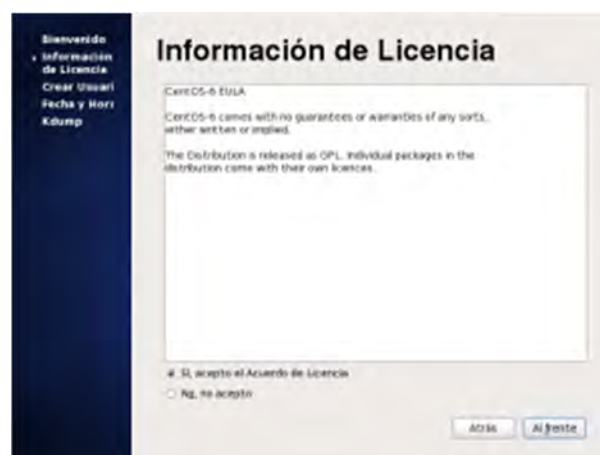
Una vez terminado el proceso de instalación, reiniciamos,



Al reiniciar después de realizar la instalación nos pide realizar unos siguientes pasos antes de iniciar el sistema,



Aceptamos el acuerdo de licencia,



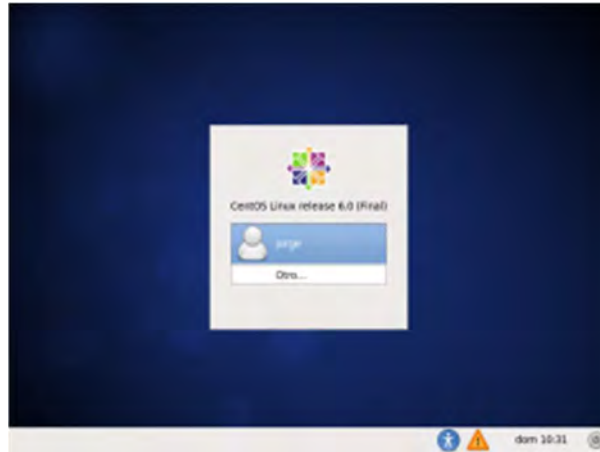
Creamos un usuario para uso no administrativo,



Ingresamos la hora del sistema nuevamente,

En la última opción nos solicita habilitar o no Kdump, Kdump es un mecanismo de volcado de fallos del kernel en caso de fallo, se encargará de recopilar la información del sistema para poder evaluar el fallo, en mi caso no voy a habilitarlo,

Vemos la pantalla de inicio de Centos 6,



Y por último os muestro el escritorio del S.O.

Referencia de instalación: (Martos, 2011)



Anexo III - Instalación IDS Suricata

1- Instalación de Suricata IDS:

-Instalar EPEL (Repositorio de librerías y archivos)

```
sudo rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-68.noarch.rpm
```

```
sudo rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

```
yum install epel-release
```

-Pre-Requisitos

```
yum -y install libpcap libpcap-devel libnet libnet-devel pcre \
```

```
pcre-devel gcc gcc-c++ automake autoconf libtool make libyaml \
```

```
libyaml-devel zlib zlib-devel file-devel
```

```
sudo yum -y install gcc libpcap-devel pcre-devel libyaml-devel file-devel \
```

```
zlib-devel jansson-devel nss-devel libcap-ng-devel libnet-devel tar make \
```

```
libnetfilter_queue-devel lua-devel
```

-Crear Directorios

```
mkdir /etc/suricata
```

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

-Descargar e instalar suricata

```
cd /usr/local/src/
```

```
wget http://www.openinfosecfoundation.org/download/suricata-2.0.5.tar.gz
```

```
tar -xvzf suricata-2.0.5.tar.gz
```

```
cd suricata-2.0.5
```



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

```
make
```

```
sudo make install
```

```
sudo ldconfig
```

```
sudo cp suricata.yaml /etc/suricata
```

-Obtener las Reglas estándar del IDS Suricata

```
cd /etc/suricata
```

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

```
tar -xzf emerging.rules.tar.gz
```

```
cd rules
```

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

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

```
sudo cp sid-msg.map /etc/suricata
```

```
sudo cp gen-msg.map /etc/suricata
```

-Configuración del archivo suricata.yaml

```
classification-file: /etc/suricata/classification.config
```

```
reference-config-file: /etc/suricata/reference.config
```

```
default-rule-path: /usr/local/etc/suricata/rules
```

```
default-log-dir: /var/log/suricata/
```

-Puesta en Marcha del IDS

```
suricata -c /etc/suricata/suricata.yaml -i eth0 --init-errors-fatal
```

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

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



-Chequeamos su funcionamiento

/var/log/suricata/fast.log

tail -f /var/log/suricata/fast.log

2- Instalación Barnyard2 (Recolector de alertas del IDS)

-Pre-Requisitos

yum install mysql mysql-devel git libtool -y

-Instalar MYSQL

sudo yum install mysql-server

sudo /sbin/service mysqld start

sudo /usr/bin/mysql_secure_installation (se requiere clave de usuario ROOT)

-Instalar Barnyard2

cd /usr/local/src

wget https://github.com/firnsy/barnyard2/archive/v2-1.13.tar.gz -O barnyard2-2.1.13.tar.gz

tar zxvf barnyard2-2-1.13.tar.gz

cd barnyard2-2-1.13

autoreconf -fvi -I ./m4

./configure --with-mysql --with-mysql-libraries=/usr/lib/mysql (sino funciona probar con ./configure --with-mysql o solo ./configure)

make

sudo make install

mkdir /etc/barnyard2



```
mkdir /var/log/barnyard2
```

```
cd /usr/local/src/barnyard2/etc
```

```
sudo cp barnyard2.conf /etc/barnyard2
```

-Editar el archivo barnyard2.conf

```
config reference_file: /etc/suricata/reference.config
```

```
config classification_file: /etc/suricata/classification.config
```

```
config gen_file: /etc/suricata/gen-msg.map
```

```
config sid_file: /etc/suricata/sid-msg.map
```

-Configuración

Crear usuario suricata:

```
useradd -M --shell /bin/false --comment "Suricata IDP account" suricata
```

```
sudo chown suricata.suricata /var/log/barnyard2
```

```
sudo touch /var/log/suricata/barnyard2.waldo
```

```
sudo chown suricata.suricata /var/log/suricata/barnyard2.waldo
```

```
sudo touch /etc/suricata/sid-msg.map
```

3- Instalación de Snorby (Dashboard que permite visualizar de forma gráfica las alertas generadas por el IDS)

-Pre-Requisitos

```
yum groupinstall "Development Tools"
```

```
yum install openssl-devel readline-devel libxml2-devel libxslt-devel mysql mysql-devel  
mysql-libs mysql-server urw-fonts libX11-devel libXext-devel fontconfig-devel  
libXrender-devel unzip wget xorg-x11-server-Xvfb libyaml libyaml-devel gdbm-devel tcl-  
devel db4-devel libffi libffi-devel
```



-Descargar e Instalar ImageMagick

cd /opt

tar xvfz ImageMagick.tar.gz

cd ImageMagick-*

./configure

make

make install

ldconfig /usr/local/lib

-Descargar e Instalar wkhtmltopdf

cd /opt

tar xvf wkhtmltox-linux-amd64_0.12.0-03c001d.tar.xz

ln -s /opt/wkhtmltox/lib/libwkhtmltox.so.0 /usr/local/lib/libwkhtmltox.so.0

ln -s /opt/wkhtmltox/bin/wkhtmltopdf /usr/local/bin/wkhtmltopdf

ldconfig /usr/local/lib

-Instalar Ruby

cd /root

mkdir -p rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}

wget ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p484.tar.gz -P rpmbuild/SOURCES

wget https://raw.githubusercontent.com/imeyer/ruby-1.9.3-rpm/master/ruby19.spec -P rpmbuild/SPECS

rpmbuild -bb rpmbuild/SPECS/ruby19.spec (Editar primero el archivo ruby19.specs en la variable %define rubyminorver p484)

yum localinstall /root/rpmbuild/RPMS/x86_64/ruby-1.9.3p484-1.el6.x86_64.rpm



-Instalar Bundler y Snorby

cd /opt

gem install bundler

**wget -O snorby.zip --no-check-certificate
https://github.com/Snorby/snorby/archive/master.zip**

unzip snorby.zip

cd snorby-master

-Crear BD Snorby

mysql -u root -p

Enter password:

mysql> create database snorby;

Query OK, 1 row affected (0.14 sec)

mysql> grant all privileges on snorby.* to snorby@localhost identified by 'snorby';

Query OK, 0 rows affected (0.06 sec)

Exit

-Configuración, hacer una copia de los archivos database.yml.example y snorby_config.yml.example dentro del directorio de snorby y renombrarlos a database.yml y snorby_config.yml

-database.yml

//This is the content of my file

snorby: &snorby

adapter: mysql

username: snorby

password: "snorby"



host: localhost

-Snorby_config.yml

development:

domain: localhost:3000

wkhtmltopdf: /usr/bin/wkhtmltopdf

test:

domain: localhost:3000

wkhtmltopdf: /usr/bin/wkhtmltopdf

production:

domain: localhost:3000

wkhtmltopdf: /usr/bin/wkhtmltopdf

"Editar GeMfile, se encuentra en carpeta Snorby"

Cambiar: gem 'rake', '0.9.2' a gem 'rake', '> 0.9.2'

Agregar: gem 'thin' despues de gem 'json':

gem 'json', '~> 1.7'

gem 'thin'

Comentar la linea gem 'thin' dentro del grupo (:development) usando a #. Los archivos se parecerán a lo siguiente:

group(:development) do

gem "letter_opener"

gem 'thin'

end

Agregar: gem 'orm_adapter' despues de gem 'netaddr':

gem 'netaddr', '~> 1.5.0'



gem 'orm_adapter'

"Editar Gemfile.lock:"

Cambiar la linea rake (0.9.2) a rake (0.9.2.2)

-Reinstalar Ruby a la version 2.1.1

wget ftp://ftp.ruby-lang.org/pub/ruby/2.1/ruby-2.1.1.tar.gz

tar xvzf ruby-2.1.1.tar.gz

./configure --prefix=/usr

make

make install

-Instalar Ruby Gems

wget http://production.cf.rubygems.org/rubygems/rubygems-1.8.25.tgz

tar xvzf rubygems-1.8.25.tgz

cd rubygems-1.8.25

ruby setup.rb config

ruby setup.rb setup

ruby setup.rb install

En gemfile.lock cambiar nokogiri a la version 1.5.10

gem install bundle

bundle install

rake snorby:setup

rails server thin -e production

-Instalar Apache2 Server (httpd)

yum install httpd



Abrir el archivo `/etc/httpd/conf/httpd.conf` .

Encontrar

#ServerName www.example.com:80” (line no : 276) .

#ServerName www.example.com:80

Y debajo agregar lo siguiente:

#ServerName www.example.com:80

ServerName 192.168.141.22:80 (Poner la IP de nuestra maquina)

service httpd start

-Instalar Passenger

gem install passenger

cd /usr/lib/ruby/gems/2.1.0/gems/passenger-5.0.25/bin

./passenger-install-apache2-module

Copiar en el archivo `/etc/httpd/conf/httpd.conf`

LoadModule passenger_module /usr/lib/ruby/gems/2.1.0/gems/passenger-5.0.24/buildout/apache2/mod_passenger.so

<IfModule mod_passenger.c>

PassengerRoot /usr/lib/ruby/gems/2.1.0/gems/passenger-5.0.24

PassengerDefaultRuby /usr/bin/ruby

</IfModule>

-Configurar archivo de Apache2

/etc/httpd/conf/httpd.conf

LoadModule passenger_module /usr/lib/ruby/gems/2.1.0/gems/passenger-5.0.24/buildout/apache2/mod_passenger.so



```
<IfModule mod_passenger.c>
```

```
    PassengerRoot /usr/lib/ruby/gems/2.1.0/gems/passenger-5.0.24
```

```
    PassengerDefaultRuby /usr/bin/ruby
```

```
</IfModule>
```

```
<VirtualHost *:80>
```

```
    ServerAdmin webmaster@localhost
```

```
    ServerName snorby.Server2
```

```
    DocumentRoot /var/www/html/snorby/public
```

```
<Directory "/var/www/html/snorby/public">
```

```
    AllowOverride all
```

```
    Order deny,allow
```

```
    Allow from all
```

```
    Options -MultiViews
```

```
</Directory>
```

```
</VirtualHost>
```

-Archivo /etc/suricata/Barnyard2.conf

output database: log, mysql, user=snorby password=snorby /

dbname=snorby host=192.168.222.132 sensor_name=sensor1

-Salir de todo y tipear el comando:

service httpd restart

-Ir a /var/www/html/snorby/config/snorby_config.yml y cambiar donde dice mailserver y poner: snorby@snorby.org



-Ir a /var/www/html/snorby y ejecutar:

sudo bundle update activesupport railties rails

sudo gem install arel ezprint && sudo bundle install

sudo bundle exec rake snorby:setup

-Después vamos a mysql y ejecutamos:

mysql -u root -p

mysql> create user 'admin'@'localhost' identified by 'admin_password';

Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on snorby.* to 'admin'@'localhost' with grant option;

Query OK, 0 rows affected (0.02 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

mysql>exit

-Después a /var/www/html/snorby/config/database.yml y cambiar por lo siguiente:

snorby: &snorby

adapter: mysql

username: admin

password: "admin_password"

host: localhost

- Luego ir a /etc/my.conf y agregar al final del archivo:

#

Instead of skip-networking the default is now to listen only on

localhost which is more compatible and is not less secure.

#bind-address = 127.0.0.1

-Después ejecutar:



sudo service mysql restart

sudo lsof -i | grep mysqld

-Crear el siguiente archivo en el directorio correspondiente /etc/apache2/mods-available/passenger.load

-Ejecutar:

sudo find / -name "*mod_passenger*"

-Devolverá la ruta de mod_passenger.so , copiar y pegar esa ruta en el archivo passenger.load creado anteriormente:

LoadModule **passenger_module**
/var/lib/gems/1.9.1/gems/passenger3.0.17/ext/apache2/mod_passenger.so

<IfModule mod_passenger.c>

PassengerRoot /var/lib/gems/1.9.1/gems/passenger-3.0.17

PassengerRuby /usr/bin/ruby

</IfModule>

-Guardar y reiniciar el servicio httpd (apache2) con el siguiente comando:

service httpd restart

-Luego ir a /etc/httpd/conf/httpd.conf y al último agregar

<VirtualHost *:80>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html/snorby/public

<Directory />

Options FollowSymLinks



AllowOverride None

</Directory>

<Directory "/var/www/html/snorby/public">

AllowOverride all

#Order deny,allow

#Allow from all

Options -MultiViews

</Directory>

</VirtualHost>

-Después ir al directorio /var/www/html/snorby y ejecutar:

sudo bundle pack && sudo bundle install --path vender/cache

service httpd restart

- Finalmente ir al navegador y poner en la url: localhost para ejecutar Snorby dashboard.

Referencias de instalación: (Open Information Security Foundation, 2012), (Miessler, 2012), (Mun, 2012), (N40LAB, 2014), (Savir, 2013)



Anexo IV – Instalación BRO Network Security Monitor

REQUISITOS PRE-INSTALACION

Instalar las siguientes librerías para el correcto funcionamiento de BRO

- Libpcap
- OpenSSL libraries
- BIND8 library
- Libmagic 5.04 or greater
- Libz
- Bash (for BroControl)
- Python (for BroControl)

```
#yum install cmake make gcc gcc-c++ flex bison libpcap-devel openssl-devel python-  
devel swig zlib-devel file-devel
```

INSTALACION Y CONFIGURACION DE BRO

Ejecutar los siguientes comandos:

```
#cd /usr/local/src  
#wget http://www.bro.org/downloads/release/bro-2.2.tar.gz  
#tar xzf bro-2.2.tar.gz  
#cd bro-2.2  
#./configure --enable-debug --enable-perftools --prefix=/usr/local/bro --with-xqilla  
#make  
#make install  
#export PATH=/usr/local/bro/bin:$PATH
```

ESTRUCTURA DEL ARCHIVO DE BRO

Ejecutar:

```
#cd /usr/local/bro/
```

/usr/local/bro es el directorio principal de instalación. Algunas veces llamado \$PREFIX

PROBANDO BRO

Para nuestro primer script - vamos a utilizar probablemente el evento más común en bro - una conexión TCP.



```
#mkdir /usr/local/bro/tmp
#cd /usr/local/bro/tmp

#vi hello.bro

event connection_established(c:connection)
{
    print fmt("Hello Bro");
}

:wq!
```

#bro -i eth0 hello.bro (eth0 nuestra interfaz de red)

Este script mostrara “Hello Bro”, cada vez que se produzca una conexión TCP.

ADMINISTRANDO BRO CON BROCONTROL:

BroControl es un shell interactivo para una fácil operación/gestión de instalaciones Bro en un solo sistema o incluso a través de múltiples sistemas de un clúster para monitorización del tráfico.

Estos son los cambios de configuración básicas para hacer una instalación mínima de BroControl que gestionará una única instancia Bro en el host local :

En **\$PREFIX/etc/node.cfg**, configurar la interfaz de red de nuestra máquina.

```
#vi $PREFIX/etc/node.cfg
```

En **\$PREFIX/etc/networks.cfg**, comentar los ajustes por defecto y añadir las redes que Bro considerará local para el medio ambiente supervisado.

Ahora iniciamos BroControl shell de la siguiente manera:

```
#broctl
```

Dado que este es el primer uso de shell, debemos realizar una instalación inicial de la configuración BroControl:

```
#[BroControl]> install
```

Luego ejecutar una instancia de bro:

```
#[BroControl]> start
```



Por ultimo:

#[BroControl]> stop

Referencia de instalación: (LINUX DIGEST, 2014)



Glosario

Algoritmos: “Un Algoritmo, se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas”.

Anomalías: “Cambio o desviación respecto de lo que es normal”.

Árbol de decisión: “Un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento”.

Ataques Informáticos: “Consiste en intentar hacer daño la información de un sistema”.

Ciber delincuencia: “Es la delincuencia o el cometimiento de delitos con la ayuda de la tecnología como ordenadores y redes de comunicación”.

Clasificación (minería de datos): “Encuentra modelos (funciones) que describen y distinguen clases o conceptos para futuras predicciones”.

Conocimiento: “El conocimiento es un conjunto de información almacenada mediante la experiencia o el aprendizaje (a posteriori), o a través de la introspección (a priori). En el sentido más amplio del término, se trata de la posesión de múltiples datos interrelacionados que, al ser tomados por sí solos, poseen un menor valor cualitativo”.

Data mining: “Es la tecnología que permite la extracción de conocimiento de un conjunto de datos”.

Falso negativo: En Informática es un error por el cual el software antivirus falla en detectar un archivo o área del sistema que está realmente infectada.

Falsos positivos: “En Informática es un error por el cual un software de antivirus informa que un archivo o área de sistema está infectada, cuando en realidad el objeto está limpio de virus; Esto también ocurre en los navegadores de internet cuando se descarga un archivo”.

Intrusión: “Introducción en una propiedad, lugar, asunto o actividad sin tener derecho o autorización para ello”.

Matriz de confusión: “Una *matriz de confusión*, también llamada *matriz de clasificación*, ordena todos los casos del modelo en categorías, determinando si el valor de predicción coincide con el valor real”.

Modelado: “Un modelo es por tanto una representación parcial o simplificada de la realidad que recoge aquellos aspectos de relevancia para las intenciones del modelador, y de la que se pretende extraer conclusiones de tipo predictivo. Se modela para comprender mejor o explicar mejor un proceso o unas observaciones”.



Modelo: “Un modelo es una representación de un objeto, sistema o idea, de forma diferente al de la entidad misma. El propósito de los modelos es ayudarnos a explicar, entender o mejorar un sistema. Un modelo de un objeto puede ser una réplica exacta de éste o una abstracción de las propiedades dominantes del objeto”.

Patrones: “Tipo de tema de sucesos u objetos recurrentes”.

Predicción: “Del latín ‘praedictio’, una predicción es una expresión que anticipa aquello que, supuestamente, va a suceder. Se puede predecir algo a partir de conocimientos científicos, relevaciones de algún tipo, hipótesis o indicios”.

Prevención: “Del latín ‘praeventio’, prevención es la acción y efecto de prevenir (preparar con antelación lo necesario para un fin, anticiparse a una dificultad, prever un daño”.

Reglas: “Es el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de la misma”.

Técnicas: “La palabra técnica proviene de ‘téchne’, un vocablo de raíz griega que se ha traducido al español como “arte” o “ciencia”. Esta noción sirve para describir a un tipo de acciones regidas por normas o un cierto protocolo que tiene el propósito de arribar a un resultado específico”.



Bibliografía

- Alegsa, L. (12 de 05 de 2015). *ALEGSA*. Recuperado el 13 de 08 de 2015, de ALEGSA:
<http://www.alegsa.com.ar/Dic/ataque%20informatico.php>
- Alexa. (08 de 08 de 2008). *SlideShare*. Recuperado el 10 de 6 de 2016, de SlideShare:
<http://es.slideshare.net/alexace84/poda-arboles>
- Arocena, G. A. (24 de 8 de 2015). *www.egov.ufsc.br*. Obtenido de *www.egov.ufsc.br*:
http://www.egov.ufsc.br/portal/sites/default/files/la_regulacion_de_los_delitos_informaticos_en_el_codigo.pdf
- ATGC. (28 de 4 de 2008). *Advanced Tech*. Recuperado el 21 de 5 de 2016, de Advanced Tech: <https://advancedtech.wordpress.com/2008/04/14/clasificacion-supervisada-y-no-supervisada/>
- Ayuso , M. A., & Barcenilla Mancha , M. Á. (07 de 01 de 2008). *www.it.uc3m.es*. Recuperado el 01 de 10 de 2016, de *www.it.uc3m.es*:
<http://www.it.uc3m.es/jvillena/irc/practicas/07-08/IntrusionesDeRed.pdf>
- Ayuso, M. A. (2010). *IT UC3M*. Obtenido de IT UC3M:
<http://www.it.uc3m.es/jvillena/irc/practicas/07-08/IntrusionesDeRed.pdf>
- Britos. (2010). *postgrado.info.unlp.edu.ar/*. Recuperado el 06 de 10 de 2015, de *postgrado.info.unlp.edu.ar/*:
http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Redes_de_Datos/Tesis/Britos_Jose_Daniel.pdf
- Cacti. (23 de 07 de 2016). *www.cacti.net*. Recuperado el 23 de 07 de 2016, de *www.cacti.net*:
<http://www.cacti.net/>
- Campoy. (2011). *openaccess.uoc.edu*. Recuperado el 06 de 10 de 2015, de
http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8167/1/dcampoy_TFM_0611.pdf
- Caparrini, F. S. (16 de 07 de 2015). *CS*. Recuperado el 20 de 5 de 28, de CS:
<http://www.cs.us.es/~fsancho/?e=75>
- ccm. (29 de 01 de 2003). *CCM - Comunidad Informatica*. Recuperado el 12 de 08 de 2015, de CCM - Comunidad Informatica: <http://es.ccm.net/contents/162-sistema-de-deteccion-de-intrusiones-ids>



- ccm. (01 de 04 de 2016). *CCM - Comunidad Informática*. Recuperado el 08 de 05 de 2016, de CCM - Comunidad Informática: <http://es.ccm.net/contents/22-ataque-por-denegacion-de-servicio>
- ConocimientosWeb. (04 de 10 de 2013). *Conocimientos Web*. Recuperado el 25 de 04 de 2016, de Conocimientos Web: <http://www.conocimientosweb.net/descargas/article323.html>
- Crece Negocios. (17 de 07 de 2014). *www.crecenegocios.com*. Recuperado el 15 de 09 de 2016, de *www.crecenegocios.com*: <http://www.crecenegocios.com/el-van-y-el-tir/>
- Cruiz. (06 de 2014). *Upct*. Recuperado el 22 de 05 de 2016, de Upct: www.dmae.upct.es/~mcruiz/Telem06/Teoria/arbol_decision.pdf
- definicion.mx. (22 de 04 de 2017). *definicion.mx*. Obtenido de definicion.mx: <https://definicion.mx/hoja-de-calculo/>
- Donostia. (2010). *Ibai Gurrutxaga Goikoetxea*. Recuperado el 02 de 07 de 2016, de Ibai Gurrutxaga Goikoetxea: <https://addi.ehu.es/bitstream/10810/13910/1/2010%20Gurrutxaga%20I.pdf>
- Esteban, S. (30 de 01 de 2016). *backtrackacademy.com*. Recuperado el 30 de 09 de 2016, de *backtrackacademy.com*: <https://backtrackacademy.com/posts/hydra-ataques-de-fuerza-bruta-online>
- Foro Kali Linux. (24 de 03 de 2013). *kalilinux.foroactivo.com*. Recuperado el 01 de 10 de 2016, de *kalilinux.foroactivo.com*: <http://kalilinux.foroactivo.com/t12-tutorial-nmap-para-kali-linux>
- Fuentes, A. (26 de 10 de 2014). *Scribd*. Recuperado el 2 de 06 de 2016, de Scribd: <https://es.scribd.com/doc/259289518/Validacion-cruzada>
- Fundacion Carlos Slim. (2009). *Académica*. Recuperado el 23 de 09 de 2015, de Académica: <http://cursoslibres.academica.mx/206/seguridad-en-redes/1-principios-de-seguridad-en-redes-de-datos/seguridad-en-redes-de-datos>
- Fundacion Carlos Slim. (2009). *Académica*. Recuperado el 23 de 09 de 2015, de Académica: <http://cursoslibres.academica.mx/206/seguridad-en-redes/1-principios-de-seguridad-en-redes-de-datos/politicas-y-procedimientos>
- Fundacion Carlos Slim. (2009). *Académica*. Recuperado el 23 de 09 de 2015, de Académica: <http://cursoslibres.academica.mx/206/seguridad-en-redes/1-principios-de-seguridad-en-redes-de-datos/1.1seguridadenred.jpg>
- Gerardo, M. (25 de 11 de 2010). *Scribd*. Recuperado el 10 de 06 de 2016, de Scribd: <https://es.scribd.com/doc/57484779/Algoritmo-c45-Arboles-de-Decision>



- Herrera, L. F. (01 de 30 de 2009). *ServidoresParedes*. Recuperado el 13 de 08 de 2015, de ServidoresParedes: <http://servidorespararedes.blogspot.com.ar/2009/01/que-es-aplicaciones-web.html>
- Highsec. (19 de 08 de 2014). *highsec.es*. Recuperado el 30 de 07 de 2016, de highsec.es: <http://highsec.es/2014/06/introduccion-a-kismet/>
- Information Security Center of eXcellence. (2015). <http://www.unb.ca/>. Recuperado el 10 de 10 de 2016, de <http://www.unb.ca/>: <http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html>
- Ireo. (23 de 07 de 2016). *www.ireo.com*. Recuperado el 23 de 07 de 2016, de <http://www.ireo.com/>: <http://www.ireo.com/fabricantes-y-productos/manageengine/opmanager/resumen/>
- LINUX DIGEST. (07 de 05 de 2014). *sathisharthars.wordpress.com*. Recuperado el 02 de 07 de 2016, de sathisharthars.wordpress.com: <https://sathisharthars.wordpress.com/2014/05/07/installing-and-configuring-bro-nids-in-centos-6/>
- Maestros del Web. (19 de 08 de 2003). *www.maestrosdelweb.com*. Recuperado el 23 de 07 de 2016, de www.maestrosdelweb.com: <http://www.maestrosdelweb.com/snort/>
- Mahbod Tavallaee, E. B. (2009). *Ryerson*. Recuperado el 02 de 07 de 2016, de Ryerson: <http://www.ee.ryerson.ca/~bagheri/papers/cisda.pdf>
- Martos, J. G. (25 de 07 de 2011). *jorgegonzalezmartos.wordpress.com*. Recuperado el 02 de 07 de 2016, de jorgegonzalezmartos.wordpress.com: <https://jorgegonzalezmartos.wordpress.com/2011/07/25/instalacion-basica-centos-6-modo-grafico/>
- Miessler, D. (18 de 10 de 2012). *danielmiessler.com*. Recuperado el 02 de 02 de 2016, de danielmiessler.com: <https://danielmiessler.com/blog/building-ids-centos-using-suricata/https://cyruslab.net/2012/10/18/building-an-ids-part-1-installing-pre-requisites-and-snorby/>
- Miranda, C. V. (2015). *Redes Telemáticas*. Paraninfo, SA.
- Moine, J. M. (05 de 2013). *Sedici*. Recuperado el 28 de 09 de 2015, de Sedici: http://sedici.unlp.edu.ar/bitstream/handle/10915/29582/Documento_completo.pdf?sequence=1
- Mun, C. L. (18 de 10 de 2012). *cyruslab.net*. Recuperado el 02 de 07 de 2016, de cyruslab.net: <https://cyruslab.net/2012/10/18/building-an-ids-part-1-installing-pre-requisites-and-snorby/>



- Munin. (23 de 07 de 2016). *www.munin-monitoring.org*. Recuperado el 23 de 07 de 2016, de *www.munin-monitoring.org*: <http://munin-monitoring.org/>
- N40LAB. (25 de 02 de 2014). *n40lab.wordpress.com*. Recuperado el 02 de 07 de 2016, de *n40lab.wordpress.com*: <https://n40lab.wordpress.com/2014/02/25/snorby-in-centos-6-5/>
- Nagios - The Industry Standard In IT Infrastructure Monitoring. (23 de 07 de 2016). *www.nagios.org*. Recuperado el 23 de 07 de 2016, de <https://www.nagios.org/>
- Nebrija. (2016). *Nebrija*. Recuperado el 20 de 05 de 2016, de Nebrija: https://www.nebrija.es/~cmalagon/inco/apuntes_mios/arboles_de_decision.pdf
- nmap.org. (01 de 10 de 2016). *nmap.org*. Recuperado el 01 de 10 de 2016, de *nmap.org*: <https://nmap.org/man/es/man-briefoptions.html>
- Open Information Security Foundation. (04 de 09 de 2012). *redmine.openinfosecfoundation.org*. Recuperado el 02 de 07 de 2016, de *redmine.openinfosecfoundation.org*: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/suricata_snorby_and_barnyard2_set_up_guide
- Ortiz, A. (30 de 11 de 2001). *florbe.com*. Recuperado el 10 de 10 de 2016, de *florbe.com*: <http://florbe.com/pe/compendio-computacion-e-informatica/warez>
- Ossec. (30 de 07 de 2016). *ossec.github.io*. Recuperado el 30 de 07 de 2016, de *ossec.github.io*: <http://ossec.github.io/>
- Perona, I. (23 de 01 de 2015). *github.com*. Recuperado el 21 de 05 de 2016, de *github.com*: <https://github.com/inigoperona/tcpdump2gureKDDCup99/blob/master/darpa2gurekddcup.bro>
- Piatetsky, G. (28 de 10 de 2014). *kdnuggets*. Recuperado el 29 de 09 de 2015, de *kdnuggets*: <http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html>
- Piatetsky, G. (03 de 06 de 2015). *Kdnuggets*. Recuperado el 29 de 09 de 2015, de *kdnuggets*: <http://www.kdnuggets.com/2013/06/kdnuggets-annual-software-poll-rapidminer-r-view-for-first-place.html>
- Quantdare. (07 de 01 de 2015). *Artificial Intelligence*. Recuperado el 11 de 06 de 2016, de *Artificial Intelligence*: <http://quantdare.com/2015/01/random-forest-vs-simple-tree/>
- Ramos, J. (19 de 04 de 2014). *rootear.com*. Recuperado el 30 de 09 de 2016, de *rootear.com*: <http://rootear.com/ubuntu-linux/ataques-de-fuerza-bruta>



- Renault, V. (2016). <http://ctb.ku.edu>. Recuperado el 08 de 04 de 2017, de <http://ctb.ku.edu>.
- Savir, L. (11 de 12 de 2013). <http://www.emind.co/>. Recuperado el 02 de 07 de 2016, de <http://www.emind.co/>: <http://www.emind.co/how-to/how-to-install-barnyard2/>
- Security Onion. (30 de 07 de 2016). securityonion.net. Recuperado el 30 de 07 de 2016, de securityonion.net: <https://securityonion.net/>
- Seguridad y Redes. (01 de 12 de 2010). seguridadyredes.wordpress.com. Obtenido de seguridadyredes.wordpress.com: <https://seguridadyredes.wordpress.com/2010/12/01/snort-snorby-un-front-end-para-analisis-y-gestian-de-alertas-para-snort/>
- Seguridad y Redes. (23 de 03 de 2011). seguridadyredes.wordpress.com. Recuperado el 30 de 07 de 2016, de seguridadyredes.wordpress.com: <https://seguridadyredes.wordpress.com/2011/03/23/bro-ids-un-sistema-de-deteccion-de-intrusiones-basado-en-politicas-especializadas/>
- Snort. (23 de 07 de 2016). www.snort.org. Recuperado el 23 de 07 de 2016, de www.snort.org: <https://www.snort.org/>
- Soporte TI - Canal YouTube. (25 de 04 de 2011). SoporteTI. Recuperado el 25 de 04 de 2016, de blog.soporteti.net: <https://blog.soporteti.net/2011/04/25/networking-en-vmware-modos-bridged-y-nat-video-powerpoint/>
- Suica, M. Y., & Naranjo Barragán, R. A. (2014). [dspace.esPOCH.edu.ec](http://dspace.esPOCH.edu.ec/bitstream/123456789/3545/1/18T00561.pdf). Recuperado el 16 de 09 de 2016, de <http://dspace.esPOCH.edu.ec/bitstream/123456789/3545/1/18T00561.pdf>
- Suricata IDS. (23 de 07 de 2016). www.suricata-ids.org. Recuperado el 23 de 07 de 2016, de www.suricata-ids.org: <https://suricata-ids.org/>
- System Admins Howto. (04 de 02 de 2016). sysadminshowto.com. Recuperado el 23 de 07 de 2016, de sysadminshowto.com: <http://www.sysadminshowto.com/how-to-install-kali-linux-2-sana-in-vmware-workstation-11-step-by-step-guide/>
- Tcpdump. (23 de 07 de 2016). www.tcpdump.org. Recuperado el 23 de 07 de 2016, de www.tcpdump.org: <http://www.tcpdump.org/>
- Tecnología, Ingeniería en sistemas. (23 de 03 de 2015). Postecnologia. Recuperado el 28 de 05 de 2016, de Postecnologia: <http://www.postecnologia.com/2015/03/rapid-miner-software-business-intelligence.html>
- Trejos, P. (2010). gredos.usal.es/. Recuperado el 06 de 10 de 2015, de gredos.usal.es/: http://gredos.usal.es/jspui/bitstream/10366/83183/1/DIA_Pin%20z%C3%B3nTrejosCI_Arquitectura%20multiagente.pdf



- Universidad de Almería. (2016). *Adminiso*. Recuperado el 25 de 06 de 2016, de Adminiso:
http://www.adminiso.es/index.php/Snort-OPCIONES_DE_UNA_REGLA
- Universidad Politécnica de Cataluña. (2014). *CS*. Recuperado el 20 de 5 de 2016, de CS:
<http://www.cs.upc.edu/~bejar/ia/transpas/teoria/6-AP-aprendizaje.pdf>
- University of California, Irvine. (28 de 10 de 1999). *kdd.ics.uci.edu/*. Recuperado el 10 de 09 de 2016, de *kdd.ics.uci.edu/*: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
- Untiveros, S. (01 de 07 de 2004). <http://www.AprendaRedes.com>. Recuperado el 28 de 08 de 2015, de <http://www.AprendaRedes.com>.
- Vallejo, & Germán. (2012). *web.usbmed.edu.co*. Recuperado el 06 de 10 de 2015, de <http://web.usbmed.edu.co/usbmed/fing/v3n1/v3n1a6.pdf>
- Vezzoni, D. (22 de 08 de 2013). *g2khosting* . Recuperado el 08 de 05 de 2016, de www.g2khosting.com: <http://www.g2khosting.com/blog/que-diferencia-hay-entre-un-dos-y-un-ddos/>
- Wenjun, L. (10 de 07 de 2010). *IEEE XPLORE - Digital Library*. Recuperado el 06 de 11 de 2015, de IEEE XPLORE - Digital Library:
<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5565749&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D5565749>
- Windump. (23 de 07 de 2016). *www.winpcap.org*. Recuperado el 23 de 07 de 2016, de www.winpcap.org: <https://www.winpcap.org/windump/>
- Wireshark. (23 de 07 de 2016). *www.wireshark.org*. Recuperado el 23 de 07 de 2016, de www.wireshark.org: <https://www.wireshark.org/>
- Zabbix - The Enterprise-class Monitoring Solution for Everyone. (23 de 07 de 2016). www.zabbix.com. Recuperado el 23 de 07 de 2016, de www.zabbix.com: <http://www.zabbix.com/>
- Zenoss - Own IT. (23 de 07 de 2016). *www.zenoss.com*. Recuperado el 23 de 07 de 2016, de www.zenoss.com: <https://www.zenoss.com/>