Universidad Católica De Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería de Sistemas

"ARQUITECTURA DE SEGURIDAD PROFUNDA CONTRA EXPLOTACIÓN DE VULNERABILIDADES MODERNAS EMPLEANDO AES-256 PARA EL MODELO TCP/IP EN REDES DE DATOS IP"

Tesis presentada por el Bachiller:

Nagata Bolivar Toshiro

Para optar el Título Profesional de:

Ingeniero de Sistemas:

Especialidad en Sistemas de Información

Asesora de Tesis:

Mg. Rosas Paredes Karina

Arequipa–Perú

2018

UNIVERSIDAD CATOLICA DE SANTA MARIA
URB. SAN JOSE S/N - UMACOLLO

FACULTAD DE CIENCIAS E INGENIERIAS FISICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

DICTAMEN DE BORRADOR DE TESIS

VISTO

El Borrador de Tesis titulado:

Arquitectura de seguridad profunda contra explotación de vulnerabilidades Zero-days empleando AES-256 para el modelo TCP/IP en redes de datos IP

Presentado por (el) (la) (los) Bachilleres

Toshiro Nagata Bolivar

Nuestro dictamen es:

Procedente

OBSERVACIONES: Ninguna

Arequipa, 15 de junio de 2018

Mario Reyes P.

1221

(5154) 382038 (5154) 252542 ucsm@ucsm.edu.pe http://www.ucsm.edu.pe

01855838

ACTA TITULO PROFESIONAL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

En Arequipa, a los 27 días del mes de JUNIO del 2018 siendo las 18:30 horas, en el local de la Universidad Católica de Santa María, se reunió el Jurado

Presidido por: ING. KARINA ROSAS PAREDES

Integrantes: ING. FREDY RAMIRO DELGADO DELGADO

ING. FERNANDO GERMÁN PAREDES MARCHENA

Actuando este último como Secretario con la finalidad de recibir las previas orales del(os) (la) (s) Señor(ita) Bachiller(s)

NAGATA BOLÍVAR TOSHIRO

Quien(es) pretende(n) optar el Título Profesional de INGENIERO DE SISTEMAS: ESPECIALIDAD EN SISTEMAS DE INFORMACION

Sustentando: LA TESIS

Titulado" ARQUITECTURA DE SEGURIDAD PROFUNDA CONTRA EXPLOTACIÓN DE VULNERABILIDADES MODERNAS EMPLEANDO AES-256 PARA EL MODELO TCP/IP EN REDES DE DATOS IP"

El Presidente del Jurado invitó al (los) Titulando(s) a hacer una exposición de su trabajo, conclusiones y recomendaciones, para luego proceder a realizar las preguntas que los Miembros del Jurado consideraron pertinente plantear. Posteriormente, se pasó a deliberar y emitir su voto en la forma establecida por el Reglamento de Grados y Títulos de la Facultad de Ciencias e Ingenierías Físicas y Formales siendo el resultado el siguiente:

APROBADO CON FELICITACIÓN PÚBLICA

Con lo que se dio por terminado el Acto siendo las 20:00 horas, para dar fe firmamos a continuación los Miembros del Jurado y el (los) Titulado(s).

PRESIDENTE

SECRETARIO

INTEGRANTE

TITULANDO

TITULANDO

OBSERVACIÓN: EL RESULTADO ES "APROBADO POR UNANIMIDAD CON FELICITACIÓN PÚBLICA".

PRESENTACIÓN

Sr. Director de la Escuela Profesional de Ingeniería de Sistemas

DR. Guillermo Enrique Calderón Ruiz

Sres. Miembros del Jurado Examinador de Tesis

Mg. Karina Rosas Paredes

Mg. Fernando Paredes Marchena

De conformidad con las disposiciones del Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, remito a vuestra consideración el presente trabajo de investigación titulado: "Arquitectura de seguridad profunda contra explotación de vulnerabilidades modernas empleando AES-256 para el modelo TCP/IP en redes de datos IP", el mismo que de ser aprobado me permitirá optar el Título Profesional de Ingeniero de Sistemas.

Arequipa, junio del 2018

Toshiro Nagata Bolivar

DEDICATORIA

Agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera y brindarme una vida llena de aprendizajes y experiencias.

Le doy gracias a mi madre Juana Bolivar, a mi padre Luis Nagata y a mi tía Janet Bolivar, por acompañarme en todos los momentos de mi vida, por su amor incondicional, apoyándome en todo momento y por darme la oportunidad de tener una excelente educación en el transcurso de mi vida.

Le agradezco a mi asesora, Mg. Karina Rosas Paredes, por la confianza, apoyo, dedicación de su tiempo, y haber compartido sus conocimientos y amistad.

RESUMEN

En la actualidad el avance de las distintas tecnologías no solo trae consigo distintos beneficios sino también problemas de seguridad de alto riesgo que deben ser controlados, mitigados y eliminados, los delitos informáticos o ciberataques son ahora más pronunciados y más peligrosos que en los últimos años como consecuencia natural del avance tecnológico, así se puede apreciar en (Centeno, 2015), donde se detalla los nuevos sistemas de delincuencia que aparecieron recientemente con el desarrollo de las nuevas tecnologías, así como las nuevas amenazas y métodos para poder explotarlas, se puede ver también en (Carrillo, 2016) el concepto de arma cibernética empleado en un ámbito internacional, en donde se analiza el concepto general, así como daños y medidas que se deben tomar contra distintas amenazas en internet, en (Carlini, 2016) se puede apreciar el motivo por el cual las ciber amenazas pueden potencialmente poner en peligro a los civiles, al medio ambiente así como también actividades gubernamentales y la economía.

El propósito de esta investigación es diseñar una arquitectura de seguridad de redes que pueda controlar, mitigar y eliminar las vulnerabilidades modernas a las que se encuentran expuestas las arquitecturas de red, para este propósito se tomará como referencia el modelo TCP/IP, cubriendo de seguridad las capas de acceso al medio, red, transporte y aplicación, esta arquitectura tomará en cuenta los tipos de conexiones IPv4 y conexiones IPv6, se analizará a profundidad los mecanismos de seguridad de dichos protocolos y se empleará el algoritmo de cifrado AES-256 en combinación con RSA, los cuales actualmente son la columna vertebral de la seguridad en las comunicaciones.

Palabras Clave: Seguridad informática, Vulnerabilidades, ciber seguridad, TCP/IP, AES-256, RSA, IP

ABSTRACT

Currently the advancement of different technologies not only brings with it different benefits but also high-security security problems that must be controlled, mitigated and eliminated, computer crimes or cyber-attacks are now more pronounced and more dangerous than in recent years as natural consequence of technological advance, as can be seen in (Centeno, 2015), which details the new crime systems that appeared recently with the development of new technologies, as well as new threats and methods to exploit them, you can see also in (Carrillo, 2016) the concept of cybernetic weapon used in an international scope, where the general concept is analyzed as well as damages and measures that must be taken against different threats on the internet, in (Carlini, 2016) the which is why cyber threats can potentially endanger civilians, the environment as well as also government activities and the economy.

The purpose of this research is to design a network security architecture that can control, mitigate and eliminate the modern vulnerabilities to which network architectures are exposed, for this purpose the TCP / IP model will be taken as a reference, covering security the layers of access to the medium, network, transport and application, this architecture will take into account the types of IPv4 connections and IPv6 connections, the security mechanisms of said protocols will be analyzed in depth and the AES-256 encryption algorithm will be used in combination with RSA, which are currently the backbone of communications security

Keywords: Informatic security, Vulnerabilities, Cybersecurity, TCP/IP, AES-256, RSA, IP

INTRODUCCIÓN

Como se aprecia en (Centeno, 2015), se detalla los nuevos sistemas de delincuencia que aparecieron recientemente con el desarrollo de las nuevas tecnologías, así como las nuevas amenazas y métodos para poder explotarlas, en (Carrillo, 2016) se observa el concepto de arma cibernética empleado en un ámbito internacional, también se analiza el daños y medidas que se deben tomar contra distintas amenazas en internet, todo esto trae consigo una gran consecuencia que impacta directamente a la sociedad y a los negocios. Por lo expuesto y debido a que la seguridad informática no se puede garantizar al 100%, la presente investigación plantea diseñar una arquitectura de seguridad comprometiendo todas las capas del modelo TCP/IP, con la finalidad de mitigar y eliminar la superficie de exposición a las vulnerabilidades modernas a las que se encuentran expuestas las arquitecturas de red, así como realizar un cifrado de conexiones inversas empleando AES-256 y RSA para gestionar y controlar los nodos confiables, los mismo que serán almacenados en una base de datos implementando un hash SHA-1 a la dirección física MAC como método de autenticación.

En el primer capítulo se realizará el planteamiento teórico, el cual implica la descripción del problema, la solución propuesta, objetivos, alcances y limitaciones, así como el tipo y nivel de investigación.

En el Segundo capítulo se realizará el análisis del marco teórico, el mismo que contendrá el marco conceptual y los aspectos relevantes para la arquitectura de seguridad propuesta.

En el tercer capítulo se analizará los protocolos de seguridad y la explotación de vulnerabilidades modernas.

En el cuarto capítulo se realizará la configuración de protocolos de seguridad por capas del modelo TCP/IP.

En el quinto capítulo se realizará la validación de pruebas de la arquitectura de seguridad propuesta.

En el sexto capítulo se realizará el análisis de los resultados del trabajo de tesis.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I

PLANTEAMIENTO TEÓRICO

1.1 Título

"Arquitectura de seguridad profunda contra explotación de vulnerabilidades modernas empleando AES-256 para el modelo TCP/IP en redes de datos IP"

1.2 Descripción del Problema

Diariamente se genera mucha información, la cual es procesada, intercambiada y conservada en redes de datos, toda organización que a la actualidad implementa tecnología en las telecomunicaciones puede ser vulnerable a cierto tipo de ataques informáticos en alguna medida. (Centeno, 2015) menciona que en los últimos tiempos los problemas de seguridad en redes o ciberataques están aumentando de forma exponencial en su faceta de sustracción de información confidencial y de secretos industriales. El continuo desarrollo de la tecnología en las redes de datos ha traído consigo muchas vulnerabilidades, desencadenando riesgos inherentes que ponen en peligro la continuidad del negocio.

El problema actual como lo exponen (Kalwar, Bohra, & Memon, 2015)en la transición de Ipv4 a Ipv6 enfrenta distintos desafíos dentro de la seguridad al dejar de usar algunos protocolos pero implementar la funcionalidad de los mismos dentro de nuevos protocolos como ICMPV6 el cual cuenta con nuevos fallos de seguridad, del mismo modo que (Vincent Nicolls, Nhien-An Le-Khac, Lei Chen, 2016) logran demostrar mediante distintas pruebas de seguridad que el protocolo

Ipv6 implementado por defecto carece de ciertas características de seguridad, en (Hong et al., 2017) se mencionan los problemas de seguridad que surgen en las tecnologías móviles al implementar Ipv6, también se mencionan problemas en la capa de aplicación como se aprecia en (Nastase, 2017), en el cual se expone problemas de seguridad, teniendo en cuenta el uso de IOT(internet de las cosas), pero este también es un riesgo en las tecnologías de comunicación actuales.

Por ello en esta investigación se propone el diseño de una arquitectura de seguridad de redes tomando como referencia el modelo TCP/IP para implementar seguridad tanto en las capas de acceso al medio, red, transporte y aplicación en redes de datos tanto Ipv4 como Ipv6, implementando seguridad en todos los niveles combinando algoritmos de cifrado tales como son RSA y AES-256, para lograr la mitigación y control de explotación de vulnerabilidades modernas que podrían poner en riesgo la continuidad de negocio

1.3 Solución Propuesta

1.3.1 Justificación Propuesta

Debido a la problemática latente por los distintos tipos de ataques y gestión de reglas de seguridad ante incidentes diversos es que la principal motivación para poder realizar esta investigación es controlar los distintos problemas latentes de seguridad en redes de datos con una arquitectura de seguridad por capas que no causen conflicto o confusión entre sí mismas, logrando de esta manera hacer frente ante distintas técnicas de explotación de vulnerabilidades modernas, las cuales representan una amenaza para los pilares de seguridad informática (CIA).

Desde el punto de vista teórico, esta investigación generará discusión sobre el conocimiento existente en el área de seguridad de redes tanto en la explotación de vulnerabilidades modernas, como en el uso de cifrado para conexiones IP, y las medidas de seguridad que se tienen que implementar en las mismas, para mitigar y eliminar los vectores de ataque que pueden poner en riesgo los pilares de seguridad (CIA).

Desde el punto de vista práctico, esta investigación pretende diseñar una arquitectura de red implementando medidas de seguridad en cada una de las capas, siendo de vital importancia implementar el cifrado AES-256 generado mediante un hash de la dirección física para el establecimiento de nodos seguros, esto puede ser utilizado independientemente de la aplicación que se utilice en las demás capas, esta arquitectura de seguridad pretende ser escalable ya que podrá ser utilizada en entornos tanto Ipv4 como Ipv6, así como su posterior implementación en redes SDN(redes definidas por software).

Desde el punto de vista profesional se pondrá en manifiesto los conocimientos adquiridos durante la carrera y permitirá sentar las bases para otros estudios que surjan partiendo de la problemática de seguridad en redes de datos IPv4/IPv6 tomando como referencia el modelo TCP/IP.

Dentro de los beneficios de la presente investigación se resolverá un problema actual en cuanto a la arquitectura y configuración por defecto de los nuevos protocolos de comunicación y sus fallos de seguridad, al igual que abrirá nuevos caminos para estudios sustantivos que presenten

situaciones similares a las que aquí se plantean, sirviendo como marco referencial a estas.

Dentro de la factibilidad para poder desarrollar esta investigación fue necesario la virtualización de las medidas de seguridad en cada una de las capas del modelo TCP/IP, así como un entorno de auditoria para poder realizar las pruebas de seguridad necesarias, el cual se realizó mediante la configuración de un *switch extreme x440,* al mismo que se conectaron distintos nodos para realizar las pruebas de seguridad y la configuración de un servidor encargado de desplegar las medidas de seguridad.

1.3.2 Descripción de Propuesta

Diseñar una arquitectura de seguridad tomando en cuenta el modelo TCP/IP, en donde se establecerá distintas medidas de seguridad dentro de las capas de acceso al medio, red, transporte y aplicación, empleando el cifrado AES-256, así como asegurar la autenticación de los nodos confiables para conexiones inversas, los mismos que se autenticaran mediante la implementación de un algoritmo de hash SHA-1 en la dirección física (MAC), así como desplegar las medidas de seguridad necesarias para asegurar las comunicaciones en redes IP, conservando los pilares de seguridad(CIA), para lograr hacer frente a la explotación de vulnerabilidades modernas, ya que estas representan una amenaza que ponen en peligro la continuidad del negocio.

1.4 Objetivos

1.4.1 Objetivo General

Diseño de una arquitectura de seguridad contra la explotación de vulnerabilidades modernas en la capa 1(subcapa de enlace de datos), 2, 3 y 4 del modelo TCP/IP en redes de datos IP empleando el algoritmo de cifrado AES-256.

1.4.2 Objetivos Específicos

a. Reconocer las amenazas de seguridad en redes de datos.

b. Analizar los métodos y aspectos técnicos de la explotación de vulnerabilidades modernas.

c. Analizar la seguridad de manera profunda en cada capa del modelo TCP/IP.

d. Dimensionar las medidas de seguridad en las capas 1, 2, 3 y 4 del modelo TCP/IP.

e. Garantizar la interconectividad entre nodos confiables mediante la implementación del algoritmo SHA-1 en la dirección física (MAC) como mecanismo de identificación.

f. Cifrar las comunicaciones entre nodos confiables utilizando los algoritmos AES-256 y RSA para el intercambio de información.

g. Comprobar el cifrado de la transmisión de datos para evitar que estos sean interceptados por nodos no confiables.

1.5 Alcances y Limitaciones

1.5.1 Alcances

La presente investigación analizará las medidas de seguridad tomadas en las capas de acceso al medio, red, transporte y aplicación del modelo TCP/IP, en conexiones IP.

La investigación tomará en cuenta la implementación de los algoritmos de cifrado RSA y AES-256, así como sus configuraciones para no disminuir la eficiencia de seguridad en otras capas del modelo TCP/IP. Así como el análisis de explotación de vulnerabilidades modernas para lograr una configuración de seguridad eficiente.

1.5.2 Limitaciones

Esta investigación se limita a dimensionar las medidas de seguridad en la capa 1(subcapa de enlace de datos), 2, 3, y 4, excluyendo el entorno físico donde se implementa la arquitectura de red.

1.6 Línea y Sub-Línea de Investigación

1.6.1 Línea

Redes y telemática

1.6.2 Sub-Línea

Seguridad informática

1.7 Tipo y Nivel de Investigación

1.7.1 Tipo de Investigación

La presente investigación es de tipo aplicada, tiene un propósito inmediato de actuar sobre la seguridad en la arquitectura de red tomando como referencia el modelo TCP/IP analizando la seguridad de las conexiones en las capas 1, 2, 3 y 4, desplegando medidas de seguridad en cada una de ellas, evaluando la seguridad de conexiones IP así como sus configuraciones de seguridad para obtener una máxima eficiencia salvaguardando los pilares de seguridad (CIA), logrando tener una línea de defensa en las distintas capas del modelo TCP/IP ante la explotación de vulnerabilidades modernas.

1.7.2 Nivel de Investigación

La presente investigación es de nivel descriptiva, la cual señala como se manifiestan los fenómenos y riesgos de seguridad en una arquitectura de red, analizando los protocolos de las conexiones para tomar las medidas de configuración más adecuadas, así como la implementación de dispositivos para salvaguardar los pilares de seguridad (CIA).

CAPÍTULO II

MARCO TEÓRICO

2.1 Estado del Arte

En vista que la información es el factor primordial por el que muchos usuarios malintencionados realizan actos ilícitos es necesario conocer y gestionar las vulnerabilidades en las redes de datos, (Narayan, Gupta, Kumar, Ishrar, & Khan, 2016) muestran el desempeño y la comparación entre el mecanismo de transición 4to6 y el mecanismo de transición 6to4 cuando es atacado por varios ciberataques como el Nmap, Zenmap, Smurf6 y el *flood* router6, también en esta investigación se compara la transición de la configuración de las redes VPN (*Virtual Private Network*) como PPTP e IPSEC y los distintos tipo de ciberataques que surgen como consecuencia de esta transición.

Las capas superiores del modelo TCP/IP también se ven afectadas con los nuevos avances de la tecnología como se demuestra en (Zalbina et al., 2017), en el cual se investiga el reconocimiento y la detección de ataques XSS mediante la asignación de patrones de expresión regular y un método de pre procesamiento, esto debido al gran avance de los ciberataques para poder evadir medidas de seguridad configuradas por defecto.

En los últimos tiempos uno de los ataques más frecuentes es el de DOS/DDOS (denegación de servicio/denegación de servicio distribuida), con el avance de las nuevas tecnologías como Ipv6 este aun es un problema latente como

lo expone (States, 2016) en cuya investigación se examina las firmas de tráfico de diferentes ataques de negación de servicio (Dos) que afectan al protocolo IPv6 mediante el uso de herramientas que analizan los paquetes en colaboración con una popular herramienta de evaluación de seguridad IPv6 y valores de correlación de tráfico para establecer las similitudes y diferencias entre varios ataques DOS. El nivel de unicidad de las firmas de tráfico de ataque se utiliza para determinar si una técnica de detección de intrusión basada en correlaciones es adecuada para mitigar los ataques de DOS en IPv6.

La seguridad es un problema vital en el nuevo protocolo de IPv6, como lo exponen (Terli, Chaganti, Alla, Sarab, & El Taeib, 2016) en el que se presenta una solución de software para ayudar a la transición entre IPv4 e IPv6 y se re aborda las necesidades básicas y finalmente establece una solución factible que aún mantiene la aplicabilidad y la capacidad de mantenimiento de ambos sistemas(IPv4/IPv6) en donde todos los dispositivos se actualizan y usan IPv6 sin comprometer a las aplicaciones existentes o nuevas aplicaciones futuras, esta solución propuesta es para ISP y los proveedores de la red principal de internet mas no para usuarios individuales.

Actualmente IPv6 es un nuevo protocolo de enrutamiento que se está desplegando dramáticamente en los últimos años, la gran mayoría de organizaciones creen que este nuevo protocolo es más seguro que IPv4, pero esta es una idea errónea como se demuestra en (Rafiqul Zaman Khan & Atena Shiranzaei, 2016)en el cual se cubre herramientas fundamentales de penetración y monitoreo en la implementación de IPv6, como conclusión este nuevo protocolo de

comunicación debe ser auditado continuamente debido a las nuevas amenazas que se exponen en internet.

Con la llegada de este nuevo protocolo IPv6 se eliminaron algunos protocolos que se veían en las conexiones de tipo IPv4, uno de los protocolos eliminados es ARP, este protocolo era utilizado para realizar ataques de *MITM*(hombre al medio), el cual permitía intervenir las comunicaciones entre dos puntos, actualmente en IPv6 no es un problema solucionado como se demuestra en la investigación de (Ouseph, 2016) IPv6 es vulnerable a los anuncios de enrutadores deshonestos (RAs), el cual es utilizado para realizar los ataques de *MITM*, dicha investigación propone una posible solución para evitar este tipo de ataques en las conexiones de tipo IPv6.

Existe una gran especulación en cuanto a que IPv6 reemplaza en su totalidad a IPv4, por ese motivo en la investigación de (Sánchez, 2015) se realizan distintas evaluaciones de seguridad contra el protocolo IPv6, teniendo en cuenta los pilares de seguridad informática tales como confidencialidad, integridad y disponibilidad (CIA).

La llegada de IPv6 vino acompañada de un protocolo de seguridad por defecto IPSEC, el cual inicialmente se introdujo como un componente adicional en IPv4, aunque este protocolo es la mejor opción para proteger el protocolo IP, su implementación y gestión es de naturaleza compleja. La implementación implica la gestión de claves y el intercambio a través de IKE, en (Shah & Parvez, 2015) se realiza una investigación empírica de los parámetros que se ven afectados por la aplicación de IPSEC en IPv6 y 6to4 . La investigación es significativa y evalúa la disminución del rendimiento que se encuentra al incorporar la seguridad.

Dentro de los problemas que abarca IPv6 existe uno que puede ser muy peligroso en cuando a la duplicación de direcciones, como se explica en (Praptodiyono et al., 2015), en donde se propone un mecanismo de seguridad ligero basado en la gestión de confianza distribuida que integra seguridad y seguridad ligera para proteger el DAD (detección de direcciones duplicadas) en IPv6 llamado Trust-ND.

En la investigación (Kamaldeep, Malik, & Dutta, 2018), se propone una implementación mejorada y más eficiente de un esquema de rastreo IP híbrido mediante un solo paquete en redes de datos IPv4 e IPv6, este esquema necesita solo de un paquete para poder realizar una trazabilidad, el mayor beneficio de esta investigación es la reducción de registro en los enrutadores. En esta investigación se registra la información de la ruta y no los paquetes individuales, de este modo se distribuyen la información de ruta entre los campos de los paquetes y tablas de registro de enrutadores.

Como se aprecia en la investigación (Fedorchenko, Kotenko, & Chechulin, 2015), la integración de las bases de datos de vulnerabilidades abiertas existentes permite incrementar la probabilidad de detectar software o hardware vulnerable que pueden estar conectados en una red de datos, de esta manera se puede mejorar el análisis de la seguridad, esta investigación prueba la utilidad de la base de datos de vulnerabilidades abiertas junto a un sistema de respuesta en tiempo real, de esta manera se puede cubrir y mejorar la seguridad dentro de una red.

Con el pasar del tiempo los ciberataques son cada vez más sofisticados y minuciosos por lo cual es recomendable el descubrimiento de patrones en las redes de taos, como se plantea en (Kao, Wang, Tsai, & Chen, 2018), en dicha

investigación se recopilan datos en tiempo real, con lo cual se pueden identificar patrones inusuales en los paquetes de red, usando los datos capturados se pueden identificar paquetes erróneos, cuellos de botella y permanecer con evidencias completas de cualquier evento que suceda en la red, esta investigación también demuestra los riesgos de la suite de protocolos TCP/IP, así como promover el análisis de los mismos con datos no estructurados.

En la siguiente investigación (Nakhla, Perrett, & McKenzie, 2017), realizada por el centro de defensa, Investigación y Desarrollo de Canadá, se propone tomar medidas defensivas efectivas, casi en tiempo real, para responder adecuadamente a los ciberataques, sin tener un impacto significativo en las operaciones, aunque existen diversas soluciones para recopilar y analizar automáticamente datos de infraestructura, pocas ofrecen análisis automatizados e implementación de medidas de seguridad, además que estos procesos pueden resultar intensivos para los operadores así como para las herramientas disponibles actualmente, ya que estas últimas suelen ser muy específicas ante ataques comunes, por ello en esta investigación se pone en marcha ARMOR, que proporciona un soporte de decisión mejorado y automatización de muchas de las tareas que los operadores de red ejecutan manualmente, esta investigación describe el marco de integración de la defensa cibernética, el conocimiento de la situación y el soporte de decisión automatizado. En esta investigación se demuestra el apoyo en la toma de decisión para la mitigación de vulnerabilidades con el objetivo de poder proteger de manera proactiva la red y poder responder de manera reactiva a los incidentes de seguridad.

En los últimos tiempos las vulnerabilidades evolucionaron y cada vez son más sofisticadas, como se demuestra en (Badea, Croitoru, & Gheorghica, 2015), en donde se destaca la captura de tráfico de red para poder identificar el rendimiento, poder detectar fallas de seguridad y evaluar las vulnerabilidades, de la misma forma que se monitorea el comportamiento de las aplicaciones, al analizar el tráfico se puede observar el comportamiento de los usuarios y poder calificar tráfico anómalo que pueden desencadenar problemas de seguridad, salvaguardar los pilares de seguridad informática mediante la recolección y análisis de tráfico de los usuarios y/o aplicaciones es el fin de esta investigación.

La detección de vulnerabilidades es un punto importante en la seguridad de los sistemas, cada vez los ciberataques son más sofisticados e inteligentes por lo cual en la investigación (Wu, Wang, Liu, & Wang, 2017), se propone un método de aprendizaje profundo para la detección de vulnerabilidades, se presentan tres modelos de aprendizaje profundo, red neuronal convolucional (CNN), memoria larga a corto plazo (LSTM) y red neuronal convolucional - larga memoria a corto plazo (CNN-LSTM), en esta investigación se extraen características dinámicas de 9872 programas binarios y con estos se entrenan los modelos de aprendizaje profundo para detectar vulnerabilidades. Los resultados experimentales muestran que la precisión de predicción de los modelos de aprendizaje profundo alcanza el 83.6%, que es mucho más alta que la del método tradicional como MLP.

Como se sabe en las redes de datos, los recursos de red se comparten entre diferentes aplicaciones en función de un esquema de prioridad de calidad de servicio (QoS), como se observa en (Mamdouh, Ghoz, & Far, 2017), se propone una técnica novedosa de algoritmo de cifrado avanzada (AES) con reconocimiento

de contexto *Cipher Block Chaining (CBC)* como un modo general para IEEE 802.11e. Este algoritmo se basa en la selección de algunos parámetros diferentes de AES según el tipo de aplicación recibida se agrega un alto nivel de seguridad dependiendo del esquema de prioridad QoS, dentro de los resultados de esta investigación sobresale el tiempo total de retraso establecido para cifrado / descifrado para AES-128 es 0.0257 segundos, mientras que para AES-192 es 0.0296 segundos y finalmente para AES-256 es 0.0351 segundos, de la misma manera el sistema propuesto se compara con la misma arquitectura de red sin algoritmo AES y con parámetros (fijos y diferentes) del algoritmo AES donde se observa que el sistema propuesto en dicha investigación mejora el rendimiento del sistema con un alto nivel de seguridad para cifrar los datos transmitidos al cumplir los parámetros de QoS.

Un tema bastante desarrollado a la actualidad es la computación en la nube, en la cual se pueden compartir recursos, servicios, etc. El problema es que al tener millones de usuarios usando la misma red para transferir los datos estos se vuelven vulnerables a distintas técnicas de ataques, proporcionar seguridad en este caso es una prioridad por lo cual en la investigación (Kumar, 2017), se aprecia que los sistemas actuales se concentran en brindar seguridad a los datos almacenados en la nube pero se deja una brecha de seguridad al momento en el que estos datos son transferidos, por ellos se propone un modelo de seguridad basado AES, tanto para el almacenamiento de datos a la nube como para el almacenamiento de los mismos.

La seguridad es uno de los aspectos importantes que debemos considerar relacionado con la comunicación de datos. La seguridad es principalmente para datos confidenciales, como datos gubernamentales o militares. AES es uno de los

métodos criptográficos que actualmente se considera seguro y lo suficientemente fuerte, aunque actualmente no existe un ataque capaz de romper AES, solo es cuestión de tiempo debido al rápido desarrollo de la velocidad de computo, los ataques de fuerza bruta también podrían romper AES rápidamente si la clave es débil. En la investigación (Irfan & Mahendra, 2017), se propone un AES modificado que altera las estructuras originales mediante el uso de *MAC Address*. El uso de la dirección MAC podría aumentar la aleatoriedad de los procesos AES en cada computadora. El uso de la dirección MAC podría aumentar la aleatoriedad de los procesos AES debido a su diferencia en cada computadora. Es probable que los atacantes no puedan descifrar fácilmente el mensaje cifrado debido a que el algoritmo AES modificado tiene mayor complejidad que el AES original, pero requiere un tiempo de ejecución más largo.

En el mundo de la seguridad informática existe la manipulación de hardware, casos de ataques de hardware con troyanos implantados, en la investigación (Chuan, Yan, & Zhang, 2017), se propone un método para activar eficientemente un troyano dentro de un hardware en el circuito criptográfico AES, de esta manera se puede verificar si el hardware se encuentra infectado.

En la actualidad la mayoría de información viaja a través de la red de datos por lo que la información que viaja por este medio es vulnerable a ser manipulada por personas no autorizadas, uno de los requisitos actuales para la comunicación y la mensajería digital cabe sobre el empleo de firmas digitales, estas buscan mantener dos aspectos de la seguridad a los que apunta la criptografía, tales como la integridad y el no repudio. La investigación (Dwi & Utama, 2017), tiene como objetivo aplicar la dirección MAC junto con AES-128 y SHA-2 de 256 bits para la

firma digital, con esta investigación se pudo realizar la firma digital de exclusividad en cada dispositivo para hacer una comunicación simple y segura.

2.2 Marco Conceptual

2.2.1 Pilares de Seguridad Informática (CIA):

- Confidencialidad: Es una propiedad que asegura el acceso a la información únicamente por personas autorizadas con los permisos que a cada una de estas personas le corresponda.
- Integridad: Esta propiedad tiene como objetivo mantener los datos libres de modificaciones las cuales no estén autorizadas.
- Disponibilidad: Esta característica permite que la información se encuentre disponible para las personas o sistemas que deseen acceder a la misma.

2.2.2 Amenazas

Hace referencia a todo evento o circunstancia que tiene la capacidad de provocar daño a un sistema en sus distintas modalidades como podrían ser, robo, destrucción, alteración de datos, denegación de servicio, etc.

2.2.3 Vulnerabilidades

Es una debilidad interna que puede ser utilizada por alguien mal intencionado para causar algún impacto sobre un sistema, poniendo en riesgo los pilares de seguridad (CIA).

2.2.4 Exploit

Es un programa o fragmento de software que puede aprovechar vulnerabilidades y pueden ser utilizados para evadir medidas de seguridad o comprometer algún nodo o sistema.

2.2.5 Superficie de Exposición

Se refiere al conjunto de vulnerabilidades y puntos de entrada o salida que pueden ser susceptibles de explotaciones de seguridad para comprometer el sistema tanto en nivel de datos como de funcionamiento.

2.2.6 Criptografía Simétrica

También conocida como criptografía de clave secreta, aquí se utiliza una sola clave para cifrar y descifrar los datos, esta clave tiene que ser conocida por el emisor y por el receptor, en la siguiente figura se puede observar el funcionamiento de la criptografía simétrica.

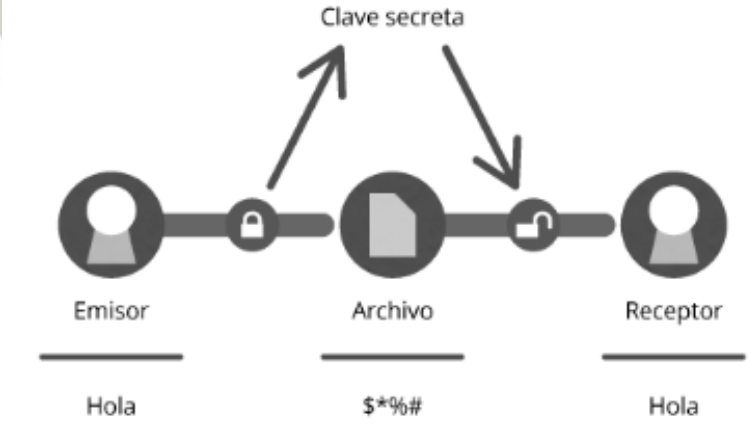


Figura 1: Criptografía simétrica

Fuente: *HTTPS*://www.genbetadev.com/seguridad-informatica/tipos-de-criptografia-simetrica-asimetrica-e-hibrida

2.2.7 Criptografía Asimétrica

Esta criptografía tiene su base al emplear dos claves una pública y una privada, la clave pública puede ser conocida, pero la privada debe mantenerse en secreto, este tipo de criptografía es más lento, pero más seguro, en la siguiente figura se observa el flujo de cómo trabaja la criptografía asimétrica.

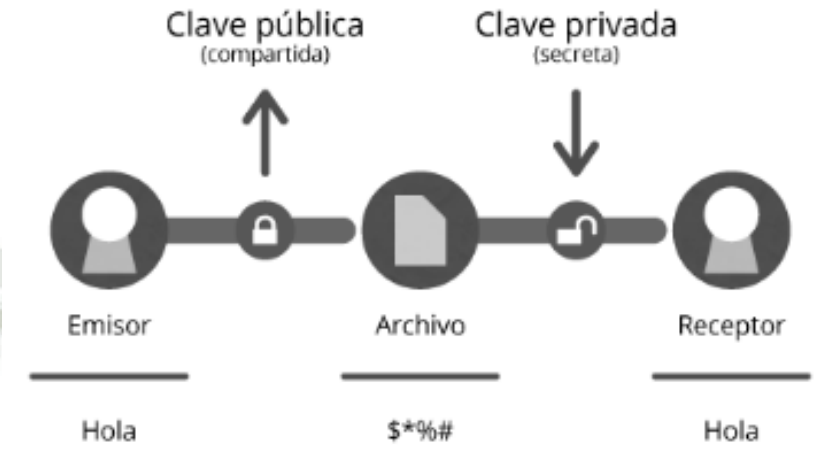


Figura 2: Criptografía asimétrica

Fuente: *HTTPS*://www.genbetadev.com/seguridad-informatica/tipos-de-criptografia-simetrica-asimetrica-e-hibrida

2.2.8 Mecanismo 6to4

Fue diseñado para enviar tráfico de IPv6 a través de una red IPv4, esto es realizado mediante túneles en donde se encapsula paquetes IPv6 dentro de paquetes IPv4 para enviarse a solo través de redes IPv4, esto es necesario ya que con la constante evolución de IPv6 puede ser necesario intercomunicar este tipo de conexiones por medio de una red que aún se encuentre en IPv4, 6to4 puede funcionar en un router dando así conectividad a toda la red, en la siguiente figura se aprecia una conexión a través de un túnel 6to4.

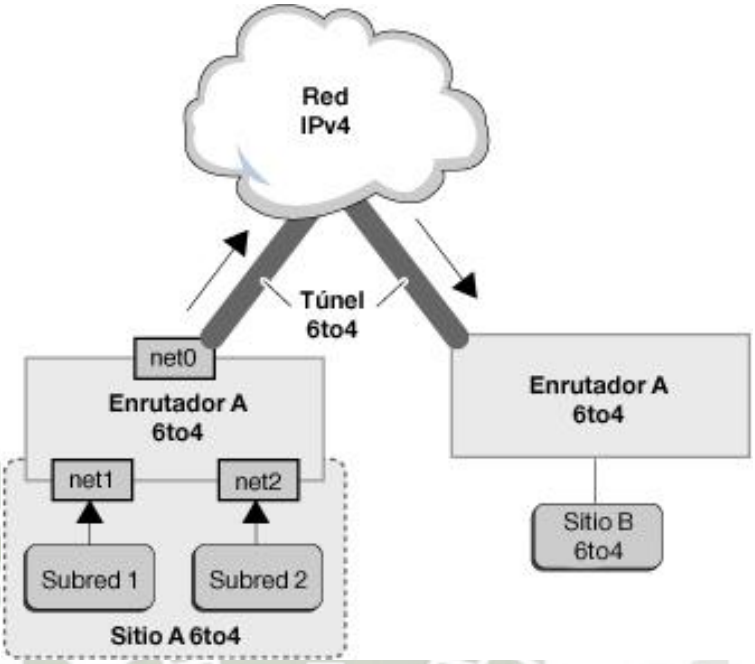


Figura 3: 6to4

Fuente: *HTTPS*://docs.oracle.com/cd/E56339_01/html/E53800/ipv6-ref-50.html

2.2.9 Cortafuegos (Firewall)

Un cortafuegos puede ser una pieza de software, un programa aplicación o hardware que es el encargado de filtrar el tráfico que se genera en una red, esta medida de seguridad es capaz de elevar la seguridad en una infraestructura, ya que se puede detallar información acerca de los patrones de tráfico, suele actuar como una puerta de seguridad entre redes de confianza y redes externas, los firewalls pueden trabajar en distintas capas como se detalla más adelante, en la siguiente figura se observa una arquitectura simple donde un firewall separara una red interna de internet.

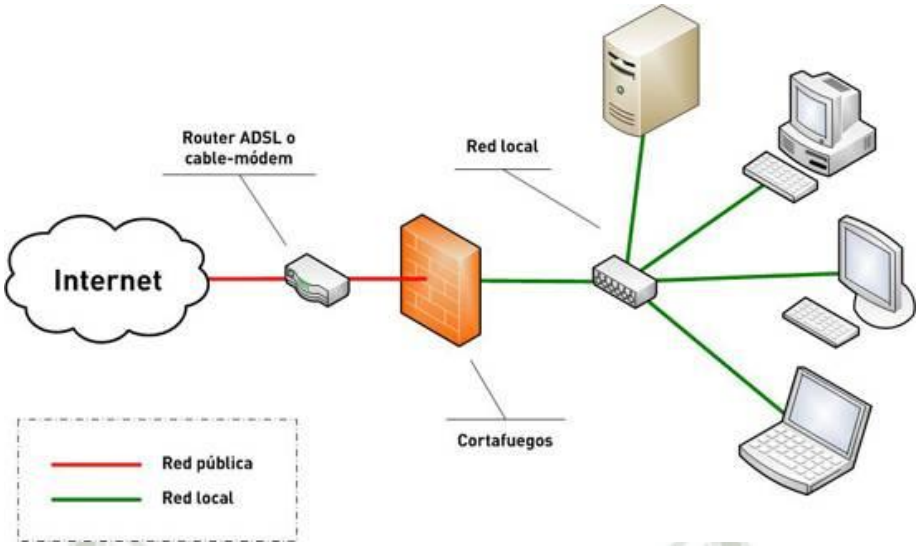


Figura 4: Firewall

Fuente: *HTTP*://blog.deservidores.com/que-es-y-para-que-sirve-un-firewall/

2.2.10 Sistema de Detección de Intrusos (IDS)

Un sistema de detección de intrusiones el cual tiene la capacidad de detectar accesos no autorizados en una red o un nodo en específico, realiza un análisis de tráfico de red y mediante el establecimiento de políticas y sensores puede detectar anomalías que pueden ser presencia de ataques o accesos no autorizados, es un sistema pasivo es decir no toma acción sobre determinados incidentes sino que almacena información de las actividades y manda distintas alertas, los IDS puedes optar por la técnica de patrón la cual analiza paquetes de red, y los compara con ataques conocidos estos patrones son denominados firmas. También se puede emplear la técnica heurística en donde se determina una actividad normal en la red basado en distintas características como el ancho de bando, protocolos, puertos, etc. Cuando este comportamiento es alterado se considera anómalo y los IDS actúan, los tipos de IDS pueden ser:

Sistema de Detección de intrusos en Red (NIDS)

Puede analizar un segmento de red capturando y analizando el tráfico que se genera, puede detectar ataques en todos los equipos conectados, usualmente trabajan mediante reglas.

Sistema de Detección de intrusos en Host (HIDS)

Este depende de los rastros que dejaron los intrusos, los HIDS intentan detectar las modificaciones en los nodos afectados para realizar un reporte, esto se da en un único nodo por lo cual cada nodo debe tener un HIDS.

En la siguiente figura se observa la diferencia entre un NIDS y un HIDS.

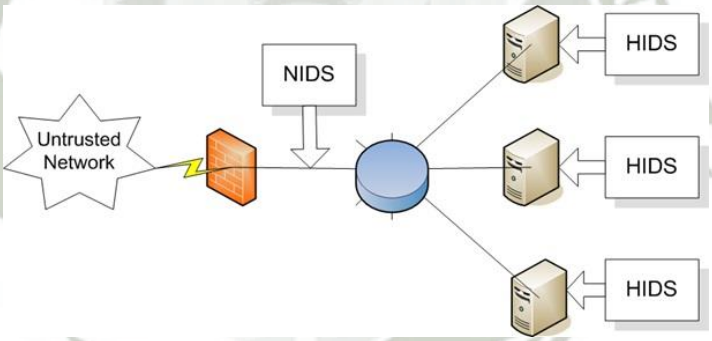


Figura 5:NIDS y HIDS

Funte:*HTTPS*://www.linkedin.com/pulse/security-device-network-nicol%C3%A1s-zanni/

2.2.11 Sistema de Prevención de Intrusos (IPS)

Un sistema de prevención de intrusos es un software que tiene como función tomar decisiones acerca de las intrusiones o accesos no autorizados y no solo reconocerlas y lanzar alertas como es el caso de los IDS, un IPS puede

bloquear inmediatamente el tráfico caracterizado como malicioso, es decir que puede filtrar el tráfico generado en una red, por lo mismo puede categorizar las detecciones basada en firmas, políticas o basadas en anomalías.

2.2.12 Listas de Control de Accesos (ACL)

Las listas de control de acceso son listas en las que se especifica los accesos para usuarios o grupos de usuarios, en la siguiente figura se observa como un router puede filtrar las conexiones según determinadas características.

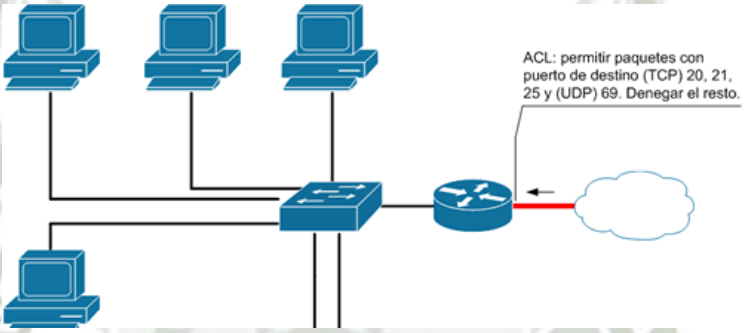


Figura 6: Listas de control de acceso

Fuente: http://www.redespracticas.com/?pag=txtACLcsco.php&Njs=t

2.2.13 Honeypot

Es un sistema trampa, es un sistema informático que tiene como objetivo simular un sistema que se encuentre actualmente en uso, como tal puede ser objetivo de posibles ataques informáticos, de esta manera se puede detectar y obtener información acerca del atacante, así los *Honeypot*s pueden detectar ataques antes de que estos sean utilizados ante servicios reales de producción.

2.2.14 Honeynet

Son un tipo de *Honeypot*s de alta interacción, que operan sobre toda una red, *honeynet* está diseñado para ser atacada y por ser de alta interacción tiene

como característica recobrar mucha información sobre posibles ataques, esto suele utilizarse para la investigación de nuevas técnicas de ataque, así como evaluar la manera en la que operan los ciber criminales.

2.2.15 Sandbox

El aislamiento de procesos es un mecanismo para ejecutar ciertos programas de manera separada tomando medidas de seguridad, ya que se encuentra en un entorno aislado se puede controlar los recursos necesarios para la ejecución como memoria, espacio en disco, etc.

2.2.16 Red de Área Local Virtual (*VLAN*)

Una red de área local virtual es una red que agrupa distintos nodos de manera lógica y no física, las *VLAN* permiten redistribuir una red y no limitarse a una arquitectura física, esto ocurre debido a que se puede definir una segmentación lógica de nodos basados en agrupamiento según algunas características como direcciones MAC, protocolos, etc. En la siguiente figura se observa una simple distribución de una red local segmentada de manera lógica empleando *VLAN*.

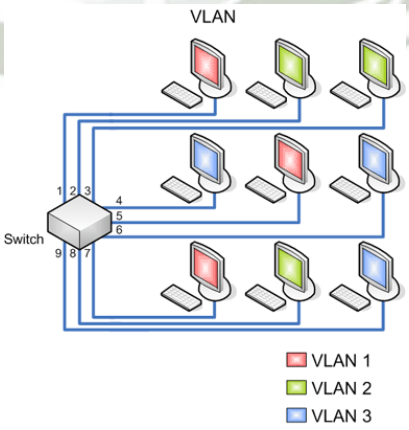


Figura 7: *VLAN*

Fuente:*HTTP*://redesconfiguracion.blogspot.pe/2015/07/que-es-una-*VLAN*-y-su-funcion.html

2.2.17 Red Privada Virtual (VPN)

Una red privada virtual es una tecnología segura que permite establecer sobre una red pública una extensión de una LAN, esto se logra mediante una conexión de punto a punto, la seguridad es un punto a favor de estas conexiones normalmente se tiene distintos tipos de verificaciones de seguridad en distintos niveles, en la siguiente figura se observa un flujo de conexión VPN.

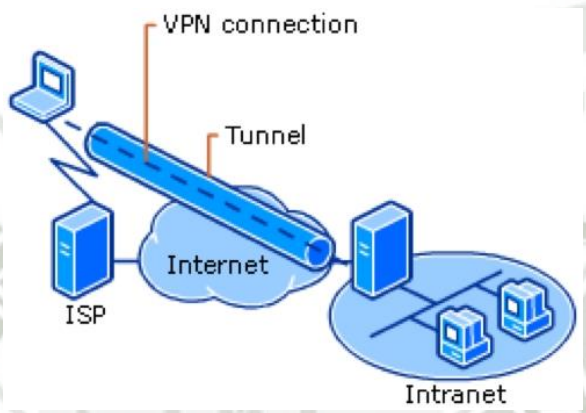


Figura 8:VPN

Fuente: *HTTP*://www.chicageek.com/que-es-vpn-para-que-sirve/

2.2.18 Balanceo de Carga

Se trata de compartir la carga entre varios recursos, estos pueden ser varios nodos, discos, etc. El balanceo de carga es muy utilizado en servidores web, ya que uno de los casos más frecuentes es recibir y gestionar muchas solicitudes lo que suele ocasionar un problema de escalabilidad, en la siguiente figura se observa el balanceo de carga entre distintos clientes y las solicitudes son distribuidas entre los distintos servidores.

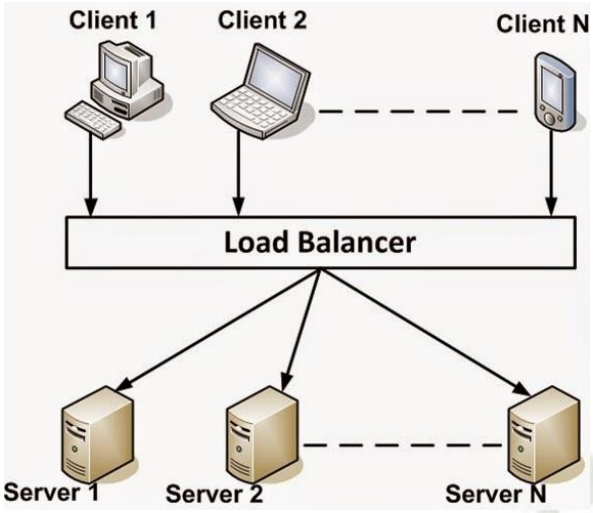


Figura 9: Balanceo de carga

Fuente: *HTTP*://redes-seguridad.blogspot.pe/2014/11/balanceo-de-carga-y-alta-disponibilidad.html

2.2.19 Autenticación, Autorización y Contabilización (AAA)

Es una familia de protocolos los cuales realizan las siguientes funciones:

- Autenticación: Hace referencia al proceso por el que una entidad es capaz de probar su identidad mediante credenciales las cuales podrían ser contraseñas, *one-time tokens*, certificados digitales, etc.
- Autorización: Hace referencia a los permisos que se tiene basándonos en la identidad de quien los solicita, estos permisos pueden estar sujetos a distintas características como pueden ser horarios, localización, etc.
- Contabilización: Hace referencia al seguimiento de recursos utilizados en la red.

Algunos ejemplos de protocolos AAA son:

- **Radius**: Este protocolo utiliza una arquitectura de cliente/servidor en donde el usuario requiere unas credenciales de acceso las cuales se

presentan ante el servidor para verificar la autenticidad del solicitante y determinar si se puede o no acceder al recurso solicitado, radius es capaz de manejar sesiones para poder determinar y notificar el tiempo de conexión desde el inicio hasta el final permitiendo sacar datos estadísticos y verificar el consumo de recursos, en la siguiente figura se observa una conexión remota a un servidor radius.

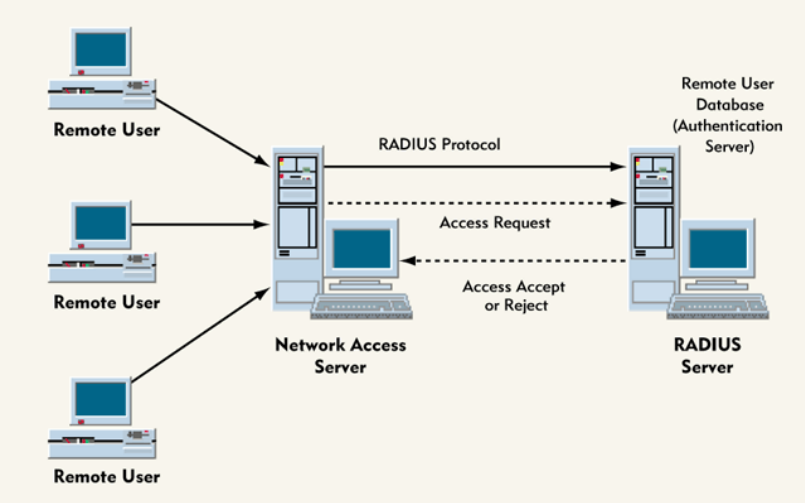


Figura 10: Autenticación RADIUS

Fuente: *HTTP*://claudelby.blogspot.pe/2011/05/radius-remote-authentication-dial-in.html

•**Diameter**: Este protocolo está basado en RADIUS, implementa algunas mejoras como pueden ser la utilización de protocolos fiables como TCP o SCTP, utiliza seguridad en la capa de transporte con IPSEC o TLS, es compatible con RADIUS, deja de ser un protocolo de cliente/servidor para ser un protocolo peer to peer, cuenta con notificaciones de errores

•**TACACS+:** Es un protocolo encargado de la autenticación de manera remota para poder realizar una gestión de accesos, puede proporcionar distintos servicios como registro autorización y autenticación.

2.2.20 Intercambio de Claves de Internet (IKE)

Es un protocolo encargado de establecer una asociación de seguridad (SA) en el cual se emplea un intercambio de llaves secreto de tipo Diffie-Helman, se suele emplear criptografía asimétrica, también se tiene como objetivo negociar una asociación de seguridad para el protocolo IPSEC permitiendo también establecer el TTL de la sesión IPSEC, en la siguiente figura se observa la interacción de IKE.

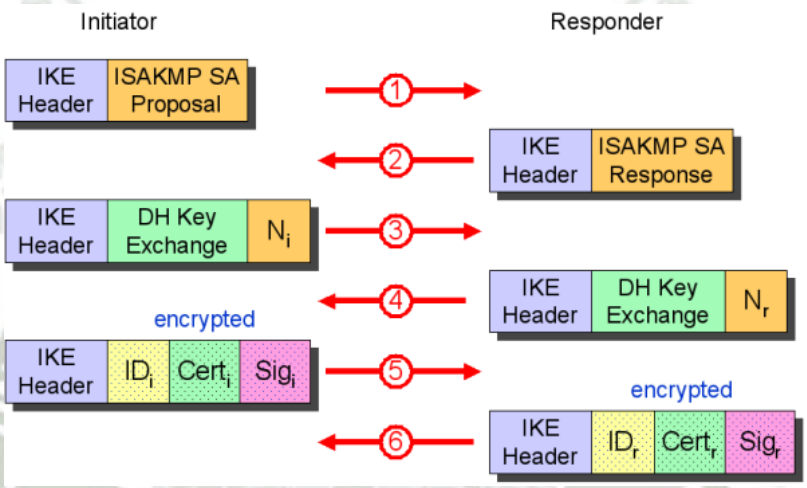


Figura 11: Internet key Exchange (IKE)

Fuente: *HTTP*://www.free-it.org/archiv/talks_2005/paper-11156/paper-11156.html

2.2.21 Protocolo de Autenticación de Contraseña (PAP)

Es un protocolo sencillo para autenticar a un usuario, puede validar el acceso a ciertos recursos mediante un usuario y contraseña los cuales son enviados sin cifrar por la red, es una autenticación en dos pasos como se observa en la siguiente figura.

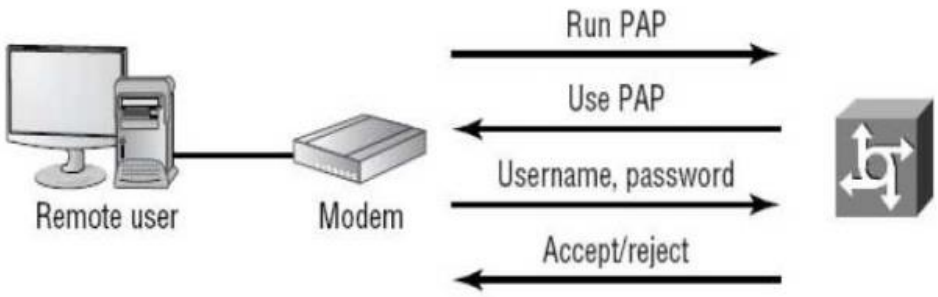


Figura 12: Password Authentication Protocol (PAP)

Fuente: *HTTP*://www.vce-download.net/study-guide/comptia-networkplus-11.8-point-to-point-authentication-protocols-ppp.html

2.2.22 Protocolo de Autenticación por Desafío Mutuo (CHAP)

Es un protocolo de autenticación por desafío mutuo, utilizado para autenticación remota normalmente utilizado para el protocolo PPP, aquí se envía un usuario un ID de usuario y luego de lanzarse un desafío se enviará la clave, pero no en texto plano, sino que se enviará un HASH generado en MD5 para realizar una comparativa y evaluar el acceso, es una autenticación realizada en 3 vías, como se observa en la siguiente figura el modo de autenticación mediante CHAP.

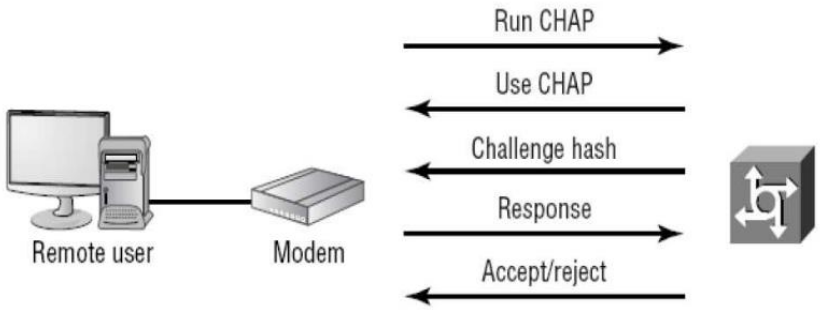


Figura 13: Challenge Handshake Authentication Protocol (CHAP)

Fuente: *HTTP*://www.vce-download.net/study-guide/comptia-networkplus-11.8-point-to-point-authentication-protocols-ppp.html

2.2.23 Privacidad Equivalente a Cableado (WEP)

Es un sistema de cifrado para redes Wireless que puede cifrar información utilizando el algoritmo RC4, utiliza claves de 64 o 128 bits, se puede emplear dos métodos de autenticación los cuales son:

- Autenticación de sistema abierto: Todo lo que se realiza es identificar un determinado nodo mediante una dirección de hardware.
- Autenticación mediante clave compartida: En este tipo de autenticación el cliente cuenta con una clave secreta la cual es verificada al momento de la conexión.

2.2.24 Protección de Acceso a Wi-Fi (*WPA*/*WPA*2)

Sistema que tiene como objetivo proteger las redes inalámbricas, *WPA*/*WPA*2 utilizan la autenticación de usuarios mediante un servidor, en donde se almacena las credenciales de seguridad, pero también puede autenticarse mediante una contraseña pre-compartida, *wpa* mejora los niveles de seguridad que establecía WEP, y *WPA*2 incrementa nuevamente este nivel de seguridad permitiendo utilizar el algoritmo AES el cual es uno de los más robustos a la actualidad.

2.3 Definición de Capas y Protocolos en el Modelo TCP/IP

2.3.1 Capa 1 Acceso al Medio

Es responsable de la transferencia confiable de información en una red de datos, para ello interactúa con la capa superior (capa de red), suministrando un

tránsito de datos confiable a través de un enlace físico. Tiene como objetivo conseguir que la información pueda circular libre de errores, entre dos nodos que estén conectados a una red, para ello se implementa las siguientes funciones:

Iniciación, terminación e identificación: En la función de iniciación se dan los procesos para activar el enlace, esto también abarca el intercambio de tramas de control para poder establecer la disponibilidad de los nodos para transmitir y recibir información. La función de terminación se encarga de liberar los recursos que son utilizados hasta el envío recepción de la última trama, al igual que usar tramas de control. La función de identificación se encarga de identificar el origen o destino de una trama, esta se lleva a cabo por la dirección MAC.

Segmentación y bloqueo: Cuando las longitudes de las tramas son demasiado extensas se da paso a la segmentación la cual realizara tramas más pequeñas con la misma información de una trama extensa. En caso contrario cuando las tramas son demasiado pequeñas se implementa una técnica de bloque, que consiste en concatenar varios mensajes cortos de nivel superior en una sola trama de la capa de enlace de datos la cual será más extensa.

Sincronización de octeto y carácter: Para la transferencia de información en esta capa es necesaria la identificación de los bits, así como identificar qué posición les corresponde en cada caracter u octeto. Esta función abarca los procesos para adquirir, mantener y recuperar la sincronización de caracter u octeto.

Delimitación de trama y transparencia: Esta capa se encarga de la delimitación y sincronización de la trama. En la sincronización se puede utilizar tres métodos:

- Principio y fin: Se utilizan caracteres específicos para identificar el principio o fin de cada trama.
- Principio y cuenta: Se utiliza un caracter para indicar el comienzo, seguido por un contador que indica su longitud.
- Guion: Se emplea una agrupación específica de bits para identificar el principio y fin mediante banderas o *flags*.

Aquí se proporciona la detección y corrección de errores en el envío de tramas entre nodos, sus funciones generales son:

- Identificar tramas de datos
- Códigos detectores y correctores de error
- Control de flujo
- Gestión y coordinación de la comunicación.

La detección de errores se puede realizar por varios tipos de código dentro de los cuales cabe resaltar los siguientes:

- Control de redundancia cíclica (CRC)
- Simple paridad
- Paridad cruzada (Paridad horizontal y vertical)
- Suma de verificación

Los métodos de control de errores son básicamente dos:

- Corrección de errores hacia adelante (*Forward Error Correction, FEC*) o corrección de errores por anticipado, no tiene control de flujo.

- Petición de repetición automática (*Automatic Repeat-reQuest, ARQ*): posee control de flujo mediante parada y espera, y/o ventana deslizante.

Las posibles implementaciones son:

- Parada y espera simple: Se envía una trama y se espera la señal de conformidad para enviar la siguiente trama, en caso se manifieste un error se enviará dicha trama nuevamente.
- Envío continuo y rechazo simple: Se envían una serie de tramas, el receptor las valida y en caso de encontrar una trama errónea eliminara todas las posteriores y enviará una petición para que se reenvíe toda la secuencia a partir de la trama errónea.
- Envío continuo y rechazo selectivo: La transmisión se realiza de manera continua, en caso de encontrarse un error pedirá solo reenviar la trama defectuosa.

Control de flujo: Esta función es necesaria para no saturar al receptor. Aunque normalmente se realiza en la capa de transporte, opcionalmente se puede realizar en la capa de enlace de datos. Utiliza mecanismos de retroalimentación, suele ir en conjunto con la corrección de errores y no debe limitar la eficiencia de un canal. Esta función lleva consigo dos acciones prioritarias las cuales son:

La detección de errores: Capaz de detectar errores al momento de enviar tramas al receptor e intentar solucionar dichos errores, esto se puede dar mediante CRC (verificación de redundancia cíclica), simple paridad, paridad cruzada o suma de verificación.

Recuperación de fallos: Esta función se utiliza para detectar y recuperar situaciones como:

- La pérdida de una trama.
- La aparición de tramas duplicadas
- La llegada de tramas fuera de secuencia.

Ventana deslizante: Es un mecanismo para el control de flujo de datos el mismo que existe entre un emisor y receptor, consta en proporcionar un buffer entre la aplicación y el flujo de datos de red, el receptor debe procesar los datos a cierta velocidad, a cada segmento del buffer se le asigna un temporizador que será un tiempo de espera para recibir confirmación del paquete. El concepto de ventana deslizante hace que exista una continua transmisión de información, mejorando el desempeño de la red.

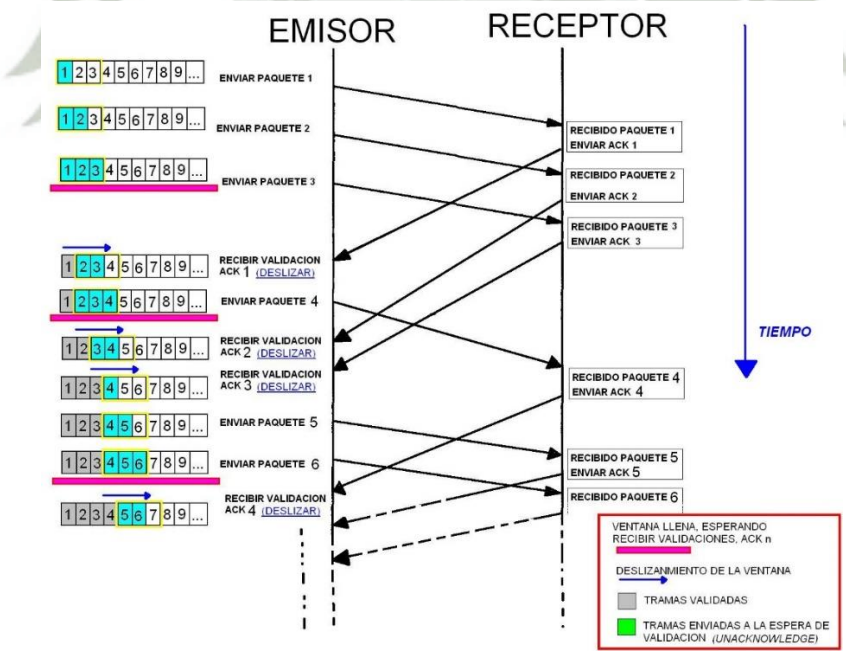


Figura 14: Ventana deslizante

Fuente: https://blogs.ua.es/redesitis/category/asignatura/

ARQ (*Automatic Repeat Request*): Protocolos de solicitud de repetición automática que se dan para el control de errores, teniendo como objetivo garantizar la integridad de los datos transmitidos, tratando de volver un enlace fiable.

Parada y espera (*Stop and Wait*): Utilizado para el control de comunicaciones está basado en el envío de un paquete y esperar hasta recibir una confirmación ACK, en caso de recibir un NACK se reenvía el paquete anterior

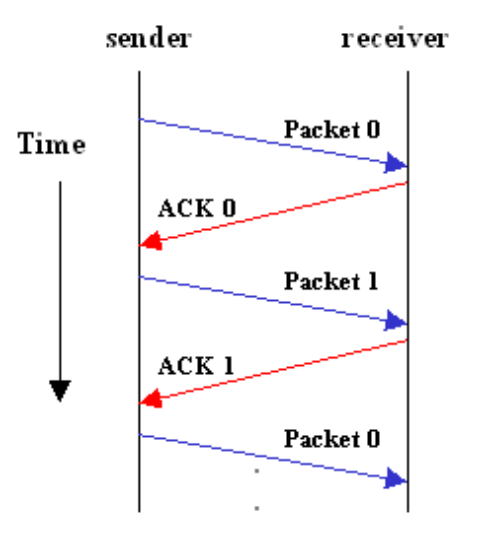


Figura 15: Parada y espera de control de comunicaciones

Fuente: *HTTPS*://sites.google.com/site/sistemasdemultiplexado/arquitecturas-de-las-redes-de--comunicacin-caractersticas/9--control-de-enlace-de-datos

ARQ con vuelta atrás N: Conocido también como *pullback NACK*, como se observa en la siguiente figura, si se envía las tramas 0, 1, 2, 3 y 4 y se recibe un NACK para la trama 2, se reenviará la trama 2, 3 y 4, aunque estas últimas hubiesen llegado de forma correcta.

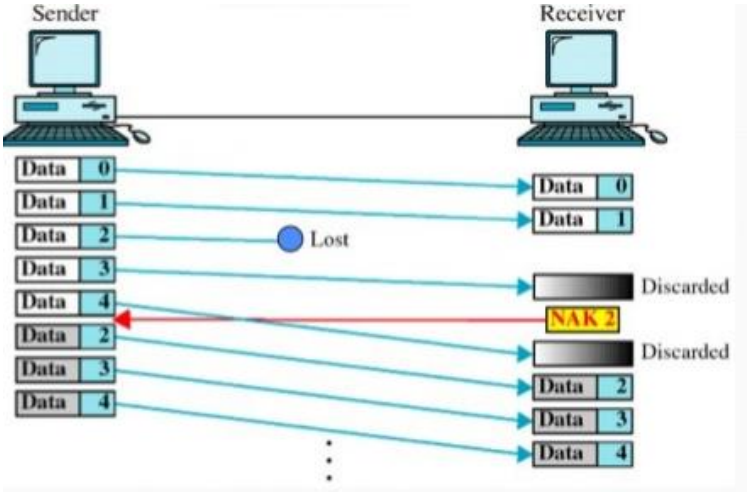


Figura 16:ARQ con vuelta atrás N

Fuente: *HTTPS*://sites.google.com/site/sistemasdemultiplexado/arquitecturas-de-las-redes-de--comunicacin-caractersticas/9--control-de-enlace-de-datos

ARQ con rechazo selectivo: Conocido como *selective repeat*, como se observa en la siguiente figura solo se reenviará la trama que contenga los errores.

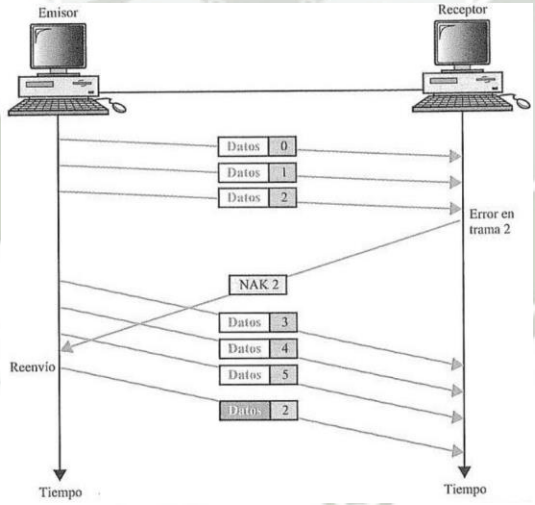


Figura 17: ARQ con rechazo selectivo

Fuente: *HTTPS*://sites.google.com/site/sistemasdemultiplexado/arquitecturas-de-las-redes-de--comunicacin-caractersticas/9--control-de-enlace-de-datos

ARP (Address resolution protocol): Este protocolo es el encargado de traducir direcciones IP a direcciones MAC las cuales son direcciones físicas, para

realizar esta conversión se utiliza tablas ARP, como se observa en la siguiente tabla cada interfaz cuenta tanto con una dirección IP como con una dirección física MAC

IP	MAC
192.168.1.1	F0:E1:D2:C3:B4:A5
192.168.1.10	E4-11-5B-2F-56-F5
192.168.1.20	10-0B-A9-87-A9-E0

Tabla 1: Tabla ARP

Fuente: Elaboración Propia

Casos en los que se puede utilizar ARP:

- Cuando dos hosts están en la misma red y uno quiere enviar un paquete a otro.
- Cuando dos hosts están sobre redes diferentes y deben usar un gateway o router para alcanzar otro host.
- Cuando un router necesita enviar un paquete a un host a través de otro router.
- Cuando un router necesita enviar un paquete a un host de la misma red.

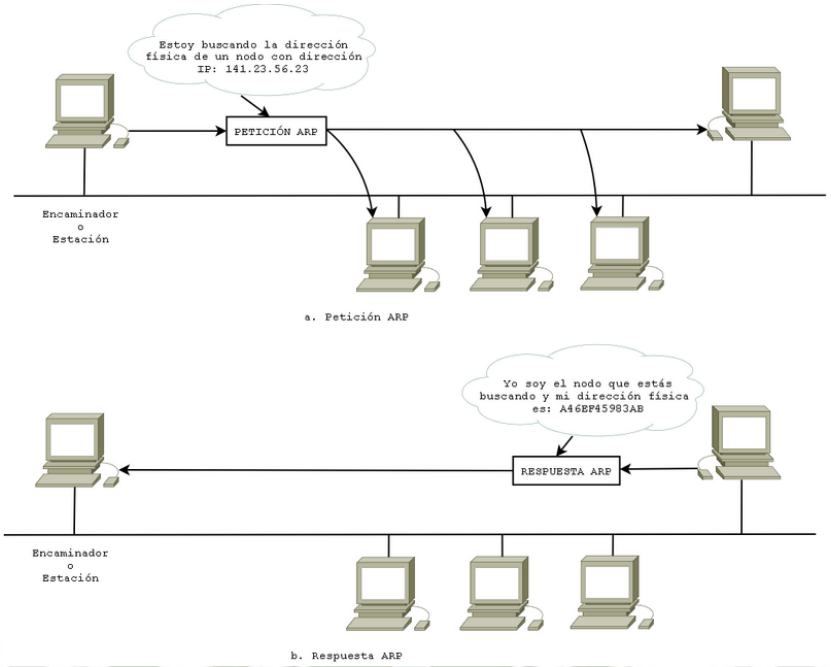


Figura 18: Funcionamiento de ARP

Fuente: geekytheory.com/redes-el-protocolo-arp

En la siguiente figura se observa la cabecera específica de una solicitud/respuesta ARP, a continuación, se especifica cada uno de sus campos.

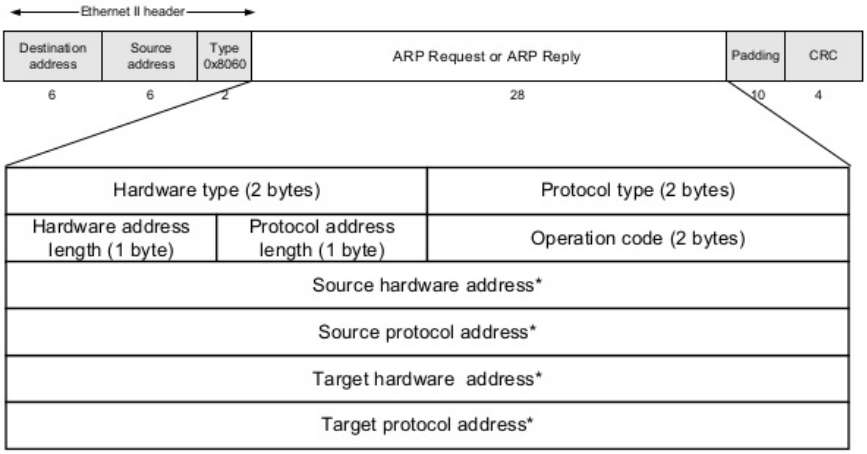


Figura 19: Cabecera ARP

Fuente: www.cisco.com

- Tipo de hardware o Hardware Type (HTYPE): Aquí se especifica el tipo de protocolo de enlace (Ethernet es 1), a continuación, se observa una tabla con los valores que puede tomar dicho campo.

Valor	Descripción	Referencia
0	reserved.	RFC 5494
1	Ethernet.	
2	Experimental Ethernet.	
4	Proteon ProNET Token Ring.	
15	Frame Relay.	
17	HDLC.	
23	Metricom.	
30	ARPSec.	
31	IPsec tunnel.	RFC 3456

Tabla 2: Valor y tipo de hardware:

Fuente: *HTTP*://www.networksorcery.com/enp/

- Tipo de protocolo o Protocol Type (PTYPE): Aquí se especifica el protocolo de interconexión de redes para las que se destina la petición ARP. Para IPv4, esto tiene el valor 0x0800.

- Longitud Hardware (HLEN): Aquí se especifica la longitud (bytes/octeto) de una dirección física, en ethernet el tamaño es de 6.

- Longitud del Protocolo (PLEN): Aquí se especifica la longitud en (bytes/octetos) de las direcciones del protocolo en el paquete de capa superior, para IPv4 será de 4

- Operación: Aquí se especifica la operación que se realizara, su contenido es de 16 bits y se aprecia en la siguiente tabla.

Valor	Descripción	Referencia
0	reserved.	RFC 5494
1	Request.	RFC 826, RFC 5227
2	Reply.	RFC 826,
3	Request Reverse.	RFC 903
4	Reply Reverse.	RFC 903
5	DRARP Request.	RFC 1931
6	DRARP Reply.	RFC 1931
7	DRARP Error.	RFC 1931
8	InARP Request.	RFC 1293
9	InARP Reply.	RFC 1293
10	ARP NAK.	RFC 1577

Tabla 3: Valor y descripción de operacion

Fuente: *HTTP*://www.networksorcery.com/enp/

- Dirección de hardware del remitente (SHA)/ Dirección de hardware de destino (THA): Contiene la dirección física del hardware del remitente, en IEEE802.3 la dirección es de 48 bits, en THA este campo es ignorado en solicitudes.
- Remitente dirección de protocolo (SPA) / Dirección de protocolo target (TPA): Contiene las direcciones del protocolo, por ejemplo, en TCP/IP son direcciones IP de 32 bits.

RARP (Reverse ARP): El Protocolo de resolución direcciones inversa, como se observa en la siguiente figura la función es inversa a ARP, es decir se tiene la dirección física (MAC) pero no conocemos la dirección lógica (IP).

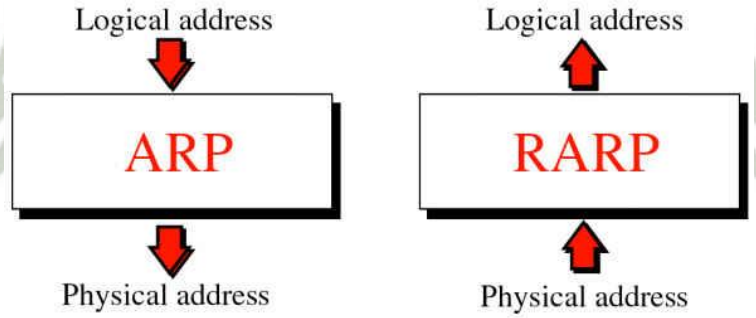


Figura 20: Función de ARP y RARP

Fuente: *HTTP*://iesramonycajaltocina.es/ciclo/Vocabulario/vocabulario_t2/html/arp.html

ARP Proxy: Se utiliza para proporcionar un enrutamiento ad hoc, como se observa en la siguiente figura un dispositivo como un router que implementa proxy ARP será el encargado de responder a las peticiones de ARP, en este caso el dispositivo podrá recibir y enviar paquetes dirigidos a los demás dispositivos.

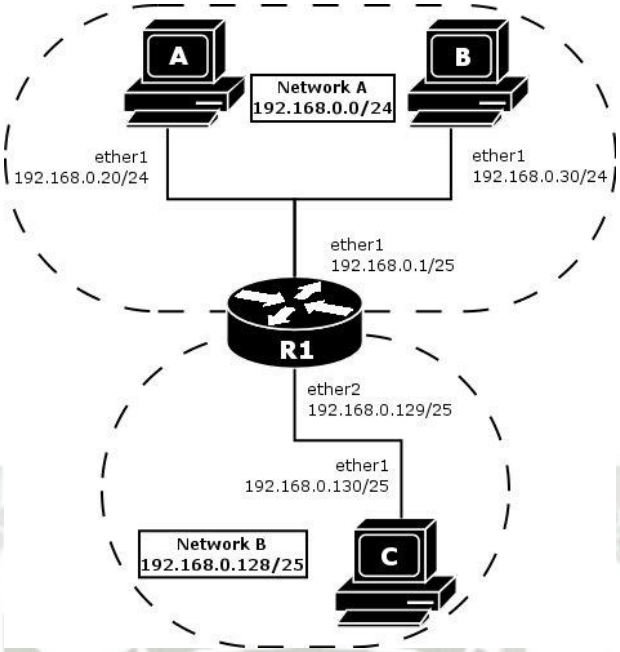


Figura 21: Proxy ARP

Fuente: *HTTP*://wirelessconnect.eu/articles/ip_addresses_and_arp

Descubrimiento de Vecino (ND): Este protocolo de IPv6, el cual es equivalente al protocolo ARP, aunque también incorpora funcionalidades del protocolo ICMP, por lo cual utiliza mensajes de ICMPv6, consiste en un mecanismo en donde al incorporar un nodo nuevo este descubre la presencia de otros nodos del mismo enlace, además de determinar las direcciones físicas para localizar routers y tener información de conectividad sobre las rutas de vecinos activos. Este protocolo es la base para permitir mecanismos de autoconfiguración en redes IPv6, así como también se emplea para mantener limpios los caches en donde se almacena información de la red y así detectar cualquier cambio, de este modo si una ruta falla se buscará nuevas alternativas.

Protocolo de Punto a Punto (PPP): Es un protocolo de punto a punto para establecer una conexión directa entre dos dispositivos, puede proveer de las siguientes funciones:

- **Autenticación de conexión:** Puede darse por PAP y por CHAP.
- **Cifrado de transmisión:** Se utiliza *encryption control protocol (ECP)* como se aprecia a continuación este protocolo se utiliza para establecer y configurar algoritmos de cifrado de datos sobre PPP, como se aprecia en la siguiente figura se examinará la cabecera ECP

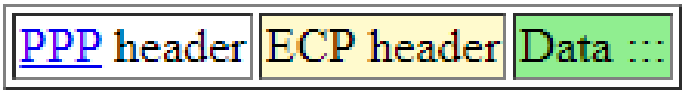


Figura 22: Cabecera ECP

Fuente: *HTTP*://www.networksorcery.com/enp/protocol/ecp.htm

Teniendo ECP la siguiente cabecera

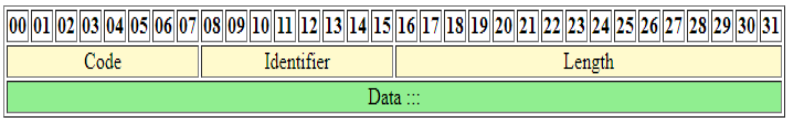


Figura 23: Cabecera ECP detallada

Fuente: *HTTP*://www.networksorcery.com/enp/protocol/ecp.htm

En donde el código tiene la función a realizar según los valores que se puede ver en la siguiente tabla.

Código	Descripción	Referencia

0	**Vendor Specific.**	**RFC 2153**
1	**Configure-Request.**	
2	**Configure-Ack.**	
3	**Configure-Nak.**	
4	**Configure-Reject.**	
5	**Terminate-Request.**	
6	**Terminate-Ack.**	
7	**Code-Reject.**	
14	**Reset-Request.**	**RFC 1968**
15	**Reset-Ack.**	**RFC 1968**

Tabla 4: Tabla de valores de campo código en cabecera ECP

Fuente: Elaboración propia

EL identificador es utilizado para hacer coincidir las solicitudes y respuestas.

Length: tamaño del paquete que incluye el encabezado

Data: puede tener cero o más bytes según lo indica el campo length, este campo puede tener una o más opciones para la configuración de ECP como se muestra en la siguiente figura de cabecera y tabla:

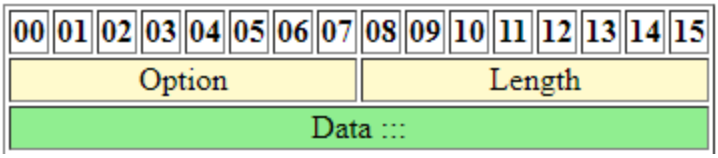


Figura 24: Cabecera de data en ECP

Fuente: *HTTP*://www.networksorcery.com/enp/protocol/ecp.htm

.

Opción	Longitud	Descripción	Referencia
0	>= 6	OUI	RFC 1968
1	10	DESE, PPP DES	RFC 1969
2	10	3DESE, PPP Triple-DES	RFC 2420
3	10	DESE-bis, PPP DES Data	RFC 2419

Tabla 5: Opciones de especificación de ECP

Fuente: Elaboración Propia

Compresión: reduce el tamaño de la trama que debe enviar de esta manera mejora el rendimiento en las conexiones PPP.

Link Control Protocol (LCP): Este protocolo de control ofrece diversas opciones de encapsulación para PPP tales como:

- **Autenticación**: Información para identificar al usuario, los dos métodos para ellos son PAP y CHAP.

- **Comprensión**: Utilizado para el aumento del rendimiento, se comprime el *payload* o carga de datos antes de ser transmitidos, al recibirlos de descomprime y recupera la trama.
- **Detección** de Errores: Detección de errores comunes de configuración
- **Multilink**: permite mantener separados canales físicos pero que se vean como un solo canal lógico en el nivel de capa de red.
- **PPP Callback**: El retorno de llamada, se da luego de la autenticación es decir luego de una conexión entre un cliente y un servidor será el servidor quien inicie la conexión entre routers.

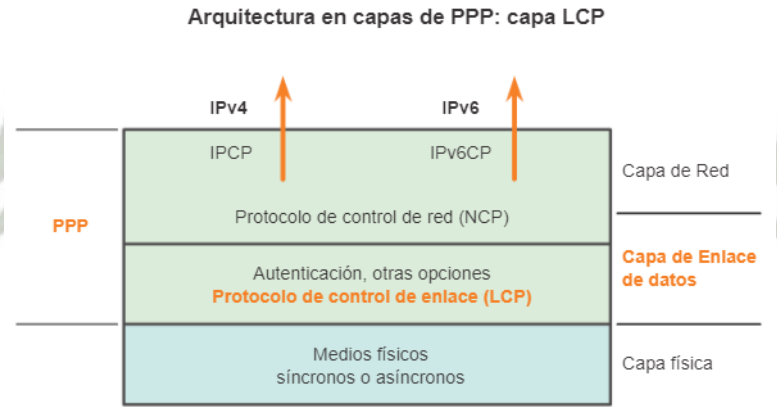


Figura 25: Arquitectura en capas de PPP

Fuente: *HTTP*://www.itesa.edu.mx/netacad/networks/course/module3/3.2.2.1/3.2.2.1.html

Protocolo de Túnel de Punto a Punto (PPTP): Es un protocolo para implementar redes privadas virtuales, permite dar un intercambio de datos seguro entre un cliente y un servidor

Reenvío de Capa 2 (L2f): Protocolo desarrollado por cisco, también utiliza PPP para autenticarse, pero añade autenticación por medio de RADIUS o TACACS+, aunque el mecanismo de autenticación es robusto no cuenta con cifrado.

Protocolo de Túnel de Capa 2(L2tp): PPTP y L2F convergieron en el protocolo de túnel de capa 2 en el cual se agrega propiedades de encapsulamiento de datos, por esto puede transmitir cualquier tipo de protocolo añadiendo una cabecera al mensaje original y transmitiendo este por el túnel, l2tp consta de mensajes de datos y mensajes de control, los primero tiene el mensaje original encapsulado y los de control aseguran que el mensaje llego a su destino de manera correcta, es importante saber que l2tp no cuenta con mecanismos de seguridad por lo cual deberá trabajar en conjunto con IPSEC.

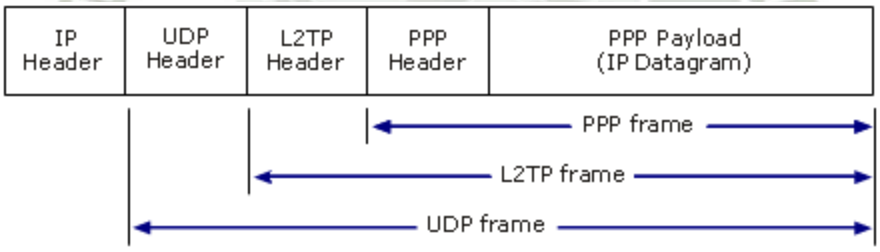


Figura 26: Estructura de cabecera L2TP

Fuente: *HTTPS*://technet.microsoft.com/dynimg/

MAC ACL (Lista de control de acceso basada en direcciones físicas): La lista de control de acceso configurada en esta capa puede aplicarse a una o más interfaces, así como establecer múltiples listas de acceso a una única interfaz, hay que tener en cuenta que no se puede configurar una ACL de MAC y una ACL de IP en una misma interfaz, puede redirigir paquetes, así como establecer reglas para inspeccionar los siguientes campos de un paquete:

- Dirección MAC origen con mascara
- Dirección MAC destino con mascara
- ID de *VLAN*

Red de Área Local Virtual en Enlace de Datos: En este tipo de *VLAN* los *switch*es son configurados o aprenden a que *VLAN* pertenece cada dirección física, en la siguiente figura se aprecia las *VLAN*S divididas según la dirección MAC de cada nodo.

VLAN1	11:22:33:44:55:66
VLAN2	22:33:44:55:66:77
VLAN3	33:44:55::66:77:88
VLAN1	00:22:33:44:55:66
VLAN3	33:44:55::66:77:99
VLAN3	00:11:55::66:77:88

Figura 27: División de *VLAN*

Fuente: *HTTP*://redesconfiguracion.blogspot.pe/2015/07/que-es-una-*VLAN*-y-su-funcion.html

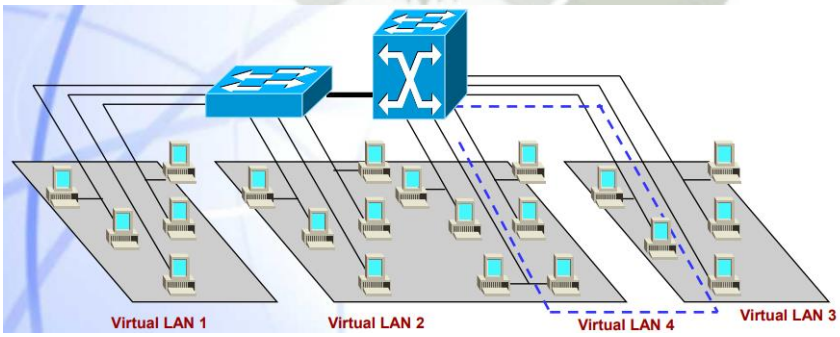


Figura 28: Estructura de VLAN

Fuente: *HTTP*://redesconfiguracion.blogspot.pe/2015/07/que-es-una-*VLAN*-y-su-funcion.html

El enlace de interconexión (inter*switch*es) utiliza un mecanismo de etiquetado o marcador para intercambiar tráfico entre *VLAN*s, los *switch*es deben saber que nodos soportan etiquetas y cuales no lo soportan, en la siguiente figura se observa el intercambio de información entre *VLAN*s, esta es transmitida a través de los *switch*es.

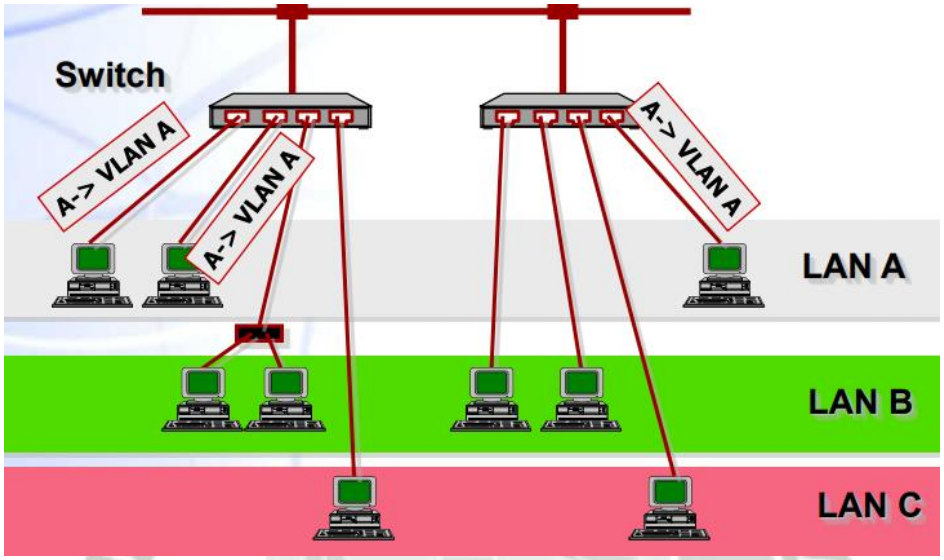


Figura 29: Conexión de *switch*es de distintas *VLAN*

Fuente: *HTTP*://redesconfiguracion.blogspot.pe/2015/07/que-es-una-*VLAN*-y-su-funcion.html

En la siguiente figura se aprecia la cabecera de etiquetado *VLAN* (*VLAN* Tag), la cual se añade a la dirección física de origen y destino.

Figura 30: Cabecera de *VLAN*

Fuente: *HTTPS*://juanmhalegre.wordpress.com/2012/01/12/ccnp-*switch*-642-813-official-certification-guide-part-ii-chapter-4-2-*VLAN*-trunks/

Firewall Transparente: Cuando dos segmentos de red físicamente separados se conectan guiados por direcciones físicas, en este nivel el firewall es casi imperceptible.

2.3.2 Capa 2 Red

La capa de Red (Internet) es la encargada de la transmisión y direccionamiento de datos entre host situados en redes diferentes.

IPv4

0-3	4-7	8-15	16-18	19-31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador			Flags	Posición de Fragmento
Tiempo de Vida		Protocolo	Suma de Control de Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones				Relleno

Figura 31: Estructura de paquete IPv4

Fuente: *HTTPS*://www.tutorialspoint.com/es/ipv4/ipv4_packet_structure.htm

- Versión: El tamaño es de 4 bits en el que se describe el formato de la cabecera que se utiliza, dependiendo de la versión que se utilice IPv6 o IPv4.
- **Tamaño Cabecera (IHL):** El tamaño es de 4 bits, aquí se encuentra la longitud del encabezado IP.
- **Tipo de Servicio (DSCP):** El tamaño es de 8 bits, existen redes capaces de ofrecer una calidad de servicio en las que se considera una prioridad con

determinado tipo de paquetes, aquí se indica varios parámetros para establecer la prioridad, como se observa en la siguiente tabla los 3 primeros bits se relacionan con el origen del mensaje, con un indicador que establece el nivel de urgencia, el cual está basado en el sistema militar de procedencia.

Bits	Descripción
000	De rutina.
001	Prioritario.
010	Inmediato.
011	Relámpago.
100	Invalidación relámpago.
101	Procesando llamada crítica y de emergencia.
110	Control de trabajo de Internet.
111	Control de red.

Tabla 6: Tipo de servicio DSCP

Fuente: Elaboración Propia

Como se observa en la siguiente tabla los siguientes 5 bits son independientes, están encargados de indicar características del servicio.

Nro. de bits	Descripción	Opciones
0 a 2	Prioridad	--

3	Retardo	0 = normal; 1 = bajo
4	Rendimiento	0 = normal; 1 = bajo
5	Fiabilidad	0 = normal; 1 = bajo
6-7	No usados, reservados para uso futuro.	--

Tabla 7: Características de servicio

Fuente: Elaboración Propia

- **Longitud Total:** Consta de un tamaño de 16 bits, indica la longitud del paquete IP incluyendo en el mismo la cabecera y carga IP, normalmente se utiliza un tamaño de 576 octetos divididos en 64 octetos de cabecera y 512 octetos de carga. En el caso de una fragmentación este campo contendrá el tamaño del fragmento no el datagrama original.

- **Identificador**: Conformado por 16 bits, es un identificador único del datagrama, si el paquete IP se fragmento todos los fragmentos tendrán el mismo identificador para saber que pertenecen al mismo paquete IP.

- ***Flags***: Conformado por 3 bits, utilizado en la fragmentación, en la siguiente tabla se puede observar los valores que toma, siempre se debe tener en cuenta la situación en la que un paquete es indivisible, si este paquete necesita ser fragmentado no se enviará.

Bit	Descripción

0	Reservado; debe ser 0
1	0 = Divisible, 1 = No Divisible (DF)
2	0 = Último Fragmento, 1 = Fragmento Intermedio

Tabla 8: Valores de *flags*

Fuente: Elaboración Propia

- **Posición de Fragmento**: Conformado por 13 bits, aquí se indica la posición del fragmento en el paquete IP.
- **Tiempo de Vida (TTL):** Consta de 8 bits, aquí se indica el número de saltos que puede dar el paquete, en cada salta el valor decrementa en 1 y al llegar a 0 el paquete es descartado, esto se realiza para para evitar bucles.
- **Protocolo**: Conformado por 8 bits, se especifica el protocolo de las capas superiores al que debe entregarse el paquete, como se observa en la siguiente tabla son algunos valores que puede tomar este campo.

Decimal	HEX	Sigla	Protocolo	Referencias
1	0x01	ICMP	Internet Control Message Protocol	RFC 792
4	0x04	IP	IP en IP (encapsulación)	RFC 2003

6	0x06	TCP	Transmission Control Protocol	RFC 793
17	0x11	UDP	User Datagram Protocol	RFC 768

Tabla 9: Valores del campo protocolo

Fuente: Elaboración Propia

- **Suma de Control de Cabecera**: Conformado por 16 bits, conocido como el *checksum* de encabezado, utilizado para saber si el paquete fue recibido sin inconvenientes.
- **Dirección IP de origen:** Consta de 32 bits, contiene la dirección del nodo que origino el paquete.
- **Dirección IP de destino:** Conformado por 32 bits contiene la dirección del nodo destino.
- **Opciones:** De longitud variable, este campo es opcional pero cualquier nodo debe ser capaz de interpretarlo. Puede contener un número indeterminado de opciones como marca de tiempo, seguridad, ruta de registro, etc.
- **Relleno:** Este campo es de longitud variable, se utiliza para verificar que el tamaño de bits en la cabecera es múltiplo de 32, el valor utilizado es el de 0.

Cabecera de IPv6

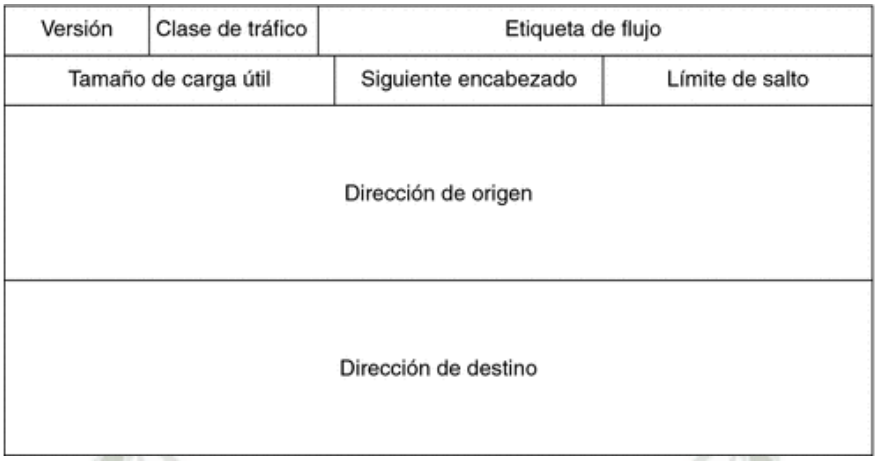


Figura 32: Estructura de IPV6

Fuente: *HTTPS*://docs.oracle.com/cd/E19957-01/820-2981/ipv6-ref-76/index.html

- Versión: Su tamaño es de 4 bits, indica la versión del protocolo en este caso IPv6.
- Clase de tráfico: Conformado por 8 bits, aquí se establece la prioridad, es equivalente a TOS en IPv4.
- Etiqueta de flujo: Su tamaño es de 20 bits, utilizado para permitir tráfico con requerimientos de tiempo real.
- Tamaño de carga útil: Consta de 16 bits, hace referencia el resto del paquete que sigue al encabezado de IPv6, en octetos.
- Encabezado siguiente: Conformado de 8 bits, utilizado para identificar el protocolo al que pertenece la información tales como puede set TCP, UDP, ICMPv6, etc. También es usado para indicar las opciones para los datos que se transportan, a diferencia de IPv4 que maneja esto mediante opciones, en

IPv6 estas se integran en el *payload* de los paquetes así se mantiene el encabezado de un tamaño fijo.

- **Límite de salto:** Conformado por 8 bits, es el TTL, el cual decrementa en uno por cada nodo que reenvía le paquete, si este límite llega a cero el paquete es descartado,
- **Dirección de origen**: Consta de 128 bits, es la dirección del nodo origen.
- **Dirección de destino**: Consta de 128 bits, es la dirección del nodo destino

Seguridad de Protocolo de Internet (IPSEC): Consiste en un grupo de protocolos encargados de asegurar las comunicaciones sobre el protocolo IP, así como el cifrado y autenticación de cada paquete IP en cada flujo de datos, también se incluyen protocolos para establecer claves de cifrado (IKE), IPSEX puede establecer dos modos básicos de operación los cuales son:

- **Modo transporte:** Utilizado para comunicaciones de nodo a nodo, aquí solo se cifra el *payload* del paquete IP, debido a que no se cifra ni modifica la cabecera IP el enrutamiento permanece intacto,
- **Modo túnel**: Utilizado para comunicaciones entre redes como túneles seguros entre routers, aquí todo el paquete IP es decir el *payload* y cabecera son cifrados, por lo cual debe ser encapsulado en un nuevo paquete IP para que el enrutamiento no tenga problemas.

IPSEC consta de los siguientes protocolos para dar seguridad a nivel de paquetes IPv4/IPv6:

Authentication Header (AH):

0 - 7 bit	8 - 15 bit	16 - 23 bit	24 - 31 bit
Next header	Payload length	RESERVED	
Security parameters index (SPI)			
Sequence number			
Hash Message Authentication Code (variable)			

Figura 33: Cabecera de AH

Fuente:
https://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html

- ***Next header*:** Identifica cual es el protocolo de los datos transferidos.
- ***Payload length***: Tamaño del paquete AH.
- ***Reserved*** Actualmente se encuentra todo en 0, ya que está reservado para emplearlo en el futuro.
- ***Security parameters index (SPI)*** Aquí se indica los parámetros de seguridad los cuales en conjunto con la dirección IP, identifican la asociación de seguridad.
- ***Sequence number***: Un número que se incrementa, se utiliza para prevenir ataques de repetición, este valor está incluido en los datos cifrados por lo que una modificación seria detectada.
- ***HMAC***: Es un código de autenticación de mensaje en hash, este campo contiene el valor para verificar la integridad.
- ***ESP***: Protocolo encargado de autenticación de origen, protección de la confidencialidad e integridad de un paquete, se puede establecer una configuración para solo cifrado o solo

autenticación, esta configuración seria insegura, en la siguiente figura se observa los campos de ESP.

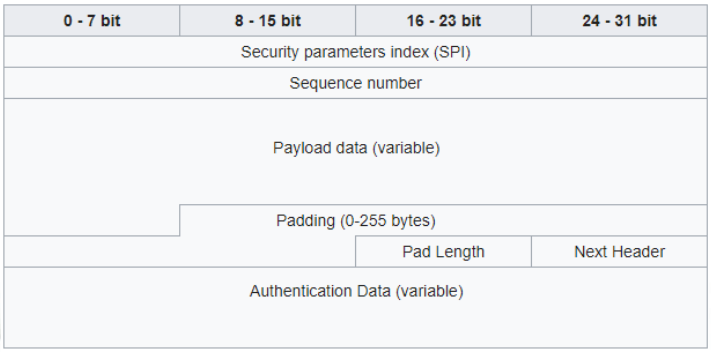


Figura 34: Cabecera ESP

Fuente: https://supportforums.cisco.com/t5/security-documents/esp/ta-p/3113341

- ***Security parameters índex (SPI)***: Aquí se dimensiona los parámetros de seguridad
- ***Sequence number:*** Número que se incrementa, es empleado para evitar la repetición
- ***Payload data***: La carga, es decir los datos que se transferirán
- ***Padding***: Es utilizado por algoritmos de cifrado para llenar los bloques.
- ***Pad length:*** Tamaño del relleno el cual se da en bytes.
- ***Next header:*** Identifica la naturaleza del *payload*, si este es de protocolo TCP o UDP.
- ***Authentication data:*** Contiene el valor de comprobación de integridad, y un código de autenticación de mensaje el cual se

emplea para verificar el nodo origen al igual que la integridad del mensaje

Protocolo de mensajes de control de internet (ICMP): Este protocolo de control de mensajería y error de internet, se utiliza para mejorar el control y errores, esto es necesario para los sistemas de la red para poder evitar o corregir errores que fueron detectados, este protocolo solo se encarga de informar la entrega de paquetes o errores en la red mas no de tomar una decisión, esto se realizara en capas superiores.

	Bit 0–7	Bit 8–15	Bit 16–23	Bit 24–31
0	Tipo	Código	Suma de verificación	
32	Datos sobre la cabecera			

Figura 35: Cabecera ICMP

Fuente: *HTTPS*://www.1and1.es/digitalguide/servidores/know-how/que-es-el-protocolo-ICMP-y-como-funciona/

En la siguiente tabla se ve el valor que puede tomar para el campo tipo de esta cabecera, e indicar el valor para los mensajes informativos.

Código	Descripción
0	Echo Reply (respuesta de eco)
3	Destination Unreacheable(destino inaccesible)
4	Source Quench (disminución del tráfico desde el origen)
5	Redirect (redireccionar - cambio de ruta)
8	Echo (solicitud de eco)
11	Time Exceeded (tiempo excedido para un datagrama)
12	Parameter Problem(problema de parámetros
13	Timestamp (solicitud de marca de tiempo)
14	Timestamp Reply (respuesta de marca de tiempo)
15	Information Request(solicitud de información) - obsoleto-

16	Information Reply (respuesta de información) - obsoleto-
17	Addressmask (solicitud de máscara de dirección)
18	Addressmask Reply(respuesta de máscara de dirección
19	Reservado para seguridad
20-29	Reservado para experimentos de robustez
30	Traceroute
31	Error de Conversión de Datagrama
32	Redirección de Host Móvil
33	IPv6
34	Petición de Registro de Móvil
35	Respuesta de registro de Móvil
36	Petición de Nombre de Dominio
37	Respuesta de Nombre de Dominio
38	SKIP Protocolo de Algoritmo de Descubrimiento
39	Photuris, Fallas de Seguridad
40-255	Reservado

Tabla 10: Valores de ICMP

Fuente: https://www.certificationkits.com/cisco-certification/ccent-640-822-icnd1-exam-study-guide/cisco-ccent-icnd1-640-822-exam-certification-guide/cisco-ccent-icnd1-tcpip-part-ii/

En la siguiente tabla se muestra el valor y la descripción de los valores para mensajes de error.

Código	Descripción
0	no se puede llegar a la red
1	no se puede llegar al host o aplicación de destino
2	el destino no dispone del protocolo solicitado
3	no se puede llegar al puerto destino o la aplicación destino no está libre
4	se necesita aplicar fragmentación, pero el *flag* correspondiente indica lo contrario
5	la ruta de origen no es correcta
6	no se conoce la red destino

7	no se conoce el host destino
8	el host origen está aislado
9	la comunicación con la red destino está prohibida por razones administrativas
10	la comunicación con el host destino está prohibida por razones administrativas
11	no se puede llegar a la red destino debido al Tipo de servicio
12	no se puede llegar al host destino debido al Tipo de servicio

Tabla 11: Valores de mensaje de error

Fuente: *HTTP*://neo.lcc.uma.es/evirtual/cdd/tutorial/red/ICMP.html

Protocolo de mensajes de control de internet versión 6 (ICMPv6): Este protocolo es una nueva versión del protocolo ICMP, en este nuevo protocolo se combinan funciones que anteriormente se encontraban divididas en varias partes de diferentes protocolos tales como ARP, IGMP o ICMP, también elimina algunos tipos de mensajes que se encontraban obsoletos o en desuso a la actualidad. En la siguiente figura se observa algunos de los valores para ICMPv6 en cuanto a los mensajes de error y mensajes informativos.

Mensajes de error ICMPv6		
Tipo	Descripción y Códigos	
1	Destino no alcanzable (Destination Unreachable)	
	Código	Descripción
	0	Sin ruta hacia el destino
	1	Comunicación prohibida administrativamente
	2	Sin asignar
	3	Dirección no alcanzable
	4	Puerto no alcanzable
2	Paquete demasiado grande (Packet Too Big)	
3	Tiempo excedido (Time Exceeded)	
	Código	Descripción
	0	Límite de saltos excedido
	1	Tiempo de desfragmentación excedido
4	Problema de parámetros (Parameter Problem)	
	Código	Descripción
	0	Campo erróneo en cabecera
	1	Tipo de "cabecera siguiente" desconocida
	2	Opción IPv6 desconocida
Mensajes informativos ICMPv6		
Tipo	Descripción	
128	Solicitud de eco (Echo Request)	
129	Respuesta de eco (Echo Reply)	

Figura 36: ICMPV6

Fuente: *HTTP*://tunnelingipv6.blogspot.pe/2009/11/ICMPv6.html

Traducción de direcciones de red (NAT): Es un mecanismo utilizado para hacer frente a la escasez de direcciones IPv4, se utiliza para conectar una o más redes LAN internas a internet mediante una o un grupo de direcciones IP, en la siguiente figura se observa la conexión de distintas maquinas a direcciones de internet de IP públicas, las direcciones son traducidas tanto al realizar la petición como al recibir la respuesta.

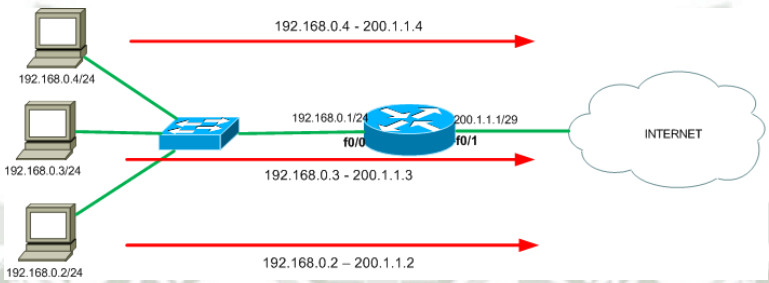


Figura 37: Funcionamiento NAT

Fuente: *HTTP*://www.redescisco.net/sitio/2010/08/18/implementando-nat-en-routers-cisco/

Firewall de capa red: En este mecanismo de seguridad se pueden establecer filtros de seguridad según los campos de los paquetes IP, aunque también tienen la capacidad de trabajar en la capa de transporte, teniendo una gran versatilidad en el tratamiento de tráfico que se transmite, algunas de estas medidas puedes ser:

- Dirección IP de origen y de destino.
- Protocolos utilizados.
- Puerto TCP-UDP de origen y de destino.

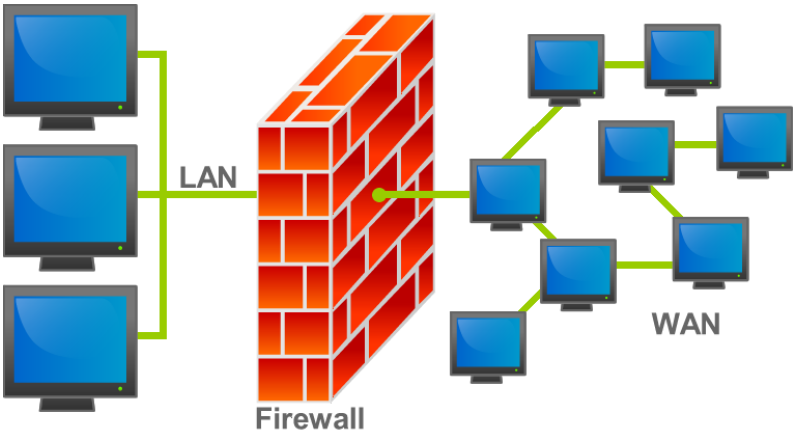


Figura 38: Firewall de capa de red

Fuente: https://threatpost.com/cisco-high-severity-flaw-lets-malware-bypass-firepower-firewall/117165/

Routing Information Protocol (RIP): Este protocolo es utilizado por los routers para transmitir información acerca de las redes IP a las que se encuentran conectados, los algoritmos de encaminamiento están basados en el vector de distancia para calcular la ruta más corta al destino, el número de saltos en RIP es de 15 por lo que al pasar este límite se considerara como ruta inalcanzable, esto también es una desventaja para RIP ya que no toma en cuenta criterios como congestión, carga o retardo, RIP solo se puede usar en redes pequeñas o medianas debido al tamaño máximo de saltos.

2.3.3 Capa 3 Transporte

Protocolo de control de transmisión (TCP): Es un protocolo que permite que los nodos conectados controlen el estado de la transmisión de datos.

Las conexiones TCP tienen tres etapas las cuales son:

- Establecer conexión.
- Transferir datos
- Finalizar conexión

Las características del TCP son:

- Orientado a la conexión: Se refiere al establecimiento de conexión entre dos nodos para realizar el intercambio de datos, esta sincronización maneja el flujo de paquetes teniendo en cuenta la congestión de la red para poder adaptarse.
- Operación Full-Duplex: Las conexiones TCP permiten recibir y transmitir datos de manera simultánea.

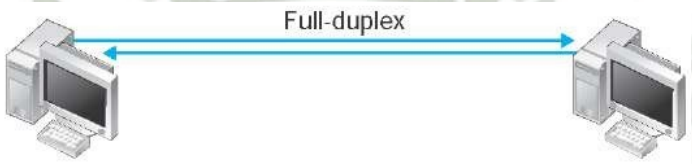


Figura 39: Conexión full-duplex

Fuente: *HTTP*://what-when-how.com/data-communications-and-networking/circuits-data-communications-and-networking/

- ***Error Checking***: Utilizado para verificar que no exista alteración en los paquetes, se utiliza la técnica de *checksum*.
- ***Acknowledgements***: Confirmación de recibo de uno o más paquetes, si esta confirmación no es enviada los paquetes pueden ser enviados nuevamente o terminar la conexión si el nodo que recibe la información ya no está conectado.
- ***Flow Control***: El control de flujo se da para evitar el desbordamiento de buffer, si no se dan las confirmaciones

de paquetes o acknowledgement fallidos el emisor puede reducir la tasa de transferencia.

- ***Servicio de recuperación de Paquete***s: El receptor puede solicitar que se vuelva a enviar un paquete, del mismo modo si no se envía un acuse de recibo el paquete será enviado nuevamente.

En la siguiente figura se observa la cabecera de TCP:

Offsets Octeto		0								1								2								3							
Octeto	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Puerto de origen																Puerto de destino															
4	32	Número de secuencia																															
8	64	Número de acuse de recibo (si ACK es establecido)																															
12	96	Longitud de Cabecera				Reservado				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Tamaño de Ventana														
16	128	Suma de verificación																Puntero urgente (si URG es establecido)															
20 ...	160 ...	Opciones (Si la Longitud de Cabecera > 5, relleno al final con "0" bytes si es necesario) ...																															

Figura 40: Cabecera TCP

Fuente: https://www.cisco.com/c/es_mx/support/docs/ip/generic-routing-encapsulation-gre/25885-pmtud-ipfrag.html

Protocolo de datagrama de usuario (UDP): Es un protocolo basado en el intercambio de datagramas, este protocolo no es orientado a conexión a diferencia de TCP, no existe el *windowing* ni retransmisión, así como tampoco existe una secuencia de paquete ni confirmaciones de recepción, usualmente son usadas por aplicaciones dedicadas a *streaming*, dentro de sus principales características podemos destacar las siguientes:

- Es un protocolo mínimo de esta capa, orientado a datagramas documentado en RFC 768

- Proporciona una interfaz sencilla que conecta la capa de aplicación con la capa de red
- No existe un control ni garantía sobre la entrega de los mensajes.
- Es bastante utilizado en aplicaciones de *streaming*, por ejemplo, al transmitir video o voz es prioritario la velocidad antes de la confirmación de la recepción de todos los bytes.

La cabecera de UDP es bastante simple como se aprecia en la siguiente figura.

+	Bits 0 - 15	16 - 31
0	Puerto origen	Puerto destino
32	Longitud del Mensaje	Suma de verificación
64	Datos	

Figura 41: Cabecera UDP

Fuente: *HTTP*://www.it.uc3m.es/lpgonzal/protocolos/transporte.php

Stream Control Transmission Protocol (SCTP): Este protocolo es una alternativa tanto a TCP como a UDP, este protocolo provee confiabilidad, control de flujo y secuenciación como TCP, pero también permite el envío de mensajes fuera de orden, y es un protocolo orientado al mensaje similar a UDP, brinda soporte de multiconexión y multi*streaming*.

Seguridad de la capa de transporte (TLS): Este protocolo permite que dos nodos se puedan comunicar garantizando la privacidad e integridad de los datos, proporciona seguridad a través de internet entre las aplicaciones cliente/servidor, utiliza criptografía asimétrica para la autenticación entre dos entidades e intercambiar una llave simétrica, esto generara una sesión que es utilizada para cifrar el flujo de datos entre nodos, en la siguiente figura se aprecia una breve descripción así como la ubicación de TLS.

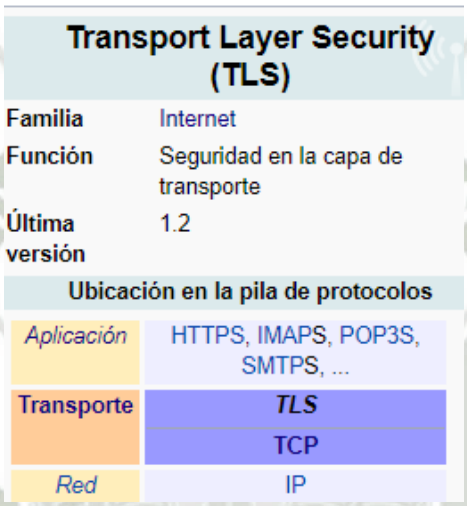


Figura 42: Ubicación de TLS

Fuente: https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/uc_system/TLS/TLS-1-2-for-On-Premises-Cisco-Collaboration-Deployments.html

Firewall Circuito a nivel de pasarela: Este mecanismo se da al establecer una conexión utilizando los protocolos TCP o UDP, una vez la conexión es establecida los nodos pueden transmitir sin más control, aquí se puede establecer conexiones de una zona segura a una no tan segura.

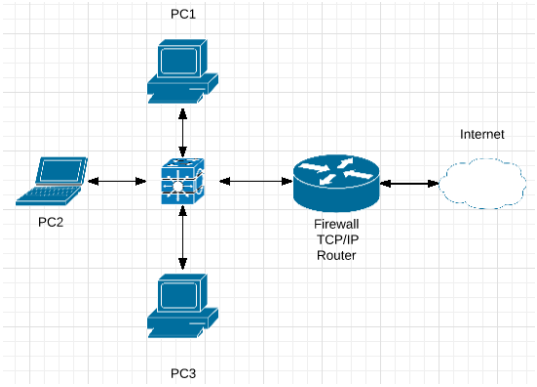


Figura 43: Firewall pasarela

Fuente: Elaboración propia

2.3.4 Capa 4 Aplicación

Capa de Sockets Seguros (*SSL*): Es un protocolo de seguridad ampliamente usado en servicios web, es capaz de brindar confidencialidad, autenticación e integridad, permite varios mecanismos de cifrado, este protocolo se compone de dos capas:

- ***SSL Record Protocol:*** Se encuentra sobre un protocolo de transporte confiable, utilizado para encapsular otros protocolos de nivel superior.
- ***SSL Handshake Protocol***: Permitido para que el cliente y el servidor se autentiquen de manera mutua, negociar un algoritmo de cifrado y compartir llaves de acceso.

En la siguiente figura se observa el nivel en el que se encuentra *SSL*.

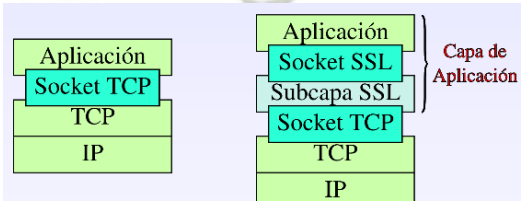


Figura 44: Nivel *SSL*

Fuente: *HTTP*://highsec.es/2014/06/seguridad-en-redes-conexiones-tcp-seguras-*SSL*-secure-sockets-layer/

Las conexiones que se realizan bajo este protocolo tienen las propiedades de ser privadas, seguras y confiables, los objetivos de *SSL* en orden de prioridad son:

- Seguridad
- Inter operatividad
- Flexibilidad
- Eficiencia

A continuación, se aprecia el funcionamiento de *SSL*.

Figura 45: Funcionamiento *SSL*

Fuente: *HTTPS*://ecolohosting.com/que-es-*SSL*/

Secure *Shell* (SSH): Es un protocolo que permite una conexión cifrada entre dos nodos a diferencia de FTP o TELNET los cuales no cuentan con esta propiedad, se utiliza una clave pública para autenticar al nodo y al usuario, usualmente se

utiliza para conexiones remotas y transferencia de ficheros de manera segura, este protocolo tiene una arquitectura en capas tales como son:

- La capa de transporte: Aquí se realiza el intercambio de las claves, así como la autentificación del servidor y negociar para establecer los métodos de compresión, cifrado y verificación de integridad, en esta capa también se organiza el nuevo intercambio de claves según determinadas características como puede ser cantidad de transferencia de tráfico (normalmente 1GB) o según el tiempo de conexión (normalmente después de una hora).
- La capa de autenticación: Algunos de los métodos que proporciona esta capa son:
  - Autenticación por contraseña
  - Autenticación por publickey (DSA o RSA)
  - Autenticación GSSAPI (permite mecanismos externos por ejemplo Kerberos)
- La capa de conexión: Aquí se define el canal, peticiones para los servicios proporcionados por SSH, varios canales pueden darse con una sola sesión

En la siguiente figura se aprecia la representación de una conexión a un servidor web mediante un túnel SSH a través de internet.

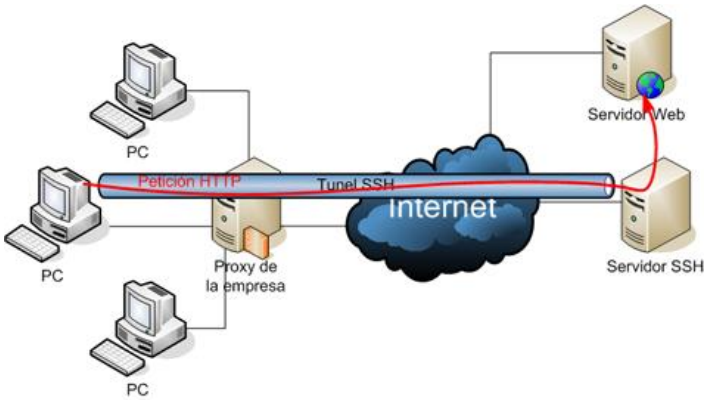


Figura 46: Conexión por medio de SSH

Fuente: *HTTP*://culturacion.com/que-es-ssh/

Protocolo de transferencia de archivos (FTP): Este protocolo tiene el objetivo de permitir el intercambio de archivos entre dos nodos a través de una red TCP/IP, está pensado para ofrecer una alta eficiencia en velocidad, como se observa en la siguiente figura la conexión es bastante simple ya que se trata de una conexión de cliente/servidor para el intercambio de archivos.

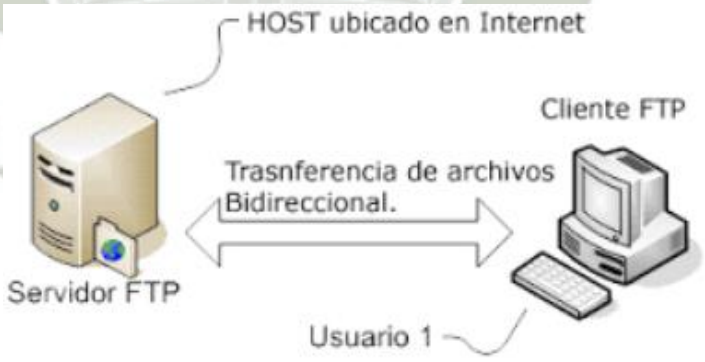


Figura 47: Conexión FTP

Fuente: *HTTPS*://elgatoinquieto.net/2013/10/21/como-montar-un-servidor-ftp-en-gnulinux/

Simple Mail Transfer Protocol (SMTP): Es un protocolo empleado para intercambiar correos electrónicos entre dos nodos, usualmente utilizado en conjunto con POP3, este protocolo es el que recibe el documento enviado por SMTP, también existe un protocolo de acceso a mensajes de internet (IMAP), es alternativo a POP3 pero puede administrar varios accesos en simultaneo, gestionar distintas entradas y mayores reglas para ordenar correos, en la siguiente figura se observa el flujo de un correo enviado por SMTP.

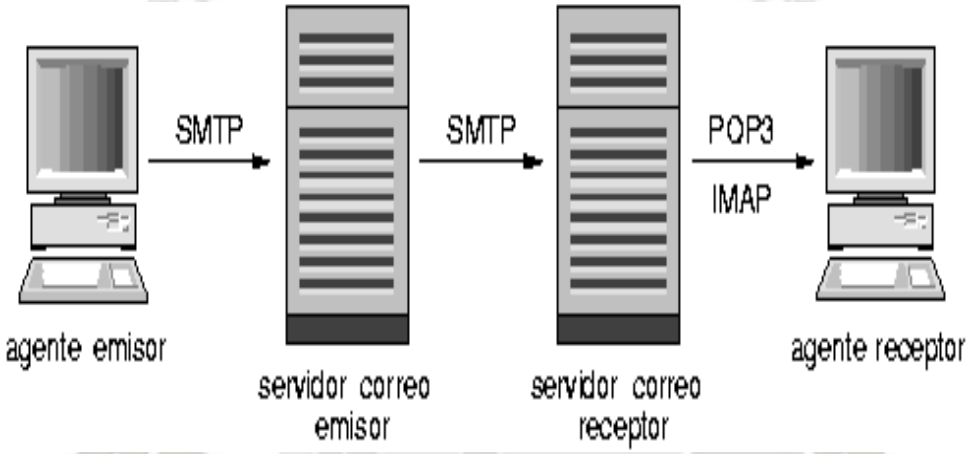


Figura 48: Conexion SMTP

Fuente: *HTTP*://infoismailyainhoa.blogspot.pe/

Dynamic Host Configuración Protocol (DHCP): El protocolo de configuración de host dinámico es el encargado de que un nodo que se encuentre en una red pueda obtener su configuración de red incluyendo direcciones IP, tiene como objetivo simplificar la administración de red, este protocolo soporta los siguientes mecanismos para localizar direcciones IP:

- Automática: Se asigna una dirección IP permanente a un nodo.

- Dinámica: Se asigna una dirección IP por un periodo de tiempo a un nodo, esto permite reutilizar la dirección, esta dirección se llama una lease.
- Manual: El administrador asigna una dirección a un nodo.

En la siguiente figura se observa el formato de un mensaje DHCP.

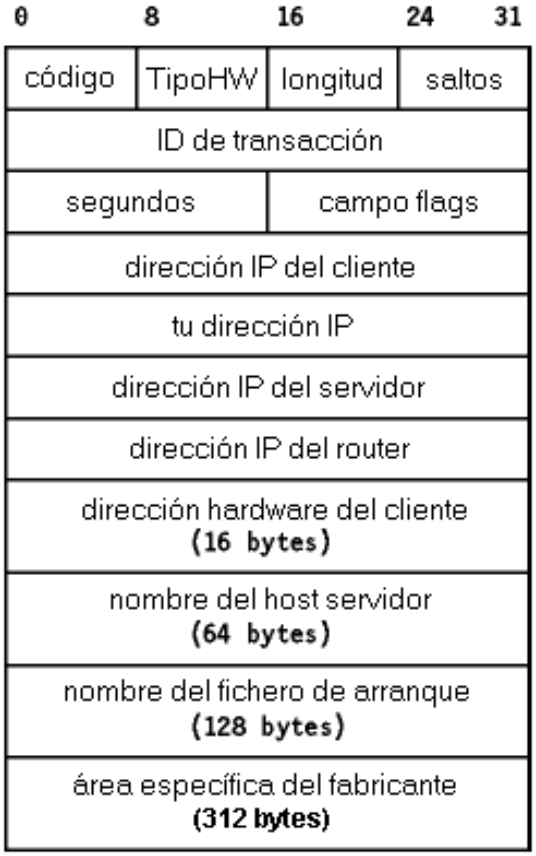


Figura 49: Formato DHCP

Fuente: HTTP://personales.upv.es/rmartin/TcpIp/cap04s10.html

Dynamic Host Configuración Protocol Versión 6 (DHCPv6): En IPv6 las direcciones disponibles aumentan a 128 bits, por lo que se busca una autoconfiguración para reducir el esfuerzo de instalación de nodos, este protocolo funciona sobre UDP y se debe determinar que procesos emplearán los mensajes

RA de ICMPv6, también permite solicitar múltiples direcciones IPv6 a diferencia de IPv4, en la siguiente figura se observa el flujo de comunicación para solicitar una dirección IPv6.

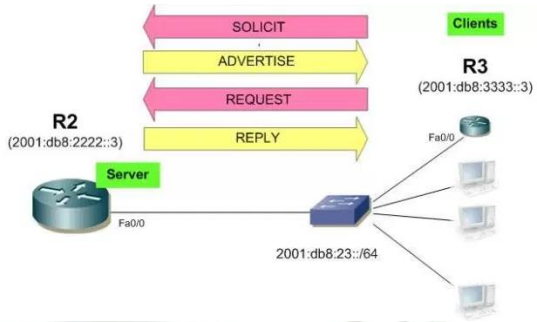


Figura 50: Flujo de comunicación DHCPV6

Fuente: *HTTPS*://cciethebeginning.wordpress.com/tag/dhcpv6/

Domain Name System (DNS): Es el encargado de traducir los nombres de dominio a direcciones IP para poder realizar una conexión, el servidor DNS suele utilizar una base de datos distribuida que contiene información sobre los nombres de dominio en internet, en la siguiente figura se observa solicitud de un usuario a un servidor DNS, el mismo que realiza consulta a distintos servidores para obtener la dirección IP del nombre de dominio dado.

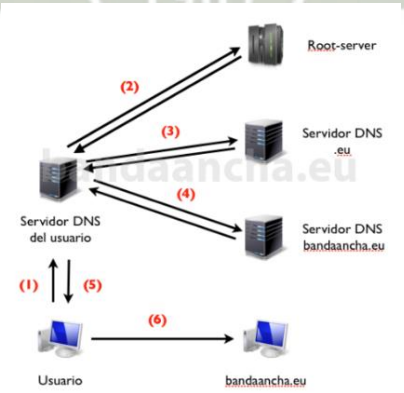


Figura 51: Solicitud a DNS

Fuente: HTTPS://bandaancha.eu/articulos/google-opendns-presentan-mejora-8027

Protocolo Simple de Manejo de Red(*SNMP*): El protocolo simple de administración de red está encargado de facilitar el intercambio de información de administración entre nodos, normalmente este protocolo es soportado por servidores, *switch*es, routers entre otros, básicamente este protocolo permite a los encargados supervisar el correcto funcionamiento de una red, así como detectar y resolver problemas, también ayuda a planear la escalabilidad de la red. Una red que se gestiona a través de *SNMP* tiene tres componentes básicos como lo son:

- ***NMS***: Es un sistema administrador de red el cual ejecuta aplicaciones que permiten controlar y supervisar los distintos nodos, en una red administrada pueden existir uno o más NMS's
- **Dispositivo administrado**: Se refiere a un nodo que se encuentra en una red administrada y que tiene un agente *SNMP*.
- **Agente**: Se refiere a un módulo de software para administración de red, este posee información de administración como paquetes IP recibidos, memoria libre, distintas rutas, etc.

Protocolo de Transferencia de Hipertexto (*HTTP*): El protocolo de transferencia de Hipertexto es un protocolo de cliente/servidor, está encargado del intercambio de información entre el nodo cliente y el nodo servidor, como se aprecia en la siguiente figura está basado en operaciones sencillas de solicitud-respuesta.

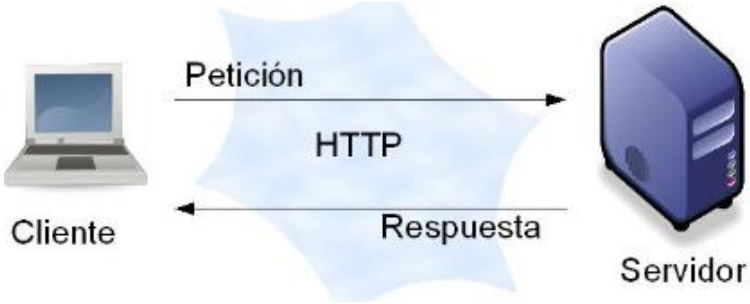


Figura 52: Operación HTTP

Fuente:
HTTP://roble.pntic.mec.es/jprp0006/tecnologia/bachillerato_tic/unidad01_navegadores/navegadores3.htm

Protocolo de Transferencia de Hipertexto Seguro (*HTTPS*): El protocolo seguro de transferencia de hipertexto está basado en el protocolo *HTTP*, al igual que este está diseñado para realizar la transferencia de archivos pero de manera segura utilizando un cifrado que está basado en *SSL*/TLS con los cuales se crea un canal cifrado, de esta manera se consigue que información sensible como credenciales de seguridad estén cifradas y se encuentren ilegibles para un atacante que podría estar interviniendo el tráfico, en la siguiente figura se observa el flujo de conexión a un sitio web que cuente con certificado de seguridad, en este caso una conexión a Facebook.

Figura 53: Flujo de conexión a una web

Fuente:
HTTPS://iw122grupo2.wikispaces.com/DESVENTAJAS++DE+LOS+PROTOCOLOS+DE+SEGURIDAD

Protocolo Trivial de Transferencia de Archivos (TFTP): El protocolo trivial de transferencia de archivos es un protocolo parecido a FTP, es un protocolo de transferencia muy simple, normalmente se utiliza para intercambio de archivos pequeños, no existen mecanismos de cifrado o de autenticación, utiliza UDP como protocolo de transporte, este protocolo no puede listar contenidos de los directorios.

Protocolo Ligero de Acceso a Directorios (LDAP): El protocolo ligero de acceso a directorios permite la administración de directorios, es decir acceder a las bases de información de usuarios en una red utilizando TCP/IP, este protocolo presenta los datos bajo una estructura jerárquica de árbol de información de directorio (DIT), en la siguiente figura se observa el flujo de conexión a un servidor LDAP.

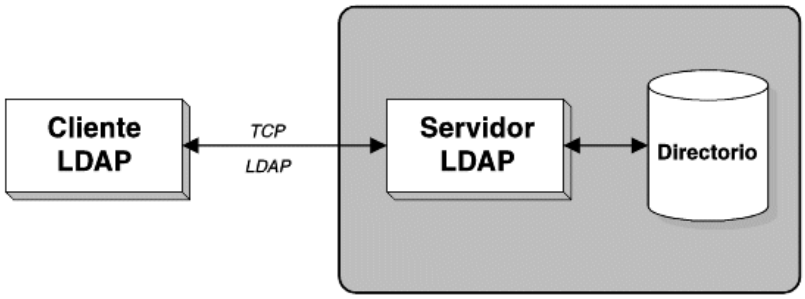


Figura 54: Flujo de conexión LDAP

Fuente: HTTP://publib.boulder.ibm.com/tividd/td/ITAME/GC23-4684-00/es_ES/HTML/adminmst11.htm

CAPÍTULO III

ANÁLISIS DE PROTOCOLOS DE SEGURIDAD Y EXPLOTACIÓN DE VULNERABILIDADES MODERNAS

En el siguiente capítulo se analizará detalladamente los protocolos de seguridad que intervienen en cada una de las capas del modelo TCP/IP, esto con el fin de revisar la seguridad en cada capa, así como los distintos tipos de ataques y explotación de vulnerabilidades existentes.

3.1 Capa de Acceso al Medio

El protocolo de resolución de direcciones ARP es el protocolo de comunicación más popular en las redes LAN, este protocolo tiene algunas limitaciones como la falta de estados y autenticación. Por lo tanto, una persona mal intencionada podría aprovechar estas vulnerabilidades para obtener acceso no autorizado a datos confidenciales y poner en riesgo los pilares de seguridad, como se aprecia en (Raviyarupal & Kumar, 2016) algunos de los ataques utilizando este protocolo pueden ser:

MAC Spoofing: Esta técnica trata de un enmascaramiento o cambio de dirección física, esto puede realizarse con fines de suplantación de identidad, aunque la dirección física viene en cada nodo configurada, existe software capaz de cambiar

dichas direcciones, esto puede desatar muchas brechas de seguridad si no se tiene un control sobre la gestión de direcciones físicas.

Man-in-the-middle (*MITM*): Los ataques de hombre al medio tratan de vincular mediante paquetes ARP su dirección física con una dirección lógica autentica, es decir utilizar el arp Spoofing para de esta manera interceptar el tráfico logrando recibir o modificar el tráfico generado en una red entre dos nodos, en la siguiente figura se observa un ataque de hombre al medio

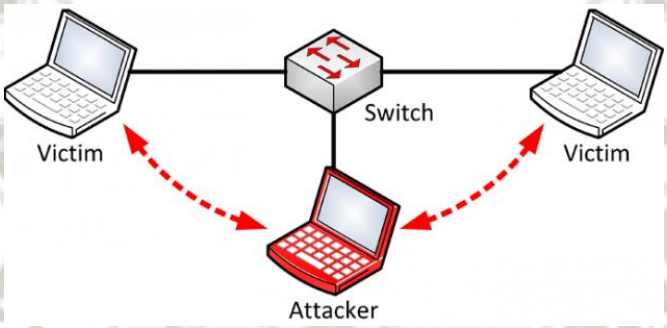


Figura 55: Ataque de hombre al medio

Fuente: *HTTPS*://www.redeszone.net/2016/01/13/bettercap-1-2-es-la-nueva-version-de-este-framework-para-ataques-man-in-the-middle/

Session hijacking: Los ataques de secuestro de sesión son realizados para robar identificadores de sesión, logrando de esta manera obtener acceso a un sistema privado y obtener los privilegios de la sesión capturada, por ejemplo, si alguien logra interceptar una cookie de sesión e inyectar la misma obtendrá acceso a la sesión que se da en ese momento, este concepto se ilustra en la siguiente figura.

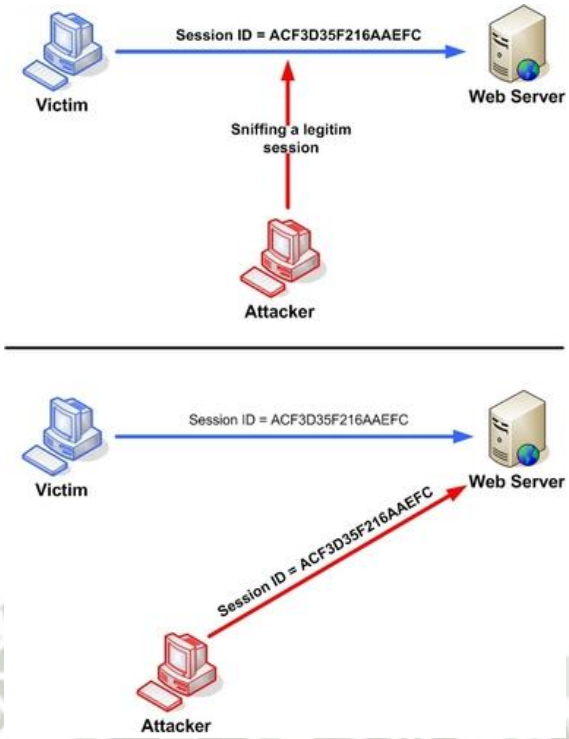


Figura 56: Ataque Session hijacking

Fuente: *HTTPS*://www.*OWASP*.org/index.php/Session_hijacking_attack

Denegación de servicios (DOS): Los ataques de denegación de servicio son muy comunes, en este caso uno de los ataques es el ARP-*FLOOD*, como se observa en la siguiente figura este consta de mandar paquetes ARP en donde se envían direcciones MAC de manera aleatoria a direcciones IPs de nuestra red.

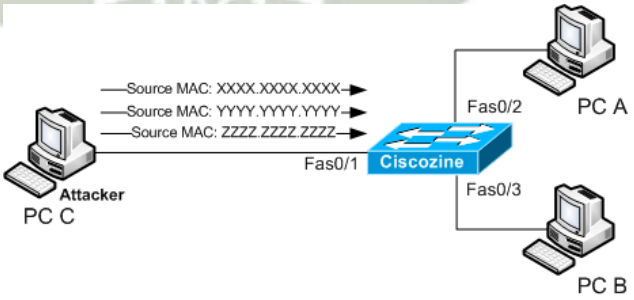


Figura 57: Ataque DOS

Fuente: *HTTP*://www.ciscozine.com/protecting-against-mac-*flood*ing-attack/

De esta manera el *switch* se verá afectado pudiendo realizar dos tipos de acciones, como son:

- Enviar todos los paquetes a todos los nodos conectados, en la siguiente figura se observa una ilustración de la redirección del tráfico a todos los nodos.

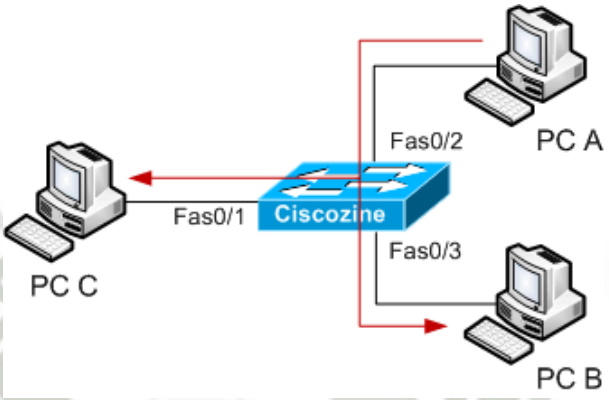


Figura 58: Redirección de tráfico a todos los nodos

Fuente: *HTTP*://www.ciscozine.com/protecting-against-mac-*flood*ing-attack/

- Establecer un estado de fuera de servicio.

Estas dos acciones que puede tomar un *switch* son perjudiciales para una red, por un lado, redirigir todo el tráfico a todos los nodos pone en riesgo la confidencialidad de la información, en este caso el *switch* perdería su funcionalidad de inteligencia comportándose como un *hub* o concentrador, en la siguiente opción que puede tomar es dejar de funcionar con lo cual se estaría logrando una denegación de servicio y no existiría la comunicación entre nodos. Los ataques de denegación de servicio son muy famosos por su versatilidad, por ejemplo, si se envía un paquete ARP diciendo que la dirección MAC del *getway* fue cambiada por una no existente, ninguno de los nodos de la sub-red podrá

comunicarse con el exterior, lo cual también daría como resultado una denegación de servicio.

Como se explica en la investigación presentada en (Tian, Butler, Choi, McDaniel, & Krishnaswamy, 2017) en donde se explica que *Neighbor Discovery Protocol (NDP)* es la base de todas las comunicaciones de red IPv6, pero este protocolo sufre las mismas vulnerabilidades que ARP, uno de los ataques modernos que se da en redes IPv6 es el de *neighbor Spoofing*, el cual se envía un paquete de *neighbor advertisement* configurando una dirección a la víctima logrado así un ataque de hombre al medio pero esta vez dado por direcciones en IPv6, en la siguiente figura se aprecia el flujo que seguiría una conexión después de un *neighbor Spoofing*.

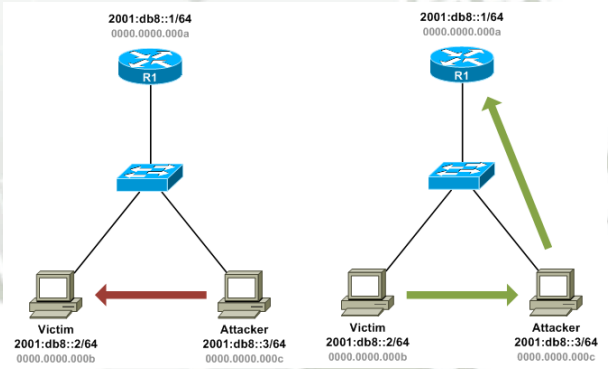


Figura 59: Flujo de conexión en *neighbor Spoofing*

Fuente: *HTTP*://packetlife.net/blog/2009/feb/2/ipv6-neighbor-spoofing/

3.2 Capa de Red

IPSOOFING: La suplantación de direcciones lógicas lo cual permite envió de paquetes con una dirección falsa, esto debido a que el protocolo IP carece de autenticación, la suplantación de direcciones lógicas es muy usada para la generación de ataques de denegación de servicio (DOS).

Ataques de fragmentación: Los ataques *teardrop* constan de enviar paquetes inválidos, esto confunde a un nodo y puede causar la ralentización de conexión. Los fragmentos sobrepuestos son mayormente usados para evadir los filtros de paquetes que suelen inspeccionar solo el primer paquete de la transmisión.

Ataques ICMP/ICMPv6: Los ataques que usan el protocolo ICMP como ataques SMURF son ataques de denegación de servicio, en este caso en concreto se envía gran cantidad de tráfico ICMP a una dirección *broadcast* para inundar un objetivo, suele realizarse desde una dirección IP suplantada, en la siguiente figura se aprecia un ataque *smurf*.

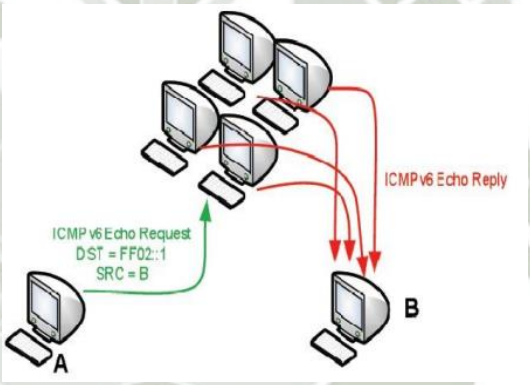


Figura 60: Ataque Smurf

Fuente: *HTTPS*://www.internet2.edu/presentations/tip2013/20130116-Navaneethan-ipv6.pdf

Otros ataques como el ping de la muerte también utilizan este protocolo, este consiste en enviar muchos paquetes ICMP bastante pesados mayor que 65536 bytes, teniendo como objetivo sobrecargar un sistema, esta técnica está actualmente obsoleta pero a la actualidad se utiliza una evolución de la misma llama Ping *Flood* que tiene los mismos principios de inundación por ICMP, con este principio nacen tecinas como, *targeted local disclosed ping flood* para este

tipo de ataques se cuenta con una seguridad perimetral mínima o nula además de conocer la dirección del nodo a objetivo ,*router disclosed ping flood* en el mismo caso que el anterior salvo que aquí el nodo objetivo es un router lo cual podría dejar sin servicio a una red o *blind ping flood* en donde no se conoce la dirección del objetivo por lo cual primero se tendrá que realizar un descubrimiento de la red. ICMP Sweep, este ataque es conocido como barrido ICMP, trata de enviar solicitudes de echo de manera continua lo que forzara al nodo victima responder continuamente manteniéndolo ocupado provocando una inundación lo que puede reducir la eficiencia de una red. *Inverse Mapping* es una técnica de mapeo inverso mediante el cual se puede obtener el mapa de nodos internos, aunque algunos de estos se encuentren bajo la protección de un firewall, esto ocurre porque un atacante envía un ICMP reply a un rango de direcciones IP que podrían estar detrás de un firewall, según las respuestas de ICMP host inaccesible para cada nodo que no se pueda alcanzar se puede dar de esta manera información al atacante de las direcciones que están detrás de un firewall. OS *fingerprinting*, conocer el perfil detallado de un sistema operativo puede proporcionar una ventaja a una persona malintencionada, cuando se realiza un escaneo de puertos también se puede utilizar paquetes ICMP para saber qué sistema operativo se está utilizando debido a que los fabricantes de cada sistema operativo tienen procedimientos distintos de comunicación, aquí el atacante realiza el envío de un paquete UDP con el bit DF configurado a un nodo cuyo puerto UDP está cerrado, la respuesta ICMP será de "puerto de destino inalcanzable", este tipo de respuesta ICMP son ligeramente distintos, los sistemas operativos se pueden determinar al examinar varios bits en el paquete que nos devuelven. Como se explicó una de las

funciones más importantes de ICMP es facilitar la redirección de enrutamiento en el caso que falle un router, un atacante podría redireccionar el enrutamiento a un nodo envenenado y de esta manera acceder a la información de los paquetes, esto conduce a un ataque de *MITM*. Estos tipos de ataques pueden realizarse tanto con *ICMPv4* como con *ICMPv6*, aunque este último fue diseñado con más funcionalidades también cuenta con algunos fallos de seguridad, existen algunos ataques que solo son posibles en la versión *ICMPv6* como se explica en la investigación (Arjuman & Manickam, 2015), ataques como *MITM with spoofed* ICMPv6 *Neighbor Advertisement*, a diferencia de redes IPv4 en donde se utilizaba ARP para realizar ataques de hombre al medio en IPv6 ARP es reemplazado por ICMPv6 *neighbor Discovery*, por lo cual este ataque solo funcionara en redes IPv6, en la siguiente figura se observa un ataque de *MITM* usando ICMPv6.

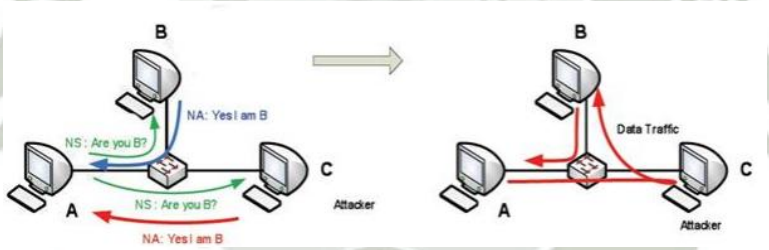


Figura 61: *MITM* por ICMPV6

Fuente: *HTTPS*://www.internet2.edu/presentations/tip2013/20130116-Navaneethan-ipv6.pdf

En la siguiente figura se puede apreciar la misma idea de *MITM* aplicada con un paquete de *router advertisement*

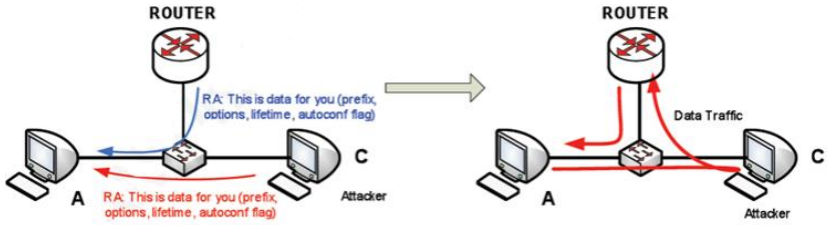


Figura 62: *MITM* con router advertisement

Fuente: *HTTPS*://www.internet2.edu/presentations/tip2013/20130116-Navaneethan-ipv6.pdf

Otro ataque que solo puede ser aplicado en redes IPv6 es utilizando el protocolo *Duplicate Address Detection* (DAD), el cual es utilizado para detectar direcciones duplicadas, DAD utiliza un *neighbor solicitation* enviado a todos los nodos en multidifusión, una atacante puede responder a todos los intentos DAD realizados por un nodo nuevo en la red, un atacante puede reclamar una dirección de dos formas, respondiendo con un NS lo cual simula que también está haciendo DAD o puede responder con un NA simulando que ya está utilizando esta dirección, este tipo de ataques puede causar una denegación de servicio, en la siguiente figura se observa un ataque utilizando DAD.

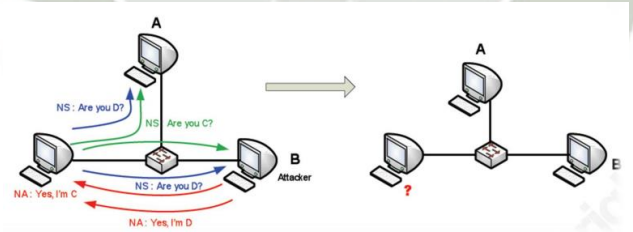


Figura 63: Ataque utilizando DAD

Fuente: Fuente: *HTTPS*://www.internet2.edu/presentations/tip2013/20130116-Navaneethan-ipv6.pdf

Otro tipo de ataque de denegación de servicio empleando el protocolo ND es aprovecharse de *Neighbor Unreachability Detection* (NUD), en este caso un nodo malintencionado puede enviar NAs fabricados específicamente como

respuesta a mensajes NS de NUD, si los NA no son protegidos de alguna forma el atacante podría llevar este ataque por periodos indeterminados. El encaminador de último salto malicioso es otro tipo de ataque en el cual un nodo de la misma subred puede suplantar un encaminador IPv6, esto se realiza enviando un RA como respuesta a un RS, configurando de esta manera la dirección del nodo malicioso.

3.3 Capa de Transporte

TCP Syn Flood este es un tipo de ataque de DOS, está directamente asociado a la capa de transporte debido a que consiste en el envío masivo de establecimiento de conexión (*SYN*), la victima tendrá que utilizar una determinada cantidad de memoria para almacenar las nuevas conexiones en curso, TCP requiere establecer una conexión en 3 pasos, en este caso la víctima se quedara esperando el último paso por lo que la conexión se quedara semi abierta lo que causara que colapse la víctima, normalmente este ataque es realizado con direcciones falsas es decir mediante *ipspoofing* para evitar ser detectados. Ataques como *Connection flood* consiste también en una denegación de servicio, los servicios de TCP son orientados a conexión por lo cual tienen un límite de conexiones al mismo tiempo, luego de este límite las conexiones serán rechazadas, una variante de este ataque es dejar la conexión en TIME_WAIT, trata de no finalizar una conexión de manera que esta siga consumiendo recursos. Algunos ataques permiten bloquear un sistema por medio del envío de paquetes *SYN* con la dirección IP origen igual a la dirección IP destino, una variación de este ataque determina que los puertos origen y destino son los mismos, existe una variante la cual en vez de enviar un solo paquete TCP se envían grupos al mismo

tiempo que se realiza un escaneo de puertos. Los secuestros de sesiones o *HIJAKING* también se pueden dar en esta capa, para tomar el control de una conexión es necesario saber la información asociada a la conexión, deben conocerse los números de secuencia actuales y el número de bytes transmitidos para luego apoderarse de una conexión, una variación es el *blind-hijacking* y su diferencia está basada en que aquí se trata de adivinar las respuestas de los sistemas que están en la comunicación, aquí es bastante utilizado los *sniffers* como wireshark. En esta capa también funciona el protocolo RIP, existen ataques dirigidos a protocolos de encaminamiento los cuales consisten en inyectar paquetes de actualización de rutas, de esta manera se puede manejar los caminos o rutas que sigue una red, el *Spoofing* RIP es uno de estos, debido a que este protocolo trabaja en el puerto 520 y se comunica mediante UDP por lo cual puede aceptar conexiones sin previa conexión, la versión 1 de RIP no proporciona un sistema de autenticación pero la versión 2 tiene un método que consiste en el envío de claves en claro de 16 bytes, al escuchar la res y detectar las actualizaciones RIP que se envían se podrá tener la tabla de rutas de ese momento, también se puede solicitar una petición RIP remota si no se encuentra en el mismo segmento de red, luego al intervenir el tráfico se puede inyectar la ruta deseada, como resultado en ese momento el tráfico de red utilizara el nuevo camino que se definió, se activa el IP *forwarding* para redirigir el tráfico a los destinos originales. Algunos ataques antiguos aún son vigentes en la actualidad como TCP *initial sequence numbers* se dan debido a que TCP genera un numero de secuencia inicial (ISN), con el fin de realizar un control de flujo, este ataque utiliza pseudo-random number generators (*PRNGs*) para poder generar *ISNs* con esto se puede llegar a modificar la

información de la conexión con *hijacking* o poder tomar control de futuras conexiones. *Tiny fragment attack*, este ataque tiene lugar en la fragmentación de paquetes TCP sobre IP, cuando un paquete IP supera el tamaño máximo de MTU este paquete se divide en paquetes más pequeños solo el primer paquete incluye la cabecera de TCP y le resto de fragmentos solo tienen la cabecera IP y los datos que se transmiten pero ninguna información sobre TCP, por medio del campo *fragment* offset que se encuentra en la cabecera IP se puede saber si hay más fragmentos y la relación que estos tienen, en los sistemas que filtran los paquetes normalmente se permite los fragmentos de un paquete IP ya que no se dispone la cabecera TCP para tomar una función de filtro como podrían ser puertos origen/destino, aquí se pretende enviar un paquete TCP inicial con información de *SYN*=0, ACK=1, FO="*more packets follow*", esto permitirá que el paquete atraviese un filtro como *stateless* al no tener el *flag SYN* activado, este paquete no resultara peligroso y es suficientemente pequeño para poder sobrescribir algunos campos de la cabecera TCP por el siguiente fragmento, este fragmento cambiara los valores a *SYN*=1, ACK=0, por lo cual establecerá una conexión en la maquina víctima, aunque las medidas de seguridad no permitan el establecimiento de conexiones en este sistema. En esta capa es bastante utilizado los e*scanners* de puertos para poder saber que puertos se encuentran abiertos y que tipo de servicios pueden correr en cada uno de ellos, aunque actualmente las medidas de seguridad pueden evitar algunos tipos de *scanners* investigaciones como (Arzhakov & Silnov, 2017), en donde propone una arquitectura multiprocesos para un scanner de red minimizando mucho el tiempo de detección de puertos y servicios de un nodo específico, otra investigación es la presentada en (Rohrmann, Ercolani, &

Patton, 2017) donde se analiza la inspección de puertos y servicios a través de nodos TOR para poder realizar un escaneo totalmente anónimos, ya que estos *scanners* tienen un tiempo de demora muy alto proponen un escaneo de forma anónima con *scanners* paralelizados como una forma eficiente de recopilar datos de escaneo de manera anónima. Por la naturaleza de TCP de ser un protocolo orientado a conexión es natural que gente mal intencionada quiere realizar el tipo de conexiones inversas, aunque técnicamente es complicado realizar este tipo de conexiones se utiliza mucho la ingeniería social para lograr una conexión inversa, esto queda demostrado en la investigación (Atwell, Blasi, & Hayajneh, 2016), en donde con un banco de pruebas pueden demostrar tomar el control de destinos nodos, aunque la intervención de los anti *malware* son importantes si el usuario acepta la ingeniera social no se podrá proteger con ninguna medida de seguridad. La investigación (Barbhuiya, Gupta, Biswas, & Nandi, 2012), tiene un punto interesante en los ataques de denegación de servicio con destino TCP a baja velocidad, en este tipo de ataques el nodo malicioso congestiona la red por un periodo de tiempo muy breve y luego se mantiene sigiloso, esto se repite después del tiempo mínimo de RTO, esto logra provocar una degradación del servicio y la denegación de servicio a los flujos TCP, estos ataques suelen ser complejos y poco escalables

3.4 Capa de Aplicación

(Makino & Klyuev, 2015) realiza una comparación entre *OWASP ZAP* y *Skipfish vulnerability scanners,* ambas herramientas muy potentes para poder descubrir brechas de seguridad en la capa de aplicación del modelo TCP/IP, como referencia para el análisis de seguridad de esta capa tomaremos el Open Web

Application Security Project (*OWASP*), el proyecto abierto de seguridad de aplicaciones web es una organización dedicada a mejorar la seguridad de aplicaciones web, tendremos en cuenta el *OWASP* Top 10 2017, donde se hace referencia a los 10 riesgos de seguridad más importantes en el la web, esta lista es enumerada del 1 al 10 por orden de importancia, en la siguiente figura se puede observar el top 10 de *OWASP*.

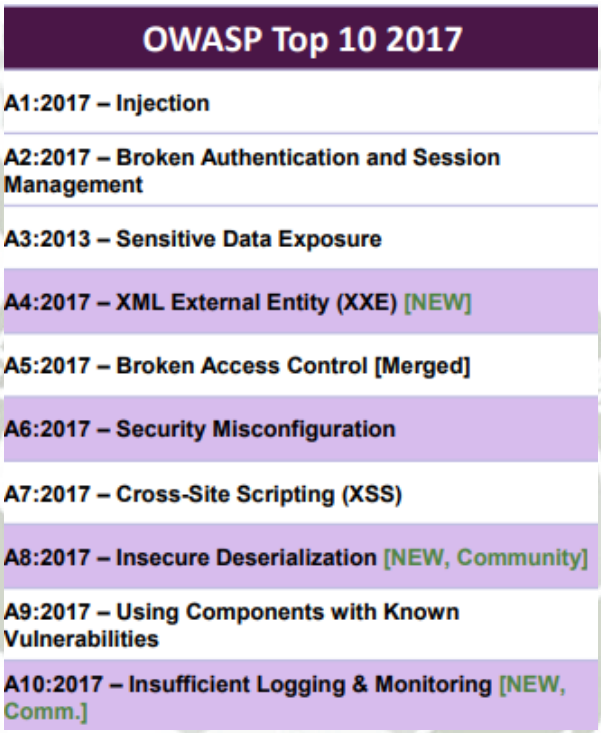


Figura 64: Top ten OWASP

Fuente:
HTTPS://www.OWASP.org/index.php/Category:OWASP_Top_Ten_Project#tab=OWASP_Top_10_for_2017_Release_Candidate_2

Los ataques de inyección son los más importantes estos ocurren cuando un atacante es capaz de enviar datos a un intérprete, estos datos serán enviados de manera maliciosa. Este tipo de inyecciones puede tener consecuencias graves como la perdida de datos o afectar la integridad de los mismos, en ocasiones una inyección puede permitir a un atacante tomar el control del sistema, con inyección

hacemos referencia por ejemplo a *Sql injection*, aunque este ataque es bastante antiguo aún se encuentra entre los de más alto riesgo ya que a la actualidad se siguen dando en gran cantidad, como se demuestra en la investigación (Gudipati, Venna, Subburaj, & Abuzaghleh, 2017), con una amplia cantidad de métodos avanzados para este tipo de ataques, también las inyecciones, también un tipo de inyección como *Command injection*,caben en esta sección, este tipo de ataque tiene como objetivo lograr ejecutar código malicioso en un sistema operativo logrando llegar a este por medio de una aplicación vulnerable, es decir pasar los datos por medio de por ejemplo cabeceras *HTTP* o cookies, estos datos van directo a una *shell* de sistema. Como segundo lugar en *OWASP* se tiene un problema que se dan por una mala gestión de la autenticación o una mala gestión de sesiones, las mismas que pueden por ejemplo no tener un tiempo de caducidad, o el envío de estas por medios sin cifrar exponiendo de esta forma a lo que conocemos como secuestro de sesiones o *hijaking*, se puede llegar a comprometer también las claves o tokens de sesión. En tercer lugar, se tiene la exposición de datos sensibles, existen medios en los que los datos personales no están bien protegidos, esto puede darse tanto en sistemas web como por ejemplo en aplicaciones móviles, existen datos que pueden quedar expuestos en internet que no se necesita más que una búsqueda web para poder encontrarlos, se pueden exponer muchos datos incluyendo contraseñas y archivos de configuración. En cuarto lugar, se tiene el *XML External Entities (XXE)*, este tipo de ataques impacta conta servicios web en particular los SOAP que procesan XML, esto se puede dar si una aplicación acepta XML de manera directa de fuentes no confiables, SOAP antes de la versión 1.2 es especialmente susceptible a XXE si las entidades XML se pasan el marco

SOAP. En la siguiente posición se tiene a *Broken Access Control*, la explotación del control de acceso hace referencia a zonas que no se encuentran correctamente protegidas y la falta de autenticación puede hacer que personas mal intencionadas ingresen, por ejemplo, modificar las url para conseguir accesos o elevación de privilegios dentro de un sistema para actuar como administrador. En la siguiente posición tenemos los fallos de configuración de seguridad estos errores son comunes en los archivos de configuración del servidor, dejar algunos archivos por defecto o mantener un entorno de desarrollo, esto cubre las cuentas por defecto y actualización de software. El siguiente fallo de seguridad es el de *Cross Site Scripting (XSS)*, este fallo de seguridad es muy antiguo pero a la actualidad aun ocurren incidentes de este tipo, este vector de ataque es capaz de ser utilizado para comprometer información confidencial, secuestrar sesiones y poder comprometer un navegador web, normalmente estas pueden ser directas o persistentes donde se inyecta el código html y esto quedara en la web para futuras conexiones, el otro tipo de XSS es indirecta o reflejada, esto puede alterar valores de la web para pasar variables entre páginas web. A continuación, tenemos la deserialización insegura, esta vulnerabilidad es nueva dentro de *OWASP*, se basa en el concepto de que no se puede confiar en que los datos que no son de confianza o malformados sean aceptados, estos datos pueden utilizarse para afectar la disponibilidad de información, afecta también el control de acceso ya que los objetos maliciosos pueden afectar la ejecución de código. Luego tenemos el uso de componentes con vulnerabilidades conocidas, esto sigue siendo un problema en las implementaciones ya que no se revisan bien los módulos, existen muchos *Exploits* en el mercado para vulnerabilidades conocidas y otras hay que realizarlas

según las especificaciones. Por último, en *OWASP* se tiene protecciones frente a ataques insuficientes, esto hace referencia a las medidas de seguridad que se tiene en el área perimetral, firewalls sistemas contra ataques DOS/DDOS, IDS/IPS, la falta de estos en la arquitectura de seguridad podría poner en riesgo el núcleo de negocio.

Adicional a este análisis también existen fallos importantes como los que se encuentran en los certificados de seguridad *SSL*, herramientas como *SSLSTRIP*, son capaces de descifrar el tráfico *HTTPS*, otra investigación como (Zhao, Lei, Yang, & Cui, 2013) en la cual se ve el *SSLSTRIP* en móviles esta investigación va dirigida a dispositivos Android, en la siguiente figura se observa el flujo realizado por *SSLSTRIP* después de realizar un ataque de *MITM*.

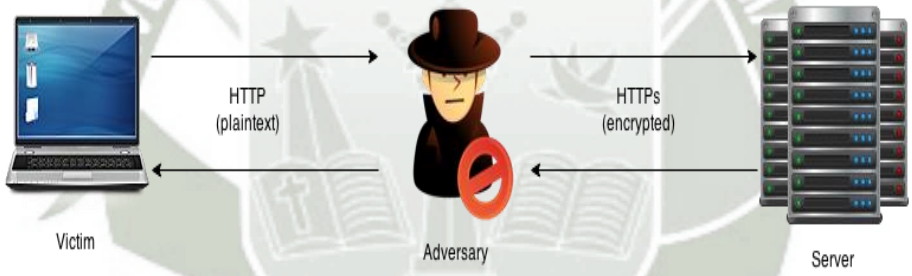


Figura 65: Flujo de ataque empleando *SSLSTRIP*

Fuente: *HTTP*://joakim.uddholm.com/posts/*SSLSTRIP*-and-ettercap.html

También se dieron ataques famosos muy importantes que se deben tener en cuenta como el expuesto en (Kyatam, Alhayajneh, & Hayajneh, 2017), que realiza una investigación ante *Heartbleed*, este ataque era capaz de recolectar los datos de un servidor que se encontraban en memoria aprovechando un fallo de Open*SSL*, la siguiente figura muestra el flujo de *Heartbleed*

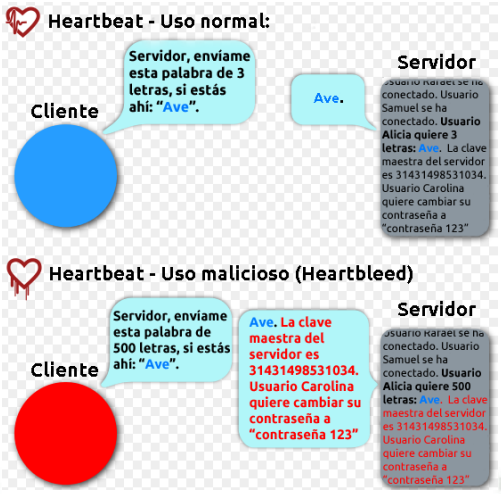


Figura 66: Flujo de ataque por *Heartbleed*

Fuente: http://blog.isecauditors.com/2015/08/la-muerte-oficial-de-ssl-criptocrisis.html

CAPÍTULO IV

CONFIGURACIÓN DE PROTOCOLOS DE SEGURIDAD POR CAPAS DEL MODELO TCP/IP

En el presente capítulo se analizarán las mejores medidas de seguridad que se pueden tomar en cada capa del modelo TCP/IP, esto implica la configuración de distintos protocolos, así como la implementación de medidas de seguridad tanto internas como externas para salvaguardar los pilares de seguridad (CIA).

En el siguiente esquema se muestra una batería de amenazas de seguridad, los mismos que serán detallados en el transcurso de este capítulo.

En la siguiente figura se puede apreciar la arquitectura de seguridad que se propone, la misma que tiene como objetivo minimizar la superficie de exposición a la que se encuentra expuesta una red, esta arquitectura pretende minimizara y mitigara las amenazas de seguridad vistas en el esquema anteriormente planteado.

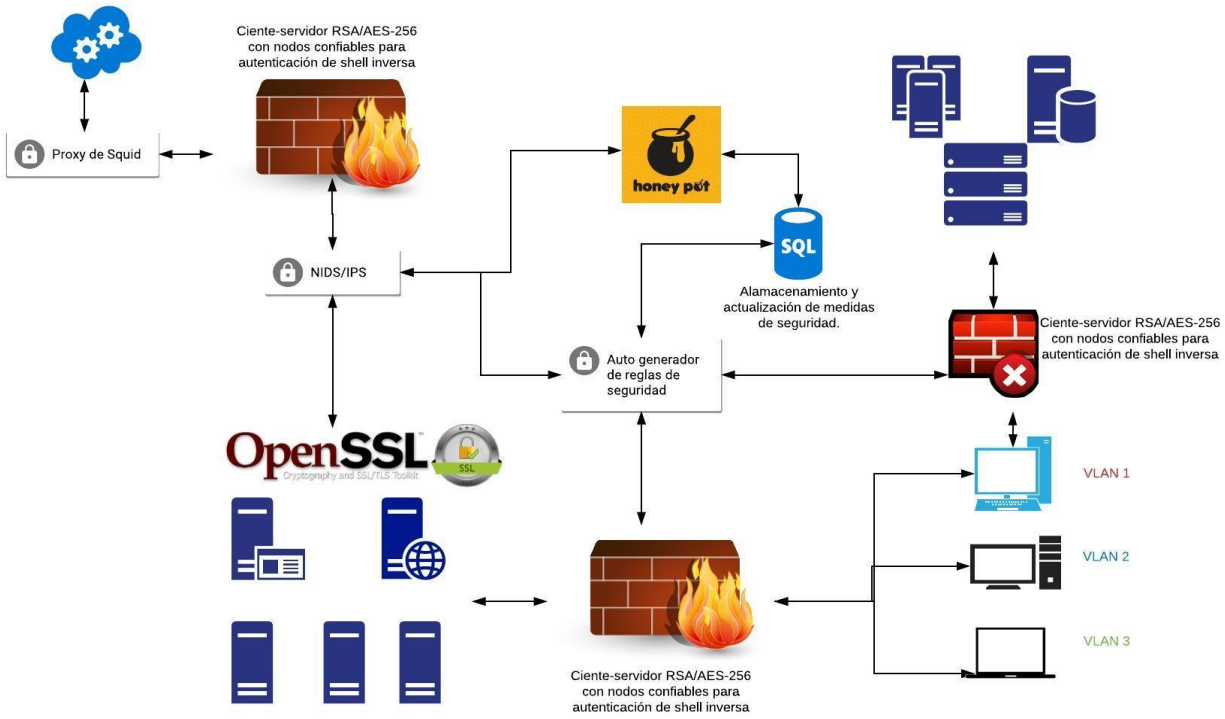


Figura 67: Arquitectura de seguridad

Fuente: Elaboración Propia

4.1 Seguridad en la Capa de Acceso al Medio

Después del análisis de las configuraciones por defecto en esta capa y las vulnerabilidades de los protocolos en conjunto con el análisis de las investigaciones mencionadas podemos tomar ciertas medidas de seguridad para prevenir o detectar anomalías en el protocolo ARP, el cual es uno de los protocolos que causara mayor impacto en la seguridad de verse comprometido. Una de las

medidas más utilizadas para prevenir ataques al protocolo ARP es establecer tablas ARP estáticas d esta manera no podrán ser reconfiguradas en ningún punto de la conexión, uno de los inconvenientes que presenta el colocar tablas ARP de manera estática es que esta se reiniciara cada vez que se reinicie un equipo por lo general se activa un script en el arranque de sistema que configure nuevamente la tabla ARP. Otra medida de seguridad como se puede dar en los dispositivos cisco es el Dynamic ARP Inspección (DAI), en donde se permite validar los paquetes ARP utilizando una tabla de asociaciones de direcciones logias con direccione físicas es decir la IP y la MAC, estas son generadas mediante DHCP, herramientas libres como *shARP* están disponibles y son capaces de detectar y mitigar ataques al protocolo ARP, esta herramienta se encuentra escrita en *bash* y actualmente se puede descargar de su repositorio oficial *HTTPS*://github.com/europa502/shARP, la manera en la que funciona hace que sea necesario instalar algunos programas como *aircrack-ng* y Python, por esto su utilización suele ser complicada si no se tiene los permisos necesarios. Como se expone en la investigación (Lootah, Enck, & Mcdaniel, 2005), se propone un protocolo TARP, el cual propone la implementación de seguridad en ARP mediante la distribución de tickets, estos tickets autentican la asociación entren una dirección MAC y una dirección IP por medio de firmas que son realizadas por un *Local Ticket Agent(LTA)*, cada ticket tiene un tiempo de validez, tras el cual deberá generarse un nuevo ticket. Otro protocolo que cumple la misma función que ARP en redes de datos IPv6 es el protocolo de *Neighbor Discovery Protocol (NDP)*, como se explica en (Zhang & Wang, 2017), para este problema de seguridad existen algunas implementaciones de seguridad como puede ser el Security *Enhaced Neghbor Discovery (SEND)*,

este protocolo resuelve muchos fallos de seguridad de NDP, sin embargo es difícil de establecer, también al analizar la investigación (Xiaorong, Jun, & Shizhun, 2013), se puede ver un mecanismo interesante de autenticación para mejorar la seguridad en NDP, dicha investigación propone un protocolo de administración de claves de capa de aplicación para resolver problemas de multidifusión, aquí se introduce también la opción de IPSEC/AH y control de MAC en NDP para realizar la autenticación del paquete y prevenir mensajes falsificados.

4.2 Seguridad en la Capa de Red

Como lo expresa (Kaushik & Joshi, 2010), los ataques en redes de datos generalmente son controlados por sistemas de seguridad como Firewalls y sistemas de detección de intrusos, no obstante, con el avance de la seguridad a la actualidad existen herramientas disponibles con los mecanismos existentes que recopilan los datos ICMPv4 y los correlacionan con los ataques relevantes, en el caso de IPv6 con sus nuevas mejoras aun presenta algunos vectores de ataque en ICMPv6 al momento de utilizar protocolos de configuración automática de direcciones, y el descubrimiento de vecinos, una propuesta interesante para minimizar el riesgo es la que se expone en la investigación (Chakraborty, Chaki, & Cortesi, 2014), en donde se implementa un IPS, este tiene una configuración para combatir eficientemente ataques DOS, *MITM*, *SPOOFING*, asociado a esta solución podemos integrar la extensión del protocolo de seguridad NDP el cual es Secure Neighbor Discovery (SEND) en donde se utiliza un par de llaves pública/privada, está basado en la utilización de Cryptographically Generated Address (CGA) que consta en la dirección IPv6 con el identificados generado criptográficamente partiendo de una llave pública, el prefijo de red y un

modificador, como se aprecia en la siguiente figura el flujo de generación una dirección IPv6 con CGA.

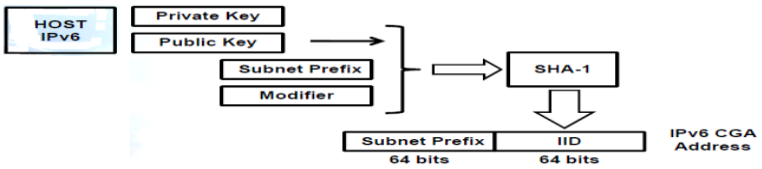


Figura 68: Generación una dirección IPv6 con CGA

Fuente: HTTP://www.6deploy.eu/workshops/20101011_santa_cruz_bolivia/DIA 2-1-Consulintel_IPv6_ES_Seguridad_IPv6.pdf

En la siguiente figura se observa un mensaje NDP que cuenta con la cabecera ICMPv6, en donde se definirán nuevas opciones para poder asegurar NDP, estas opciones son Parámetros de CGA como Modificador, prefijo de subnet, llave pública. Nonce es un número aleatorio con el fin de evitar ataques por re-actuación. Firma contiene los parámetros de CGA y NONCE los cuales están firmados con la llave privada.

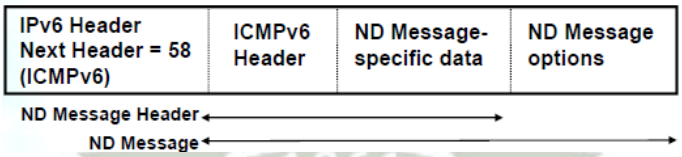


Figura 69: Mensaje NDP con cabecera ICMPV6

Fuente: *HTTP*://www.6deploy.eu/workshops/20101011_santa_cruz_bolivia/DIA2-1-Consulintel_IPv6_ES_Seguridad_IPv6.pdf

En la siguiente figura podemos observar el flujo final de un nodo a otro el cual quiere saber la dirección física mediante IPv6, por lo cual se enviará un *Neighbor Solicitation (NS)*, esta solicitud se realizará usando SEND.

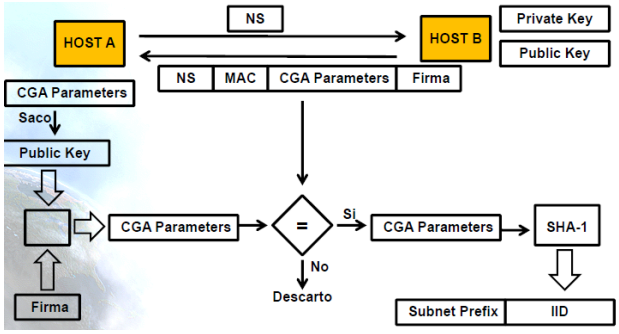


Figura 70: Flujo de descubrimiento por IPV6

Fuente: *HTTP*://www.6deploy.eu/workshops/20101011_santa_cruz_bolivia/DIA2-1-Consulintel_IPv6_ES_Seguridad_IPv6.pdf

En esta capa también se tendría un problema similar con los RA, estos pueden ser cubiertos de manera similar, los RA son firmados por los routers que necesitan un certificado el cual será asociado con el par de llaves, esto se realiza para que los nodos puedan confiar en ellos, el certificado es generado por un CA en el que los nodos deben confiar, se crearán dos nuevos mensajes de ICMPv6 los *Certification Path Solicitation* (CPS), este mensaje será utilizado por el nodo para obtener certificado del router. También se crea Certification Path Advertisement (CPA), esta será la respuesta del router con el certificado, este flujo puede apreciarse en la siguiente figura.

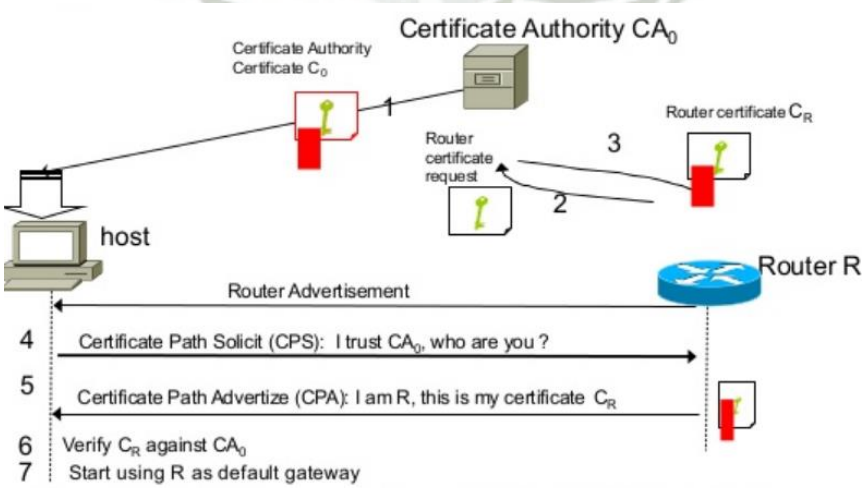


Figura 71: Flujo de certificado de autorización

Fuente: *HTTPS*://www.slideshare.net/getyourbuildon/preparing-for-byod-and-ipv6-with-a-single-security-policy-techadvantage-webinar

Agregar a estas medidas de seguridad investigaciones como (Najjar & Kadhum, 2015) en la cual se desarrolla un conjunto de datos confiable de IPv6 NDP (*Neighbor Discovery Protocol*) capturando los comportamientos normales y anormales de NDP utilizando herramientas confiables específicas. Un conjunto de datos confiable ayuda a entender y distinguir entre comportamiento normal y anomalías en IPv6 NDP. También existen opciones de seguridad basadas en estenografía para IPv6 como lo demuestra (Bobade & Goudar, 2015) en la cual se implementa estenografía para la comunicación encubierta para proporcionar más seguridad que es muy útil en áreas como militares, sectores bancarios. Para una seguridad mejorada se aplica una técnica de cifrado RSA.

Esta capa es importante porque es donde resaltara el uso del protocolo IPSEC, este protocolo opera en un nodo como una pasarela de seguridad o como un dispositivo independiente, la protección de este protocolo se basa en los requerimientos que se tengan definidos en la *Security Policy Database* (SPD), IPSEC puede emplearse tanto en la protección de comunicación entre un par de nodos, así como proteger la comunicación entre pasarelas de seguridad o entre la combinación de una pasarela y un nodo, este protocolo brinda una solución muy buena para la seguridad del protocolo IP en la propia capa de red a diferencia de otros protocolos como TLS o SSH, para proteger una aplicación a través de una red IP se utiliza una asociación de seguridad (SA) cuyo concepto fundamental es una conexión simple proporciona servicios de seguridad al tráfico que transporta,

tanto AH como ESP utilizan SAs, el establecer y mantener las SAs es una de las principales funciones de IKE.

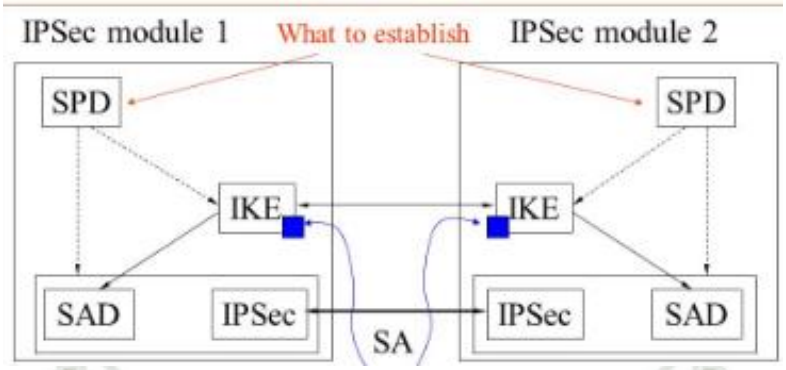


Figura 72: Asociación e seguridad SA

Fuente: *HTTPS*://flylib.com/books/en/4.178.1.77/1/

En donde IKE se encargará de cómo establecer IPSEC SAs mediante los siguientes pasos.

- Algoritmo de cifrado a utilizar
- Algoritmo de hash a utilizar
- D-H group
- Método de autenticación

Dentro de las configuraciones de seguridad de un firewall se recomienda el filtrado de dirección tanto origen como destino, procesar las cabeceras de extensión IPv6, filtrar paquetes tomando en cuenta la información de protocolo de capa superior, la inspección de tráfico encapsulado, según el tipo de distribución del firewall dentro de la arquitectura de red podremos tener:

En la siguiente figura se tiene una arquitectura de Internet/router/firewall/red(es)

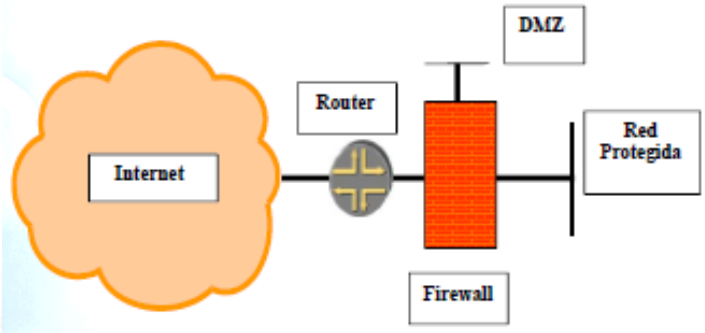


Figura 73: Arquitectura de internet/router/firewall/red

Fuente: *HTTPS*://sergvergara.wordpress.com/2011/03/14/arquitectura-y-diseno-de-seguridad-de-red-perimetral/

- Este tipo de arquitectura debe poder reconocer filtrado de ND/NA
- Soportar mensajes MLD si se requiere
- Soportar RS/RA en el caso que se emplee SLAAC

En la siguiente figura se tiene una arquitectura de Internet/firewall/router/red(es)

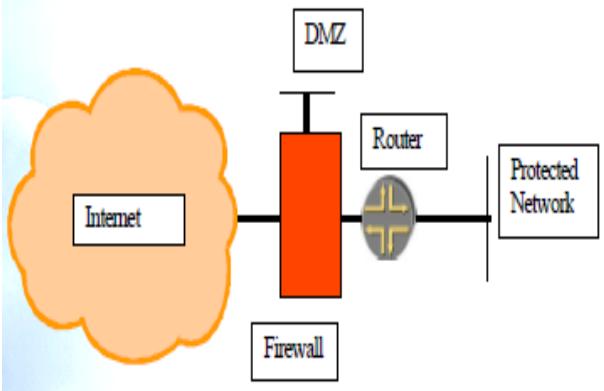


Figura 74: Arquitectura de Internet/firewall/router/red

Fuente: *HTTPS*://sergvergara.wordpress.com/2011/03/14/arquitectura-y-diseno-de-seguridad-de-red-perimetral/

- Soportar ND/NA
- Soportar protocolo de encaminamiento dinámico
- Debe tener una variedad de tipos de interfaces.

En la siguiente figura se tiene una arquitectura de Internet/router-firewall/red(es)

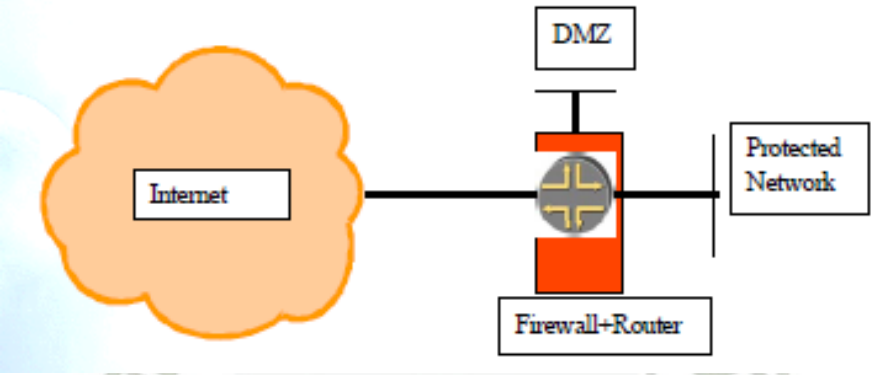


Figura 75: Arquitectura de Internet/router-firewall/red

Fuente: *HTTPS*://sergvergara.wordpress.com/2011/03/14/arquitectura-y-diseno-de-seguridad-de-red-perimetral/

- Este tipo de arquitectura puede tener un dispositivo muy potente el cual se utilizará para encaminamiento y establecer políticas de seguridad, es común en redes SOHO.

Debe tener capacidad de soportar lo que normalmente puede soportar tanto un router como un firewall.

4.3 Seguridad en la Capa de Transporte

En las medidas de seguridad dispuestas en esta capa podríamos poner como primera línea los sistemas de detección de intrusos (IDS), estos suelen tener *sniffer* de red para poder verificar el tráfico que se transmite en la red gracias a esto se pueden detectar anomalías que pueden ser resultado de presencia de ataques o incidentes, el tráfico puede ser comparado contra ataques conocidos, paquetes malformados que pueden indicar cierto tipo de comportamientos sospechosos, normalmente está integrado en conjunto con un firewall el mismo que puede establecer determinado tipo de reglas ante algunos eventos sospechosos, se sugiere la implementación de NIDS que están distribuidos en un segmento de red, este no solo debe contar con mecanismos basado en patrones que comparan paquetes con ataques conocidos sino también en heurística que lanzara alertas cuando ocurran eventos que no son usuales en la red, al tomar acción sobre el tráfico se volverán nodos activos de seguridad por lo que serán sistemas de prevención de intrusos (IPS),como se observa en la siguiente figura es una arquitectura de un IDS/IPS antes de impactar con los filtro de un firewall.

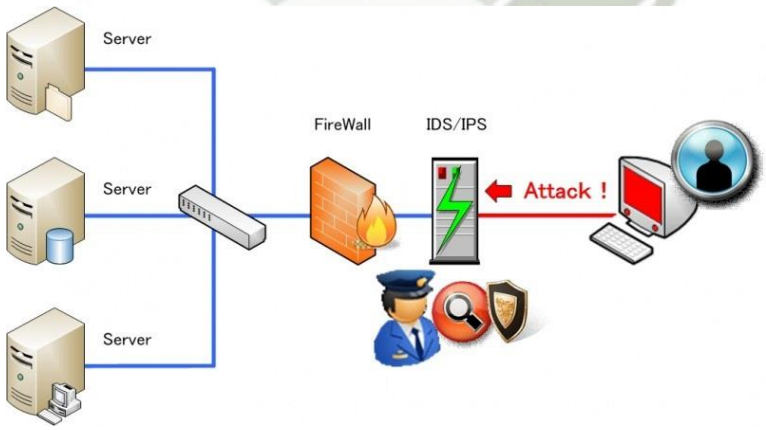


Figura 76: Arquitectura de un IDS/IPS antes de impactar con un firewall.

Fuente: HTTPS://infosecprimer.wordpress.com/2013/07/09/introducing-ids-and-ips/

Estas medidas de seguridad también disminuirán la eficiencia de los ataques modernos de denegación de servicio como el de ataques DOS con destino TCP de baja velocidad, aunque existen algunas medidas de seguridad como estas suelen ser complejas para una red de alta demanda, algunas medidas como la modificación del formato del protocolo TCP o la modificación de implementación de TCP en router intermedios, estas medidas son inutilizables en las redes actuales (Bhosale & Mane, 2015) demuestra que utilizar sólo técnicas de prevención de intrusiones no es suficiente. Para detectar distintas vulnerabilidades es necesaria la detección de intrusiones, se realiza una comparativa de las distintas soluciones de NIDS existentes en el mercado pero estas soluciones no cubren en su totalidad las capas del modelo TCP/IP, por lo que es necesario la interacción entre distintos mecanismos de seguridad, con respecto al análisis puertos abiertos y sus servicios juega un papel importante la configuración de firewall que se cuenta en la arquitectura, esta debería evitar las conexiones sospechosas o el intento de las mismas, en una arquitectura se puede implementar los *Honeypot*s, estos son un mecanismo de seguridad que se despliegan en una red y tienen como objetivo detectar un posible ataque y poder analizarlo para obtener información acerca de este, antes que pueda corromper otros sistemas, se pueden configurar *Honeypot*s de baja interacción los que trabajan emulando servicios para poder recopilar información sobre el ataque, son utilizados en *Honeypot*s de investigación. Los *Honeypot*s de alta interacción son redes trampa las cuales tienen la capacidad de recolectar mucha información para luego poder tomar medidas de seguridad y analizar el modo en el que operan los ciber criminales, los *Honeypot*s pueden ser de gran ayuda contra ataques automatizados, y pueden ofrecer métodos de

detección precisa por su ubicación pueden encontrarse en el área de producción que son fundamentales ya que protegerán este ambiente con una disponibilidad de 24/7, los *Honeypot*s de investigación permiten los ataques e intrusiones en sistemas simulados para poder analizarlos, para la protección de una red los *Honeypot*s pueden estar distribuidos de la siguiente manera:

Antes del firewall, como se aprecia en la siguiente figura se encuentra fuera de la zona protegida por el firewall por lo que puede ser atacado sin tomar mayores medidas y se podría analizar el tráfico para observar los comportamientos sospechosos que llegan de internet, esto evitara analizar los IDS ya que los ataques estarán dirigidos al *Honeypot*, pero este tipo de arquitectura no permitirá analizar el tráfico generado internamente.

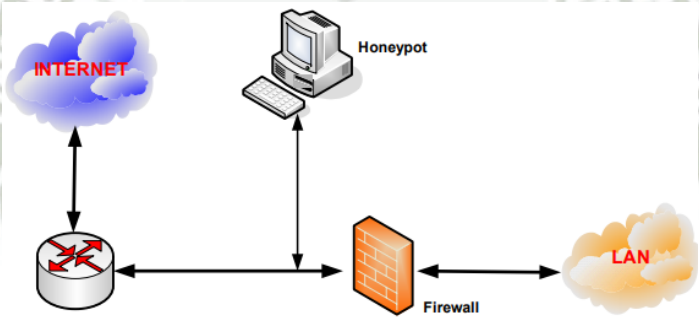


Figura 77: Arquitectura *Honeypot* fuera de firewall

Fuente: *HTTP*://www.cybsec.com/upload/ESPE_*Honeypot*s.pdf

En la siguiente figura se aprecia un arquitectura en donde el *Honeypot* se encuentra detrás del firewall, esto significa que el primer impacto de las conexiones de internet serán contra el firewall lo que significaría que se establecerían ciertas reglas en el firewall para permitir el acceso al *Honeypot* en caso de posibles ataques externos, por otro lado al ingresar al *Honeypot* se puede

poner en riesgo la red de producción debido a que están conectados y ya pasaron la seguridad del firewall, esta arquitectura permite también poder detectar atacantes en la red interna así como poder ver si el firewall se encuentra bien configurado o existen maquina infectadas por algún tipo de *malware*, aunque este *Honeypot* puede ayudar en la identificación de atacantes internos se recomienda poner más firewalls para gestionar la comunicación entre el *Honeypot* y la red de producción.

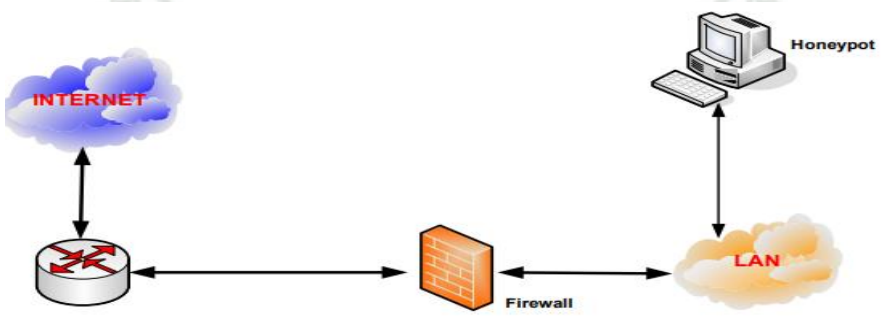


Figura 78: Arquitectura *Honeypot* dentro de la LAN

Fuente: *HTTP*://www.cybsec.com/upload/ESPE_*Honeypot*s.pdf

En la siguiente figura se observa una arquitectura de seguridad donde el *Honeypot* está en la zona desmilitarizada (DMZ), esto permite agrupar al *Honeypot* junto con los servidores de producción y poder controlar el peligro que puede exponer un *Honeypot* ya que este se encuentra detrás de un firewall, debido a que se encuentra en la DMZ se puede detectar tanto ataques externos como internos si el firewall es configurado correctamente, al no estar en contacto directo con la red interna disminuye las alertas de seguridad de los IDS, un atacante de la red interna podría intentar comprometer los servidores públicos pero se encontrarían con un *Honeypot* que podría recabar información para identificar al nodo malicioso.

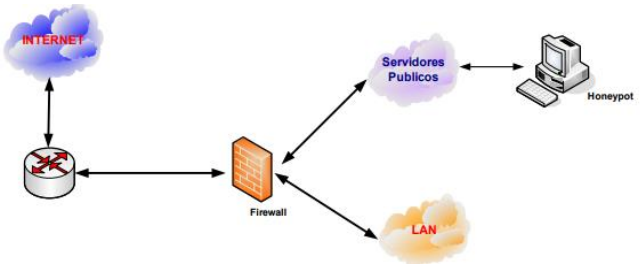


Figura 79: Arquitectura *Honeypot* protegido por un firewall separado de la LAN

Fuente: *HTTP*://www.cybsec.com/upload/ESPE_*Honeypot*s.pdf

En la siguiente figura se observa un *HONEYNET*, es un tipo de *Honeypot* altamente interactivo diseñado para investigar nuevos tipos de ataques y comportamientos, es necesaria una configuración de un entorno real para lograr una alta interacción, esta red está separada de una red de producción, debido a de que el objetivo de este *Honeypot* es poder ser comprometido.

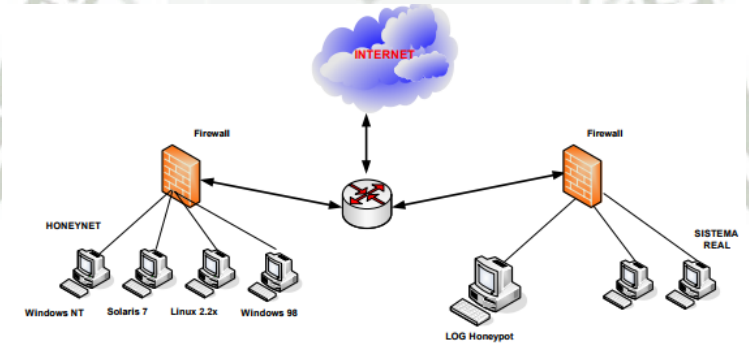


Figura 80: *Honeynet*

Fuente: *HTTP*://www.cybsec.com/upload/ESPE_*Honeypot*s.pdf

En cuanto a las medidas de seguridad de *Honeypot*s en IPv6 se tienen una solución interesante propuesta en la investigación (Schindler, Schnor, & Scheffler, 2015) en la que se presenta una nueva arquitectura *Honeypot* híbrida que se centra en la cobertura de grandes espacios de direcciones IPv6, propone el uso de *Honeypot*s de alta interacción configurados dinámicamente que pueden

cubrir grandes trozos del espacio de direcciones IPv6. Un nuevo mecanismo de proxy se utiliza para transferir transparentemente y reenviar el tráfico de *Honeypot*s de baja a alta interacción bajo demanda para proporcionar la mejor granularidad de servicio posible, en la siguiente figura se observa un esquema de la capa de *Honeypot* de baja interacción.

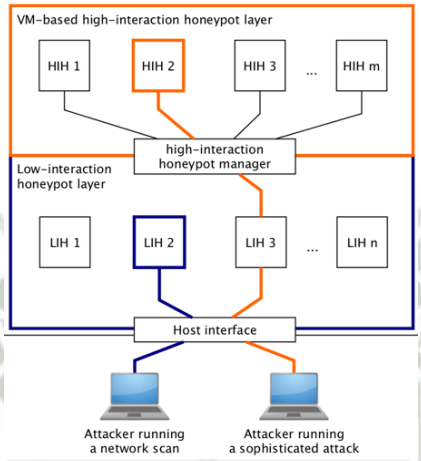


Figura 81: *Honeypot* de baja interacción

Fuente: *HTTP*://www.cs.uni-potsdam.de/bs/research/projectIpv6.html

En la siguiente figura se observa la arquitectura de *Honeypot* de IPv6 y los mecanismos internos de seguridad, así como la virtualización de un nodo para poder guardar la mayor cantidad de información posible.

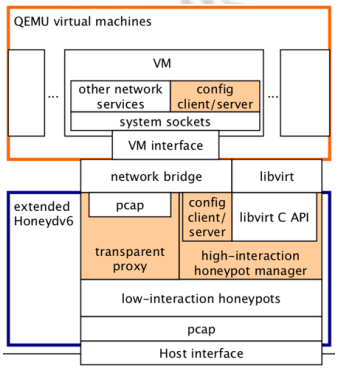


Figura 82: *Honeypot* de IPv6

Fuente: *HTTP*://www.cs.uni-potsdam.de/bs/research/projectIpv6.html

4.4 Seguridad en la Capa de Aplicación

En casos como las inyecciones es recomendable mantener los datos separados de las consultas, se requiere pasar una validación de entrada de datos, este es un tema complicado debido a que muchas aplicaciones requieren algunos caracteres especiales en texto o API de aplicaciones móviles, tomar en cuenta solo la cantidad mínima de privilegios necesarios al conectarse a una fuente de información, desactivar el modo de desarrollo porque estos entornos usualmente dan información detallada acerca de los errores que ocurren, es una buena medida también el usar procedimientos almacenados ya que estos usualmente no son afectados por las inyecciones, pero hay que ser cautelosos ya que se podría activar uno de estos a través por ejemplo del uso de *exec*(), la verificación de parámetros es una medida importante debido a que si ocurre un *command injection*, este podrá ejecutar comando en un sistema operativo pero los privilegios que se tendrán serán los mismo que tiene la aplicación que se vulnero, en el problema sobre el manejo de autenticación y sesiones se debe tomar medidas de seguridad desde las más básicas como establecer contraseñas complejas como primera medida, establecer políticas para cambiar estas de manera periódica, contar con más de un solo factor de autenticación, esta última medida es muy importante ya que mientras más factores de autenticación se implemente la seguridad se elevara exponencialmente, una autenticación segura se basa en tres factores algo que sabemos cómo podría ser una contraseña robusta, algo que somos que podría ser un factor de biometría, y algo que tenemos que podría ser una clave al teléfono móvil, la combinación de por lo menos dos de estas medidas elevaran el nivel de seguridad de manera exponencial, estas medidas también son válidas para mitigar

la exposición de datos sensibles, además de asegurar que estos datos se encuentren cifrados, y cifrar las conexiones por medio de certificados de seguridad *SSL*/TLS y también utilizar *Strict Transport Security HSTS*, deshabilitar de la memoria cache las respuestas que puedan contener datos confidenciales. Contra XXE se tienen que tomas medidas de actualización como actualizar SOAP a la última versión, parchar los últimos procesadores y bibliotecas XML. LA implementación de controles de acceso es primordial para garantizar la seguridad de las aplicaciones, así como registrar los errores de conexión o autenticación, esto implica también tener bien parametrizados los privilegios para cada usuario. Las medidas de seguridad contra las configuraciones de seguridad son las de eliminar características innecesarias ya sean de componentes o algunos documentos, eliminar también las dependencias que no son utilizadas, establecer un proceso para actualizar el sistema, contar también con una sólida arquitectura de aplicaciones. LA prevención del ataque XSS se puede dar mediante la división de los datos del contenido web apartado de información sensible, también se debe aplicar una política de seguridad de contenido (CSP), con esto se puede mitigar y controlar los incidentes de XSS, adicionalmente en la investigación (Zalbina et al., 2017) se muestra el reconocimiento y la detección de ataques XSS mediante la asignación de patrones de expresión regular y un método de pre procesamiento. Los experimentos se realizan en un banco de pruebas con el objetivo de revelar el comportamiento del ataque. Es de suma importancia no aceptar objetos serializados de fuentes consideradas no confiables, también se puede implementar comprobaciones de integridad, así como aislar el código que se deserializar en un entorno con bajos privilegios y monitorizar la actividad que realiza para hacer

frente al problema de la deserialización insegura. Las actualizaciones de software son importantes, como lo es el eliminar las dependencias que no se utilizan y obtener componentes solo de fuentes confiables esto apoyara la protección contra el uso de componentes vulnerables que menciona *OWASP*. Es primordial la seguridad en la web no solo implementando firmas de seguridad como *SSL* sino también *HTTP* Strict Transport Security (*HSTS*), pensando en la intervención de tráfico *HTTPS*, también puede prevenir el secuestro de sesiones, su principal objetivo es evitar que un nodo malintencionado convierta una conexión *HTTPS* en *HTTP* lo cual generaría que el tráfico viaje en texto plano.

En la siguiente figura se aprecia el flujo de conexión de *HSTS*.

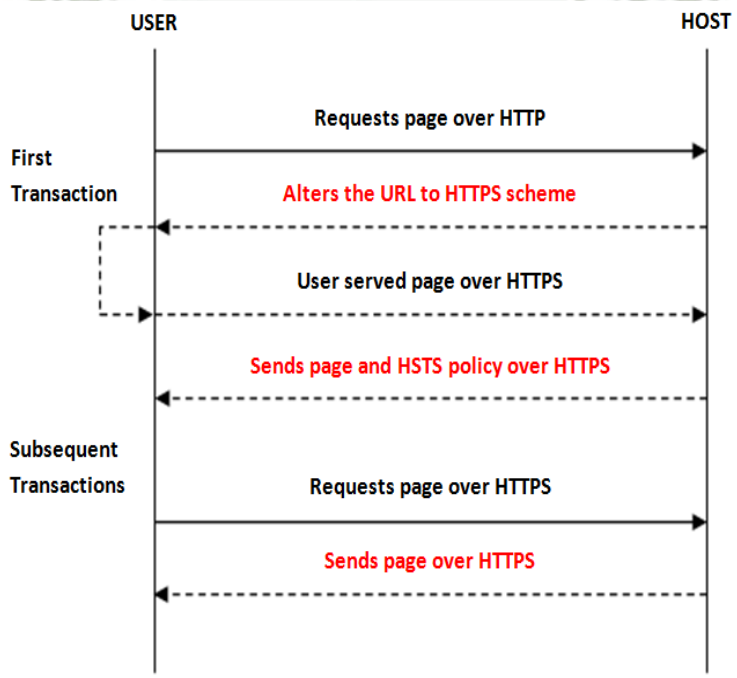


Figura 83: Flujo de conexión *HSTS*

Fuente: *HTTPS*://www.*SSL*2buy.com/wiki/*HTTP*-strict-transport-security-*HSTS*-better-security-for-applications

En la siguiente figura se muestra una conexión a la página web de *twitter*, intentando primero una conexión sin *SSL* que enviaría los datos por texto plano lo cual es rechazado ya que *HSTS* solo permite enviar datos de manera segura.

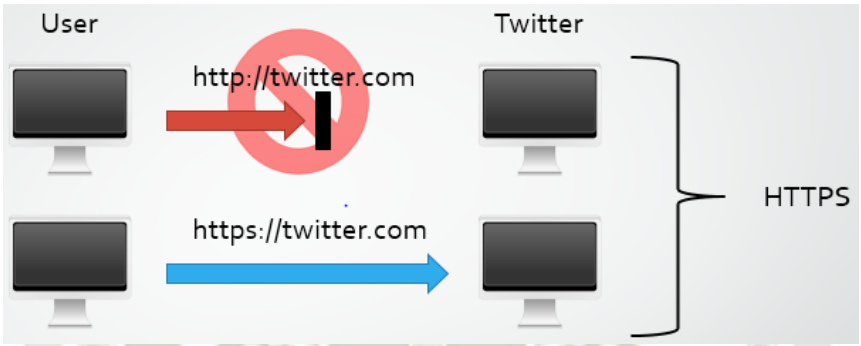


Figura 84: Conexión *HSTS*

Fuente: *HTTP*://unaaldia.hispasec.com/2017/10/google-forzara-*HSTS*-en-los-tld-con-los.html

CAPÍTULO V

VALIDACIÓN DE PRUEBAS DE LA ARQUITECTURA DE SEGURIDAD

El presente capítulo presentara las pruebas realizadas para verificar la eficiencia del despliegue de las medidas de seguridad, para esto se creara una red con un *switch Extreme 440*, en el que se crearán distintas VLANS, se pondrá en producción un servidor vulnerable para realizar las pruebas de seguridad dentro y fuera de la arquitectura propuesta, se establecerá los nodos confiables para conexiones inversas mediante RSA y AES-256, este último generado de una manera específica empleando las direcciones físicas(MAC) de dichos nodos, así como la aplicación del algoritmo SHA-1.

5.1 Estructura de la Red

En la siguiente figura se muestra la configuración de un *switch Extreme*-440, el mismo que cuenta con la configuración de *snmp* para la descripción de la configuración, así como también se observa el módulo de configuración de *VLAN*, se observa del mismo modo que el único puerto "*tagged*", es el puerto 24, esto debido a que es el puerto por el cual se administra el *switch*, pro el mismo motivo este puerto se encuentra dentro de la *vlan* "Tesis_Nagata_Mgmt", la cual es la vlan de administración.

```
SWX440-ToshiroNagataTesis.2 # show configuration
#
# Module devmgr configuration.
#
configure snmp sysName "SWX440-ToshiroNagataTesis"
configure snmp sysLocation "UCSM"
configure snmp sysContact "Toshiro Nagata Bolivar"
configure timezone -300 noautodst
configure sys-recovery-level switch reset

#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-24
configure vr VR-Default add ports 1-24
configure vlan default delete ports 1-24
create vlan "Tesis_Nagata_Mgmt"
configure vlan Tesis_Nagata_Mgmt tag 90
create vlan "Tesis_Nagata_V1"
configure vlan Tesis_Nagata_V1 tag 10
create vlan "Tesis_Nagata_V2"
configure vlan Tesis_Nagata_V2 tag 20
create vlan "Tesis_Nagata_V3"
configure vlan Tesis_Nagata_V3 tag 30
create vlan "Tesis_Nagata_V4"
configure vlan Tesis_Nagata_V4 tag 40
configure vlan Tesis_Nagata_Mgmt add ports 24 tagged
configure vlan Tesis_Nagata_V1 add ports 1-6 untagged
configure vlan Tesis_Nagata_V2 add ports 7-12 untagged
configure vlan Tesis_Nagata_V3 add ports 13-18 untagged
configure vlan Tesis_Nagata_V4 add ports 19-23 untagged
configure vlan Tesis_Nagata_Mgmt ipaddress 10.0.102.2 255.255.255.0
```

Figura 85: Configuración de swtich

Fuente: Elaboración Propia

A continuación, se muestran las *VLAN*S creadas con el comando "show *VLAN*", como podemos observar en la figura 86, las *VLANS* creadas son distintas y mantienen un *tag* de 10,20,30,40, con excepción de la *VLAN* de administración la cual contiene el *tag* 90 y la dirección ip 10.0.102.2 con mascara 24b

```
SWX440-ToshiroNagataTesis.2 # show vlan
-------------------------------------------------------------------------------------------------
Name            VID  Protocol Addr       Flags                        Proto  Ports  Virtual
                                                                             Active router
                                                                             /Total
-------------------------------------------------------------------------------------------------
Default         1    ------------------------------------------------- ANY    0 /0   VR-Default
Mgmt            4095 ------------------------------------------------- ANY    0 /1   VR-Mgmt
Tesis_Nagata_Mgmt 90   10.0.102.2     /24  ---------------------------- ANY    0 /1   VR-Default
Tesis_Nagata_V1 10   ------------------------------------------------- ANY    0 /6   VR-Default
Tesis_Nagata_V2 20   ------------------------------------------------- ANY    0 /6   VR-Default
Tesis_Nagata_V3 30   ------------------------------------------------- ANY    0 /6   VR-Default
Tesis_Nagata_V4 40   ------------------------------------------------- ANY    0 /5   VR-Default
-------------------------------------------------------------------------------------------------
Flags : (B) BFD Enabled, (c) 802.1ad customer VLAN, (C) EAPS Control VLAN,
        (d) Dynamically created VLAN, (D) VLAN Admin Disabled,
        (e) CES Configured, (E) ESRP Enabled, (f) IP Forwarding Enabled,
        (F) Learning Disabled, (i) ISIS Enabled, (I) Inter-Switch Connection VLAN for MLAG,
        (k) PTP Configured, (l) MPLS Enabled, (L) Loopback Enabled,
        (m) IPmc Forwarding Enabled, (M) Translation Member VLAN or Subscriber VLAN,
        (n) IP Multinetting Enabled, (N) Network Login VLAN, (o) OSPF Enabled,
        (O) Flooding Disabled, (p) PIM Enabled, (P) EAPS protected VLAN,
        (r) RIP Enabled, (R) Sub-VLAN IP Range Configured,
        (s) Sub-VLAN, (S) Super-VLAN, (t) Translation VLAN or Network VLAN,
        (T) Member of STP Domain, (v) VRRP Enabled, (V) VPLS Enabled, (W) VPWS Enabled,
        (Z) OpenFlow Enabled

Total number of VLAN(s) : 7
SWX440-ToshiroNagataTesis.3 #
```

Figura 86: Mostrar *VLAN*S creadas en *switch*

Fuente: Elaboración Propia

En la siguiente figura se observa la asignación de puertos a las distintas *VLAN* contando con el puerto 24 asignado a la *VLAN* de administración,

```
SWX440-ToshiroNagataTesis.3 # show ports no-refresh
Port Summary
Port   Display          VLAN Name            Port   Link
#      String           (or # VLANs)         State  State
=============================================================
1                       Tesis_Nagata_V1      E      R
2                       Tesis_Nagata_V1      E      R
3                       Tesis_Nagata_V1      E      R
4                       Tesis_Nagata_V1      E      R
5                       Tesis_Nagata_V1      E      R
6                       Tesis_Nagata_V1      E      R
7                       Tesis_Nagata_V2      E      R
8                       Tesis_Nagata_V2      E      R
9                       Tesis_Nagata_V2      E      R
10                      Tesis_Nagata_V2      E      R
11                      Tesis_Nagata_V2      E      R
12                      Tesis_Nagata_V2      E      R
13                      Tesis_Nagata_V3      E      R
14                      Tesis_Nagata_V3      E      R
15                      Tesis_Nagata_V3      E      R
16                      Tesis_Nagata_V3      E      R
17                      Tesis_Nagata_V3      E      R
18                      Tesis_Nagata_V3      E      R
19                      Tesis_Nagata_V4      E      R
20                      Tesis_Nagata_V4      E      R
21                      Tesis_Nagata_V4      E      R
22                      Tesis_Nagata_V4      E      R
23                      Tesis_Nagata_V4      E      R
24                      Tesis_Nagata_Mgmt    E      R
=============================================================
```

Figura 87: Mostrar puertos de *switch*

Fuente: Elaboración Propia

5.2 Análisis de Explotación de Vulnerabilidades y Seguridad por Capas

En la siguiente figura se observa la puesta en producción de una máquina virtual para realizar pruebas de seguridad la misma a la cual se realizará auditorías de seguridad informática tanto dentro como fuera de la arquitectura de seguridad propuesta para evaluar los resultados en cuanto a la configuración y nivel de seguridad otorgado

```
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:02:66:c1
          inet addr:192.168.81.134  Bcast:192.168.81.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe02:66c1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1048 (1.0 KiB)  TX bytes:1298 (1.2 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:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figura 88: Configuración de red de maquina vulnerable

Fuente: Elaboración Propia

En la siguiente figura se observa el portal web de la máquina virtual de pentesterlab(https://pentesterlab.com/exercises/web_for_pentester), la misma que cuenta con distintas pruebas para distintos fallos de seguridad.

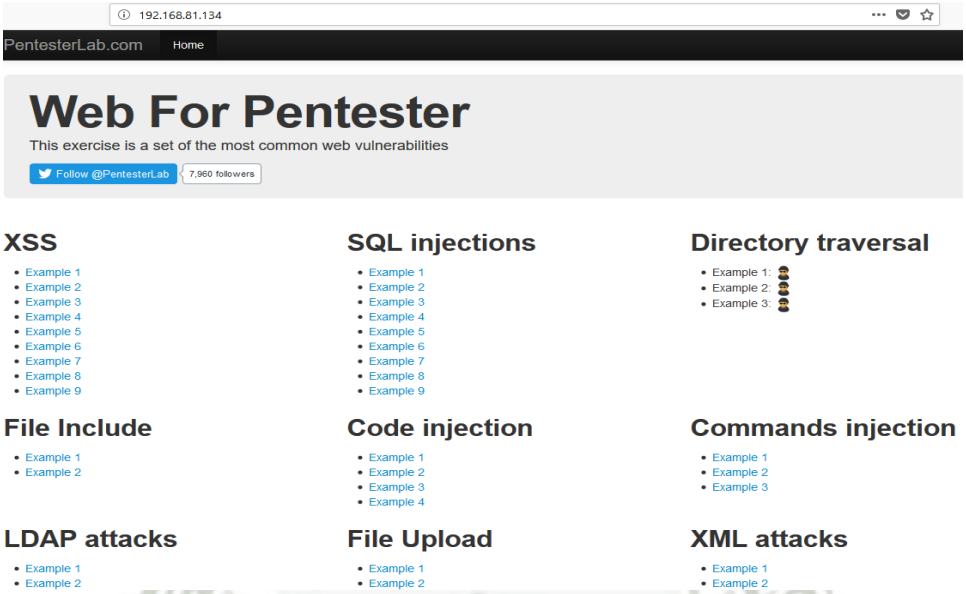


Figura 89: Servicio web levantada en servidor vulnerable

Fuente: Elaboración Propia

Como se mencionó en el capítulo anterior unos de los fallos de seguridad más utilizados y que puede causar un gran impacto sobre una organización es la inyección de códigos SQL, en esta figura se explota el ejemplo 9, el cual tienen una vulnerabilidad de inyección de código SQL, la cual será auditada.

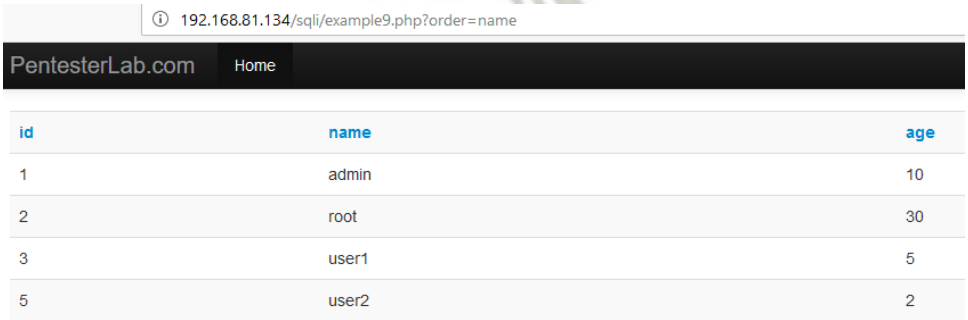


Figura 90: Pagina web con inyección SQL

Fuente: Elaboración Propia

En la siguiente figura se observa el uso de una herramienta llamada SQLAMP, en este caso se probó desde un sistema operativo orientado para auditorías de seguridad Kali Linux (https://www.kali.org/), se observa que se da los parámetros de la maquina auditada con la dirección URL.

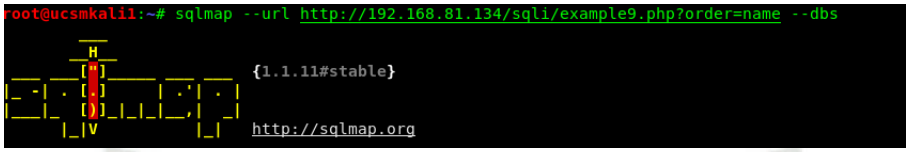


Figura 91: Uso de herramienta SQLMAP

Fuente: Elaboración Propia

Los resultados se pueden observar en la siguiente figura en la cual nos brinda información del motor de base de datos utilizado "MySQL", así como las bases de datos disponibles, siendo uso de este ejemplo la base de datos "*exercises*".

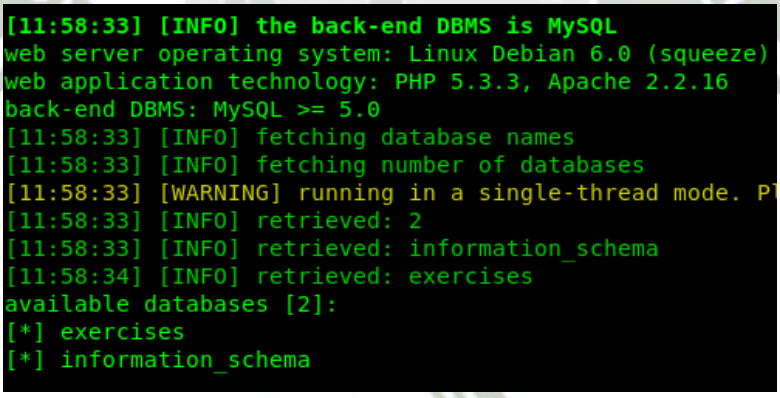


Figura 92: Listado de bases de datos obtenidas

Fuente: Elaboración Propia

A continuación, se brinda el parámetro de la base de datos como se observa en la siguiente figura nos da información sobre las tablas existentes en este caso la tabla encontrada es "*users*"

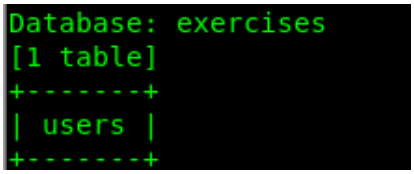


Figura 93: Selección de la tabla

Fuente: Elaboración Propia

En la siguiente figura se observa los detalles de la tabla *users* para posteriormente extraer los datos que esta tabla contiene.

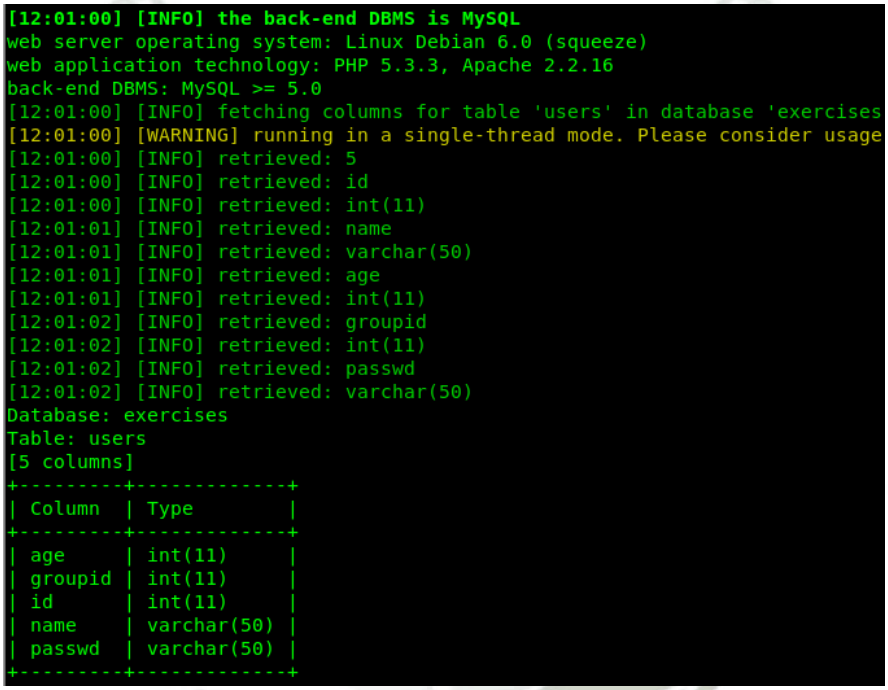


Figura 94: Columnas de la tabla

Fuente: Elaboración Propia

Como se observa en la siguiente figura los datos son extraídos en su totalidad teniendo 4 elementos los cuales contienen credenciales de seguridad de usuarios que podrían contar con privilegios para realizar distintas actividades dentro de la base de datos u otros elementos dentro del servidor, por lo que podrían comprometer el mismo, así como la red en la cual se encuentra.

```
Database: exercises
Table: users
[4 entries]
+----+---------+-----+-------+---------+
| id | groupid | age | name  | passwd  |
+----+---------+-----+-------+---------+
| 1  | 10      | 10  | admin | admin   |
| 2  | 0       | 30  | root  | admin21 |
| 3  | 2       | 5   | user1 | secret  |
| 5  | 5       | 2   | user2 | azerty  |
+----+---------+-----+-------+---------+
```

Figura 95: Extracción del contenido de la tabla

Fuente: Elaboración Propia

A continuación, se observa un fallo de seguridad sobre ejecución de comandos, como se observa en la siguiente figura, se podrían inyectar comandos en la *url*, haciendo que estos sean ejecutados en el servidor donde se encuentra la página web, los mismos que podrían revelar credenciales de seguridad e información confidencial

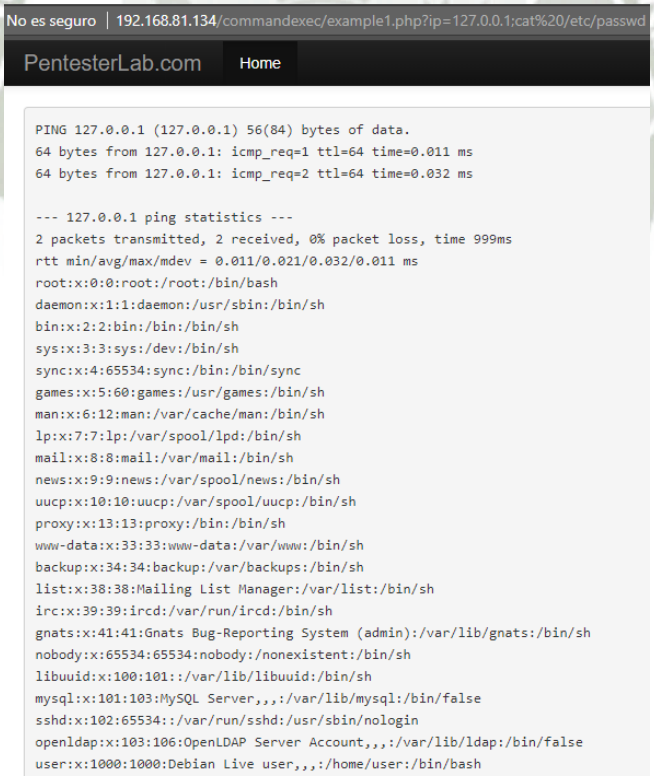


Figura 96: Explotación de ejecución de comandos

Fuente: Elaboración Propia

A continuación, se observa un fallo de seguridad sobre "*File Upload*", el mismo que consiste en subir un archivo que puede contener un código malicioso para tomar control sobre el servidor en el cual se encuentra alojado una página web.

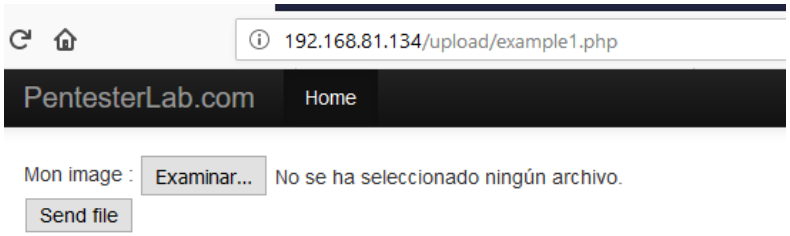


Figura 97: Explotación de *File Upload*

Fuente: Elaboración Propia

A continuación, se muestra un código de *Shell* en php, este archivo denominado "c99.php", es capaz de otorgar un control parcial o total sobre el servidor en el que se ejecuta.

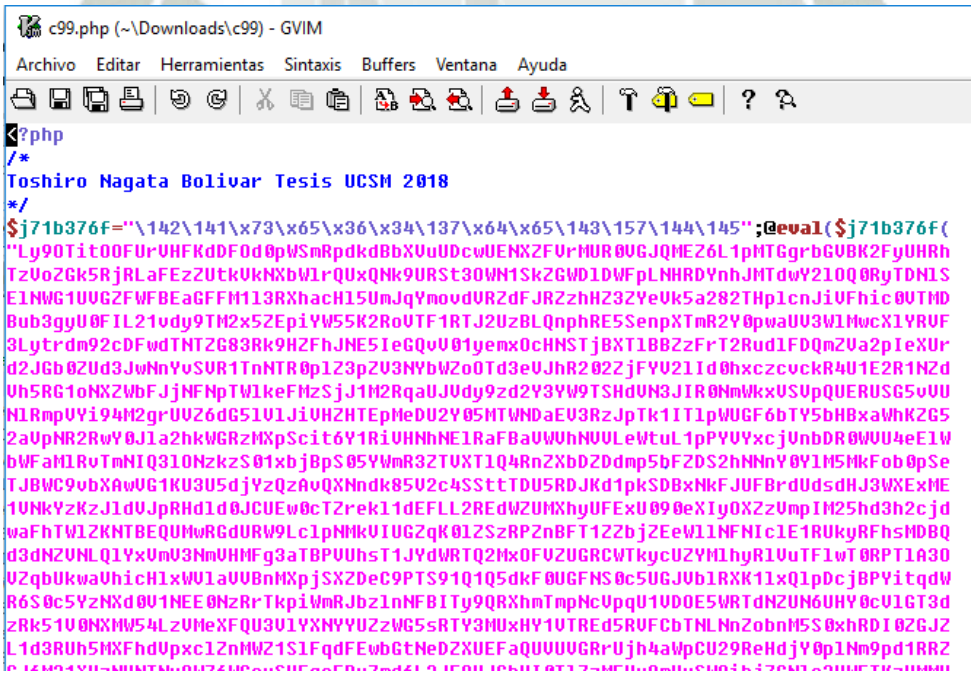


Figura 98: *Shell* c99.php

Fuente: Elaboración Propia

A continuación, se observa la pantalla de administración de la *Shell* la cual fuel subida en la página web, al conectarse a este archivo directamente ofrece una consola de administración sobre las páginas web alojadas en el servidor

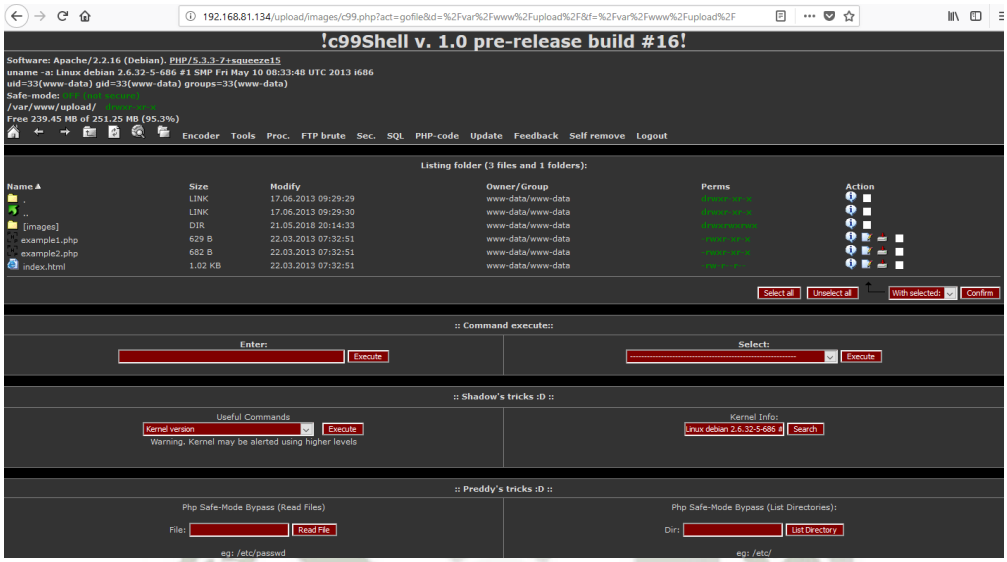


Figura 99: Subida de shell para explotación de File Upload

Fuente: Elaboración Propia

En la siguiente figura se observa lo que se conoce como una desfiguración web, comúnmente consiste en cambiar el index.html de una página web, para este caso se subió una figura de prueba, la cual es redireccionada al index.html, lo mismo que demostrara que se pudo tomar un cierto control sobre la web y/o servidor sobre el que se encuentra alojada la misma, dependiendo de los permisos obtenidos se podrá acceder a las credenciales de seguridad de dicho servidor, o pivotar a la configuración de distintas páginas que estén alojadas dentro del servidor, esto es conocido como "Desfiguración en masa".

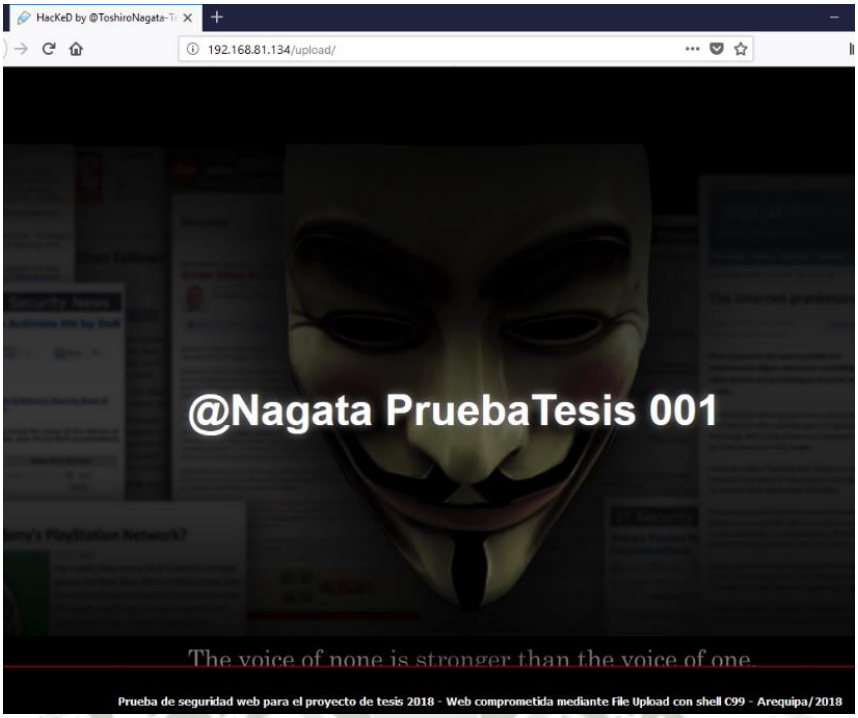


Figura 100: Desfiguramiento de web para prueba de seguridad

Fuente: Elaboración Propia

En la siguiente figura se puede apreciar la captura de un paquete que viaja sin certificado de seguridad (SSL), esto provoca que el tráfico generado viaje a través de la red en texto plano, el cual puede ser capturado y leido por cualquier nodo malicioso que se encuentre en la red en la que se estan transmitiendo los datos, en la figura se observa la captura y examinación del paquete que contiene las credenciales de seguridad, esta superficie de exposición categoriza un gran riesgo dentro de la red, debido a la transmisión de información sensible sin un metodo de cifrado entre el cliente y el servidor, en la figura se puede apreciar los campos de “textBoxUsername” y “textBoxPassword”, asi como el valor que contienen dichos campos, esta captura de tráfico se puede obtener mediante ataques de hombre al medio (MITM), y la posteior captura de tráfico.

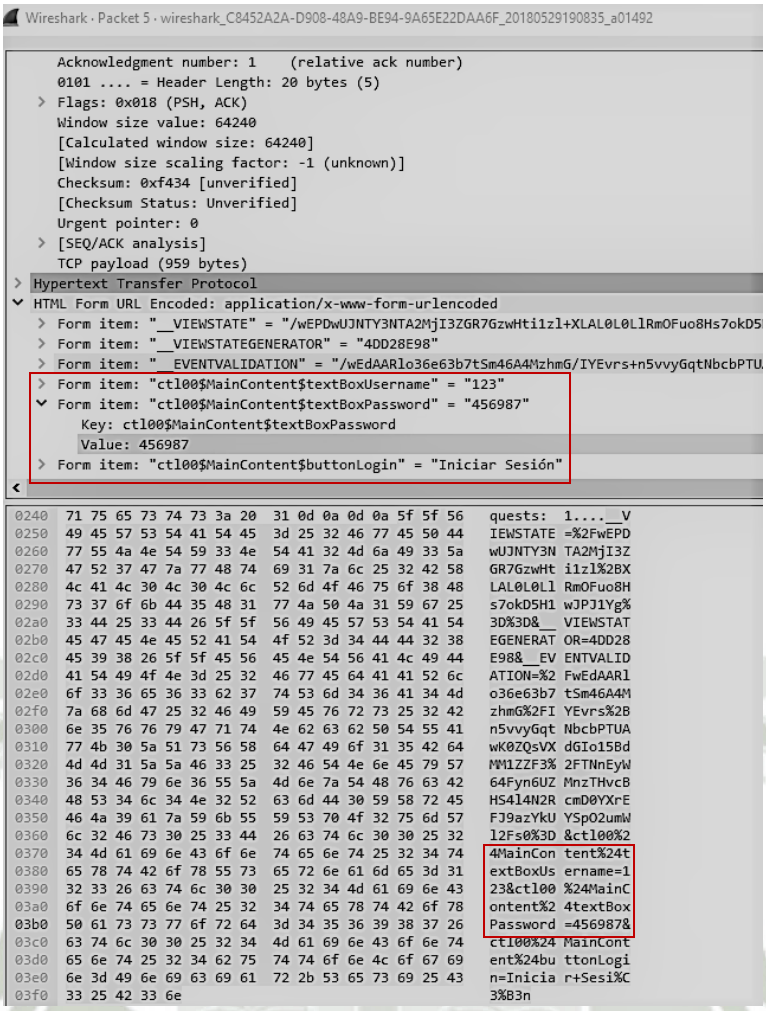


Figura 101: Paquete capturado con credenciales de seguridad

Fuente: Elaboración Propia

A continuación se observa un nivel de seguridad aosicado en esta capa, los certificados de seguridad(*SSL*),para la configuracion de estos certificados en la siguiente figura se observa la direccion IP del servidor que lojara un APACHE2, osbre el cual se montara la estructura web, se observa que esta por defecto direccionado en el puerto 80 el cual utiliza un protocolo *HTTP*, como se explico en el cpaitulo anterior este protocolo no cuenta con ningun nivel de seguridad por lo que el tráfico desde y hacia el se ecnontraron en formato de texto plano,

sometiendo la arquitectura a un fallo de seguridad que podría ser aprovechado por los atacantes.

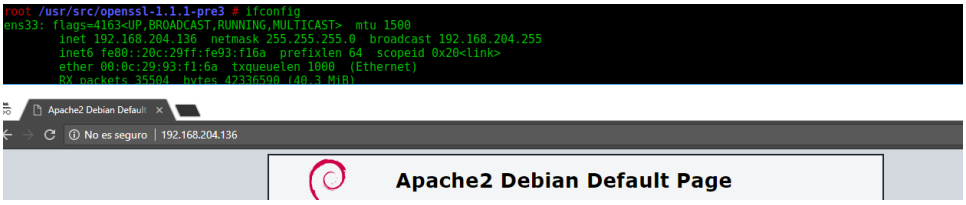


Figura 102: Conexión web sin *SSL*

Fuente: Elaboración Propia

A continuación, se muestra la generación de un certificado de seguridad, generando una llave privada RSA, la cual deberá ser resguardada en dicho servidor.

```
root /etc/apache2 # cat NagataKey1.key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,AB8463C611CC40C4

pEfsoqLy5usdw13q9FqRSBumUYNettDszT+zOP1feeOIuiauuYPS9Hbz8VPF9Vf3
1KKuseNogacFPL8xhf7RWiVb4ihPPfmzrip6gF/aHwtojuzvChxsrqLbeVZryLYc
wpOWGmeONfgrK6BSR9BDheICpOK+1yCjTSEjX9U2FQLsXgd8ETPeE/Fv9BH1EuM6
M+vC+9RH5kRC88yYJuuALtZDWW0IYWzA/0JzZSH5dwJrZH68A17CI1negwmpYEqy
JSyrdoRkE2AJvqxM0Q3RX80E/4HsFBr2CRvVRy+XhM88YmM+n2oZSR9lox6G1AZa
vXuQS1jRbjySkr2V8x/kdhHy7YMyeyfZOkJS65Yrf01DsNXpIv9OrLFPWldTbEon
cq3y0543ePdaa+fQX+Iquqsh7NosKmga/zdKqTysQzXmAHzRen+/jPdTXHczLWWn
NbXWXvxIsRXpXPKm+HHkLGrN05Ro8wRMsCZPNScVpB/LeR2fm3D9DxIyniPOwAKE
yMqDKdPN5bx0jBuHfByxCpxckF0pEgHLKy3px6i+dGBqGW2NWeTzyRAkH3pILPpd
/nvPSxSmzon450keevLcIGQpOGMA84ahR+ebYb7CxIPufMSXldMI1lfFhEZyj/Ak
10BTuQB8HDzgBBUn2VcPVIPN4b6fRQVaOJvlKkBFzuUQKoi2xmOpfhhfTry7s2/k
S9JNCxMDUsq3/Ht3SGreSQlS54fznabWI7jhrWcx6xv4+gCsQ/xYuqg1ImBahtin
cWPhKacjl7suLOzrqzwfv8V9VhCDJN4SfJNp3Swmr88DtpOh/Ae84PSJQkPFHjeU
7OMThB/byGcMBu7WPm9K5QK6+NskmgsAaciaphuFbPKNOEfMdxR9nlJlknmuHoYL
Beyv2S2LtM5TCG6WyWM2iO/IUt89ZqlVcpHsGk4XuyHD8LCvMzA8MqHwHhuZCUxZ
eBRtx6Ioxb8Xcg2RfFIfBo+vRB9eO7YW/UP/zCN+CguIxpWtGeHPiff+JRmKkSlT
TeDf40OCkafmVvAOHIS7S6kvS1nRrbbHaPGulLxFtnYHwuxqg0jYBBBHMQXT/5YT
DjbOkUoK9rVgrL91p4ByuIlfmnUl4bMYexVMW/WXhYvnvx+CMKYB2otHScP42Eo8
eAR8xNRUUjuesrp2sRklq5IBUMpiaVTML9WagyOQNQFzYm2r4EVBhhtp3khDn45c
vbXAjs05M5AmF3GLXsssygGR0KLC1fnQMxQU4aYd9mavCDDc1CJNViMzRuqBHvqN
7oHpZfyPCOi0aJTDC+Zbq00OXuS2cW3+Ajefblvbsa4DGipnewx311xI8bjo3bpZ
qnUX2L4N725Y2OUbIpOEHWW24xcG3OHScUfPTZA1k3AvV7K/yKFxCz8G2u+9F+7Y
iJx/Ki8oNqfjUPZS6Sde9w0qcVxbwFjOLdmoAEB8/n2hfPBMUrwa1l66XbL54eGA
RufO0KU+3/KdvfQmzIJOHw8mme/g555srz1LYTrIvyflrn3Ba/oo2kGL2fCWqk4S
9K7BMwj366rObOg5qcAzLzNdgRA2ld48Fslbbq73xKq0Vpejn7nEVfbVsxP2HDeg
-----END RSA PRIVATE KEY-----
```

Figura 103: Llave privada RSA

Fuente: Elaboración Propia

Como se puede observar en la siguiente figura se utilizó OPEN*SSL*, el mismo que se puede encontrar mayor documentación en www.openssl.org, se observa que una vez generada la llave privada se generara un certificado de seguridad., en el cual se ingresara los parámetros que se solicita para validar dicho certificado de seguridad.

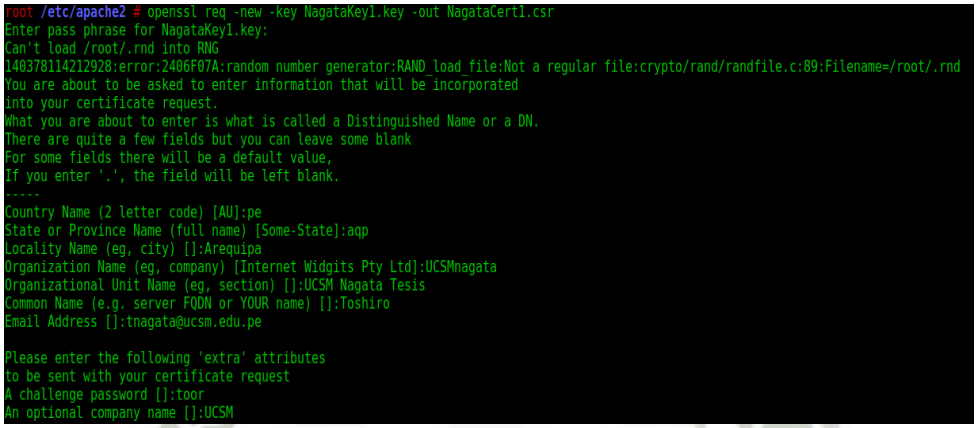


Figura 104: Configuración de la llave privada con el certificado de seguridad

Fuente: Elaboración Propia

En la siguiente figura se observa los parámetros adicionales que serán integrados en el certificado de seguridad tales como cuentas y duración de dicho certificado.

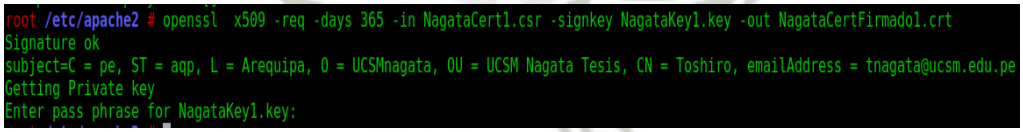


Figura 105: Configuración de datos para el certificado de seguridad *SSL*

Fuente: Elaboración Propia

A continuación, se procede con la configuración final sobre el certificado de seguridad copiando los archivos en el directorio "etc/ss", donde estos serán almacenados para la colocación dentro del archivo de configuración.

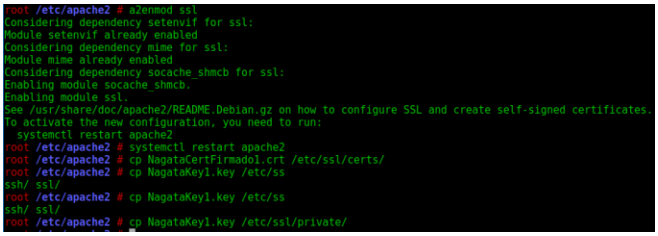


Figura 106: Configuración de certificado de seguridad y llave privada en carpeta necesarias

Fuente: Elaboración Propia

A continuación, se observa el archivo de configuración donde se apuntan los certificados de seguridad al directorio donde fueron copiados anteriormente en el cual se encuentra tanto el certificado de seguridad como la llave privada RSA.

```
#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

SSLOptions +fakeBasicAuth +ExportCertData+StrictRequire
SSLCertificateFile /etc/ssl/certs/NagataCertFirmado1.crt
SSLCertificateKeyFile /etc/ssl/private/NagataKey1.key
```

Figura 107: Configuración de apache con certificado de seguridad *SSL*:

Fuente: Elaboración Propia

En la siguiente figura se muestran los detalles del certificado de seguridad datos como por quien fue emitido, así como la fecha de expiración de este certificado.

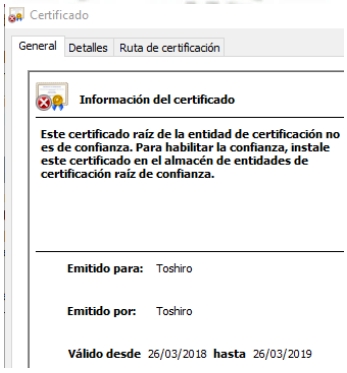


Figura 108: Detalles de certificado de seguridad

Fuente: Elaboración Propia

En la siguiente figura se pueden observar algunos detalles del certificado de seguridad como la llave pública y los datos de contacto, así como la validación de fechas vigentes para este certificado de seguridad.

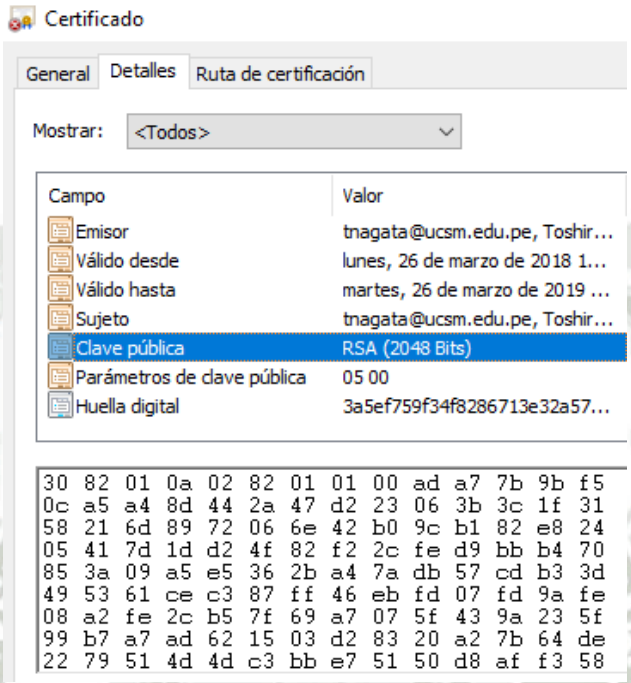


Figura 109: Información de clave pública y parámetros de certificado *SSL*

Fuente: Elaboración Propia

5.3 Dimensionamiento de Medidas de Seguridad en Capas

A continuación se observa el uso de PostgreSQL, para lograr almacenar las medidas de seguridad de una manera mas inteligente, integrando todos los incidentes y direcciones tanto maliciosas ocmo direcciones confiables entre servidores los cuales tienen un tipo especial de comunicación.

En la sguiente figura se observa la base de datos creada para este proyecto la cual es "*Malware*_DB_tesis01", dentro de esta se pueden encontrar las tablas, *deslisted*, hosts, ip, medidasdeseguridad y nodoconfiable, las mismas que analizaremos acontinuación.

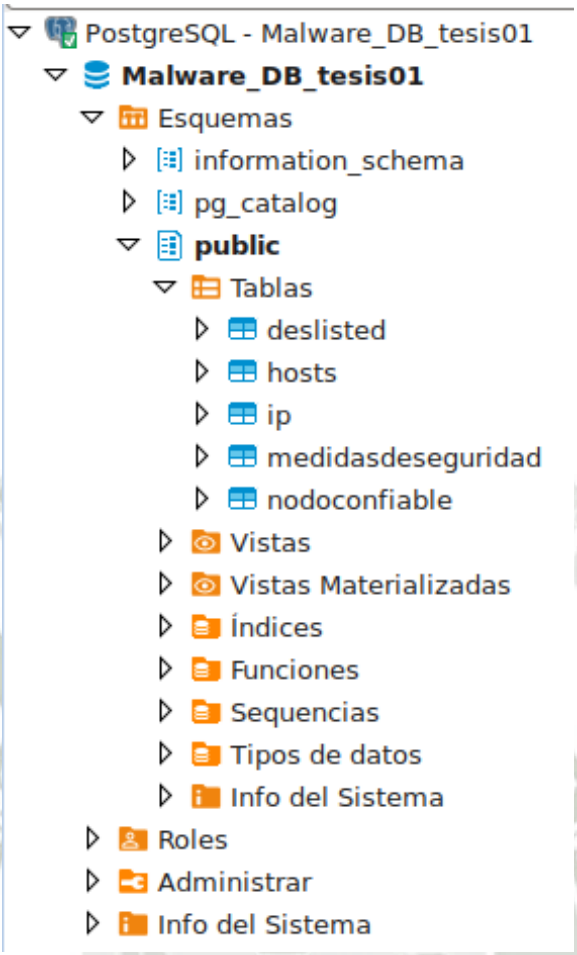
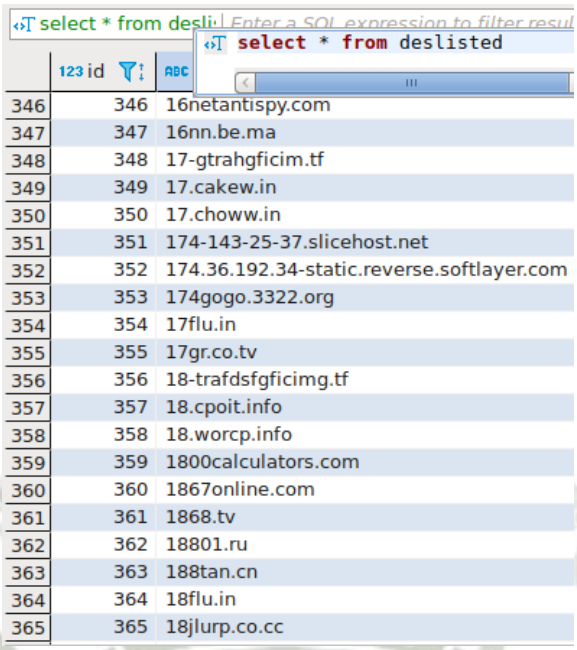


Figura 110: Base de datos *Malware*_DB_tesis01

Fuente: Elaboración Propia

Como se observa en la figura 111 tenemos la tabla *deslisted*, la misma que es actualizada desde "www.malwaredomainlist.com/hostslist/delisted.txt", mediante un script desarrollado en python encargado de revisar actualizaciones mediante *checksum* y buscar las actualizaciones de sitios web maliciosos, de encontrarse actualizaciones, este mismo script es el ecnargado de ingresar a dicha tabla y actualizarla, esta logica es aplicada para la tabla hosts e ip, como se observa en las figuras 112 y 113, estos datos almacenados seran de uso posterior para la generacion de reglas en NIDS y proxy de manera dinamica.

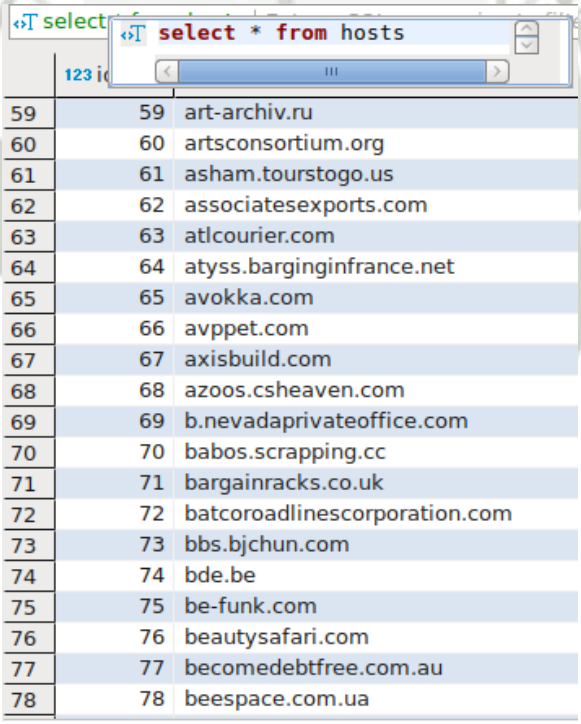


```
select * from deslisted
```

	id	
346	346	16netantispy.com
347	347	16nn.be.ma
348	348	17-gtrahgficim.tf
349	349	17.cakew.in
350	350	17.choww.in
351	351	174-143-25-37.slicehost.net
352	352	174.36.192.34-static.reverse.softlayer.com
353	353	174gogo.3322.org
354	354	17flu.in
355	355	17gr.co.tv
356	356	18-trafdsfgficimg.tf
357	357	18.cpoit.info
358	358	18.worcp.info
359	359	1800calculators.com
360	360	1867online.com
361	361	1868.tv
362	362	18801.ru
363	363	188tan.cn
364	364	18flu.in
365	365	18jlurp.co.cc

Figura 111: Contenido tabla *deslisted*

Fuente: Elaboración Propia



```
select * from hosts
```

59	59	art-archiv.ru
60	60	artsconsortium.org
61	61	asham.tourstogo.us
62	62	associatesexports.com
63	63	atlcourier.com
64	64	atyss.barginginfrance.net
65	65	avokka.com
66	66	avppet.com
67	67	axisbuild.com
68	68	azoos.csheaven.com
69	69	b.nevadaprivateoffice.com
70	70	babos.scrapping.cc
71	71	bargainracks.co.uk
72	72	batcoroadlinescorporation.com
73	73	bbs.bjchun.com
74	74	bde.be
75	75	be-funk.com
76	76	beautysafari.com
77	77	becomedebtfree.com.au
78	78	beespace.com.ua

Figura 112: Contenido de tabla hosts

Fuente: Elaboración Propia

select * from ip | Enter a SQL expre

select * from ip

	123 id	
124	124	174.127.70.195
125	125	174.129.234.86
126	126	174.132.149.189
127	127	174.140.17.100
128	128	174.140.171.207
129	129	174.141.228.147
130	130	174.142.240.91
131	131	174.37.172.162
132	132	174.37.194.151
133	133	175.182.230.91
134	134	176.104.18.152
135	135	176.106.31.227
136	136	176.221.32.120
137	137	176.31.28.226
138	138	176.31.73.196
139	139	176.58.111.253
140	140	176.9.198.14
141	141	176.9.90.164
142	142	177.12.163.81
143	143	178.150.192.50

Figura 113: Contenido de tabla IP

Fuente: Elaboración Propia

En la siguiente figura se muestra el almacenamiento de los nodosconfiables, estos son almacenados siguiendo unos pasos de autenticacion entre nodos.

select * from nodoconfiable | Enter a SQL expression to filter results (use Ctrl+Space)

select * from nodoconfiable

	123 id	ABC di		ash
1	192.168.159.136	00:0c:29:ed:b4:12	4698fdd5ae327b7cfd98d6cefbf283b9077f5127	
2	192.168.159.135	00:0c:29:93:f1:6a	b1790ff38771a3553c241d9743d4ae050915a1c3	
3	192.168.159.134	1C:66:6D:8E:91:E8	c045977cb7a45afaebc20dc4091fe54d3efd36a7	

Figura 114: Contenido de tabla nodo confiable

Fuente: Elaboración Propia

5.4 Despliegue de Medidas de Seguridad por Capas

En el despliegue de medidas de seguridad se tomará una principal importancia a los datos almacenados en las tablas vistas anteriormente (*deslisted*, *hosts*, ip), ya que serán la fuente de alimentación para poder escribir reglas de seguridad en las distintas capas.

En la siguiente figura se aprecia el host vulnerable, el mismo que será redirigido a nuestro servidor encargado del despliegue de medidas de seguridad como se aprecia la nueva ruta de entrada/salida será por medio del host encargado de la seguridad informática.

```
root@debian:/home/user# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.159.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.159.2   0.0.0.0         UG    0      0        0 eth0
root@debian:/home/user# ip r change default via 192.168.159.128
root@debian:/home/user# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.159.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.159.128 0.0.0.0         UG    0      0        0 eth0
root@debian:/home/user# ping 192.168.1.6
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data.

From 192.168.159.128: icmp_seq=40 Redirect Host(New nexthop: 192.168.159.2)
64 bytes from 192.168.1.6: icmp_req=40 ttl=128 time=0.884 ms
From 192.168.159.128: icmp_seq=41 Redirect Host(New nexthop: 192.168.159.2)
64 bytes from 192.168.1.6: icmp_req=41 ttl=128 time=0.937 ms
64 bytes from 192.168.1.6: icmp_req=42 ttl=128 time=0.715 ms
^C
--- 192.168.1.6 ping statistics ---
42 packets transmitted, 3 received, 92% packet loss, time 41126ms
rtt min/avg/max/mdev = 0.715/0.845/0.937/0.097 ms
root@debian:/home/user# _
```

Figura 115: Redirección de rutas de servidor vulnerable

Fuente: Elaboración Propia

En la figura 116 se observa la habilitación de "*ip_forward*", el cual habilitará la conectividad para el servidor que se encuentra detrás de este, encargándose este servidor primario de la revisión de tráfico y el despliegue de medidas de seguridad.

Figura 116: Habilitación de forwarding

Fuente: Elaboración Propia

En las siguientes figuras tanto en 117 como en 118, se puede apreciar la ejecución del script de Python el cual tiene la función de recolectar los datos de las tablas mencionadas para poder escribir una tabla de "medidas de seguridad.

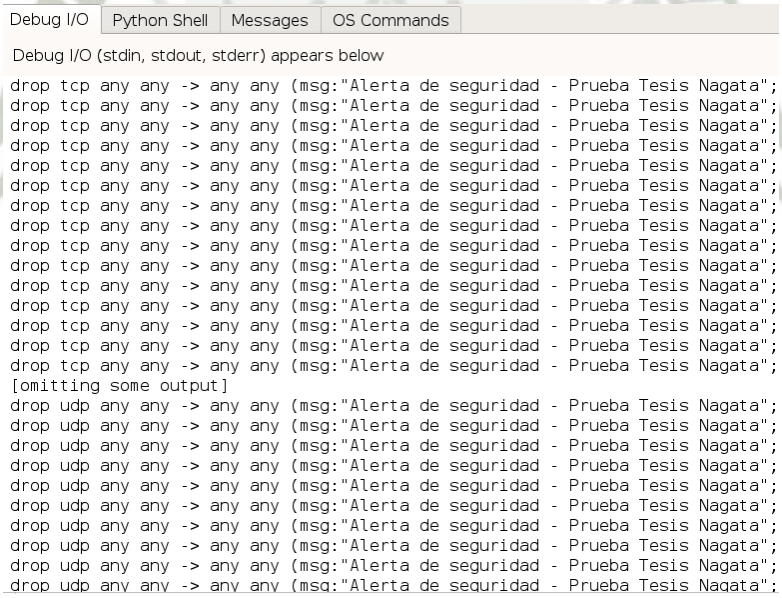


Figura 117: Creación de tabla "medidas de seguridad" parte1

Fuente: Elaboración Propia

```
content:"ze2lagow.info"; classtype:policy-violation; sid:99376;)
content:"zealotbbd6.info"; classtype:policy-violation; sid:99377;)
content:"zebowyhamu.9cy.com"; classtype:policy-violation; sid:99378;)
content:"zebra.ignorelist.com"; classtype:policy-violation; sid:99379;)
content:"zebrabel1.co.cc"; classtype:policy-violation; sid:99380;)
content:"zedexstore.com"; classtype:policy-violation; sid:99381;)
content:"zedmnnuthsvuxo.info"; classtype:policy-violation; sid:99382;)
content:"zedoze9.co.cc"; classtype:policy-violation; sid:99383;)
content:"zeetce.oaktuna.top"; classtype:policy-violation; sid:99384;)
content:"zeetto.1gb.ru"; classtype:policy-violation; sid:99385;)
content:"zeferesds.com"; classtype:policy-violation; sid:99386;)
content:"zehava.in"; classtype:policy-violation; sid:99387;)
content:"zeis.org.ua"; classtype:policy-violation; sid:99388;)
content:"zeitsignale.de"; classtype:policy-violation; sid:99389;)
content:"zeix.cz.cc"; classtype:policy-violation; sid:99390;)

content:"addvertseense.co.uk"; classtype:policy-violation; sid:101606;)
content:"addweb.ru"; classtype:policy-violation; sid:101607;)
content:"ade2e4.info"; classtype:policy-violation; sid:101608;)
content:"adedkiwabvf.com"; classtype:policy-violation; sid:101609;)
content:"adedklwabvf.com"; classtype:policy-violation; sid:101610;)
content:"adekaacamcp.com"; classtype:policy-violation; sid:101611;)
content:"adekniep.info"; classtype:policy-violation; sid:101612;)
content:"adelcio.dias.sites.uol.com.br"; classtype:policy-violation; sid:101613;)
content:"adeliouotre.com"; classtype:policy-violation; sid:101614;)
content:"adelsonalves.hospedagemdesites.ws"; classtype:policy-violation; sid:101615;)
```

Figura 118: Creación de tabla “medidas de seguridad” parte2

Fuente: Elaboración Propia

Como se observa en la figura 119, se realiza una selección según los protocolos TCP y UDP, esto es consecuencia de la ejecución del script en Python, el cual se encuentra configurado para poder anular peticiones en los protocolos TCP, UDP y DNS de cada uno de los datos guardados en las tablas anteriormente mencionadas, los cuales son considerados peligrosos y por ello se niega cualquier tipo de acceso

```
select * from medidasdeseguridad where protocolo='tcp'
union
select * from medidasdeseguridad where protocolo='udp'
```

	id	accion	protocolo	orig	classtype	sid
1011	2.029	drop	tcp	any a	classtype:Violacion de seguridad-Prueba de tesis Nagata001;	sid:2029;)
1012	2.030	drop	tcp	any a	classtype:Violacion de seguridad-Prueba de tesis Nagata001;	sid:2030;)
1013	2.031	drop	tcp	any any -> any any	(msg:"Alerta de seguridad"; content:"98.131.132.1";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2031;)
1014	2.032	drop	tcp	any any -> any any	(msg:"Alerta de seguridad"; content:"98.131.172.1";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2032;)
1015	2.033	drop	tcp	any any -> any any	(msg:"Alerta de seguridad"; content:"98.131.229.2";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2033;)
1016	2.034	drop	tcp	any any -> any any	(msg:"Alerta de seguridad"; content:"98.158.178.231";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2034;)
1017	2.035	drop	tcp	any any -> any any	(msg:"Alerta de seguridad"; content:"1.1.1.1";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2035;)
1018	2.036	drop	tcp	any any -> any any	(msg:"Alerta de seguridad"; content:"2.2.2.2";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2036;)
1019	2.037	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.14.120.121";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2037;)
1020	2.038	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.19.89.55";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2038;)
1021	2.039	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.224.212.222";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2039;)
1022	2.040	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.24.13.91";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2040;)
1023	2.041	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.31.186.207";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2041;)
1024	2.042	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.31.186.29";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2042;)
1025	2.043	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.4.16.91";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2043;)
1026	2.044	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.4.218.22";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2044;)
1027	2.045	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.6.196.156";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2045;)
1028	2.046	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.8.127.189";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2046;)
1029	2.047	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"103.8.127.205";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2047;)
1030	2.048	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.152.215.90";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2048;)
1031	2.049	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.200.67.194";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2049;)
1032	2.050	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.245.239.7";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2050;)
1033	2.051	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.27.163.228";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2051;)
1034	2.052	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.28.14.104";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2052;)
1035	2.053	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.28.15.104";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2053;)
1036	2.054	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"104.31.75.147";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2054;)
1037	2.055	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"107.161.144.14";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2055;)
1038	2.056	drop	udp	any any -> any any	(msg:"Alerta de seguridad"; content:"107.180.26.77";	classtype:Violacion de seguridad-Prueba de tesis Nagata001; sid:2056;)

Figura 119: Selección de la tabla de medidas de seguridad unidos por protocolos

Fuente: Elaboración Propia

En la siguiente figura se muestra la creación del archivo de reglas de seguridad donde se escribieron las reglas que fueron grabadas en la tabla "medidas de seguridad", esta secuencia se realiza por los siguientes motivos:

- Prevenir eliminación de las reglas de seguridad del archivo de reglas.
- Mantener actualizadas las reglas de seguridad.
- Mantener un control y orden sobre la creación de reglas de seguridad.
- Mantener la persistencia e integridad de las reglas de seguridad para que no sean manipuladas.

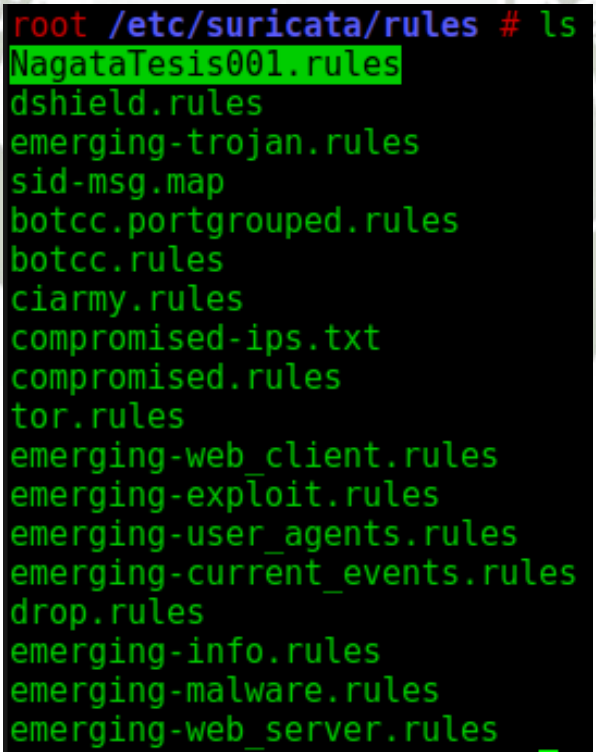


Figura 120: Creación del archivo de reglas de seguridad

Fuente: Elaboración Propia

En la siguiente figura se observa el archivo de configuración de suricataIDS, el mismo que se encuentra ubicado en "/etc/suricata/suricata.yaml", esta configuración se realiza para activar las políticas de seguridad ya creadas.

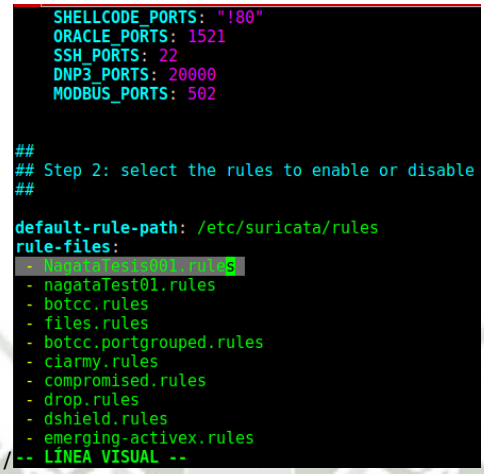


Figura 121: Configuración para añadir reglas a suricataIDS

Fuente: Elaboración Propia

A continuación, tanto en la figura 122 como en la figura 123, se aprecia el archivo con las reglas que fueron generadas y agregadas desde la tabla "medidas de seguridad"

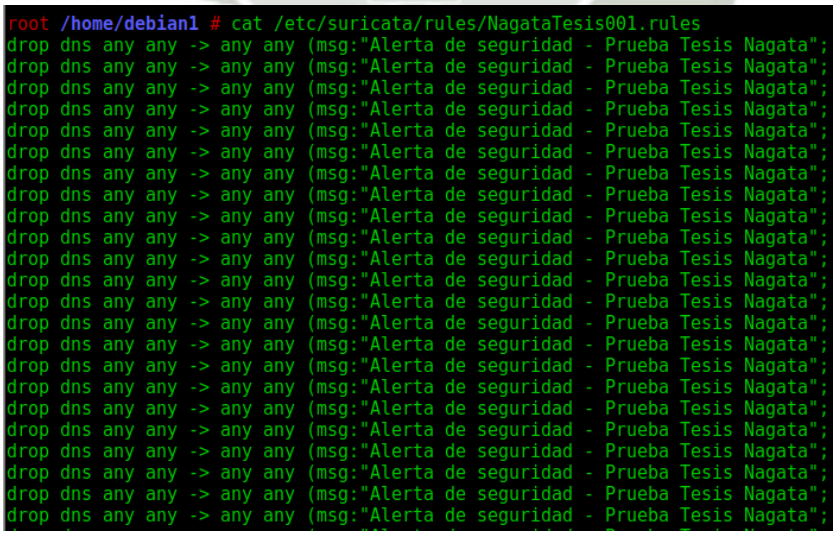


Figura 122: Reglas generadas en suricataIDS parte I

Fuente: Elaboración Propia

```
content:"103.14.120.121"; classtype:policy-violation; sid:1;)
content:"103.19.89.55"; classtype:policy-violation; sid:2;)
content:"103.224.212.222"; classtype:policy-violation; sid:3;)
content:"103.24.13.91"; classtype:policy-violation; sid:4;)
content:"103.31.186.207"; classtype:policy-violation; sid:5;)
content:"103.31.186.29"; classtype:policy-violation; sid:6;)
content:"103.4.16.91"; classtype:policy-violation; sid:7;)
content:"103.4.218.22"; classtype:policy-violation; sid:8;)
content:"103.6.196.156"; classtype:policy-violation; sid:9;)
content:"103.8.127.189"; classtype:policy-violation; sid:10;)
content:"103.8.127.205"; classtype:policy-violation; sid:11;)
content:"104.152.215.90"; classtype:policy-violation; sid:12;)
content:"104.200.67.194"; classtype:policy-violation; sid:13;)
content:"104.245.239.7"; classtype:policy-violation; sid:14;)
content:"104.27.163.228"; classtype:policy-violation; sid:15;)
content:"104.28.14.104"; classtype:policy-violation; sid:16;)
content:"104.28.15.104"; classtype:policy-violation; sid:17;)
content:"104.31.75.147"; classtype:policy-violation; sid:18;)
content:"107.161.144.14"; classtype:policy-violation; sid:19;)
content:"107.180.26.77"; classtype:policy-violation; sid:20;)
content:"108.162.198.96"; classtype:policy-violation; sid:21;)
content:"108.162.199.96"; classtype:policy-violation; sid:22;)
content:"108.163.178.131"; classtype:policy-violation; sid:23;)
content:"108.168.210.189"; classtype:policy-violation; sid:24;)
content:"108.179.202.25"; classtype:policy-violation; sid:25;)
```

Figura 123: Reglas generadas en *suricataIDS* parte II

Fuente: Elaboración Propia

En la siguiente figura se sigue observando la continuación del archivo de reglas de seguridad autogenerado, esto debido a que se eliminará cualquier tipo de acceso no solo a las direcciones IP, sino a la resolución de nombres de dominio de alguna de las paginas categorizadas como peligrosas.

```
content:"www.sftcp.cn"; classtype:policy-violation; sid:50461;)
content:"www.sg2321.cn"; classtype:policy-violation; sid:50462;)
content:"www.sg2678.cn"; classtype:policy-violation; sid:50463;)
content:"www.sh1908.org"; classtype:policy-violation; sid:50464;)
content:"www.shanghai.co.kr"; classtype:policy-violation; sid:50465;)
content:"www.shangzhuan.com"; classtype:policy-violation; sid:50466;)
content:"www.shanyrack.com"; classtype:policy-violation; sid:50467;)
content:"www.sharingpath.org"; classtype:policy-violation; sid:50468;)
content:"www.sharma.com.ua"; classtype:policy-violation; sid:50469;)
content:"www.sharon.or.kr"; classtype:policy-violation; sid:50470;)
content:"www.shaynetjensen.com"; classtype:policy-violation; sid:50471;)
content:"www.shefler.net"; classtype:policy-violation; sid:50472;)
content:"www.shells.kit.net"; classtype:policy-violation; sid:50473;)
content:"www.shells4you.net"; classtype:policy-violation; sid:50474;)
content:"www.shemel.co.cc"; classtype:policy-violation; sid:50475;)
content:"www.shhdyb.cn"; classtype:policy-violation; sid:50476;)
content:"www.shiko181.cn"; classtype:policy-violation; sid:50477;)
content:"www.shilee.com"; classtype:policy-violation; sid:50478;)
content:"www.shinchang.es.kr"; classtype:policy-violation; sid:50479;)
content:"www.shinsungbuk.com"; classtype:policy-violation; sid:50480;)
content:"www.shiratech.com"; classtype:policy-violation; sid:50481;)
content:"www.shkolyarsotakyu.cn"; classtype:policy-violation; sid:50482;)
content:"www.shogunlevallois.com"; classtype:policy-violation; sid:50483;)
content:"www.shomaliha.com"; classtype:policy-violation; sid:50484;)
content:"www.shop.beastieboys.com"; classtype:policy-violation; sid:50485;)
content:"www.shop86.biz"; classtype:policy-violation; sid:50486;)
content:"www.shopdirect4u.com"; classtype:policy-violation; sid:50487;)
```

Figura 124 Reglas generadas en *suricataIDS* para resoluciones DNS

Fuente: Elaboración Propia

En la siguiente figura se observa parte del archivo de configuración de prueba donde se puede apreciar la acción "*drop*" para cierto tipo de páginas y envió de alertas también en este caso la implementación de reglas para "Facebook.com"

```
drop dns any any -> any any (msg:"DNS Facebook"; content:"facebook"; classtype:policy-violation; sid:39398144; rev:1;)
drop tls any any -> any any (msg:"SSL Facebook"; tls.subject:"facebook"; classtype:policy-violation; sid:39398145; rev:1;)
drop tcp any any -> any any (msg:"facebook is blocked"; content:"facebook.com"; http_header; nocase; classtype:policy-violation; sid:1;
drop udp any any -> any any (msg:"facebook is blocked"; content:"facebook.com"; http_header; nocase; classtype:policy-violation; sid:12
```

Figura 125: Configuración de reglas de seguridad para "*facebook*"

Fuente: Elaboración Propia

La configuración del NIDS utilizado en este caso es *suricataIDS*, el mismo que al entrar en modo de alerta se configura para que pueda realizar un análisis de las conexiones y si se detecta una imagen poder mantener un registro de que imagen es la que se está transmitiendo por medio de la red, como se ve en la siguiente figura al revisar el registro de *"/var/log/suricata/files*", contiene las imágenes que se transmitieron en la red para poder llevar un control sobre las mísmas.

```
root /var/log/suricata/files # ls
file.1          file.12.meta  file.15.meta
file.10         file.13       file.16
file.10.meta    file.13.meta  file.16.meta
file.11         file.14       file.17
file.11.meta    file.14.meta  file.17.meta
file.12         file.15       file.18
```

Figura 126: Imágenes capturadas por *SuricataIDS*

Fuente: Elaboración Propia

A continuación, en la figura 127 se muestra la revisión de un registro en específico de una imagen, teniendo en cuenta los metadatos para poder realizar un análisis detallado.

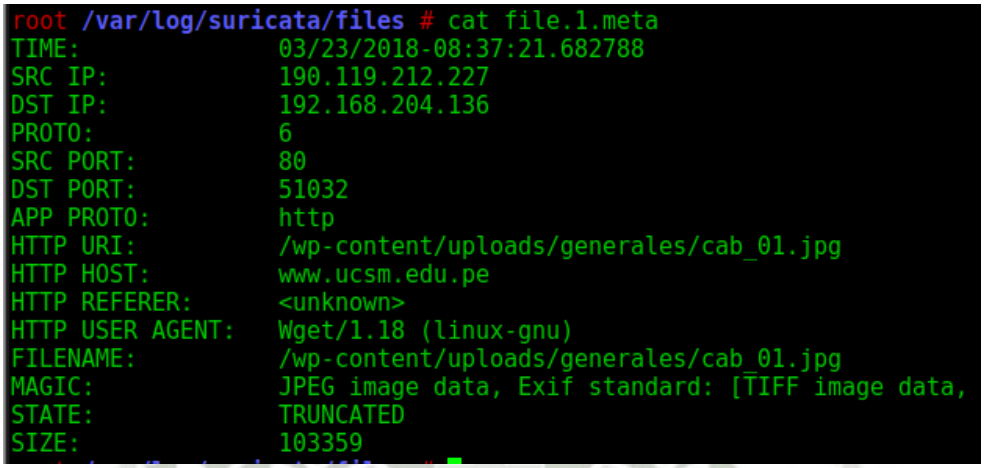


Figura 127: Inspección de un archivo de metadatos de figura capturada por *suricataIDS*

Fuente: Elaboración Propia

En la siguiente figura se puede observar como entra en funcionamiento las reglas de *suicataIDS* configuradas, las mismas que al realizar la resolución de nombre de dominio de Facebook lanza alertas de seguridad y se interrumpe dicha conexión.

Figura 128: Eliminación de conexiones DNS a "Facebook"

Fuente: Elaboración Propia

En la siguiente figura se observa clasificación de imágenes activada, se lanzan las alertas de seguridad cada vez que se realiza el transporte de imágenes dentro del tráfico de la red.

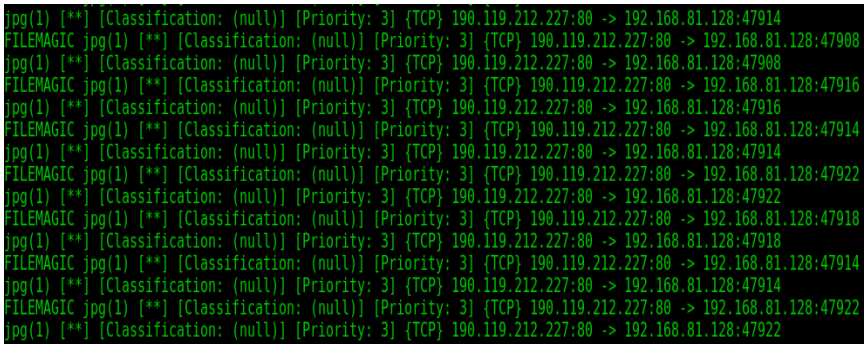


Figura 129: Alertas de *suricataIDS* sobre tráfico que contiene imágenes

Fuente: Elaboración Propia

Luego de ver la implementación de reglas auto generadas en *suricataIDS*, se observa que al cambiar de nivel en las capas se necesita cubrir distintos puntos de seguridad, para ello con la misma lógica de auto generación de reglas para el *IDS*, se generara reglas de seguridad en el firewall empleando en este caso *IPTABLES*, en la siguiente figura se observa el resultado de la generación de reglas en *IPTABLES*, el bloqueo de estas rutas fue generado alimentándose con las tablas de , *deslisted*, *hosts*, ip, por ello se tomará la acción *DROP*, al realizar solicitudes de entrada o salida a la red.

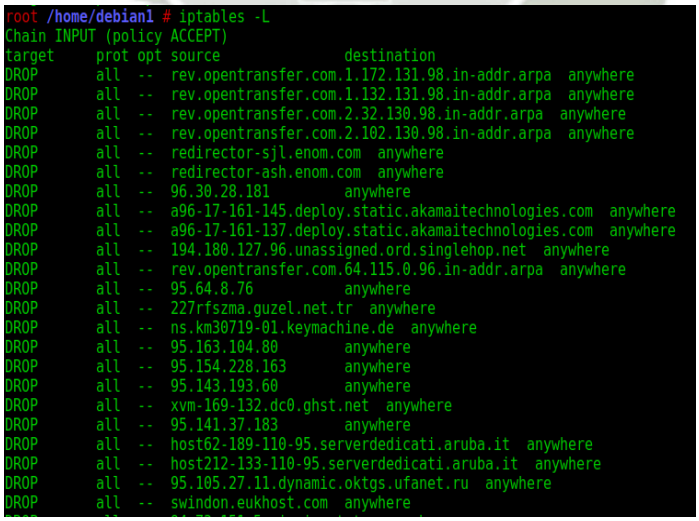


Figura 130: Iptables con reglas de seguridad autogeneradas para IPv4

Fuente: Elaboración Propia

IP6TABLES es el firewall configurado para poder registrar conexiones mediante IPV6, en este protocolo existen algunas variantes en cuanto a los ataques debido a los nuevos protocolos, en este caso luego de generar las reglas para conexiones IPv4, se generarán una serie de reglas para mantener las conexiones IPv6 seguras, permitiendo solo acceso a determinados protocolos muy específicos y denegando el tráfico distinto al especificado.

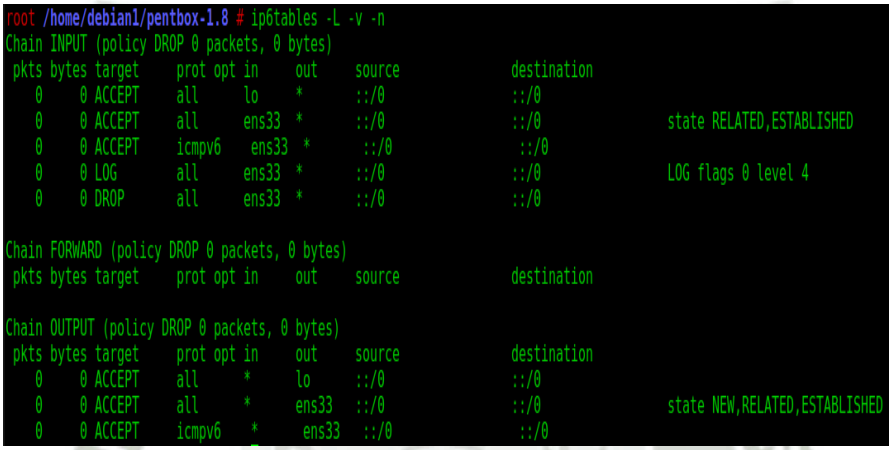


Figura 131: Iptables con reglas de seguridad autogeneradas para IPv6

Fuente: Elaboración Propia

En la siguiente figura se puede apreciar la misma lógica de auto generación de reglas de seguridad para el proxy utilizado en este caso *SQUID*, se genera un archivo de sitios bloqueados obtenido de la tabla de *deslisted*, así como de la tabla host, estos serán bloqueados en caso se logre evadir las anteriores medidas d seguridad y se logre resolver el nombre de dominio, estos sitios no podrán ser visitados, debido a la configuración de sitios bloqueados en *SQUIDPROXY*.

```
root /etc/squid # cat blocksitesTesisNagata1
0koryu0.easter.ne.jp
109-204-26-16.netconnexion.managedbroadband.co.uk
1866809.securefastserver.com
2amsports.com
4dexports.com
50efa6486f1ef.skydivesolutions.be
61kx.uk-insolvencydirect.com
6b8a953b2bf7788063d5-6e453f33ecbb90f11a62a5c376375af3.r71.cf5.rackcdn.com
97b1c56132dfcdd90f93-0c5c8388c0a5897e648f883e2c86dc72.r54.cf5.rackcdn.com
999fitness.com
a.update.51edm.net
ab.usageload32.com
abcdespanol.com
above.e-rezerwacje24.pl
absurdity.flarelight.com
achren.org
acool.csheaven.com
ad-beast.com
ad.9tv.co.il
ad.getfond.info
adgallery.whitehousedrugpolicy.gov
adlock.in
adobeflashupdate14.com
ads.wikipartes.com
adserving.favorit-network.com
adv.riza.it
advancetec.co.uk
afa15.com.ne.kr
```

Figura 132: Lista de sitios bloqueados para *SQUIDProxy*

Fuente: Elaboración Propia

El siguiente despliegue de seguridad consiste en la integración y levantamiento de servicio de un *Honeypot*, el cual servirá para poder tomar en cuenta algunos puntos vulnerables para el análisis y posteriormente tomar medidas de seguridad ante estos puntos vulnerables, este acceso por medio del *Honeypot* no interrumpirá los servicio del servidor oficial, como se puede observar en la siguiente figura se levanta el servicio y se realiza una prueba de conexión al puerto 23, identificado como un puerto vulnerable empleando el servicio de telnet para una conexión remota, esta conexión es detectada y almacenada en los logs como se puede apreciar tanto en la figura 133 como en la figura 134.

```
root@ngtK001:~/Escritorio/TesisNagata# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.159.136  netmask 255.255.255.0  broadcast 19
        inet6 fe80::20c:29ff:feed:b412  prefixlen 64  scopeid 0x2
        ether 00:0c:29:ed:b4:12  txqueuelen 1000  (Ethernet)
                                                    root@ngtK0
root@ngtK001:~/Escritorio/TesisNagata# telnet 192.168.159.128 23
Trying 192.168.159.128...
Connected to 192.168.159.128.
```

Figura 133: Conexión de prueba a puerto vulnerable

Fuente: Elaboración Propia

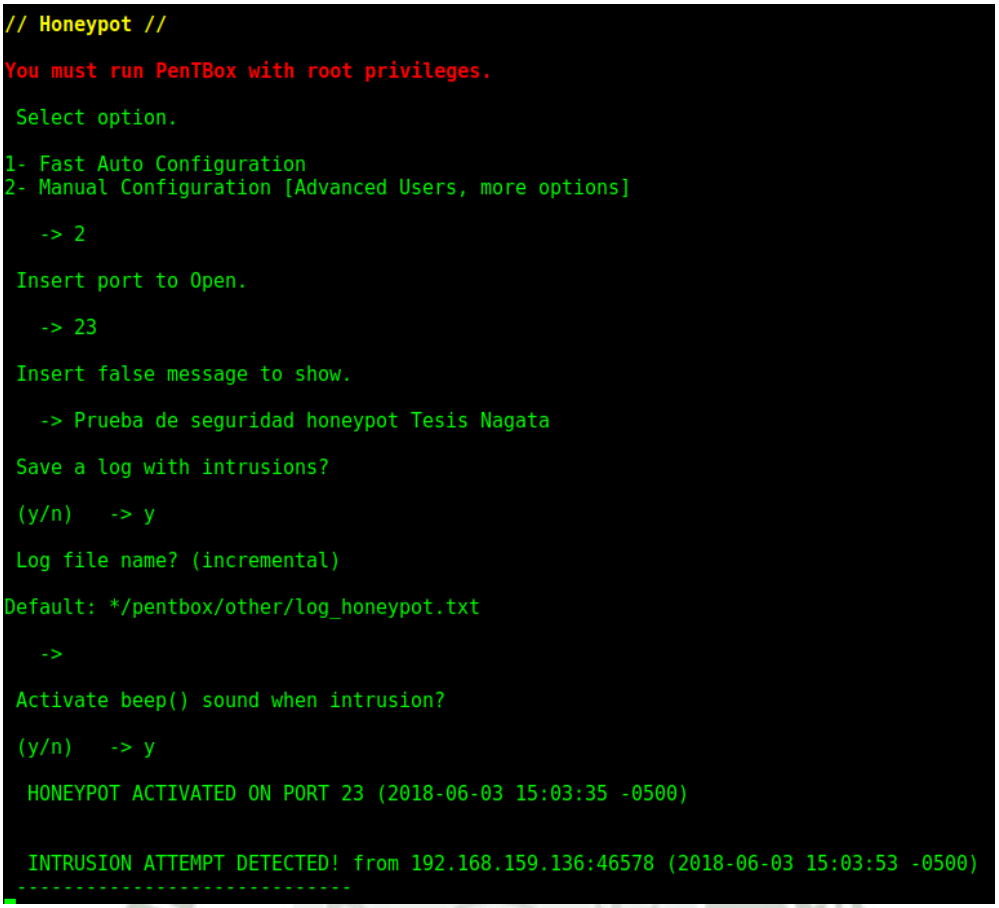


Figura 134: Establecimiento de *Honeypot* y detección de conexión remota.

Fuente: Elaboración Propia

5.5 Limitaciones de la Arquitectura de Seguridad

Aunque el despliegue de seguridad se realizó de manera satisfactoria, una arquitectura de seguridad por más eficiente y minuciosa no podrá cubrir el 100% de las vulnerabilidades, esto es debido a que tiene que existir una concientización de seguridad para los usuarios finales, para probar este punto se realizado la programación de un *keylogger* en el lenguaje C++, el cual puede ejecutarse en distintos sistemas operativos, en la siguiente figura se observa el ejecutable de este *malware* (KL001nagata.exe), así como la creación automática de un archivo log(*AppLog*)

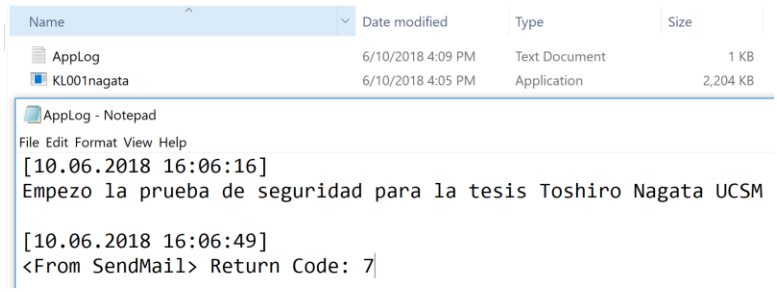


Figura 135: Ejecutable (.exe) y log de *keylogger*

Fuente: Elaboración Propia

En la siguiente figura se aprecia la ejecución de este *malware* como un proceso en el sistema, el mismo que consume una cantidad de recursos mínimo.

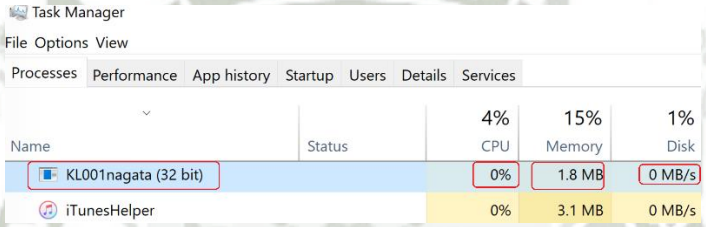


Figura 136: Ejecución de *malware* como proceso y uso de recursos

Fuente: Elaboración Propia

A continuación, para las pruebas se observa en la siguiente figura la escritura de un texto normal en un bloc de notas, el texto será capturado por este *malware* y enviado a un correo electrónico.

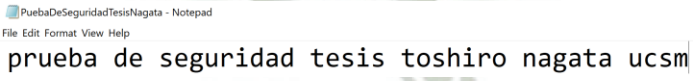


Figura 137: Escritura de texto para prueba de *malware*

Fuente: Elaboración Propia

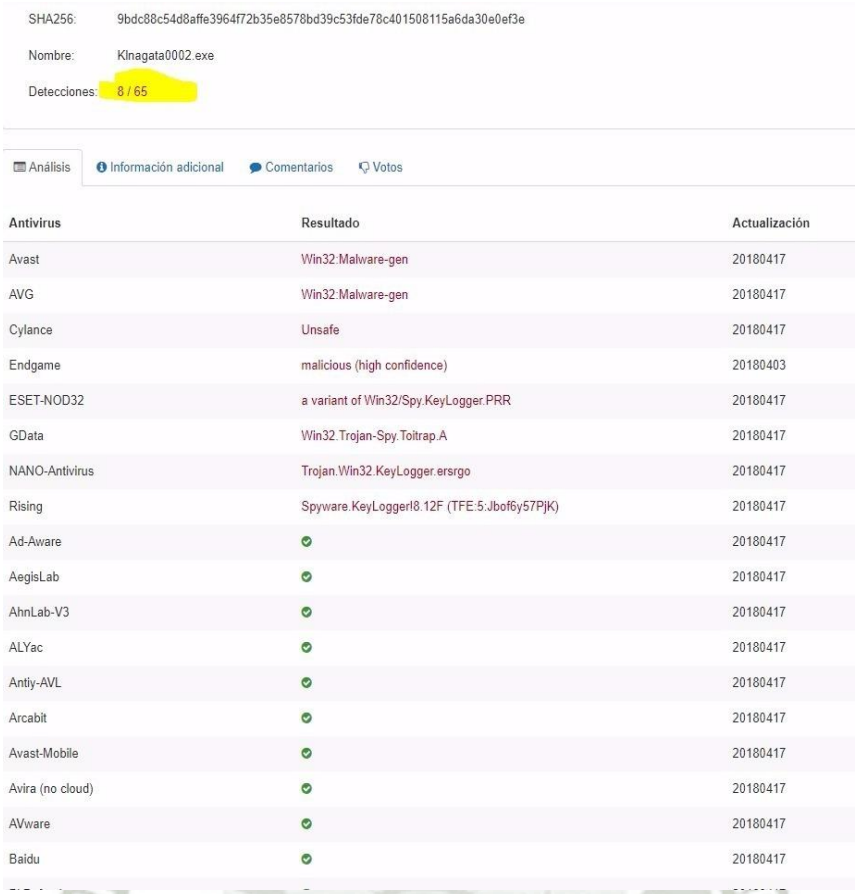
En la siguiente figura se muestra el correo enviado por el *malware*, este correo contiene todas las teclas pulsadas por el usuario infectado, evadiendo las medidas de seguridad ya que en este caso la infección es por medio del correo

electrónico, ejecutando un archivo para el que no se encuentran registros de seguridad, lo que es equivalente a una explotación por *0 day*, es decir un ataque que no tiene antecedentes históricos motivo por el cual no se pueden tomar medidas de seguridad.

Figura 138: Revisión de *log* con letras pulsadas enviadas por el *malware*

Fuente: Elaboración Propia

En este caso como los anti*malware* no cuentan con la firma de seguridad existentes motivo por el que pocos *antimalware* son capaces de detectarlo, así se muestra en la siguiente figura, este *malware* fue probado en www.virustotal.com, el cual es un proyecto mantenido actualmente por Google, esto nos permite obtener distintas opiniones de distintas firmas de *antimalware* acerca de si nuestro archivo podría ser maliciosos o no, teniendo en este caso un total de 8 *antimalware* de 65 que lo reconocen como una amenaza, este ejemplo puede verse en la actualidad como los ataques por medio de ransomware.



SHA256: 9bdc88c54d8affe3964f72b35e8578bd39c53fde78c401508115a6da30e0ef3e

Nombre: Klnagata0002.exe

Detecciones: 8 / 65

Análisis | Información adicional | Comentarios | Votos

Antivirus	Resultado	Actualización
Avast	Win32:Malware-gen	20180417
AVG	Win32:Malware-gen	20180417
Cylance	Unsafe	20180417
Endgame	malicious (high confidence)	20180403
ESET-NOD32	a variant of Win32/Spy.KeyLogger.PRR	20180417
GData	Win32.Trojan-Spy.Toitrap.A	20180417
NANO-Antivirus	Trojan.Win32.KeyLogger.ersrgo	20180417
Rising	Spyware.KeyLogger!8.12F (TFE:5:Jbof6y57PjK)	20180417
Ad-Aware	✓	20180417
AegisLab	✓	20180417
AhnLab-V3	✓	20180417
ALYac	✓	20180417
Antiy-AVL	✓	20180417
Arcabit	✓	20180417
Avast-Mobile	✓	20180417
Avira (no cloud)	✓	20180417
AVware	✓	20180417
Baidu	✓	20180417

Figura 139: Validación de *malware* en virustotal.com

Fuente: Elaboración Propia

Aun en este caso de infección que no puede controlarse la arquitectura de seguridad propuesta permite mantener este riesgo de manera aislada evitando que pueda extenderse a toda la red, debido a la segmentación de redes por medio d *VLANS* y medidas de seguridad tomadas en el *firewall.*

CAPÍTULO VI

ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En el presente capítulo se presentará los resultados obtenidos por medio de la implementación de la arquitectura de seguridad en un en un entorno real, se realizaron distintos tipos de ataques de seguridad informática contra el servidor encargado de desplegar las medidas de seguridad, logrando disminuir la superficie de exposición y minimizar/mitigar el riesgo ante la explotación de vulnerabilidades modernas, del mismo modo se comprobaran las conexiones inversas sobre nodos confiables las cuales se mantendrán cifradas empleando los algoritmos de AES-256 y RSA.

En la siguiente tabla se aprecia una comparación entre distintos algoritmos de cifrado simétricos.

Algoritmos de cifrado	Descripción
Aes128cbc Aes192cbc Aes256cbc	AES (Advanced Encryption Standerd) es un nuevo estándar adoptado por el gobierno de U.S.A. que remplazara a DES/3DES. AES también es conocido como el algoritmo de encrypcion Rijndael, el cual es muy veloz. El número después del nombre del algoritmo es el tamaño de cifrado que usa.
3des-cbc	DES y 3DES fueron los primeros 2 algoritmos de cifrado disponibles. Creados por la NSA (Nacional Security Agency de USA), a principios de los de la década de los 70. estos algoritmos han sido susceptibles a análisis criptográfico, por lo que

	no son muy recomendados si se tiene otros algoritmos disponibles
blowfishcbc	Blowfish es un algoritmo de cifrado en bloque con llave simétrica diseñado por Schneier que usa una llave de tamaño variable y un bloque de 64 bits. La llave puede ser de 32 a 448 bits. Los datos son pasados por la función de encriptamiento 16 veces
cast128cbc	CAST128 es un algoritmo parecido a DES que tiene una buena resistencia al análisis criptográfico.
arcfour	Arcfour es un algoritmo basado en RC4 y SHA1, simétrico y de stream. Tiene un nivel bajo en seguridad

Tabla 12: Comparación de algoritmos simétricos

Fuente: Elaboración Propia

A continuación, se aprecia una tabla con la comparación entre los algoritmos de cifrado simétrico y la cantidad de bits que se requieres para realizar el cifrado de los datos.

Algoritmo de Cifrado	Cantidad de bits para cifrado
AES	128, 192, 256
DES	8 a 64
TRIPLEDES	8 a 192
BLOWFISH	8 a 448
RIJNDAEL-256	8 a 256
SERPENT	8 a 256
TWOFISH	8 a 256

Tabla 13: Comparación entre algoritmos simétricos y bits de cifrado

Fuente: Elaboración Propia

Debido a la polémica existente entre el algoritmo de cifrado DES contra el algoritmo de cifrado AES, en la siguiente tabla se puede apreciar las posibles combinaciones para estos cifrados según el tamaño de la llave, demostrando ser AES el más eficiente.

Tamaño de Llave	Posibles combinaciones
56-bit (DES)	$7.2x10^{9}$
128 bit (AES)	$3.4x10^{38}$
128 bit (AES)	$6.2x10^{57}$
128 bit (AES)	$1.1x10^{77}$

Tabla 14: Posibles combinaciones entre DES y AES

Fuente: Elaboración Propia

A continuación, se muestra una tabla comparativa entre los algoritmos de cifrado asimétricos, tomando en cuenta los parámetros de generación de llave, cantidad de recursos utilizados en los ciclos del CPU, así como el tamaño de la llave privada y pública, se puede apreciar que RSA es el más eficiente en relación con la seguridad en el cifrado de datos y el uso óptimo de recursos, del mismo modo la generación de un par de llaves robustas tanto privada como pública.

Security level (power of 2)	Algorithm	KeyGen	Sign 59 bytes (CPU cycles)	Verify (CPU cycles)	Private key size (bytes)	Public key size (bytes)	Signature size (bytes)
80	RSA1024	102,869,553	2,213,112	60,084	1024	128	128
	ECDSA160	1,201,188	944,364	1,083,060	60	40	40
	MQQSIG160	799,501,482	6,534	92,232	401	137,408	40
	RainbowBinary256181212	30,311,648	38,784	43,800	23,408	30,240	42
96	RSA1536	322,324,721	5,452,076	87,516	1536	192	192
	ECDSA192	1,799,284	1,390,560	1,662,664	72	48	48
	MQQSIG192	800,724,096	7,938	138,972	465	222,360	48
112	RSA2048	786,466,598	11,020,696	125,776	2048	256	256
	ECDSA224	2,022,896	1,555,740	1,821,348	84	56	56
	MQQSIG224	1,107,486,126	9,492	184,392	529	352,828	56
128	RSA3072	2,719,353,538	31,941,760	230,536	3072	384	384
	ECDSA256	2,296,976	1,780,524	2,085,588	96	64	64
	MQQSIG256	1,501,955,022	9,138	218,700	593	526,368	64
	TTS6440	60,827,704	84,892	76,224	16,608	57,600	43
	3ICP	15,520,100	1,641,032	60,856	12,768	35,712	36

Tabla 15: Tabla comparativa de algoritmos asimétricos

Fuente: Elaboración Propia

6.1 Conexión Inversa de Nodos Confiables con RSA y AES-256

A continuación se observa la autenticación entre nodos antes de ser guardados en la base de datos, esta autenticación consiste en la conexión remota entre servidores confiables, estos generan una llave privada y una llave pública ambas RSA, esto garantizara la seguridad en el intercambio de mensajes, debido a que los algoritmos asimétricos son computacionalmente muy costosos en el primer mensaje protegido por RSA se intercambiara una llave de cifrado AES-256, esta llave será utilizada para garantizar las siguientes conexiones, la autenticación entre estos dos nodos se da generando una llave AES de 32 bits totalmente aleatoria más un hash generado con el algoritmo SHA-1 el cual es aplicado directamente a la dirección física (MAC), estos dos se juntan y son transmitidos bajo RSA, el servidor al recibir esta cadena sabe los parámetros específicos, es decir el rango en el que se encuentra el hash de autenticación del nodo el cual es comparado en la base de datos para poder autenticarlo y realizar una conexión inversa para poder tomar control o transferir archivos, etc. Una vez autenticado se procede al almacenamiento de la clave AES-256 para poder realizar las siguientes conexiones como una *Shell* inversa, todo lo mencionado anteriormente está gestionado por un script desarrollado en Python 2.7, se puede observar esta autenticación en la figura 140 donde se observa las llaves pública y privada, el mensaje cifrado sobre RSA, el mismo mensaje descifrado en el servidor y la división del hash del nodo confiable para la autenticación así como clave AES-256 a utilizarse en la próxima conexión, en esta figura también se puede observar el apartado de "[*shell*]", el mismo que representa una conexión inversa establecida mediante la llave AES-256, la cual fue compartida en la

primera conexión con RSA, se observa la ejecución de comandos que brinda la información de otra máquina a la que se encuentra conectada como lo demuestra la figura 141, en este caso la conexión inversa se da con las credenciales de ejecución actual, es decir como super usuario del sistema (root), es por ello que antes de aceptar la conexión inversa son comprobadas las credenciales de seguridad, esto garantiza que solo los nodos confiables serán capaces de conectarse de manera remota para realizar distintas operaciones, las mismas que serán almacenadas en un log guardando el usuario que se conectó así como la dirección desde donde se ejecutaron dichos comandos, estos nodos confiables son almacenados en una base de datos con el motivo de que ante cualquier falla por la cual se caigan los enlaces, al momento de reestablecerse la comunicación estos nodos puedan autenticarse y levantar los servicios de la amanera más eficiente posible.

Figura 140: Conexión inversa con RSA y AES-256

Fuente: Elaboración Propia

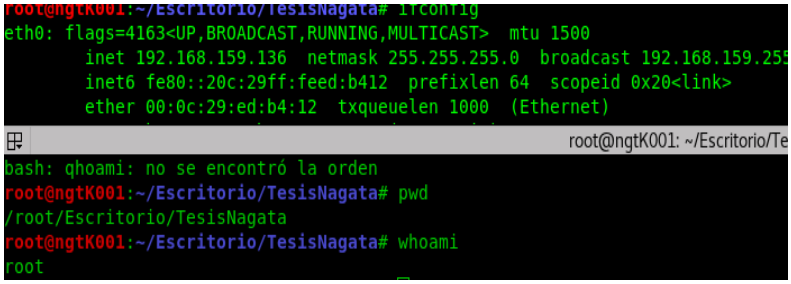


Figura 141: Confirmación de comandos remotos

Fuente: Elaboración Propia

A continuación, se observa el intercambio de clave pública RSA, el cual no existe ningún problema en que sea obtenida por otros usuarios debido a que puede ser conocida por todos.

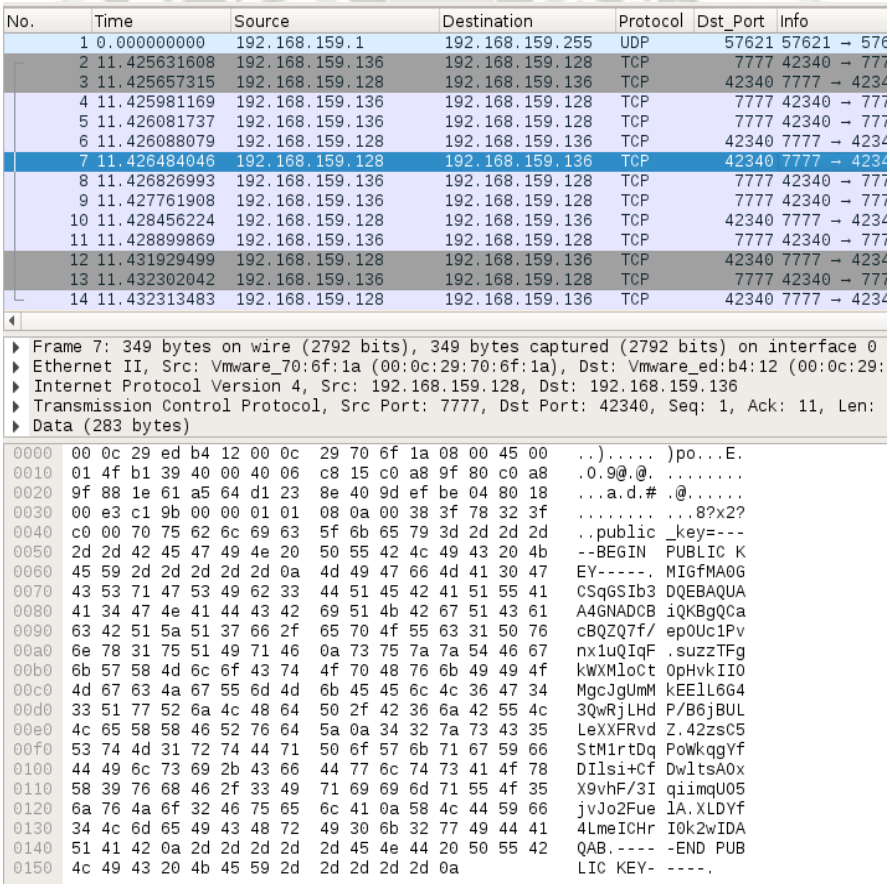
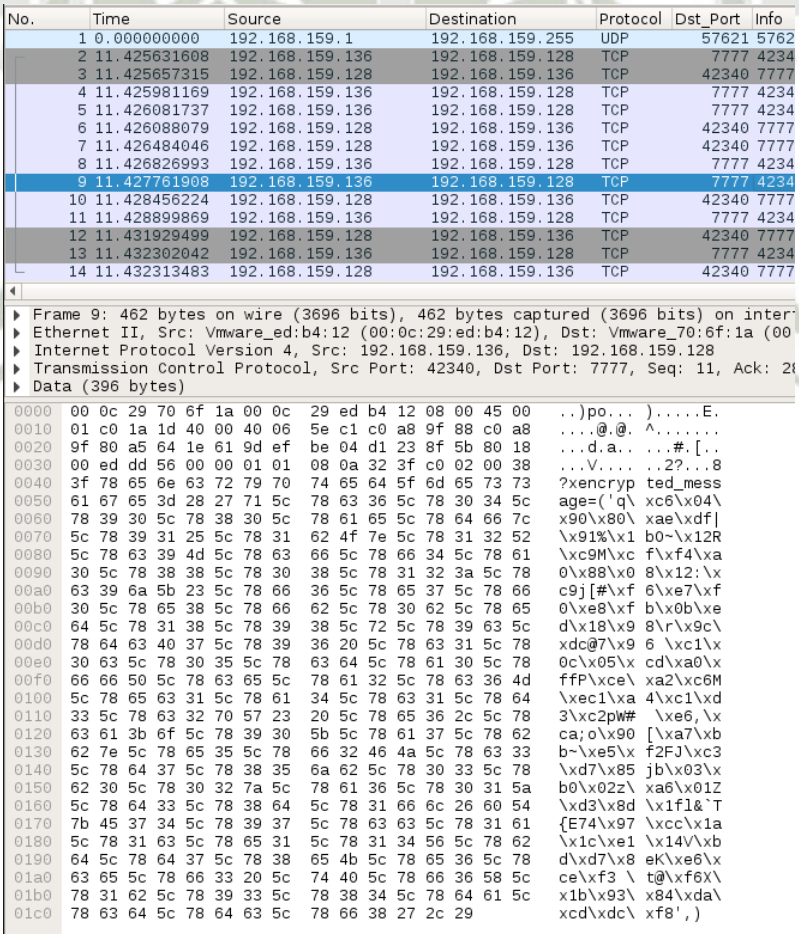


Figura 142: Captura de traza de envió de llave pública

Fuente: Elaboración Propia

En la siguiente figura se observa el tráfico generado en la autenticación de nodos, esta figura puede simular un ataque de hombre al medio(*MITM*), es decir en el caso que se logre intervenir y comprometer la comunicación de autenticación entre nodos lo que se obtendría seria el intercambio de información que se encuentra cifrado por RSA, el mismo que solo se podrá descifrar con la llave privada del servidor, esto se puede observar en la trama seleccionada(9), el cual tiene la conexión validad por RSA desde el nodo 192.168.159.136 con destino al nodo 192.168.159.128, cuyo contenido de dicha conexión es ilegible para alguien que no tenga la llave privada, la cual solo es conocida por el servidor.



No.	Time	Source	Destination	Protocol	Dst_Port	Info
1	0.000000000	192.168.159.1	192.168.159.255	UDP	57621	5762
2	11.425631608	192.168.159.136	192.168.159.128	TCP	7777	4234
3	11.425657315	192.168.159.128	192.168.159.136	TCP	42340	7777
4	11.425981169	192.168.159.136	192.168.159.128	TCP	7777	4234
5	11.426081737	192.168.159.136	192.168.159.128	TCP	7777	4234
6	11.426088079	192.168.159.128	192.168.159.136	TCP	42340	7777
7	11.426484046	192.168.159.128	192.168.159.136	TCP	42340	7777
8	11.426826993	192.168.159.136	192.168.159.128	TCP	7777	4234
9	11.427761908	192.168.159.136	192.168.159.128	TCP	7777	4234
10	11.428456224	192.168.159.128	192.168.159.136	TCP	42340	7777
11	11.428899869	192.168.159.136	192.168.159.128	TCP	7777	4234
12	11.431929499	192.168.159.128	192.168.159.136	TCP	42340	7777
13	11.432302042	192.168.159.136	192.168.159.128	TCP	7777	4234
14	11.432313483	192.168.159.128	192.168.159.136	TCP	42340	7777

```
▸ Frame 9: 462 bytes on wire (3696 bits), 462 bytes captured (3696 bits) on inter
▸ Ethernet II, Src: Vmware_ed:b4:12 (00:0c:29:ed:b4:12), Dst: Vmware_70:6f:1a (00
▸ Internet Protocol Version 4, Src: 192.168.159.136, Dst: 192.168.159.128
▸ Transmission Control Protocol, Src Port: 42340, Dst Port: 7777, Seq: 11, Ack: 2
▸ Data (396 bytes)

0000  00 0c 29 70 6f 1a 00 0c  29 ed b4 12 08 00 45 00   ..)po... ).....E.
0010  01 c0 1a 1d 40 00 40 06  5e c1 c0 a8 9f 88 c0 a8   ....@.@. ^.......
0020  9f 80 a5 64 1e 61 9d ef  be 04 d1 23 8f 5b 80 18   ...d.a.. ...#.[..
0030  00 ed dd 56 00 00 01 01  08 0a 32 3f c0 02 00 38   ...V.... ..2?...8
0040  3f 78 65 6e 63 72 79 70  74 65 64 5f 6d 65 73 73   ?xencryp ted_mess
0050  61 67 65 3d 28 27 71 5c  78 63 36 5c 78 30 34 5c   age=('q\ xc6\x04\
0060  78 39 30 5c 78 38 30 5c  78 61 65 5c 78 64 66 7c   x90\x80\ xae\xdf|
0070  5c 78 39 31 25 5c 78 31  62 4f 7e 5c 78 31 32 52   \x91%\x1 bO~\x12R
0080  5c 78 63 39 4d 5c 78 63  66 5c 78 66 34 5c 78 61   \xc9M\xc f\xf4\xa
0090  30 5c 78 38 38 5c 78 30  38 5c 78 31 32 3a 5c 78   0\x88\x0 8\x12:\x
00a0  63 39 6a 5b 23 5c 78 66  36 5c 78 65 37 5c 78 66   c9j[#\xf 6\xe7\xf
00b0  30 5c 78 65 38 5c 78 66  62 5c 78 30 62 5c 78 65   0\xe8\xf b\x0b\xe
00c0  64 5c 78 31 38 5c 78 39  38 5c 72 5c 78 39 63 5c   d\x18\x9 8\r\x9c\
00d0  78 64 63 40 37 5c 78 39  36 20 5c 78 63 31 5c 78   xdc@7\x9 6 \xc1\x
00e0  30 63 5c 78 30 35 5c 78  63 64 5c 78 61 30 5c 78   0c\x05\x cd\xa0\x
00f0  66 66 50 5c 78 63 65 5c  78 61 32 5c 78 63 36 4d   ffP\xce\ xa2\xc6M
0100  5c 78 65 63 31 5c 78 61  34 5c 78 63 31 5c 78 64   \xec1\xa 4\xc1\xd
0110  33 5c 78 63 32 70 57 23  20 5c 78 65 36 2c 5c 78   3\xc2pW#  \xe6,\x
0120  63 61 3b 6f 5c 78 39 30  5b 5c 78 61 37 5c 78 62   ca;o\x90 [\xa7\xb
0130  62 7e 5c 78 65 35 5c 78  66 32 46 4a 5c 78 63 33   b~\xe5\x f2FJ\xc3
0140  5c 78 64 37 5c 78 38 35  6a 62 5c 78 30 33 5c 78   \xd7\x85 jb\x03\x
0150  62 30 5c 78 30 32 7a 5c  78 61 36 5c 78 30 31 5a   b0\x02z\ xa6\x01Z
0160  5c 78 64 33 5c 78 38 64  5c 78 31 66 6c 26 60 54   \xd3\x8d \x1fl&`T
0170  7b 45 37 34 5c 78 39 37  5c 78 63 63 5c 78 31 61   {E74\x97 \xcc\x1a
0180  5c 78 31 63 5c 78 65 31  5c 78 31 34 56 5c 78 62   \x1c\xe1 \x14V\xb
0190  64 5c 78 64 37 5c 78 38  65 4b 5c 78 65 36 5c 78   d\xd7\x8 eK\xe6\x
01a0  63 65 5c 78 66 33 20 5c  74 40 5c 78 66 36 58 5c   ce\xf3 \ t@\xf6X\
01b0  78 31 62 5c 78 39 33 5c  78 38 34 5c 78 64 61 5c   x1b\x93\ x84\xda\
01c0  78 63 64 5c 78 64 63 5c  78 66 38 27 2c 29         xcd\xdc\ xf8',)
```

Figura 143: Captura de traza cifrada de envió de llave AES-256 mediante RSA

Fuente: Elaboración Propia

6.2 Pruebas de Seguridad Contra Impacto de Ataques por Capas.

Una vez implementado el certificado de seguridad (SSL), se comprobará que las conexiones de origen y destino se darán de manera cifrada, para esto se observa en la siguiente figura que al reestablecer el servicio de "apache2", este pide la contraseña dada en la configuración del certificad de seguridad (SSL), el mismo que habilitará conexiones en el puerto 443 estableciendo las conexiones a la web de manera segura ya que el tráfico se encontrará de manera cifrada, siendo ilegible en tránsito.

Figura 144: Restablecimiento de servicio apache con SSL

Fuente: Elaboración Propia

En la siguiente figura se observa la confirmación del navegador que se encuentra en una conexión segura, el mismo que contiene un certificado verificado por "UCSMnagata"

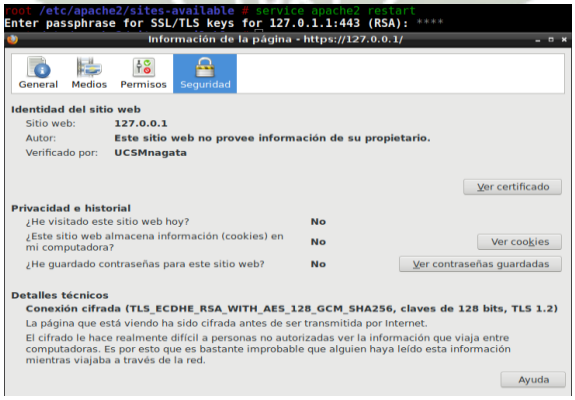


Figura 145: Confirmación de certificado de seguridad

Fuente: Elaboración Propia

A continuación, se observa en la siguiente figura una conexión al sitio web bajo el certificado de seguridad implementado, se observa la origen y destino de los puertos de la conexión del servidor, siendo estos el puerto 443, se observa también que todo el tráfico de esta conexión se encuentra cifrado, por lo que aun siendo interceptado no será legible.

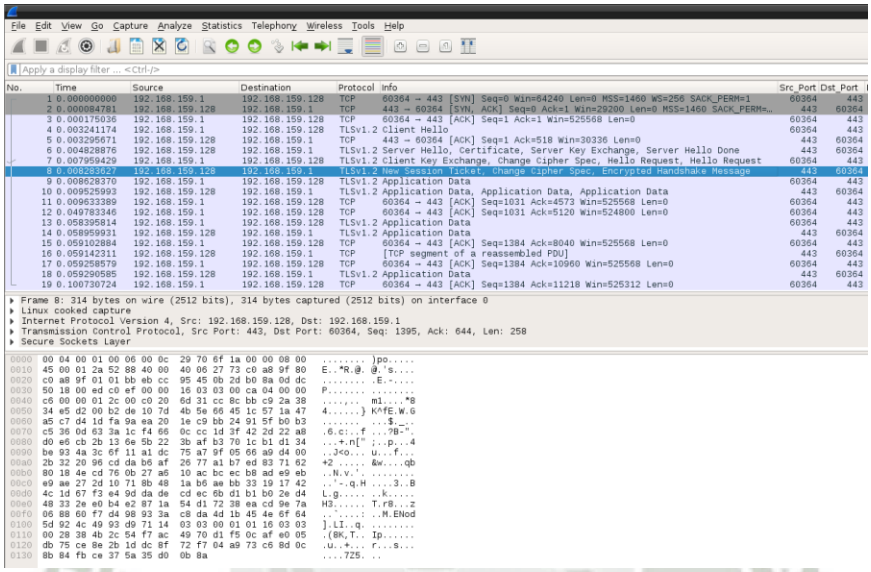


Figura 146: Captura de paquete con certificado de seguridad SSL

Fuente: Elaboración Propia

En la siguiente figura se observa un ataque realizado contra el host configurado para mantener la capa de seguridad de las comunicaciones, se observa que se realiza un ataque *flood*, el mismo que falsificará direcciones IP para simular un desbordamiento en la cantidad de solicitudes al servidor.

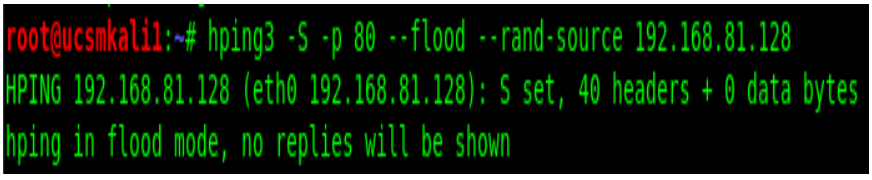


Figura 147: Ataque flood con hping para pruebas de seguridad

Fuente: Elaboración Propia

En las siguientes figuras se observa el impacto del ataque de *flood* mencionado anteriormente, el mismo que es detectado y se establece la regla de "*drop*", la misma que se encargará de eliminar dichas conexiones maliciosas, esto puede observarse en la figura 148 y 149.

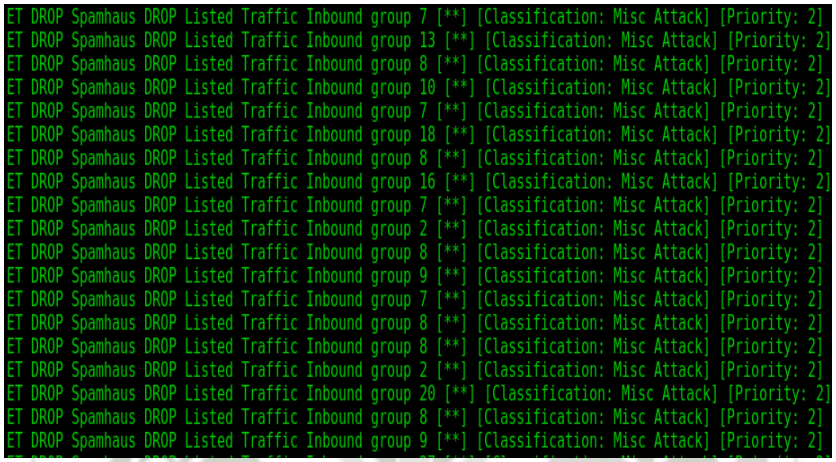


Figura 148: Detección de tráfico malicioso y eliminación de conexión parte I

Fuente: Elaboración Propia

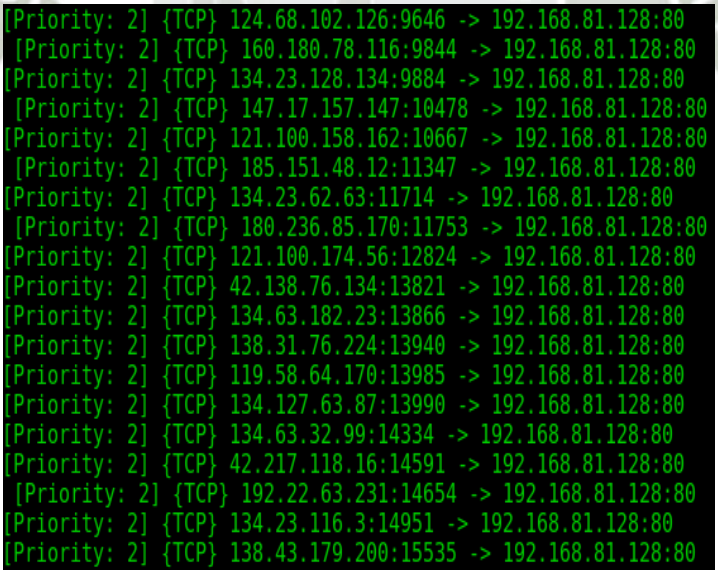


Figura 149: Detección de tráfico malicioso y eliminación de conexión parte II

Fuente: Elaboración Propia

En la siguiente figura 150 se pude observar al sistema de detección de intrusos descubriendo nuevos nodos conectados a la red, los cuales tendrán que pasar por la arquitectura de seguridad propuesta para poder realizar peticiones y conexiones tanto a servicios que se encuentren dentro como fuera de la red, se observa que el descubrimiento otorga información tanto de la dirección IP, así como la dirección física (MAC), y un descriptor del nodo, en este caso los nodos conectados son dispositivos de la marca "Dell", en este descubrimiento también se realiza la asociación de nombres de los nodos a sus direcciones IP, esto se puede apreciar en la figura 151, esta información puede ayudar a monitorizar y detectar con mayor precisión la ubicación del nodo que podría estar realizando algún tipo de actividad maliciosa.

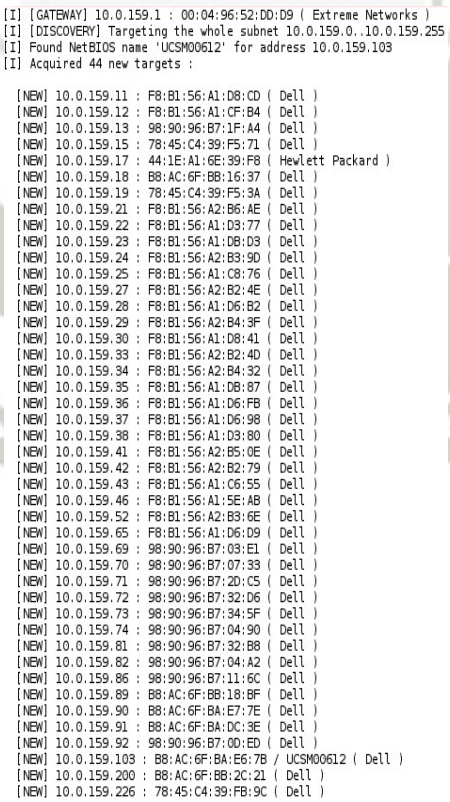


Figura 150: Detección de direcciones IP y direcciones MAC de nodos en red

Fuente: Elaboración Propia

```
[I] Found NetBIOS name 'UCSM00616' for address 10.0.159.19
[I] Found NetBIOS name 'UCSM' for address 10.0.159.34
[I] Found NetBIOS name 'UCSM00531' for address 10.0.159.25
[I] Found NetBIOS name 'UCSM' for address 10.0.159.43
[I] Found NetBIOS name 'UCSM00594' for address 10.0.159.86
[I] Found NetBIOS name 'UCSM00535' for address 10.0.159.29
[I] Found NetBIOS name 'UCSM00530' for address 10.0.159.24
[I] Found NetBIOS name 'UCSM00533' for address 10.0.159.27
[I] Found NetBIOS name 'UCSM00551' for address 10.0.159.12
[I] Found NetBIOS name 'UCSM00526' for address 10.0.159.11
[I] Found NetBIOS name 'UCSM00559' for address 10.0.159.52
[I] Found NetBIOS name 'UCSM00528' for address 10.0.159.22
[I] Found NetBIOS name 'UCSM00577' for address 10.0.159.69
[I] Found NetBIOS name 'UCSM' for address 10.0.159.42
[I] Found NetBIOS name 'UCSM00541' for address 10.0.159.35
[I] Found NetBIOS name 'UCSM00624' for address 10.0.159.226
[I] Found NetBIOS name 'UCSM00543' for address 10.0.159.37
[I] Found NetBIOS name 'UCSM00536' for address 10.0.159.30
[I] Found NetBIOS name 'DESKTOP-PVN56GG' for address 10.0.159.18
[I] Found NetBIOS name 'UCSM00527' for address 10.0.159.21
[I] Found NetBIOS name 'UCSM00617' for address 10.0.159.200
[I] Found NetBIOS name 'UCSM00579' for address 10.0.159.71
[I] Found NetBIOS name 'UCSM00544' for address 10.0.159.38
[I] Found NetBIOS name 'UCSM' for address 10.0.159.23
[I] Found NetBIOS name 'UCSM00542' for address 10.0.159.36
[I] Found NetBIOS name 'UCSM00553' for address 10.0.159.46
```

Figura 151: Descubrimiento de nombre de nodos para direcciones IP

Fuente: Elaboración Propia

Luego de la detección de los nuevos nodos que pasan sus conexiones por medio de la arquitectura de seguridad propuesta, en la siguiente figura se observa las alertas de intento de conexión con el protocolo UDP, para realizar la resolución de nombres de dominio de "*facebook*", debido a que esta regla de seguridad fue implementada para eliminar este tipo de conexiones la resolución de nombres de dominio no podrá realizarse.

```
[1:100032:0] Facebook DNS [**] [Priority: 0] {UDP} 10.0.159.15:17223 -> 10.0.2.12:53
[1:100032:0] Facebook DNS [**] [Priority: 0] {UDP} 10.0.159.19:54347 -> 10.0.2.12:53
[1:100032:0] Facebook DNS [**] [Priority: 0] {UDP} 10.0.159.15:60835 -> 10.0.2.11:53
[1:100032:0] Facebook DNS [**] [Priority: 0] {UDP} 10.0.159.15:60835 -> 10.0.2.12:53
```

Figura 152: Alertas de conexión sobre reglas de seguridad

Fuente: Elaboración Propia

A continuación se muestra las alertas de seguridad del sistema de detección de intrusos, estas alertas se manifiestan debido a que distintos nodos de la red se encuentran infectados con un troyano de red como se muestra en la siguiente figura, en la misma que se aprecia cómo se envía todo el tráfico generado a un nodo que no es la puerta de enlace de la red, esto es calificado como un ataque de hombre al medio, este intento de ataque fue detectado por el sistema de detección de intrusos que se encuentra dentro de nuestra arquitectura de red, el mismo que elimino dichas conexiones maliciosas.

Figura 153: Detección de ataques de troyanos en red

Fuente: Elaboración Propia

6.3 Análisis de Eficiencia de las Medidas de Seguridad

En la siguiente figura se observa la detección de tráfico anómalo o fuera de control, el mismo que al ser detectado por la arquitectura de seguridad propuesta genera medidas de seguridad ante posibles ataques y nodos maliciosos, estas medidas de seguridad son aplicadas tanto para conexiones internas como externas, se observa que una vez establecidas las medidas de seguridad el tráfico baja rotundamente, la reconexión solo se dará por medio de nodos confiables, los

mismos que aseguraran el funcionamiento de los servicios necesarios para mantener la continuidad del negocio.

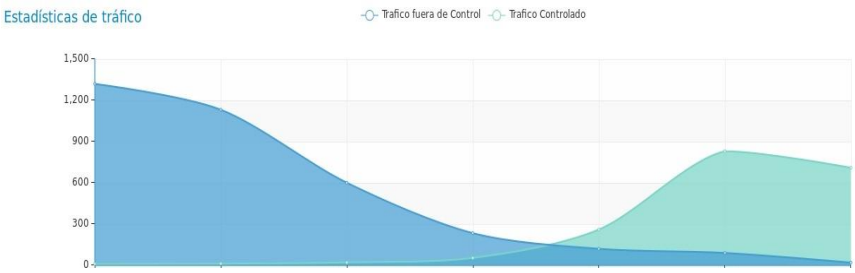


Figura 154: Control de tráfico

Fuente: Elaboración Propia

En la siguiente figura se aprecia el análisis de amenazas de seguridad, las mismas que fueron detectadas y mitigadas por la arquitectura de seguridad propuesta, este análisis aborda tanto el descontrol de tráfico como incidentes menores, nuevos ataques los cuales fueron recientemente actualizados en las políticas de seguridad, así como también los posibles datos sensibles extraídos, esto es posible debido a que suricataIDS, es capaz de guardar un historial de las imágenes que se transmiten como ya se vio en capítulos anteriores.

Figura 155: Análisis de amenazas de seguridad

Fuente: Elaboración Propia

Al igual que la figura anterior, en la siguiente figura se puede observar un análisis detallado expresado de una manera numérica para poder tomar una mayor conciencia sobre los incidentes de seguridad ocurridos.

Figura 156: Análisis de amenazas de seguridad detallado

Fuente: Elaboración Propia

Debido a que los ataques de más alto riesgo se dan dentro de la organización, a continuación, se muestra una comparativa entre los ataques detectados de manera interna frente a los ataques detectados de manera externa, el hecho de poder ofrecer un análisis gráfico facilita el poder tomar medidas de seguridad tanto dentro como fuera de la organización, así como concientizar a los usuarios finales del riesgo al que se encuentran expuestos.

Figura 157: Estadística de ataques internos

Fuente: Elaboración Propia

CONCLUSIONES

a) Se logró diseñar una arquitectura de seguridad capaz de minimizar la superficie de exposición ante la explotación de vulnerabilidades modernas, esta arquitectura logró cubrir. la capa 1(subcapa de enlace de datos), 2, 3 y 4 del modelo TCP/IP empleando el algoritmo de cifrado AES-256 en redes de datos IP.

b) Se logró reconocer las amenazas de seguridad en redes datos, tomando como referencia la metodología *OWASP* para el nivel de la capa de aplicación.

c) Se analizó en su totalidad los métodos y aspectos técnicos de la explotación de vulnerabilidades modernas, realizando auditorías de seguridad con herramientas de *pentesting* contra la arquitectura propuesta.

d) Se analizó minuciosamente la seguridad en cada capa del modelo TCP/IP, identificando todas las amenazas de seguridad en redes de datos, cubriendo en su totalidad las capas de:

   - Enlace de datos
   - Red
   - Transporte
   - Aplicación

e) Se logró dimensionar y desplegar distintos tipos de medidas de seguridad en todas las capas del modelo TCP/IP partiendo de la capa de enlace de datos, garantizando la eficiencia de esta arquitectura al realizar las pruebas de seguridad pertinentes.

f) Se logró garantizar la conectividad únicamente de nodos seguros, mediante una autenticación propuesta en esta investigación considerando la implementación del algoritmo SHA-1 a la dirección física (MAC).

g) Se logró cifrar las comunicaciones entre los nodos confiables, realizando las pruebas pertinentes para validar que la comunicación realiza el intercambio de claves AES-256, el mismo que va cifrado por RSA en la primera conexión.

h) Se comprobó el cifrado de la transmisión de datos de los nodos confiables desde el primer intercambio de información, por ello se demuestra la seguridad total en las capas de enlace de datos, red y transporte, debido a que no se utiliza la capa de aplicación en este punto.

RECOMENDACIONES

a) Establecer doble factor de autenticación en cuentas de usuarios.

b) Eliminar IPv6 si no se está utilizando, debido a que este protocolo tiene nuevas brechas de seguridad.

c) Revisión de la configuración de los *switches* que trabajan en la capa de enlace de datos para revisar las configuraciones de seguridad, implementar SPB que ofrece una mayor seguridad y eficiencia en vez de STP.

d) Actualización periódica de todo el software para evitar fallos de seguridad ya solucionados.

e) Actualización periódica de las políticas de seguridad, incluyendo políticas de BYOD (trae tu propio dispositivo), ya que estos dispositivos se suelen conectar a la red interna.

f) Preparar la arquitectura de red para poder migrar a redes SDN, sin perder la seguridad en la misma.

g) Concientización a los usuarios que utilizan los distintos sistemas informáticos al riesgo de seguridad al que se encuentran expuestos.

REFERENCIAS BIBLIOGRÁFICAS

Arjuman, N. C., & Manickam, S. (2015). A review on ICMPv6 vulnerabilities and its mitigation techniques: Classification and art. *I4CT 2015 - 2015 2nd International Conference on Computer, Communications, and Control Technology, Art Proceeding*, (I4ct), 323–327. https://doi.org/10.1109/I4CT.2015.7219590

Arzhakov, A. V., & Silnov, D. S. (2017). Architecture of multithreaded network scanner. *International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices, EDM*, 43–45. https://doi.org/10.1109/EDM.2017.7981704

Atwell, C., Blasi, T., & Hayajneh, T. (2016). Reverse TCP and Social Engineering Attacks in the Era of Big Data. *Proceedings - 2nd IEEE International Conference on Big Data Security on Cloud, IEEE BigDataSecurity 2016, 2nd IEEE International Conference on High Performance and Smart Computing, IEEE HPSC 2016 and IEEE International Conference on Intelligent Data and S*, 90–95. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.60

Badea, A., Croitoru, V., & Gheorghica, D. (2015). Computer network vulnerabilities and monitoring. *2015 9th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2015*, 49–54. https://doi.org/10.1109/ATEE.2015.7133678

Barbhuiya, F. A., Gupta, V., Biswas, S., & Nandi, S. (2012). Detection and mitigation of induced low rate TCP-targeted denial of service attack. *Proceedings of the 2012 IEEE 6th International Conference on Software Security and Reliability, SERE 2012*, 291–300. https://doi.org/10.1109/SERE.2012.27

Bhosale, D. A., & Mane, V. M. (2015). Comparative study and analysis of network intrusion detection tools. *2015 International Conference on Applied and Theoretical*

Computing and Communication Technology (ICATccT), 312–315. https://doi.org/10.1109/ICATCCT.2015.7456901

Bobade, S., & Goudar, R. (2015). Secure data communication using protocol steganography in IPv6. *Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015*, (D), 275–279. https://doi.org/10.1109/ICCUBEA.2015.59

Carlini, A. (2016). Ciberseguridad: un nuevo desafío para la comunidad internacional. *REVISTA DEL INSTITUTO ESPAÑOL DE ESTUDIOS ESTRATÉGICOS*, 1–16. Retrieved from http://www.ieee.es/Galerias/fichero/docs_opinion/2016/DIEEEO67-2016_Ciberseguridad_Desafio_ComunidadInt_ACarlini.pdf

Carrillo, M. R. (2016). El concepto de arma cibernética en el marco internacional: una aproximación funcional. *REVISTA DEL INSTITUTO ESPAÑOL DE ESTUDIOS ESTRATÉGICOS*, 1–19. Retrieved from http://www.ieee.es/Galerias/fichero/docs_opinion/2016/DIEEEO101-2016_Arma_Cibernetica_MargaritaRobles.pdf

Centeno, F. J. U. (2015). Opinión. *Instituto Español de Estudios Estratégicos*, 1–18. Retrieved from http://www.ieee.es/Galerias/fichero/docs_opinion/2015/DIEEEO09-2015_AmenazaCiberataques_Fco.Uruena.pdf

Chakraborty, M., Chaki, N., & Cortesi, A. (2014). A New Intrusion Prevention System for Protecting Smart Grids from ICMPv6 Vulnerabilities, *2*, 1539–1547. https://doi.org/10.15439/2014F287

Chuan, X., Yan, Y., & Zhang, Y. (2017). An efficient triggering method of hardware Trojan in AES cryptographic circuit. *2017 2nd IEEE International Conference on Integrated Circuits and Microsystems (ICICM)*, 91–95. https://doi.org/10.1109/ICAM.2017.8242145

Dwi, K., & Utama, B. (2017). Digital Signature using MAC Address based AES- 128 and SHA-2 256-bit. *International Seminar on Application for Technology of Information and Communication (ISemantic) Digital*, 72–78.

Fedorchenko, A., Kotenko, I., & Chechulin, A. (2015). Design of integrated vulnerabilities database for computer networks security analysis. *Proceedings - 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2015*, 559–566. https://doi.org/10.1109/PDP.2015.38

Gudipati, V. K., Venna, T., Subburaj, S., & Abuzaghleh, O. (2017). Advanced automated SQL injection attacks and defensive mechanisms. *2016 Annual Connecticut Conference on Industrial Electronics, Technology and Automation, CT-IETA 2016*. https://doi.org/10.1109/CT-IETA.2016.7868248

Hong, H., Choi, H., Kim, D., Kim, H., Hong, B., Noh, J., & Kim, Y. (2017). When Cellular Networks Met IPv6: Security Problems of Middleboxes in IPv6 Cellular Networks. *Proceedings - 2nd IEEE European Symposium on Security and Privacy, EuroS and P 2017*, 595–609. https://doi.org/10.1109/EuroSP.2017.34

Irfan, L., & Mahendra, B. (2017). Enhanced AES using MAC Address for Cloud Services, 66–71.

Kalwar, S., Bohra, N., & Memon, A. A. (2015). A survey of transition mechanisms from IPv4 to IPv6 - Simulated test bed and analysis. *2015 3rd International Conference*

on Digital Information, Networking, and Wireless Communications, DINWC 2015, 30–34. https://doi.org/10.1109/DINWC.2015.7054212

Kamaldeep, Malik, M., & Dutta, M. (2018). Implementation of single-packet hybrid IP traceback for IPv4 and IPv6 networks. *IET Information Security*, *12*(1), 1–6. https://doi.org/10.1049/iet-ifs.2015.0483

Kao, D., Wang, Y., Tsai, F., & Chen, C. (2018). Forensic Analysis of Network Packets from Penetration Test Toolkits, 363–368.

Kaushik, A. K., & Joshi, R. C. (2010). Network Forensic System for ICMP Attacks. *International Journal of Computer Applications*, *2*(3), 14–21. https://doi.org/10.5120/649-906

Kumar, P. (2017). Enhanced Cloud Data Security Using AES Algorithm.

Kyatam, S., Alhayajneh, A., & Hayajneh, T. (2017). Heartbleed attacks implementation and vulnerability. *2017 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2017*. https://doi.org/10.1109/LISAT.2017.8001980

Lootah, W., Enck, W., & Mcdaniel, P. (2005). TARP : Ticket-based Address Resolution Protocol, (Acsac).

Makino, Y., & Klyuev, V. (2015). Evaluation of web vulnerability scanners. *Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015*, *1*(September), 399–402. https://doi.org/10.1109/IDAACS.2015.7340766

Mamdouh, M., Ghoz, H. El, & Far, Y. El. (2017). A new dynamic secured IEEE 802 . 11e AES based system, 101–107. https://doi.org/10.1109/CICN.2017.24

Najjar, F., & Kadhum, M. M. (2015). Reliable behavioral dataset for IPv6 neighbor discovery protocol investigation. *2015 5th International Conference on IT Convergence and Security, ICITCS 2015 - Proceedings*. https://doi.org/10.1109/ICITCS.2015.7293014

Nakhla, N., Perrett, K., & McKenzie, C. (2017). Automated computer network defence using ARMOUR: Mission-oriented decision support and vulnerability mitigation. *2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment, Cyber SA 2017*. https://doi.org/10.1109/CyberSA.2017.8073389

Narayan, S., Gupta, R., Kumar, A., Ishrar, S., & Khan, Z. (2016). Cyber security attacks on network with transition mechanisms. *2015 International Conference on Computing and Network Communications, CoCoNet 2015*, 163–169. https://doi.org/10.1109/CoCoNet.2015.7411182

Nastase, L. (2017). Security in the Internet of Things: A Survey on Application Layer Protocols. *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, 659–666. https://doi.org/10.1109/CSCS.2017.101

Ouseph, C. (2016). Prevention of MITM attack caused by Rogue Router Advertisements in IPv6, 952–956.

Praptodiyono, S., Hasbullah, I. H., Kadhum, M. M., Murugesan, R. K., Wey, C. Y., & Osman, A. (2015). Improving Security of Duplicate Address Detection on IPv6 Local Network in Public Area. *2015 9th Asia Modelling Symposium (AMS)*, 123–128. https://doi.org/10.1109/AMS.2015.28

Rafiqul Zaman Khan, & Atena Shiranzaei. (2016). IPv6 Security Tools - A Systematic Review, *202002*, 459–464.

Raviyarupal, D., & Kumar, H. (2016). Detection and Prevention of ARP Poisoning in Dynamic IP configuration, 1240–1244.

Rohrmann, R. R., Ercolani, V. J., & Patton, M. W. (2017). Large Scale Port Scanning Through Tor: Using Parallel Nmap Scans to Scan Large Portions of the IPv4 Range, 185–187.

Sánchez, J. R. P. (2015). Analysis of the Security IPv6 and Comparative Study between Two Routing Protocols Oriented to IPv6. *Proceedings - 2015 Asia-Pacific Conference on Computer-Aided System Engineering, APCASE 2015*, 374–379. https://doi.org/10.1109/APCASE.2015.73

Schindler, S., Schnor, B., & Scheffler, T. (2015). Taming the IPv6 address space with Hyhoneydv6. *2015 World Congress on Internet Security, WorldCIS 2015*, 113–118. https://doi.org/10.1109/WorldCIS.2015.7359425

Shah, J. L., & Parvez, J. (2015). Impact of IPSec on Real Time applications in IPv6 and 6to4 Tunneled Migration Network. *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 0–5. https://doi.org/10.1109/ICIIECS.2015.7193114

States, U. (2016). Analysing Denial of Service Attack Traffic Signature in IPv6 Local Network using Correlation Inspection, (December), 1008–1013.

Terli, V. K. K., Chaganti, S. P., Alla, N. B., Sarab, S., & El Taeib, T. (2016). Software implementation of IPv4 to IPv6 migration. *2016 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2016*. https://doi.org/10.1109/LISAT.2016.7494160

Tian, D. J., Butler, K. R. B., Choi, J. I., McDaniel, P., & Krishnaswamy, P. (2017).

Securing ARP/NDP from the Ground Up. *IEEE Transactions on Information Forensics and Security*, *12*(9), 2131–2143. https://doi.org/10.1109/TIFS.2017.2695983

Vincent Nicolls, Nhien-An Le-Khac, Lei Chen, M. S. (2016). IPv6 security and DNS security, 743–748.

Wu, F., Wang, J., Liu, J., & Wang, W. (2017). Vulnerability Detection with Deep Learning, 1298–1302.

Xiaorong, F., Jun, L., & Shizhun, J. (2013). Security analysis for IPv6 neighbor discovery protocol. *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, 303–307. https://doi.org/10.1109/IMSNA.2013.6743275

Zalbina, M. R., Septian, T. W., Stiawan, D., Idris, M. Y., Heryanto, A., & Budiarto, R. (2017). *Payload* recognition and detection of Cross Site Scripting attack. *2017 2nd International Conference on Anti-Cyber Crimes, ICACC 2017*, 172–176. https://doi.org/10.1109/Anti-Cybercrime.2017.7905285

Zhang, T., & Wang, Z. (2017). Research on IPv6 Neighbor Discovery Protocol (NDP) security. *2016 2nd IEEE International Conference on Computer and Communications, ICCC 2016 - Proceedings*, 2032–2035. https://doi.org/10.1109/CompComm.2016.7925057

Zhao, Y., Lei, Y., Yang, T., & Cui, Y. (2013). A new strategy to defense against SSLStrip for Android. *International Conference on Communication Technology Proceedings, ICCT*, 70–74. https://doi.org/10.1109/ICCT.2013.6820349

ANEXOS

ANEXO A

GLOSARIO DE TÉRMINOS

ACL	Access control list
AES-256	Advanced Encryption Standard (256 bits)
ARP	Address Resolution Protocol
CGA	Cryptographically Generated Address
CIA	Confidencialidad/Integridad/Disponibilidad
CPA	Certification Path Advertisement
CPS	Certification Path Solicitation
DAD	Duplicate Address Detection
DAI	Dynamic ARP Inspection
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuración Protocol Versión 6
DMZ	Demilitarized zone
DNS	Domain Name System
DOS	Denegación de servicios
ECP	Encryption control protocol
FTP	Protocolo de transferencia de archivos
HONEYPOT	Sistema trampa o señuelo

HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol Overview.
HTTPS	Hypertext Transport Protocol Secure
ICMP	Internet Control Message Protocol
ICMPv6	Address Resolution Protocol Version 6
IDS/IPS	Intrusion Detection System/Intrusion Prevention System
IKE	Internet key exchange
IPSEC	Internet Protocol Security
IPTABLES	Firewall integrado en el kernel de Linux
IPv4/IPv6	Internet Protocol version 4/Internet Protocol version 6
L2f	Layer Tow Forwarding
L2tp	Layer 2 Tunneling Protocol
LDAP	Lightweight Directory Access Protocol
LTA	Local Ticket Agent
MAC ACL	Lista de control de acceso basada en direcciones físicas
MAC	Media Access Control
MITM	Man in the middle
NAT	Traducción de direcciones de red
NDP	Neighbor Discovery Protocol
NUD	Neighbor Unreachability Detection

OWASP	Open Web Application Security Project
PPP	Point to Point protocol
PPTP	Point To Point Tunneling Protocol
RARP	Reverse ARP
RIP	Routing Information Protocol
RSA	Rivest, Shamir y Adleman (sistema criptográfico de clave pública)
SA	Asociación de Seguridad
SCTP	Stream Control Transmission Protocol
SDN	Software Defined Networking
SEND	Secure Neighbor Discovery
SHA	Dirección de hardware del remitente
SLAAC	Stateless Address Auto configuration
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SPB	Shortest Path Bridging
SPI	Security Parameters Index
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP/IP	Protocolo de Control de Transmisión/Protocolo de Internet
TFTP	Trivial File Transfer Protocol

TFTP	Trivial File Transfer Protocol
THA	Dirección de hardware de destino
TLS	Seguridad de la capa de transporte
TTL	Time To Live
UDP	Protocolo de datagrama de usuario
VLAN	Virtual Local Area Network

ANEXO B

CÓDIGO FUENTE DE ACTUALIZACION DE RIESGOS

```
'''
@author: Nagata
'''
# -*- coding: utf-8 -*-
import wget
import hashlib
import psycopg2
import time
import os.path

def DescargarActualizar(RepositorioWeb):
    filename=wget.download(RepositorioWeb)
    print "Descarga completa"


def InsertarBaseRiesgo(Archivo,Actualizar):
    try:
        conn = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
    except:
        print "Error de Base de datos"

    if(Archivo=="hosts"):
        if(Actualizar==0):
            fh = open(Archivo+'.txt')
        else:
            fh = open(Archivo+' (1).txt')

        for line in fh:
            if(line.startswith("127")):
                direccion=(line.split())
                if(direccion[1]=='localhost')or(direccion[1]=='127.0.0.1'):
                    pass;
                else:
                    cur = conn.cursor()
                    ##
                    if(Actualizar==1):
                        cur.execute("SELECT hosts_direccion FROM hosts WHERE hosts_direccion ='"+direccion[1]+"' ")
                        if(cur.fetchone()==None):
                            print"Insertando nuevo registro -> "+direccion[1]
                            cur.execute("INSERT INTO hosts (hosts_direccion)  VALUES('"+direccion[1]+"')")
                            conn.commit();
                        else:
                            pass;
                    else:
                        cur.execute("INSERT INTO hosts (hosts_direccion)  VALUES('"+direccion[1]+"')")
                        conn.commit()
            else:
                pass;
        fh.close()
        print "Hosts Maliciosos actualizados"

    if(Archivo=="ip"):
        if(Actualizar==0):
            fh = open(Archivo+'.txt')
        else:
            fh = open(Archivo+' (1).txt')

        for line in fh:
            if(line=='localhost')or(line=='127.0.0.1'):
                    pass;
            else:
                direccion=line.split()
                cur = conn.cursor()
                if(Actualizar==1):
                    cur.execute("SELECT ip_direccion FROM ip WHERE ip_direccion ='"+direccion[0]+"' ")
                    if(cur.fetchone()==None):
                        print"Insertando nuevo registro -> "+direccion[0]
                        cur.execute("INSERT INTO ip (ip_direccion)  VALUES('"+direccion[0]+"')")
                        conn.commit();
                    else:
                        pass;
                else:
                    cur.execute("INSERT INTO ip (ip_direccion)  VALUES('"+direccion[0]+"')")
                    conn.commit()
        fh.close()
        print "Direcciones IP Maliciosas Actualizadas"

    if(Archivo=="delisted"):
        if(Actualizar==0):
            fh = open(Archivo+'.txt')
        else:
            fh = open(Archivo+' (1).txt')

        for line in fh:
            if(line=='localhost')or(line=='127.0.0.1'):
                    pass;
            else:
                direccion=line.split()
                cur = conn.cursor()
                if(Actualizar==1):
                    cur.execute("SELECT deslisted_direccion FROM deslisted WHERE deslisted_direccion ='"+direccion[0]+"' ")
                    if(cur.fetchone()==None):
                        print"Insertando nuevo registro -> "+direccion[0]
                        cur.execute("INSERT INTO deslisted (deslisted_direccion)  VALUES('"+direccion[0]+"')")
                        conn.commit();
                    else:
```

```
                    pass;
                cur.execute("INSERT INTO deslisted (deslisted_direccion)  VALUES('"+direccion[0]+"')")
                conn.commit()
        fh.close()
        print("Actualizando lista de nodos sospechosos")

def filemd5(filename, block_size=2**20):
    f = open(filename)
    md5 = hashlib.md5()
    while True:
        data = f.read(block_size)
        if not data:
            break
        md5.update(data)
    f.close()
    print "Checksum -> "+md5.hexdigest()
    return md5.digest()

def VerificarActualizacion(Archivo):

    if(os.path.isfile(Archivo+' (1).txt')):
        a = filemd5(Archivo+'.txt')
        b = filemd5(Archivo+' (1).txt')
        if(a==b):
            print "No se registraron actualizaciones para "+Archivo+" -> "+ time.strftime("%c")
        else:
            print "Actualizaciones detectadas..."
            InsertarBaseRiesgo(Archivo,1)
    else:
        print "Esta es la primera insercion de "+Archivo;
        InsertarBaseRiesgo(Archivo,0)


if __name__ == "__main__":

    DescargarActualizar("http://www.malwaredomainlist.com/hostslist/delisted.txt")
    VerificarActualizacion("deslisted")
    DescargarActualizar("http://www.malwaredomainlist.com/hostslist/ip.txt")
    VerificarActualizacion("ip")
    DescargarActualizar("http://www.malwaredomainlist.com/hostslist/hosts.txt")
    VerificarActualizacion("hosts")

```

Resultado de la ejecución de la actualización de riesgos

```
Console  Hierarchy View
<terminated> MalwareDownload.py [/usr/bin/python]
Descarga completa
Esta es la primera insercion de deslisted
Descarga completa
Esta es la primera insercion de ip
Direcciones IP Maliciosas Actualizadas
Descarga completa
Esta es la primera insercion de hosts
Hosts Maliciosos actualizados
```

CÓDIGO FUENTE PARA LA ALIMENTACIÓN DE LA TABLA MEDIDAS DE SEGURIDAD

```
'''
@author: Nagata
'''

import psycopg2
from __builtin__ import True, str
from _elementtree import tostring
from twisted.web.test.test_stan import proto

global sid
i=0
def Reglas_DB(Tabla,protocolo,origen,destino):

    Accion="drop"
    conn = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
    cur = conn.cursor()
    cur.execute("SELECT * from "+Tabla)

    conn2 = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
    cur2 = conn2.cursor()

    row = cur.fetchone()
    print "Actualizando reglas de seguridad de tabla -> ",Tabla
    global i

    msg="Alerta de seguridad"

    classtype ="Violacion de seguridad-Prueba de tesis Nagata001";

    while row:
        i=i+1
        contenido=row[1]

        mensaje= "(",'msg:"',msg,'";'
        Mensaje_DB= ''.join(mensaje)

        contenido='content:"',contenido,'";'
        Contenido_DB=''.join(contenido)

        Classtype='classtype:',classtype,';'
        Classtype_DB=''.join(Classtype)

        Sid="sid:",str(i),';',')'
        Sid_DB=''.join(Sid)

        insertarDB= Accion,protocolo,origen,destino,Mensaje_DB,Contenido_DB,Classtype_DB,Sid_DB

        cur2.execute("INSERT INTO medidasdeseguridad (accion,protocolo,origen,destino,msg,contenido,classtype,sid) VALUES('"+
                     Accion+"','"+
                     protocolo+"','"+
                     origen+"','"+
                     destino+"','"+
                     Mensaje_DB+"','"+
                     Contenido_DB+
                     "','"+Classtype_DB+"','"+Sid_DB+"')")



        conn2.commit();

        row = cur.fetchone()


    cur.close()
    conn.close()

    cur2.close()
    conn2.close()
    print "Termino"


Reglas_DB("ip", "dns","any any ->","any any")
Reglas_DB("ip", "tcp","any any ->"," any any")
Reglas_DB("ip", "udp","any any ->"," any any")


Reglas_DB("hosts", "dns", "any any ->","any any")
Reglas_DB("hosts", "tcp", "any any ->","any any")
Reglas_DB("hosts", "udp", "any any ->","any any")

Reglas_DB("deslisted","dns", "any any ->","any any")
Reglas_DB("deslisted","tcp", "any any ->","any any")
Reglas_DB("deslisted","udp", "any any ->","any any")
```

Resultado de alimentación de la tabla medidas de seguridad:

```
select * from medidasdeseguridad
```

medidasdeseguridad

select * from medidasdeseguridad | Enter a SQL expression to filter results (use Ctrl+Space)

	id	accion	protocolo	origen	destino	msg	contenido	classtype
1	1	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.14.120.121";	classtype:Vi
2	2	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.19.89.55";	classtype:Vi
3	3	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.224.212.222";	classtype:Vi
4	4	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.24.13.91";	classtype:Vi
5	5	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.31.186.207";	classtype:Vi
6	6	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.31.186.29";	classtype:Vi
7	7	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.4.16.91";	classtype:Vi
8	8	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.4.218.22";	classtype:Vi
9	9	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.6.196.156";	classtype:Vi
10	10	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.8.127.189";	classtype:Vi
11	11	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"103.8.127.205";	classtype:Vi
12	12	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.152.215.90";	classtype:Vi
13	13	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.200.67.194";	classtype:Vi
14	14	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.245.239.7";	classtype:Vi
15	15	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.27.163.228";	classtype:Vi
16	16	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.28.14.104";	classtype:Vi
17	17	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.28.15.104";	classtype:Vi
18	18	drop	dns	any any ->	any any	(msg:"Alerta de seguridad";	content:"104.31.75.147";	classtype:Vi

CÓDIGO FUENTE PARA GENERACIO DE REGLAS EN SURICATA(IDS)

```
'''
@author: Nagata
'''
import psycopg2

file = open("/etc/suricata/rules/NagataTesis001.rules","a")

try:
    conn = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
except:
    print "Error de Base de datos"

conn = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
cur = conn.cursor()

cur.execute("SELECT * from MedidasDeSeguridad")
row = cur.fetchone()


while row:

    Regla=row[1]+" "+row[2]+" "+row[3]+" "+row[4]+" "+row[5]+" "+row[6]+" "+row[7]+" "+row[8]

    print(Regla)
    file.write(Regla)
    file.write("\n")

    row = cur.fetchone()

cur.close()
conn.close()

file.close()
```

Resultado de la generación de reglas en suricata(IDS):

```
root /home/debian1 # iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  rev.opentransfer.com.1.172.131.98.in-addr.arpa  anywhere
DROP       all  --  rev.opentransfer.com.1.132.131.98.in-addr.arpa  anywhere
DROP       all  --  rev.opentransfer.com.2.32.130.98.in-addr.arpa  anywhere
DROP       all  --  rev.opentransfer.com.2.102.130.98.in-addr.arpa  anywhere
DROP       all  --  redirector-sjl.enom.com  anywhere
DROP       all  --  redirector-ash.enom.com  anywhere
DROP       all  --  96.30.28.181         anywhere
DROP       all  --  a96-17-161-145.deploy.static.akamaitechnologies.com  anywhere
DROP       all  --  a96-17-161-137.deploy.static.akamaitechnologies.com  anywhere
DROP       all  --  194.180.127.96.unassigned.ord.singlehop.net  anywhere
DROP       all  --  rev.opentransfer.com.64.115.0.96.in-addr.arpa  anywhere
DROP       all  --  95.64.8.76           anywhere
DROP       all  --  227rfszma.guzel.net.tr  anywhere
DROP       all  --  ns.km30719-01.keymachine.de  anywhere
DROP       all  --  95.163.104.80        anywhere
DROP       all  --  95.154.228.163       anywhere
DROP       all  --  95.143.193.60        anywhere
DROP       all  --  xvm-169-132.dc0.ghst.net  anywhere
DROP       all  --  95.141.37.183        anywhere
DROP       all  --  host62-189-110-95.serverdedicati.aruba.it  anywhere
DROP       all  --  host212-133-110-95.serverdedicati.aruba.it  anywhere
DROP       all  --  95.105.27.11.dynamic.oktgs.ufanet.ru  anywhere
DROP       all  --  swindon.eukhost.com  anywhere
```

CÓDIGO FUENTE PARA GENERACIO DE REGLAS EN *SQUID* PROXY

```
'''
@author: Nagata
'''
import psycopg2


file = open("/etc/squid/blocksitesTesisNagata1","a")

try:
    conn = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
except:
    print "Error de Base de datos"

conn = psycopg2.connect("dbname='Malware_DB_tesis01' user='postgres' host='localhost' password='toor'")
cur = conn.cursor()

cur.execute("SELECT * from hosts")
row = cur.fetchone()


while row:
    print row[1]
    file.write(row[1])
    file.write("\n")
    row = cur.fetchone()

cur.close()
conn.close()
file.close()
```

Resultado de la generación de reglas en Squid Proxy:

```
root /etc/squid # cat blocksitesTesisNagata1
0koryu0.easter.ne.jp
109-204-26-16.netconnexion.managedbroadband.co.uk
1866809.securefastserver.com
2amsports.com
4dexports.com
50efa6486f1ef.skydivesolutions.be
61kx.uk-insolvencydirect.com
6b8a953b2bf7788063d5-6e453f33ecbb90f11a62a5c376375af3.r71.cf5.rackcdn.com
97b1c56132dfcdd90f93-0c5c8388c0a5897e648f883e2c86dc72.r54.cf5.rackcdn.com
999fitness.com
a.update.51edm.net
ab.usageload32.com
abcdespanol.com
above.e-rezerwacje24.pl
absurdity.flarelight.com
achren.org
acool.csheaven.com
ad-beast.com
ad.9tv.co.il
ad.getfond.info
adgallery.whitehousedrugpolicy.gov
adlock.in
adobeflashupdate14.com
ads.wikipartes.com
adserving.favorit-network.com
adv.riza.it
advancetec.co.uk
afa15.com.ne.kr
```