UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA DE FORMACIÓN PROFESIONAL DE

INGENIERÍA DE SISTEMAS

TÉCNICAS DE PROTECCIÓN PARA MEJORAR LA SEGURIDAD DE UNA APLICACIÓN WEB, EN LA CIUDAD DE AYACUCHO, 2012.

TESIS PRESENTADO POR:

Bach. JAVIER CRUZ AYALA

Para optar el título profesional de:

INGENIERO DE SISTEMAS

AYACUCHO-PERU

2012

"TÉCNICAS DE PROTECCIÓN PARA MEJORAR LA SEGURIDAD DE UNA APLICACIÓN WEB, EN LA CIUDAD DE AYACUCHO, 2012"

RECOMENDADO : 09 DE AGOSTO DEL 2012

APROBADO : 24 DE AGOSTO DEL 2012

Ing. Dr. JAIME A. HUAMAN MONTES
(Presidente)

Ing. EDITH F. GUEVARA MOROTE
(Miembro)

Ing. ELINAR CARRILLO RIVEROS
(Miembro)

Ing. FLORO N. YANGALI GUERRA
(Secretario Docente)

Según el acuerdo constatado en el Acta, levantada el 24 de agosto del 2012, en la Sustentación de Tesis presentado por el Bachiller en Ingeniería de Sistemas Sr. Javier CRUZ AYALA, con el Trabajo Titulado "TÉCNICAS DE PROTECCIÓN PARA MEJORAR LA SEGURIDAD DE UNA APLICACIÓN WEB, EN LA CIUDAD DE AYACUCHO, 2012", fue calificado con la nota de TRECE (13) por lo que se da la respectiva APROBACIÓN.

Ing. Dr. JAIME A. HUAMAN MONTES
Presidente)

Ing. EDITH F. GUEVARA MOROTE
(Miembro)

Ing. ELINAR CARRILLORIVEROS
(Miembro)

Ing. FLORO N. YANGALI GUERRA
(Secretario Docente)

DEDICATORIA

Al "SEÑOR DE QUINUAPATA", por brindarme la fortaleza necesaria y la sabiduría.

AGRADECIMIENTO

A nuestra "UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA", y a nuestros profesores quienes nos impartieron con sus conocimientos y experiencias, a mi padre: Víctor Cruz Pariona, y mi madre: Victoria Ayala Gómez, quienes con su apoyo incondicional, lograron sacarme adelante, como también mi hermano: Gerardo Cruz Ayala, quien también me supo ayudar y a mi señora esposa: Adolfa Condori Landa, quien con su paciencia y apoyo me supo sobrellevar en los momentos más críticos de mi vida, y a mi hija: Annarie Jhoanna Cruz Condori, que es la razón de mi vida.

CONTENIDO **Pag.**

CAPITULO I

PROBLEMA DE INVESTIGACION

CAPITULO II

MARCO TEORICO

CAPITULO III

METODOLOGIA DE LA INVESTIGACIÓN

CAPITULO IV

ANALISIS Y RESULTADOS DE LA INVESTIGACION

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

RESUMEN

Los sitios web actuales son complejos, la lógica de servidor es la inteligencia de las aplicaciones, estas pueden ser vulnerables a numerosos tipos de ataques, es por ello se aplicó las técnicas de protección que a continuación se especifica.

Para la recolección de datos se elaboró un cuestionario de preguntas, donde se realizó una entrevista a los diferentes personas comunes que realizan transacciones de información, y mencionaban que no confiaban en las tecnologías de internet, ya que es un medio no seguro, para realizar cualquier transacción, también se realizó las entrevistas a los consultores de software, en este caso especificaban que no conocían métodos seguros en las aplicaciones webs

Para este proyecto, utilizaremos técnicas de protección para dar seguridad en las aplicaciones webs, creando algunas configuraciones y/o implementando código en Java, para la protección de los datos, en una aplicación web, necesariamente se protegerá el servidor de aplicaciones, el Apache TOMCAT, para proteger los datos de las peticiones y respuestas de los usuarios frente al servidor.

Se va a implementar un certificado digital, utilizando el JDK de java, ya que este software de programación nos ayuda a crear nuestro certificado para nuestro servidor, y para la encriptación de datos se implementara un algoritmo de encriptación de texto plano como es el RSA, a texto cifrado, para lo cual se podrá enviar de manera segura los datos cifrados.

Para configurar los certificados SSL (Secure Socket Layer), donde se tendrá un estación cliente seguro, para la navegación de nuestra aplicación.

La técnica de autenticación se realizara utilizando la librería Captcha, en la cual se tendrá una imagen aleatoria, donde el usuario ingresara el texto de la imagen. Como también se implementara un teclado virtual, con la finalidad de tener una captura de datos de manera encapsulada, al momento de la autenticación.

Para la inyección de SQL, se implementara un procedimiento almacenado en PostgreSQL, para la consulta de la Base de Datos.

Para la manipulación URL, se usara el FRON CONTROLLER en java, en la cual no se mostrara la dirección correcta de la URL.

En el marco teórico se investigara la parte fundamental del tema, donde nos informaremos de todos los indicadores que se obtuvo tanto de las variables independientes y dependientes, como también sus indicadores.

De esta manera se podrá tener un modelo de una aplicación web, con la finalidad de brindar a los programadores las técnicas utilizadas, para su seguridad.

PALABRAS CLAVE

Aplicación web, Técnicas de protección, Seguridad web, Servidor web, Seguridad informática, Seguridad a la CID.

INTRODUCCION

Si se analiza de manera detenida a las aplicaciones webs, que se desarrolla en nuestro medio, como es el caso de nuestra Universidad Nacional de san Cristóbal de Huamanga, sobre el uso de algunas técnicas y/o configuración, para dar la seguridad, no existen, por ello para solventar la problemática existente en la protección de la información. La razón por la que se ha elegido la investigación, es por la necesidad del desarrollador tener un prototipo de técnicas de desarrollo de una aplicación web.

El objetivo de este proyecto consiste en establecer un conjunto de lineamientos necesarios, para implementar: Seguridad en aplicaciones web, para lograr su Confiabilidad, Integridad y Calidad. Estos lineamientos están acompañados por técnicas y herramientas de apoyo a la seguridad de los desarrollos tecnológicos, resultado de sus proyectos de investigación para lograr un trabajo confiable y de calidad.

CAPITULO I

1.1 DIAGNOSTICO Y ENUNCIADO DEL PROBLEMA

Hoy en día en la ciudad de Ayacucho, existen diversas aplicaciones webs de las diferentes instituciones estatales como privadas que se encuentran en la red, y esas aplicaciones fueron desarrolladas por consultores en el desarrollo de software, dichas consultoras no conocían a fondo las mejores prácticas de seguridad, enfocadas al desarrollo web, sin considerar los controles básicos de seguridad. y/o con algunas técnicas en protección.
Podemos evaluar la seguridad de un activo en base a tres aspectos principales como son la integridad, disponibilidad y confidencialidad.

Según la ISO 27001 en su apartado dedicado a la seguridad de la información. La confidencialidad, donde permite el acceso únicamente a los datos a los cuales el usuario está permitido, la integridad que asegura que los datos no se falsifican o alteran por usuarios no autorizados, y la disponibilidad que asegura los sistemas y datos están disponibles para los usuarios autorizados cuando lo necesiten.

Generalmente la seguridad se basa en la protección de los activos, estos elementos pueden ser elementos tangibles como un servidor o base de datos, en donde el robo de información es muy continuo por parte de los expertos, aprovechando las vulnerabilidades de una aplicación web, para luego obtener accesos a una aplicación, robo de información de la empresa (números de tarjetas de crédito, cuentas bancarias, información clasificada, información personal, correo electrónico, etc.), afectar el funcionamiento normal del servicio que presta la organización.

Los ataques informáticos más comunes en las aplicaciones web, que se dan en la ciudad de Ayacucho, y que se deben considerar son los siguientes:
Ataques anónimos, ataques a la integridad, ataques de denegación de

servicios, ataques a la autenticación de sesión, las debilidades de los protocolos HTTP, ataques al servidor web y ataques a la base de datos en donde los hackers realizan las técnicas de intrusión como son inyección SQL, fraudes por ausencia de "Roles", manipulación de URL, etc.

Presentamos un cuadro estadístico, en donde con la ayuda de la herramienta ACUNETIX, se hicieron las pruebas de escaneo de vulnerabilidades, de las aplicaciones web de las instituciones de nuestra región Ayacucho, en donde nos dío los siguiente resultados.

	TIPOS DE ATAQUES			
EMPRESAS	**Ataque a la confidencialidad**	**Ataques anónimos**	**Ataques de integridad**	**Ataques a la Autenticación**
Municipalidad Provincial de Huamanga	5%	15%	24%	66%
Universidad Nacional de San Cristóbal de Huamanga	26%	10%	13%	51%
Gobierna Regianal de Ayacucha	20%	18%	26%	36%

Tabla N° 1.1: Principales ataques en aplicaciones web, Región Ayacucho, 2012.

Observamos la tabla 1.1, los ataques a la autenticación es el más recurrente en las aplicaciones Web, en la ciudad de Ayacucho.

1.2 DEFINICION DEL PROBLEMA

PROBLEMA GENERAL

¿De qué manera las técnicas de protección mejoran la seguridad de la aplicación web, en la ciudad de Ayacucho, 2012?

PROBLEMAS SECUNDARIOS

a. ¿Cómo la encriptación de datos, mejora la seguridad de la aplicación web?
b. ¿Cómo la configuración de SSL (Secure Socket Layer) en el servidor mejora la seguridad de la aplicación web?
c. ¿Cómo las técnicas de autenticación, permite reducir los riesgos en la

seguridad de la aplicación web?

d. ¿Cómo las técnicas contra ataques de inyección SQL, mejoran en la seguridad de una aplicación web?
e. ¿De qué manera las técnicas contra ataques de manipulación de comandos, aportan en la seguridad en una aplicación web?

1.3 DELIMITACION DE LOS OBJETIVOS DE LA INVESTIGACION

OBJETIVO GENERAL

Implementar las técnicas de protección para mejorar la seguridad de la aplicación web, en la ciudad de Ayacucho, 2012. A través de las tecnologías de internet, la utilización de herramientas JDK (Java Deveploment Kit), con el entorno integrado de desarrollo (IDE) NetBeans), base de datos relacional libre PostgreSQL, Servidor de aplicaciones (Apache Tomcat v7.0.8) y herramientas de escaneo de vulnerabilidades (Acunetix v7.0); con el propósito de disminuir los ataques a las aplicaciones web que funcionan en la Ciudad de Ayacucho; y la finalidad de brindar al desarrollador de software las diferentes técnicas de protección investigadas.

1.3.2 OBJETIVOS ESPECIFICOS

a. Implementar la encriptación de datos, para mejorar la seguridad de la aplicación web, con la finalidad de proteger la información confidencial transmitida por la red.
b. Configurar e implementar los certificados SSL (Secure Socket Layer), para mejorar la seguridad de una aplicación web, con la finalidad de asegurar, una transmisión segura de la información, contra la interceptación de los datos (ataques a la integridad), desde el cliente hacia el servidor.
c. Implementar una técnica de autenticación, para mejorar la seguridad en la aplicación web, con la finalidad de tener un acceso al sistema más seguro.
d. Implementar una técnica, contra ataques de inyección SQL, para mejorar la seguridad de una aplicación web, con la finalidad de contar con procedimientos almacenados, para proteger la información de la base de datos.

e. Implementar una técnica contra ataques de manipulación de comandos, para mejorar la seguridad de una aplicación web, con la finalidad de contar con una URL inactivo, para proteger la información de la base de datos.

1.4.1 HIPOTESIS DE LA INVESTIGACION

HIPOTESIS GENERAL

Si implementamos las técnicas de protección de información, entonces se mejora la seguridad de la aplicación web en la Ciudad de Ayacucho, 2012.

1.4.2 JUSTIFICACION Y DELIMITACION DE LA INVESTIGACION

IMPORTANCIA

En nuestra ciudad de Ayacucho, los ataques más comunes a las páginas webs, son mayormente en las instituciones financieras o donde se realizan el comercio electrónico, donde son las más afectadas por los hackeadores de páginas webs, que necesitan técnicas de protección, a fin de mejorar en la seguridad de una aplicación web.

No considerar las técnicas de protección en la parte de desarrollo de los proyectos, tienen como consecuencia, las diversas vulnerabilidades de la aplicación, y al no incluir esas protecciones, se tendrá problemas en el funcionamiento de la aplicación, ya sea el robo de información.

JUSTIFICACION

La importancia técnica que se implementara todo los mecanismos y/o técnicas de seguridad, a quien apoyará a los investigadores, desarrolladores de aplicaciones, en el mejoramiento de la protección de sus aplicaciones web, se tendrá un procedimiento detallado e implementado como una guía que pueda ser adaptado, para su posterior aplicación, y de esta manera dar una buena seguridad al momento de su implantación de la aplicación.

DELIMITACION

La investigación se desarrollará para una aplicación web, que se ejecutará en la ciudad de Ayacucho de las instituciones públicas.

CAPITULO II

MARCO TEORICO

2.1 ANTECEDENTES DE LA INVESTIGACION

En la actualidad en nuestra ciudad de Ayacucho, no se encuentra ninguna tesis sobre la seguridad en aplicaciones web, pero si tenemos algunas tesis que se elaboraron en la ciudad de Lima, pero precisamente no es sobre la seguridad en las aplicaciones web, sino más bien tesis sobre los comercios electrónicos, y lineamientos acerca del uso de las herramientas que se utilizan para aminorar los peligros que estas pueden tener.

Según Mattos, E. (2005). "Las instituciones bancarias son las que han realizado mayor avance en el comercio electrónico, priorizando el tema de seguridad", "Incluir en la normatividad referencia a estándares internacionales no sólo de certificación y de evaluación, también para la definición clara de la PKI nacional"

Según Sebastián, L. (2006). "Muchas de las vulnerabilidades que se pueden presentar son propias de la plataforma sobre la que se desarrolla la aplicación (Sistema Operativo, software de base, herramientas de desarrollo), otras son negligencia por parte de jefes de proyecto, arquitectos, diseñadores, programadores, administradores y usuarios del sistema".

Según Sanctum, H. (2004). "AppScan DE es un instrumento para probar la seguridad de las aplicaciones Web. Al saber más del producto de AppScan DE, será difícil de evitar la impresión de que AppScan DE es como un tipo de "pirata informático".

Según Cabrera, S., García, MC., y Salinas, J. (2009). "A lo largo de la aplicación del modelo en el desarrollo que está siendo realizado por Dominio se mencionaron procedimientos planteados en el modelo de seguridad Web, estos deben ser utilizados para reducir las vulnerabilidades de una aplicación. Se

demostró que la aplicación que se examinó no cumple con varios puntos del modelo, lo que la hace vulnerable a varios tipos de ataques, si se hubiese implementado el modelo desde el inicio de habrían ahorrado tiempo en re trabajo".

Vemos que varias medidas de control, implementadas en los demás investigaciones mencionadas, en el comercio electrónico deben ser implementadas en el marco de políticas de seguridad establecidas, ejecutadas en varias fases distintas del ciclo de vida de la aplicación, y controladas por un auditor, que permiten disminuir considerablemente los riesgos e impacto de estas amenazas vistas, aunque difícilmente sea posible asegurar la invulnerabilidad de una aplicación.

Los protocolos y arquitecturas para asegurar las comunicaciones en una red son, al igual que el resto de los componentes de software y hardware, propensos a ataques malintencionados.

2.2 MARCO TEORICO

TÉCNICA DE PROTECCIÓN

Según Benson, S. (2001). "Las técnicas de protección está diseñada para ayudar a los profesionales de la seguridad para desarrollar una estrategia para proteger la disponibilidad, integridad y confidencialidad de los datos de los sistemas de información de las organizaciones que tienen acceso con el medio dela web".

Según Schneir. (2004). "La metodología ofrece un acercamiento sistemático a esta importante tarea y, como precaución final, también implica el establecimiento de planes de contingencia en caso de desastre".

A. ENCRIPTACIÓN DE DATOS

Según Salcedo, H. (2008). "Los datos que se enviaran mediante la red de la web se tienen que estar cifrado por cualquier algoritmo de encriptación de los datos".

Según Technet, M (2006). "El cifrado se puede entender como el hecho de guardar algo valioso dentro de una caja fuerte cerrada con llave. Los datos confidenciales se cifran con un algoritmo de cifrado y una clave que los hace

ilegibles si no se conoce dicha clave. Las claves de cifrado de datos se determinan en el momento de realizar la conexión entre los equipos. El uso del cifrado de datos puede iniciarse en su equipo o en el servidor al que se conecta."

CRIPTOGRAFÍA

Según Schneir. (1998). "La criptografía es la ciencia que trata los problemas teóricos relacionados con la seguridad en el intercambio de mensajes en clave entre un emisor y un receptor a través de un canal de comunicaciones"

Según Gómez, A. (2007). "La criptografía es la ciencia que se encarga de estudiar las distintas técnicas empleadas para transformar la información y hacerla irreconocible a todos aquellos usuarios no autorizados de un sistema informático, de modo que solo los legítimos propietarios puedan recuperar la información original".

Es un conjunto de técnicas empleadas para conservar la información de forma segura. Está basada en la transformación de los datos de forma tal que sean incomprensibles para los receptores no autorizados, en cambio para aquellos receptores que posean la autorización correspondiente, los datos que conforman dicha información resultarán perfectamente comprensibles.

En la transformación se pueden identificar 2 procesos bien definidos

Encriptación.- Proceso mediante el cual un conjunto de datos se transforman en un conjunto cifrado de datos mediante una función de transformación y una llave de codificación.

Desencriptación.- Proceso inverso a la encriptación, en el cual el conjunto cifrado de datos se convierte en el texto original mediante una segunda función de transformación y una llave de desencriptación. La llave puede ser la misma para ambos procesos o distinta.

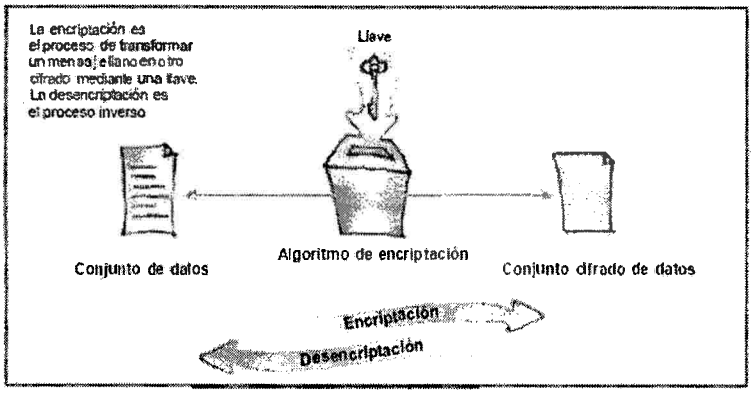


Figura: Nº 2.1: Encriptación y desencriptación de datos (Gómez, 2007)

CRIPTOANÁLISIS

Según Schneir. (1998). "El criptoanálisis es la ciencia opuesta a la criptografía, ya que si se trata principalmente de crear y analizar criptosistemas seguros, la primera intenta romper esos sistemas, demostrando su vulnerabilidad: dicho de otra forma, trata de descifrar los criptogramas".

Según Gómez, A. (2007) "El criptoanálisis es la ciencia que se ocupa de estudiar herramientas y técnicas que permitan romper los códigos y sistemas de protección definidos por la criptografía".

CRIPTOSISTEMA

CRIPTOSISTEMA DE CLAVE SECRETA

Según Whitfiel, D. y Hellman, M. (1976). "Denominamos criptosistema en el que la clave de cifrado, puede ser calculada a partir de la de descifrado y viceversa".

CRIPTOSISTEMA DE CLAVE PÚBLICA

Según Whitfiel, D. y Hellman, M. (1976). "La clave de cifrado se hace de conocimiento general (se le llama clave pública). Sin embargo, no ocurre lo mismo con la clave de descifrado(clave privada), que se ha de mantener en secreto. Ambas claves no son independientes, pero del conocimiento de la pública no es posible deducir la privada sin ningún otro dato (recordemos que en los sistemas de clave privada sucedía lo contrario)."

Según Arcert. (2008). "Este sistema se basa en una llave de sesión, que es una llave pública aleatoria que se utiliza para crear un sistema de llaves simétricas.

Cada vez que se inicie un intercambio de datos, la llave aleatoria habrá cambiado y se generará una nueva llave simétrica. Este sistema es uno de los más utilizados ya que combina las ventajas de ambos sistemas".

B. MECANISMO DE AUTENTICACIÓN

Según Fernández (2005)."La autenticación me permite identificar quien es el usuario que está ingresando".

Según Salcedo, H. (2008). "Los Servlets implementan la autenticación, y tenemos 4 mecanismos están basados en el usuario y contraseña, y el servidor Web mantiene la lista de usuarios y contraseñas.".

HTTP BASIC

Según Fernández (2005). "Es simple y es el más usado para proteger los recursos. Consta de una ventana pop up que pide el usuario y contraseña. El problema es que no encripta los datos y no se puede modificar (personalizar) la ventana pop up".

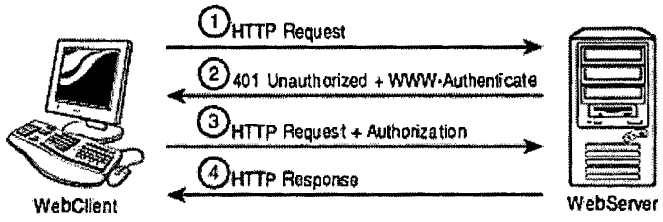


Figura N° 2.2: Autenticación básica - RFC 2617

HTTP DIGEST

Según Fernández (2005). "Es muy parecido al HTTP Basic, solo que este mecanismo si es encriptado (usa MD5). Lo malo es que sólo lo soporta Internet Explorer, y tampoco lo soportan todos los contenedores de servlets".

HTTPS CLIENT

Según Fernández (2005)."Se trata del HTTP sobre SSL (Secure Socket Layer). Los datos que viajan entre el servidor y el cliente son encriptados usando la criptografía de llave pública. Es en definitiva el más seguro y es soportado por la mayoría de los navegadores. Las desventajas son que requiere un certificado de

una autoridad certificadora como Verisign, y además, es muy costoso su implementación y mantenimiento".

AUTENTICACIÓN HTTP BASADA EN FORMULARIOS

Según Fernández (2005). "Es un similar a HTTP Basic pero utiliza un formulario en HTML para ingresar el usuario y contraseña. Es fácil de implementar y todos los navegadores lo soportan. Lo malo es que no es seguro (los datos no viajan encriptados), y además se requiere que los clientes soporten cookies".

Según Gonzales, C. (2008). "Con este tipo de autenticación tenemos una situación muy clara: El proceso de autenticación no depende de las características del protocolo HTTP o SSL.

Si no existen fallos en la selección del protocolo de autenticación o en su forma de implementarlo, los ataques por adivinación de contraseña deben ser evitados.

C. TECNICAS DE ATAQUES DE INYECCIÓN DE CÓDIGO SQL

Según Meiners, L. (2005). "Es una técnica cuyo objetivo es el de "inyectar" consultas SQL arbitrarias en páginas vulnerables que interactúan con una Base de Datos, logrando de esta forma obtener, modificar y/o eliminar información sensible. Atacando ciertos motores de Bases de Datos es posible, también, lograr la ejecución de comandos del Sistema Operativo".

Según Sebastián, L. (2006). "SQL Inyección es una vulnerabilidad que afecta aplicaciones a nivel de base de datos. Dicha vulnerabilidad consiste en enviar instrucciones SQL adicionales a partir de parámetros entrada ingresados por el usuario".

Según Gómez, A. (2007). "El ataque por inyección de código SQL se produce cuando no se filtra de forma adecuada la información enviada por un usuario. Un usuario malicioso podría incluir y ejecutar textos que representen nuevas sentencias SQL que el servidor no debería aceptar".

Inyección SQL es una vulnerabilidad que afecta aplicaciones a nivel de base de

datos. Dicha vulnerabilidad consiste en enviar instrucciones SQL adicionales a partir de parámetros entrada ingresados por el usuario.

Al "inyectar" el código SQL malicioso dentro de estos campos, el código "invasor" se ejecuta dentro del código SQL propio de la aplicación para alterar su funcionamiento normal, de acuerdo con el propósito del atacante.

Este tipo de ataques es independiente al sistema de base de datos subyacentes, ya que depende únicamente de una inadecuada validación de datos de entrada.

D. TECNICAS DE ATAQUES DE EJECUCIÓN DE COMANDOS

Según López, S. (2006). "Las técnicas de manipulación de entrada vistas son sólo algunas que llevan a la posibilidad de ejecutar remotamente comandos del Sistema Operativo de la víctima.

Determinados caracteres especiales pueden ser interpretados por scripts de validación poco seguros como una instrucción al SO de esperar un comando arbitrario a continuación. En particular, el punto y coma o la barra | son en Unix caracteres que permiten encadenar comandos. Por tanto incluir en el campo de entrada un; seguido del comando que se desea ejecutar puede tener éxito si el mecanismo de validación no es lo bastante robusto"

E. HERRAMIENTA DE SEGURIDAD

Según Carcert. (2005). "Las herramientas sirven para administrar la seguridad en todo el sistema de defensa por niveles y administrar las amenazas en curso".

Según Stalling. (2005). "Son programas que se utilizan para evaluar o mejorar la seguridad de unsitio. Las podemos dividir en cuatro categorías:

Herramientas de instantánea.- Estos sistemas toman una foto del servidor y buscan debilidades potenciales notificándolas a la persona que la ejecuta, se caracterizan por hacer muchas revisiones en poco tiempo. También se las conoce como herramienta de auditoría estática. Estos programas deben ejecutarse con regularidad y se debe evaluar con mucho cuidado la salida que producen, ya que en ellos se pueden encontrar los agujeros de seguridad del

sistema.

Herramientas de detección de cambios no autorizados.- Cuando un atacante ingresa en un sitio restringido lo primero que suele hacer escanear puertas traseras que le faciliten futuros ingresos al sistema y también modificar el mismo para ocultar evidencia de intrusión. Dejando a su paso cambios no autorizados en el sistema. El hecho de encontrar estos cambios no evita la interrupción pero puede indicar que el sistema ha sido violado.

Herramientas que escudriñan la red, buscando debilidades en ella.- Estas herramientas buscan errores de programación conocidos relacionados con la seguridad. Es sabido que los crackers escudriñan la red con estas herramientas, así que lo mejor es utilizar uno mismo estos programas para mantenerse consciente de las falencias de nuestro sistema.

Herramientas que monitorean el sistema y la red buscando ataques en progreso: Estas herramientas funcionan como alarma antirrobo registrando la computadora mientras opera, en busca de señales de una irrupción".

ACUNETIX - ESCANEO DE VULNERABILIDADES WEB

Según Todoroms, E. (2005). "Acunetix Web Vulnerability Scanner, es un escáner de vulnerabilidades en aplicativos Web (CMS, Sistemas de Información Web, Web dinámicas), permite llevar a cabo de manera automática detecciones de vulnerabilidades del tipo SQL Injection, XSS, Cifrado, descubrimiento de directorios, etc".

SEGURIDAD EN UNA APLICACIÓN WEB

Sebastián, L. (2006). "El cerebro de los web sites actuales es la lógica de servidor. Comprender las arquitecturas n-niveles es importante desde la seguridad de las aplicaciones".

Según Jeri, F. (2006). "La seguridad, en informática como en otras áreas, se basa en la protección de activos. Estos activos pueden ser elementos tan tangibles como un servidor o una base de datos, o pueden ser la reputación de una empresa. Generalmente podemos evaluar la seguridad de un activo en base a

tres aspectos principales que no necesitan explicación: integridad, disponibilidad, confidencialidad.

BASES DE LA SEGURIDAD DE LA INFORMACIÓN

Según Howard y LeBlanck. (2003). "La seguridad de la información se ha mantenido sobre los siguientes pilares, planteados en la normativa ISO 27001, en su apartado dedicado a la Seguridad de la Información".

Confidencialidad.- Permitir acceso únicamente a los datos a los cuales el usuario está permitido.

Integridad.- Asegurar que los datos no se falsifican o alteran por usuarios no autorizados.

Disponibilidad.- Asegurar que los sistemas y datos están disponibles para los usuarios autorizados cuando lo necesiten.

ARQUITECTURA DE SEGURIDAD

Según Howard y LeBlanck. (2003). "Las aplicaciones sin una arquitectura de seguridad son como puentes construidos sin un análisis finito de elementos ni tests de túneles de viento. Seguramente, parecerán puentes, pero caerán a la primera sacudida de las alas de una mariposa. La necesidad de la seguridad de aplicaciones en forma de arquitectura de seguridad están grande como en la construcción de puentes o edificios"

Sebastián, L. (2006). "Los arquitectos de aplicaciones son los responsables de su construcción y diseño para cubrir los típicos riesgos tanto de uso como de ataques extremos. Los diseñadores de puentes necesitan superar cierta cantidad de coches y tráfico a pie, pero también ciclones, terremotos, fuegos, accidentes de tráfico e inundaciones. Los diseñadores de aplicaciones deben superar eventos extremos como fuerza bruta o ataques de inyección y fraude. Los riesgos de los diseñadores de aplicaciones son bien conocidos".

La seguridad ahora es algo esperado, y no un caro complemento o algo dejado de lado. La arquitectura de seguridad se refiere a los pilares fundamentales: la

aplicación debe proporcionar controles para proteger la confidencialidad de la información, integridad de los datos, y proporcionar acceso a los datos cuando se requiera (disponibilidad) y solamente a los usuarios apropiados.

La arquitectura de seguridad empieza el día en que se modelan los requisitos del negocio, y no termina nunca hasta que la última copia de su aplicación es retirada. La seguridad es un proceso de larga vida y no un disparo por accidente.

ARQUITECTURA

Sebastián, L. (2006). "La arquitectura de software son las técnicas formas y guías generales que en base a estas se pueden resolver problemas, se define arquitectura porque asemeja planos de una construcción (casa o edificio) y se plasma el funcionamiento la estructura e interacción entre las partes del software".

Capa de presentación: Tiene capacidades para aceptar entradas y desplegar los resultados

Capa lógica: Captura las entradas de la capa de presentación, realizando operaciones diversas sobre los datos y devuelve el resultado a la presentación.

Capa de datos: Permite el almacenamiento permanente de la información. La capa lógica puede consultar o actualizar.

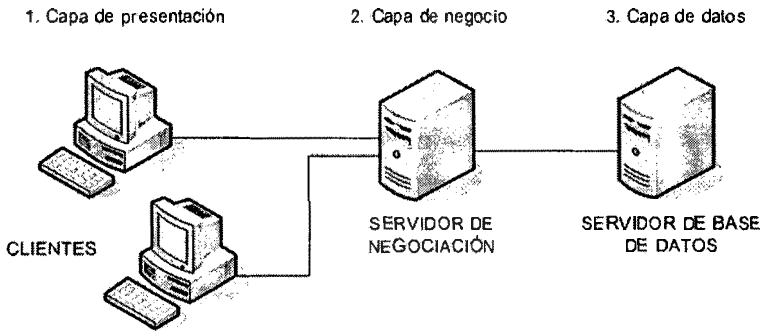


Figura N° 2.3: Arquitectura del software (LeBlanck., 2003)

APLICACIÓN WEB

Según Sebastián, L (2006). "Una aplicación escrita para el Internet, incluyendo aquellas construidas con tecnologías Java como Java Server Pages y

Servlets, así como los construidos con las tecnologías no Java como CGI y Perl".

Según Magazine, Pc. (2005). *Web Application Definition.* "En Ingeniería de software una aplicación web es una aplicación a la cual se tiene acceso vía un navegador Web sobre una red, como Internet o una intranet. Además, es una aplicación de software codificada en un lenguaje soportado por un browser o navegador Web (Como HTML, JavaScript, Java), en la que se confía la ejecución al navegador."

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responde a cada una de sus acciones, como por ejemplo; rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

LA ARQUITECTURA CLIENTE SERVIDOR.

Para Pressman, R. (2002). "Los sistemas basados en la arquitectura cliente/servidor están formados por dos partes lógicas: un servidor que proporciona servicios, y un cliente que solicita servicios del servidor o servidores. Los dos, juntos, forman un sistema de computación completo con una clara división de responsabilidades.

La computación cliente servidor es un intento de equilibrar el proceso de una red hasta que se comparta la potencia de procesamiento entre computadoras que llevan a cabo servicios especializados tales como acceder a bases de datos (servidores), y aquellos que llevar a cabo tareas tales como la visualización GUI que es el más adecuado para el punto final dentro de la red. Por ejemplo permite que las computadoras se ajusten a tareas especializadas tales como el procesamiento de bases de datos en donde se utilizan hardware y software de propósito especial para proporcionar un procesamiento rápido de base de datos comparado con el hardware que se encuentra en las mainframes que tienen que enfrentarse con una gran gama de aplicaciones".

CLIENTE

Según Pressman, R. (2002). "Si se considera que el usuario, a través de una computadora local correspondiente al cliente, es el interesado en interactuar con los programas que existen en el Internet, el cliente tiene como función primordial facilitar la interacción del usuario. En otras palabras, el objetivo básico del cliente en la arquitectura cliente servidor es facilitar la presentación y control de la información administrada por la aplicación, algo similar al rol de las clases borde en relación con un actor de tipo usuario. Por tal motivo, la mayoría de las tecnologías que se procesan en el cliente están dirigidas a facilitar la visualización y control de la información, como en el caso de HTML, Flash, Javascript, VBScript, JScript".

SERVIDOR

Según Pressman, R. (2002). "El servidor es el responsable de prestar los servicios requeridos por los clientes. Es común que la mayor parte de una aplicación en una arquitectura cliente/servidor se encuentre del lado del servidor. Esto se hace por razones de costo, eficiencia y facilidad para dar servicio a múltiples clientes de manera concurrente. En general en muchos de los lenguajes tradicionales de programación han sido utilizados para programar los servidores, pero existen ciertas tecnologías diseñadas específicamente para arquitecturas cliente servidor en Internet".

Para Fernández, R. (2002). "Un cliente es una computadora que solicita los servicios que proporciona uno o más servidores y que también lleva a cabo algún tipo de procesamiento por sí mismo".

Un Servidor es una computadora que lleva a cabo un servicio que normalmente requiere mucha potencia de procesamiento.

ESTRUCTURA DE LA APLICACIÓN WEB

Para Pressman, R. (2002). "Una aplicación Web está normalmente estructurada como una aplicación de tres capas. En su forma más común, el navegador Web ofrece la primera capa y un motor capaz de usar alguna tecnología Web dinámica (PHP, ASP, ASP.NET, CGI, ColdFusion, embPerl, Python

(programming language) o Ruby on Rails); que constituye la capa de un medio. Por último, una base de datos constituye la tercera y última capa".

Según Wikipedia, (2008). "El navegador Web manda peticiones a la capa de en medio que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario".

LENGUAJES DE PROGRAMACIÓN

Según Maestro, W. (2008). "Existen numerosos lenguajes de programación empleados para el desarrollo de Aplicaciones Web, entre los que destacan: PHP, ASP/ASP.NET, JAVA, con sus tecnologías Java Servlets y Java Server Pages (JSP) Perl, HTML, XML, Ruby, Python; ASP no es un lenguaje de programación en sí mismo, sino una arquitectura de desarrollo Web en la que se pueden usar por debajo distintos lenguajes (por ejemplo, VB.NET o C# para ASP.NET o VBScript/JScript para ASP)".

INTERNET

Para Stallings, W. (2000). "Una colección de redes de comunicación interconectadas por puentes o dispositivos de encaminamiento".

Según CEBALLOS (2006). "Es una red de redes informáticas distribuidas por todo el mundo que intercambian información entre sí mediante la familia de protocolos TCP/IP".

TCP / IP

Para Stallings, W. (2000). "TCP/IP es la arquitectura más adoptada para la interconexión de sistemas. Es el resultado de la investigación y desarrollo llevados a cabo en la red experimental de conmutación de paquetes ARPANET, y se denomina globalmente como la familia de protocolos TCP/IP Esta familia consiste en una extensa colección de protocolos que se han erigido como estándares de Internet".

Según Hernández (2005). "Es un conjunto de protocolos de comunicaciones que definen como se pueden comunicar entre si los ordenadores y otros dispositivos

de distinto tipo"

Al contrario que en OSI, no hay un modelo oficial de referencia TCP/IP. No obstante, basándose en los protocolos estándar que se han desarrollado, todas las tareas involucradas en la comunicación se puede organizar en cinco capas relativamente independientes:

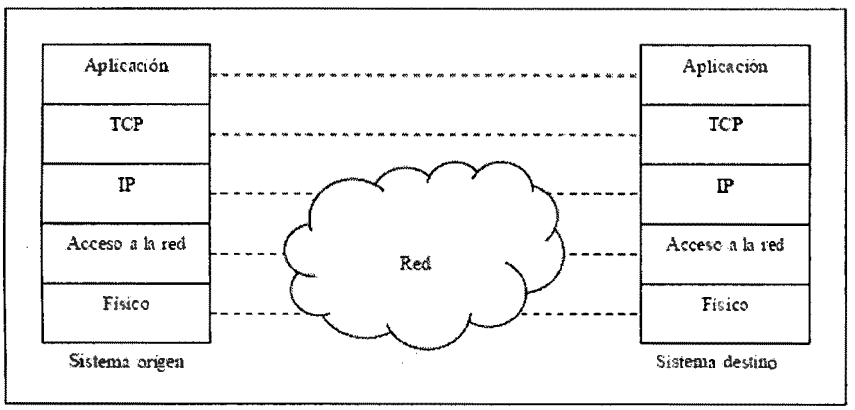


Figura N° 2.4: Modelo de arquitectura del protocolo (Stallings, 2000)

La Figura 2.4 muestra cómo se implementan los protocolos TCP/IP en los sistemas finales, a la vez que relaciona la arquitectura con el modelo para las comunicaciones. Nótese que las capas física y de acceso a la red proporcionan la interacción entre el sistema final y la red, mientras que las capas de aplicación y transporte albergan los protocolos denominados "extremo a extremo", ya que facilitan la interacción entre los dos sistemas finales. La capa Internet tiene algo de las dos aproximaciones anteriores. En esta capa, los sistemas origen y destino proporcionan a la red la información necesaria para realizar el encaminamiento, pero a la vez, deben proporcionar algunas funciones adicionales de intercambio entre los dos sistemas finales.

SERVIDOR WEB

Según PRESSMAN, R, (2002). "Los documentos Web se almacenan como páginas en una computadora conocida como servidor Web. Cuando se utiliza un navegador para ver las páginas Web normalmente pincha sobre un enlace en un documento Web existente. Esto dará como resultado un mensaje que se enviará al servidor Web que contiene la página. Este servidor responderá entonces enviando una página a su computadora, donde el navegador pueda

visualizarlo. De esta manera los servidores Web actúa como una forma de servidor de archivos, administrando archivos relativamente pequeños a usuarios, administrando archivos relativamente pequeños a usuarios, quienes entonces utilizan un navegador para examinar estas páginas".

JAVA

Según Deitel, H., Harvey, M., Paul, J. (2005). "Java es un lenguaje completamente orientado a objetos, la cual ofrece un sólido soporte para las técnicas apropiadas de ingeniería de Software. Se ha convertido en un lenguaje de elección para desarrollar aplicaciones en Internet. Utilizando una técnica de subprocesamiento permite escribir programas con actividades paralelas".

Según Ceballos, J. (2006). "Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para Internet".

A. **ATAQUE A LA CONFIDENCIALIDAD:**

Según Jeri, F. (2008). "Se refiere a que sólo el usuario indicado debe acceder a la información sensible. La diferencia con autorización es que la confidencialidad asegura que incluso si la información cae en malos manos, esta sea inutilizable".

Según Magazine, Pc. (2005). "Un atacante podría robar información sensible como contraseñas u otro tipo de datos que viajan en texto claro a través de redes confiables, atentando contra la confidencialidad al permitir que otra persona, que no es el destinatario, tenga acceso a los datos."

B. **ATAQUES ANÓNIMOS**

Según Sebastián, L. (2006). "Los ataques anónimos donde tratan de obtener información confidencial analizando las comunicaciones entre 2 computadoras".

C. **ATAQUES DE INTEGRIDAD DE DATOS**

Según Sebastián, L. (2006). "Este tipo de ataques altera la información en tránsito con fines maliciosos".

Según Jeri F. (2008). "Mientras la información se transmite a través del protocolo de comunicación, un atacante podría interceptar el mensaje y realizar cambios en determinados bits del texto cifrado con la intención de alterar los datos del criptograma. Este tipo de ataques se denomina Bit Flipping y son considerados ataques contra la integridad de la información."

D. **ATAQUES A LA BASE DE DATOS**

Según Sebastián, L. (2006). "Los ataques de inyección SQL atacan los sitios web que dependen de bases de datos relacionadas".

En este tipo de páginas Web, los parámetros se pasan a la base de datos como una consulta de SQL. Si un diseñador no verifica los parámetros que se pasan en la consulta de SQL, un hacker puede modificar la consulta para acceder a toda la base de datos e incluso modificar su contenido.
Algunos caracteres posibilitan coordinar varias consultas de SQL o ignorar el resto de la consulta. Al insertar este tipo de carácter en la consulta, un hacker puede ejecutar potencialmente la consulta que elija.

E. **ATAQUE A LA MANIPULACIÓN DE PARÁMETROS URL**

Según Sebastián, L. (2006). "Es un conjunto de técnicas que atacan la lógica de negocio de la aplicación, tomando ventaja del uso de campos ocultos o fijos para la transferencia de información sensible entre browser y servidor. En particular, tags ocultos en un form, cookies o parámetros anexados a la URL son fácilmente modificables por un atacante".

Según Magazine, Pc. (2005). "Al manipular ciertas partes de una URL, un hacker puede hacer que un servidor Web le permita acceder a páginas Web a las que supuestamente no tenía acceso.

2.2.1 METODO

UTILIZACIÓN DE SSL (Secure Sockets Layer) EN LOS SERVIDORES

Según Wikipedia. (2008). "Consiste en la encriptación de los datos que viajan desde el servidor al cliente es comúnmente conocido como HTTPS. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o

IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA".

Según Carcert (2005). "El protocolo SSL es un sistema diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos HTTP, FTP, SMTP"

FIRMAS DIGITALES

Según Cabrera (2002). "Consiste en la transformación del mensaje utilizando un sistema de cifrado asimétrico".

La firma digital es una tecnología que consta de:

Una llave privada.- Utilizada para firmar un bloque de datos.

Una llave pública.- Utilizada para verificar la firma.

Supongamos que un sujeto **A,** como el de la Figura 2.5, distribuye una llave pública a prueba de alteración. Como esta llave sólo sirve para comprobar si la llave privada que el sujeto **A** conserva es realmente la llave privada del sujeto **A,** sí alguien intercepta la llave pública no le serviría de nada, por lo tanto el sujeto **A** podría distribuir la llave pública por cualquier medio.

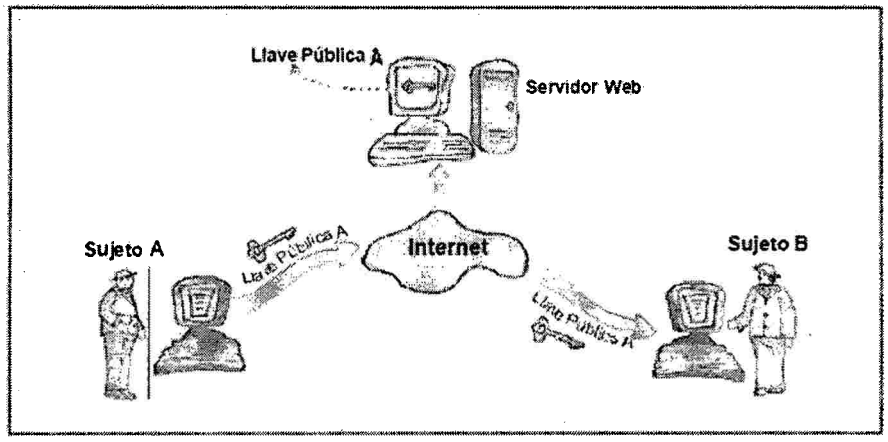


Figura: Nº 2.5: Distribución de una llave pública.

Supongamos ahora que un sujeto **B**, como el de la Figura 2.6, necesita corroborar que el sujeto **A** ha leído un documento **X**, para esto envía el documento **X** por mail al sujeto **A**, el sujeto **A** recibe el documento **X** lo lee y anexa su firma generada con la llave secreta. El sujeto **A** reenvía el documento firmado al sujeto

B que mediante la llave pública corrobora la legitimidad de la firma.

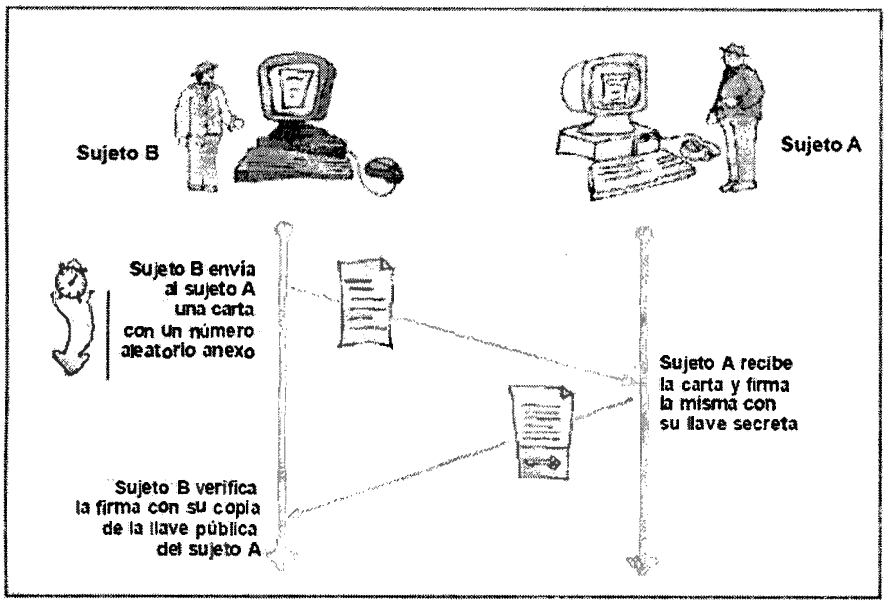


Figura: N° 2.6: Uso de la firma digital para comprobar la identidad.

CERTIFICADOS DIGITALES

Según VeriSign. (2008). "Existen organizaciones, llamadas autoridades certificadoras, que se encargan de comprobar si una llave pública específica es propiedad de un individuo u organización en particular. Como resultado de esta comprobación emiten un "Certificado de llave pública" que contiene el nombre de la persona, la llave pública, número de serie, la fecha de creación del certificado y la fecha de vencimiento.

Según Narvaes, (2005). "Un Certificado Digital, también llamado Certificado de Autenticidad o ID Digital, es un desarrollo de última tecnología que usa criptografía de clave pública para identificar personas, sus privilegios y relaciones; estos certificados son el equivalente de documentos de identidad como los DNI, licencias de conducir, pasaportes u otros".

El certificado comprueba que una llave pública específica es propiedad de un individuo u organización en particular. Los navegadores de Internet reconocen a muchas de estas autoridades certificadoras automáticamente y permiten agregar manualmente a las que no reconocen.

Si la llave privada del tenedor ha sido violada, si la persona ha dado datos

incorrectos o si hay copias falsas de ésa llave, el certificado puede ser revocado. Las llaves revocadas que aún no están vencidas son colocadas en una Lista de Revocación de Certificados (CRL, Certificate Revocation List), estas listas tienden a crecer con rapidez pero su actualización es lenta."

TÉCNICAS DE AUTENTICACIÓN

Según Gonzalo, C. (2010)."El proceso de autenticación es muy importante en la seguridad de las aplicaciones web. En general, la solicitud de usuario y contraseña, es la modalidad más implementada en web; existen otras: Autenticación HTTP Básica, Autenticación HTTP Digest, Autenticación basada en certificado, Autenticación basada en formularios".

Según Narvaes (2005). "Para la autenticación del cliente que se conecta a un Web es necesario utilizar los mecanismos que están basados en el usuario y contraseña".

2.2.2 HERRAMIENTAS

a) JDK JAVA DEVELOPMENT KIT

Según Sebastián, L. (2006). "Es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación".

Según Deitel (2008). "Java es un lenguaje completamente orientado a objetos, la cual ofrece un sólido soporte para las técnicas apropiadas de ingeniería de Software. Se ha convertido en un lenguaje de elección para desarrollar aplicaciones en Internet. Utilizando una técnica de subprocesamiento permite escribir programas con actividades paralelas".

Según Monson, J. (2008) "Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para Internet"

b) IDE NETBEANS

Según Sebastián, L. (2006). "La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software".

c) ADMINISTRADOR DE BASE DE DATOS (PostgreSQL)

POSTGRESQL

Según PostgreSQL, es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales, las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

CARACTERÍSTICAS:

Alta Concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés). PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para: Números de precisión arbitraria, Texto de largo ilimitado, Figuras geométricas, Direcciones IP (IPv4 e IPv6), Bloques de direcciones estilo CIDR, Direcciones MAC, Arrays.

Otras características: entre otras características básicas se tiene: las Vistas, Integridad transaccional, Herencia de tablas, Tipos de datos y operaciones

geométricas.

d) APACHE TOMCAT v7.0.8

Según Apache Tomcat (2008) "Servidor de aplicaciones de Sun Microsystems que implementa las tecnologías definidas en la plataforma Java E.E y permite ejecutar aplicaciones que siguen esta especificación.

Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems".

CAPITULO III

METODOLOGIA DE LA INVESTIGACION

3.1 TIPO DE INVESTIGACION

La presente investigación es aplicada, donde se tendrá que implementar algunas técnicas para dar seguridad a una aplicación web, y de esta manera se estaría dando un modelo para el desarrollador de software web, para que pueda aplicar como un prototipo de desarrollo.

3.2 DISEÑO DE LA INVESTIGACION

El diseño es una estrategia general de trabajo que el investigador determina una vez que haya alcanzado suficiente claridad respecto a su problema y que orienta y esclarece las etapas que habrán de realizarse posteriormente.

El diseño de la presente investigación no es experimental, pues no se manipulan las variables, sino que más bien se estudian tal como se presentan.

3.3 POBLACION Y MUESTRA

POBLACION

20 Aplicaciones web, de las instituciones públicas, de la ciudad de Ayacucho 2012.

MUESTRA

Se tomará una muestra de 19 aplicaciones webs, de la ciudad de Ayacucho, con el 95% de significancia y el 5% de error de una aplicación web, de la ciudad de Ayacucho 2012.

FÓRMULA PARA CALCULAR EL TAMAÑO DE LA MUESTRA

$n = Z^2 a^2 N / (e^2 + Z(N - 1) Z^2 a^2)$

Donde

n = Tamaño de la muestra

N = Tamaño de la población

a = Desviación estándar de la población, suele usarse un valor constante: 0.5

Z = Valores obtenidos mediante niveles de confianza

e = error de muestreo. En términos de proporción (tanto por uno)

Ahora vemos el valor de una población de 20 y un error de 5%, y el resultado de la muestra es: 19 aplicaciones webs.

e	0.05
N	20
o	0.5
Confianza	95
Area a la izquierda	0.025
-z	-1.95996398
z	-1.95996398
formula	19.05741317

3.4 VARIABLES E INDICADORES

3.4.1 DEFINICION CONCEPTUAL DE LAS VARIABLES

VARIABLE INDEPENDIENTE

X: TÉCNICAS DE PROTECCIÓN

En esta técnica, consiste en la utilización de diversos métodos o procedimientos para proteger nuestra aplicación web.

INDICADORES

X1: ENCRIPTACIÓN DE DATOS

Consiste en utilizar un algoritmo de encriptación, donde realiza unas transformaciones sobre el texto claro, para obtener un texto modificado, conocido como texto cifrado.

X2: CRIPTOGRAFÍA ASIMÉTRICA (Secure Socket Layer) SERVER

Consiste de dos claves, que son públicas y privadas, donde la clave pública puede ser usada para encriptar mensajes y verificar firmas. Mientras la clave privada solamente es conocida por el receptor, usada para desencriptar

mensajes y crear firmas.

X3: TÉCNICAS DE AUTENTICACIÓN

Se define la autenticación y la autorización como algo interiormente ligado, debido a que la autenticación es el establecimiento y la confirmación de algo y la autorización es aquella que después de verificar la identidad auténtica permite el acceso a zonas restringidas de información.

X4: TECNICAS CONTRA ATAQUES DE INYECCIÓN SQL:

Consiste en una consulta a través del lenguaje SQL textual, utilizado para interactuar con bases de datos relacionales, es un conjunto de instrucciones que permiten modificar la estructura de la base de datos o manipular el contenido de la base de dato.

En los servidores web se utiliza estos lenguajes para acceder a base de datos y ofrecer páginas dinámicas o nuevas funcionalidades a sus usuarios.

X5: TECNICAS CONTRA ATAQUES DE EJECUCIÓN DE COMANDOS:

Es donde el atacante realiza algunos métodos para poder inyectar códigos a la línea de comandos, y a veces, las aplicaciones invocan comandos externos a través de un intérprete de comandos.

VARIABLE DEPENDIENTE

Y: SEGURIDAD DE LA APLICACIÓN WEB

La importancia de la seguridad y la administración de vulnerabilidades de una aplicación Web es de suma importancia, y los equipos de desarrollo y seguridad deben identificar las vulnerabilidades en la etapa de desarrollo y en producción para arreglarlas de manera rápida y eficiente.

Los expertos en seguridad informática, deben analizar vulnerabilidades, técnicas y lógicas en Websites de producción y desarrollo, utilizando herramientas automatizadas, para ver la seguridad y de ser requerido (en ocasiones estas herramientas detectan algunas vulnerabilidades, más no todas), técnicas de análisis manuales realizadas por nuestros expertos.

INDICADORES

Y1: ATAQUES ANÓNIMOS

Los ataques anónimos, donde tratan de obtener información confidencial analizando las comunicaciones entre 2 computadoras.

Y2: ATAQUES A LA INTEGRIDAD DE DATOS

Es cuando en este tipo de ataque, se altera la información en tránsito con fines maliciosos.

Y3: ATAQUES A LA BASE DE DATOS

Los ataques de inyección de código (SQL), atacan los sitios web que dependen de bases de datos relacionadas. Pues en este tipo de ataques a las páginas Web, los parámetros se pasan a la base de datos como una consulta de SQL. Mientras un hacker puede modificar la consulta para acceder a toda la base de datos e incluso modificar su contenido.

Y4: ATAQUE A LA MANIPULACIÓN DE URL

Es un conjunto de técnicas, que atacan la lógica de negocio de la aplicación, tomando ventaja del uso de campos ocultos o fijos para la transferencia de información sensible entre browser y servidor. En particular, tags ocultos en un form, cookies o parámetros anexados a la URL son fácilmente modificables por un atacante.

Y5: ATAQUES A LA AUTENTICACIÓN DE SESIÓN

La autenticación puede ser definida como un proceso mediante el cual se intenta verificar y asegurar la identidad de un usuario, es decir, que el usuario realmente sea el dueño de la cuenta con la que se está firmando a una aplicación.

Relacionando las definiciones anteriores, se definen los ataques a la autenticación de sesión como el robo o plagio de identidad al firmarse en una aplicación, correo o sistema de seguridad con un acceso que no es propio.

3.4.2 DEFINICION OPERACIONAL DE LAS VARIABLES

VARIABLE INDEPENDIENTE

X: Técnicas de protección

INDICADORES

X1: Encriptación de datos

X2: Criptografía asimétrica (Secure Socket Layer)

X3: Técnicas de autenticación

X4: Técnicas contra ataques Inyección SQL

X5: Técnicas contra ataques de ejecución de comandos

VARIABLE DEPENDIENTE

Y: seguridad de la aplicación web

INDICADORES

Y1: Ataques anónimos

Y2: Ataques de integridad de datos.

Y3: Ataques a la base de datos.

Y4: Ataque a la manipulación de URL

Y5: Ataques a la autenticación de sesión

3.5 TECNICAS E INSTRUMENTOS PARA LA RECOLECCION DE DATOS

Se hará uso de las técnicas de entrevista, encuesta a los administradores de los webs master en el área de informática de las instituciones de la ciudad de Ayacucho, y el análisis documental para identificar los peligros y establecer los mecanismos para evitar y/o reducir las vulnerabilidades para posteriormente definir el nivel de riesgo al que expondrá la aplicación web.

En el anexo A del Capítulo V podemos encontrar los instrumentos para recolectar los datos e información.

3.7 HERRAMIENTAS TECNOLOGICAS

Software	Versión	Descripción
Win32OpenSSL_Light	1.0.0	OpenSSL provee muchas operaciones criptográficas como encriptación y desencriptación de datos, creación y

		verificación de resúmenes (digests), cálculo de pares de claves públicas y privadas, manipulación de certificados, etc.
Captcha 1.1	1.5	Es una librería que es utilizada para evitar que los robots, puedan utilizar ciertos servicios en una aplicación web, como enviar mensajes basuras, etc.
Acunetix	1.7	Acunetix Web Vulnerability Scanner es un escáner de vulnerabilidades en aplicativos Web (CMS, Sistemas de Información Web, Web dinámicas etc.), permite llevar a cabo de manera automática detecciones de vulnerabilidades del tipo SQL Injection, XSS, Cifrado, descubrimiento de directorios, etc.
Netbeans IDE	6.9	Interfaz de desarrollo integral. Herramienta para programadores pensada para escribir, compilar, depurar programas. Escrito en Java pero sirve para cualquier otro lenguaje.
PostgreSQL	1.10.0	Sistema Gestor de base de datos.
J2EE	1.6	Herramienta para desarrollo de aplicaciones utilizando tecnologías como: Servlets, JSP, Java beans. Permite desarrollo de aplicaciones multicapas.
Apache tomcat	7.0.8	Servidor de aplicaciones de Sun Microsystems que implementa las

		tecnologías definidas en la plataforma Java E.E y permite ejecutar aplicaciones que siguen esta especificación.
AESCrypt	3.0.8	Es una aplicación que encripta ficheros en AESCrypt y es útil para autenticidad de los datos.

Tabla N° 3.1 – Herramientas utilizadas para realizar la auditoria de las aplicaciones web

3.7 TECNICAS PARA PROCESAR LA INFORMACION

a. ALGORITMO DE ENCRIPTACION DE DATOS

Se implementará un software, que permita realizar el cifrado y descifrado de datos aplicando el cifrado RSA, se desea tener seguridad informática; puede ser también entendida como un medio para garantizar las propiedades de confidencialidad, integridad y disponibilidad de los recursos de un sistema.

b. CRIPTOGRAFIA ASIMETRICA (Secure Socket Layer) EN SERVIDORES

Se usara la firma digital para poder dar seguridad, existen organizaciones, llamadas autoridades certificadoras, que se encargan de comprobar si una llave pública específica es propiedad de un individuo u organización en particular. Como resultado de esta comprobación emiten un "Certificado de llave pública" que contiene el nombre de la persona, la llave pública, número de serie, la fecha de creación del certificado y la fecha de vencimiento.

El certificado comprueba que una llave pública especificada es propiedad de un individuo u organización en particular.

Si la llave privada del cliente ha sido violada, si la persona ha dado datos incorrectos o si hay copias falsas de ésa llave, el certificado puede ser revocado. Las llaves revocadas que aún no están vencidas son colocadas en una lista de Revocación de Certificados.

También se utilizara el KeyTool de Java, para poder crear nuestro certificado

digital, para poder dar seguridad a nuestra aplicación, en el servidor de aplicaciones.

c. TECNICAS DE AUTENTICACION

Implementaremos una técnica de autenticación, usando métodos eficaces en la generación de contraseñas para cada usuario, utilizando la criptografía MD5 y de esta manera acceder al sistema de modo más seguro.

Se implementara un generador de claves de texto en imágenes de forma aleatoria, y para esto utilizaremos la librería captcha versión 1.1, que se encuentra distribuida en la red, donde usaremos con nuestro lenguaje de programación, en este caso Java.

Se implementara la utilización de un teclado virtual, para ingresar la clave de acceso al sistema, utilizando JavaScript.

d. TECNICAS CONTRA ATAQUES DE INYECCION DE CODIGO

La técnica más adecuada para la protección contra Inyección SQL, es utilizando procedimientos almacenados, para poder proteger los datos, donde se tendrá las variables de alcance para imposibilitar la concatenación de instrucciones SQL, donde puedan aplicarse variables en las instrucciones anexadas.

La validación de la entrada es necesaria una validación fuerte en lado de servidor para entrada de usuario, validación de datos filtrar la entrada del usuario de caracteres SQL, verificar tanto el tamaño como el tipo de los datos y sintaxis de las entradas de usuario. Este punto se aplica a muchos ataques similares, en particular lo reafirmaremos para los próximos ataques analizados.

e. TECNICAS CONTRA MANIPULACION DE PARAMETROS

La configuración más adecuada contra la manipulación de parámetros, para la protección, es utilizando el Fron Controller de java avanzado, donde podremos esconder la URL de nuestro navegador.

CAPITULO IV

ANALISIS Y RESULTADOS DE LA INVESTIGACION

4.1 RESULTADOS

4.1.1 ENCRIPTACION DE DATOS - TECNICAS A LA CONFIDENCIALIDAD

a) ALGORITMO DE ENCRIPTACION DE DATOS

Para realizar el encriptado, se seleccionan dos numero primos largos, p y q de forma aleatoria y se calcula el producto de n=p*q. Donde n se le denomina módulo.

Se elige un número d, menor que el valor n

Calculamos un número entero e que esté dentro del rango

• $1 \leq e \leq (p-1)(q-1)$ con la siguiente operación:

•e d=1 mod [(p-l)(q-l)]

• A d y e se les conoce como los exponentes privado y público respectivamente.

calculando anteriormente y que sea relativamente primo al producto de (p-l)(q-l).

La clave pública la formarán la pareja (n,e).

• La clave privada viene dada por (n,d).

• Con la clave pública procederíamos a encriptar el mensaje deseado y con la clave privada se desencriptaría.

El factor p y q debe ser guardado en secreto o destruido, para evitar que cualquiera intente reventar el sistema, ya que podría obtener la clave privada.

Presumiblemente es difícil obtener la clave privada, d, a partir de la clave pública (n,e).

• Si pudiéramos factorizar n en p y q, podríamos obtenerla.

• La seguridad del RSA se basa en la idea de que esta factorización es sumamente complicada de realizar.

La relación entre e y d asegura una correcta reconstrucción de los mensajes.

• Debido a que solo el receptor conoce la clave secreta d, solamente el puede desencriptarlo. Es responsabilidad el usuario guardar la clave secreta.

- El cifrado de mensajes tiene lugar sin ningún tipo de compartición de claves privadas.

• Cada persona usa solamente la clave pública del receptor al que quiere enviar el mensaje y su propia clave privada

b) ALGORITMO PARA GENERAR LAS CLAVES RSA

Supongamos que "Javier" desea permitir que "Nilo" le envíe un mensaje cifrado sobre un canal inseguro. Lo primero que ha de hacer es generar los pares de claves pública (n,e) y privada(n,d):

1. Elegimos dos números grandes para *p* y *q* (entre más grandes es más segura la encriptación, pero es más demorado el proceso de encriptar/desencriptar) que sean diferentes y totalmente independientes el uno del otro. Calculamos $n=p*q$
2. Calculamos la función de Totient den: $totient(n)=(p-1)*(q-1)$
3. Elegimos un entero e, de tal forma que $1 < e < totient(n)$ y además que sea con primo con $totient(n)$.
4. Calculamos un *d*, de tal forma que $d*e= 1 \bmod totient(n)$
5. La clave pública será entonces (*n*,e) y la privada (*n*,*d*)
6. En dicho caso "Javier" puede compartir su clave pública con "Nilo" y guardar celosamente la privada.

Encriptando datos

Ahora supongamos que "Nilo" desea mandarle un mensaje a "Javier", lo único que tendrá que hacer es consultar la clave pública de "Javier", dividir el mensaje que quiere enviarle, asignarle un alfabeto numérico a cada trozo y calcular para cada división: $c=ne \bmod n$

Desencriptando

Con el mensaje que le ha llegado a "Javier", lo que tiene que hacer es dividirlo y usar su clave privada para calcular: $n=cd \bmod n$

c) IMPLEMENTACION DEL ALGORITMO RSA PARA EL CIFRADO EN JAVA

El código utilizado para esta función es el indicado para el cifrado de texto y a continuación se detalla.

Pero el algoritmo para esta implementación realizada en java, se tiene en el apartado a, donde se detalla el algoritmo.

```
public BigInteger[] cifrarTextoClaro(String mensaje, BigInteger ee, BigInteger nn) {
    int i;
    byte[] temp = new byte[1];
    byte[] digitos = mensaje.getBytes();
    BigInteger[] bigdigitos = new BigInteger[digitos.length];
    JOptionPane.showMessageDialog(null, digitos.length );

    for (i = 0; i < bigdigitos.length; i++) {
        temp[0] = digitos[i];
        bigdigitos[i] = new BigInteger(temp);
    }

    BigInteger[] encriptado = new BigInteger[bigdigitos.length];

    for (i = 0; i < bigdigitos.length; i++) {
        encriptado[i] = bigdigitos[i].modPow(ee, nn);
    }

    return (encriptado);
}
```

d) FUNCIONES REALIZADAS PARA EL DESCIFRADO

El algoritmo se mostró en el apartado a para realizar el descifrado de texto.

```
public String desCifrarTextoCifrado(BigInteger[] txtcifrado, BigInteger dd, BigInteger nn)
    BigInteger[] desencriptado = new BigInteger[txtcifrado.length];

    for (int i = 0; i < desencriptado.length; i++) {
        desencriptado[i] = txtcifrado[i].modPow(dd, nn);
    }

    char[] charArray = new char[desencriptado.length];

    for (int i = 0; i < charArray.length; i++) {
        charArray[i] = (char) (desencriptado[i].intValue());
    }

    return (new String(charArray));
}
```

e) GENERADOR DE LLAVES EN JAVA

Implementación del algoritmo, para generar las claves públicas, claves privada y clave general.

El algoritmo para generar las claves RSA se encuentra en el apartado b, donde se indica la forma de cómo realizar la generación de las llaves, tanto pública como privada.

```
public class clave {
    //donde e : es la clave publica
    //donde d: es la clave privada

    int limite;//es la potencia a la cual se eleva
    BigInteger n, q, p;
    //donde q y p son numeros primos
    BigInteger on; //declaracion de la variable TOTIENT
    BigInteger e, d;

    public clave(int limite) {
        this.limite = limite;
        generaPrimos();              //Genera p y q
        generaClaves();              //Genera e y d
    }
 public void generaPrimos() {
        p = new BigInteger(limite, 10, new Random());
        do {
            q = new BigInteger(limite, 10, new Random());
        } while (q.compareTo(p) == 0);
    }

public void generaClaves() {
    n = p.multiply(q);//Calculando n=p.q
    //Calculando ø(n) = (p-1)(q-1)
    on = p.subtract(BigInteger.valueOf(1));//
    on = on.multiply(q.subtract(BigInteger.valueOf(1)));
    //e: Seleccionando aleatoriamente la clave de cifrado
    //e |donde 1<e<ø(n), y MCD(e,ø(n))=1
    do {
        e = new BigInteger(2 *limite, new Random());
    } while ((e.compareTo(on) != -1) || (e.gcd(on).compareTo(BigInteger.valueOf(1))
   /*
    Resolviendo la siguiente ecuacion para encontrar clave de cifrado d:
     e.d mod ø(n) = 1 , y 0<d<ø(n)
    */
    d = e.modInverse(on);
}
```

Compilación del algoritmo en JAVA, luego nos envía a que Ingresemos las llaves para el cifrado.

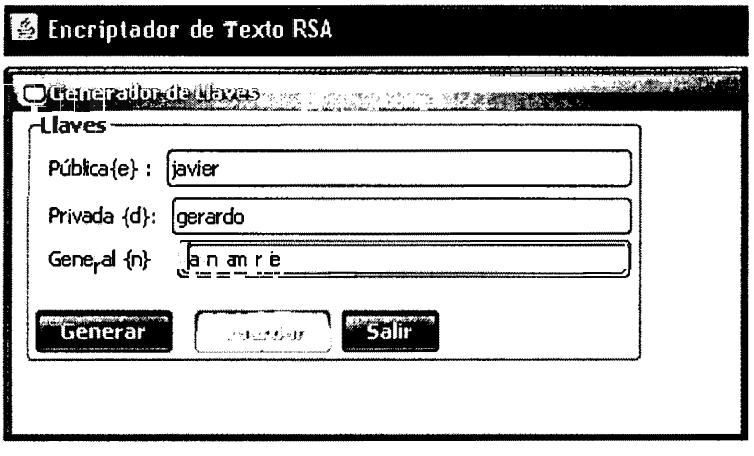


Figura N° 4.1: Generador de llaves antes de cifrar (Elaboración propia)

Resultado del cifrado de las llaves

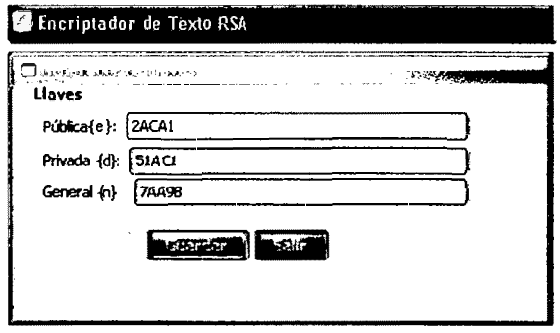


Figura N° 4.2: Generador de llaves cifrado (Elaboración propia)

ENCRIPTADO DE TEXTO

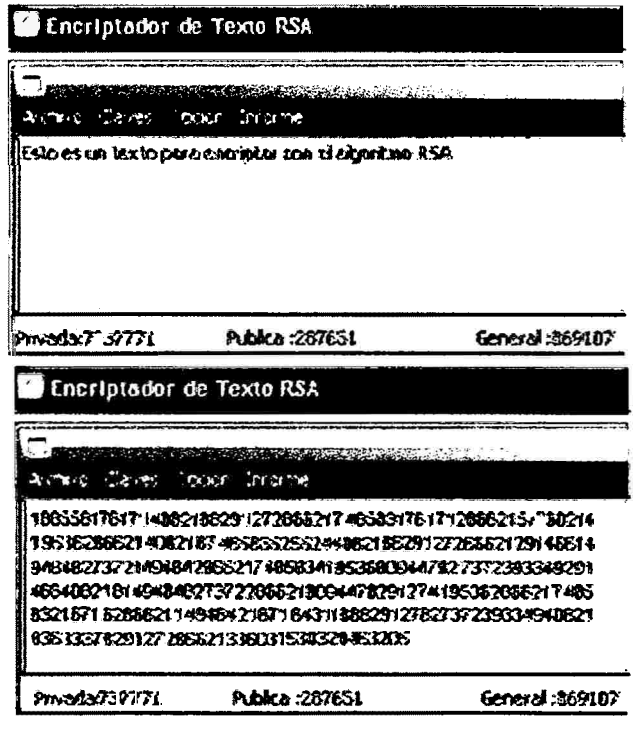


Figura N° 4.3: Texto cifrado (Elaboración propia)

4.1.2 CRIPTOGRAFIA SIMETRICA – ASIMETRICA EN EL SERVIDOR PARA PROTEGER LA DISPONIBILIDAD INTEGRIDAD Y CONFIDENCIALIDAD DE LOS DATOS.

a. INSTALAR EL SERVIDOR TOMCAT 7.0.8

1. Instalar el Apache Tomcat en la unidad c:\apache-tomcat-7.0.8
2. Configurar correctamente las variables de entorno, para poder correr el contenedor web Tomcat

   Para esto tenemos que realizar clic derecho – propiedades – pestaña

(opciones avanzadas) –clic en variables de entorno

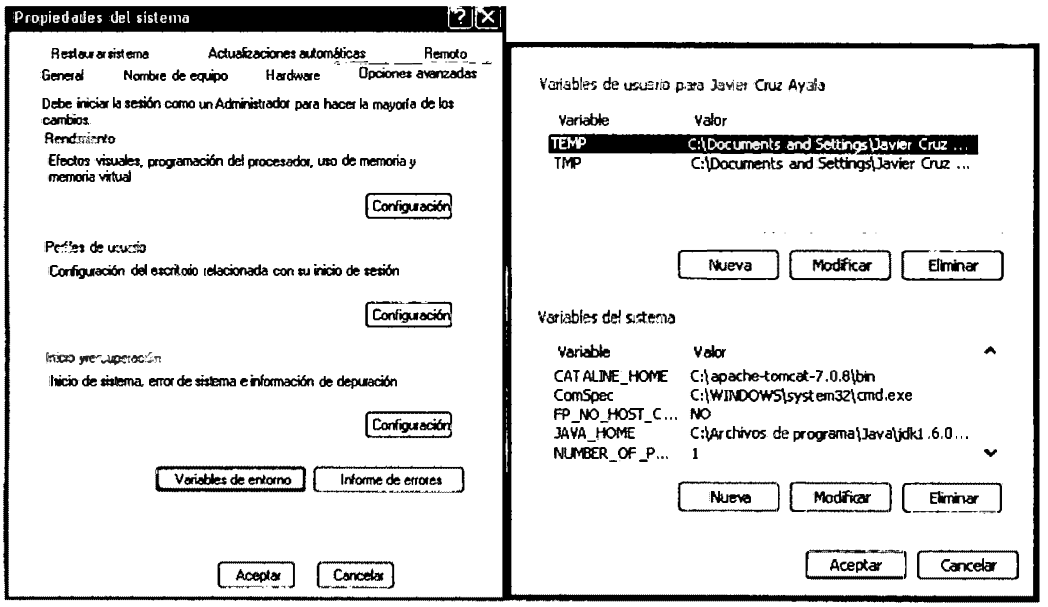


Figura N° 4.4: variables de entorno (Elaboración propia)

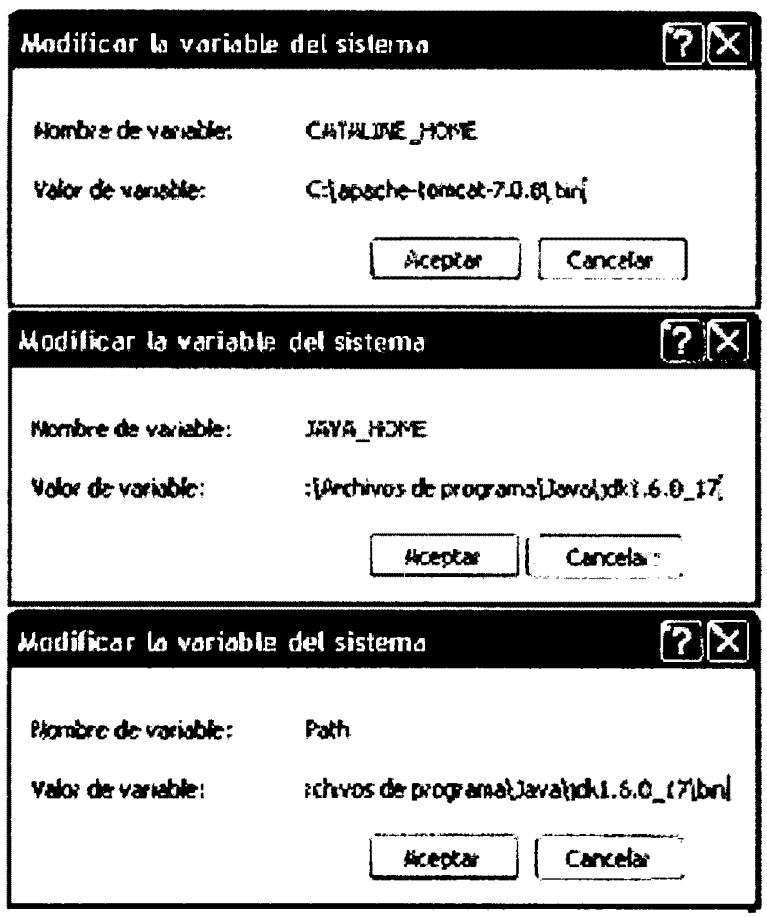


Figura N° 4.5: Modifica las variables del sistema (Elaboración propia)

3. Ejecutamos el archivo "C:\apache-tomcat-7.0.8\bin\startup.bat" para correr el servidor.

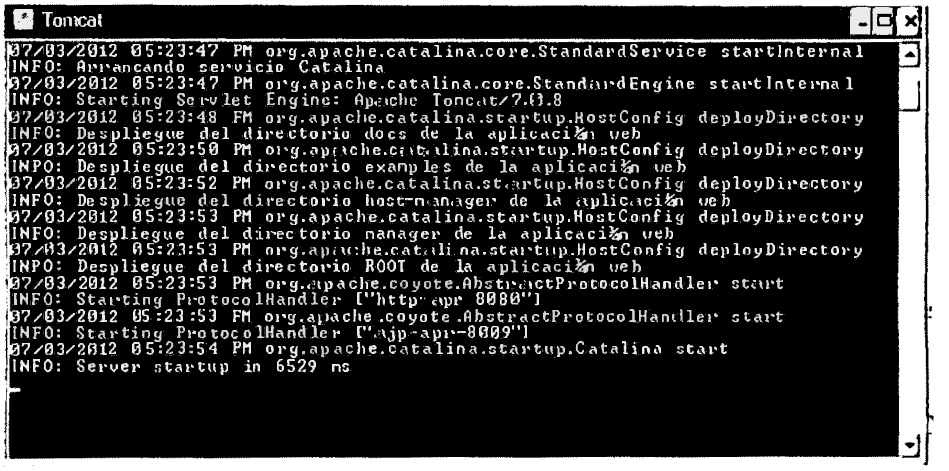


Figura N° 4.6: Startup de apache tomcat

4. Pruebe en una página de internet Explorer la siguiente dirección http://localhost:8080/.

   Si el servidor se configuro correctamente deberíamos tener como nos muestra la figura N°4.7

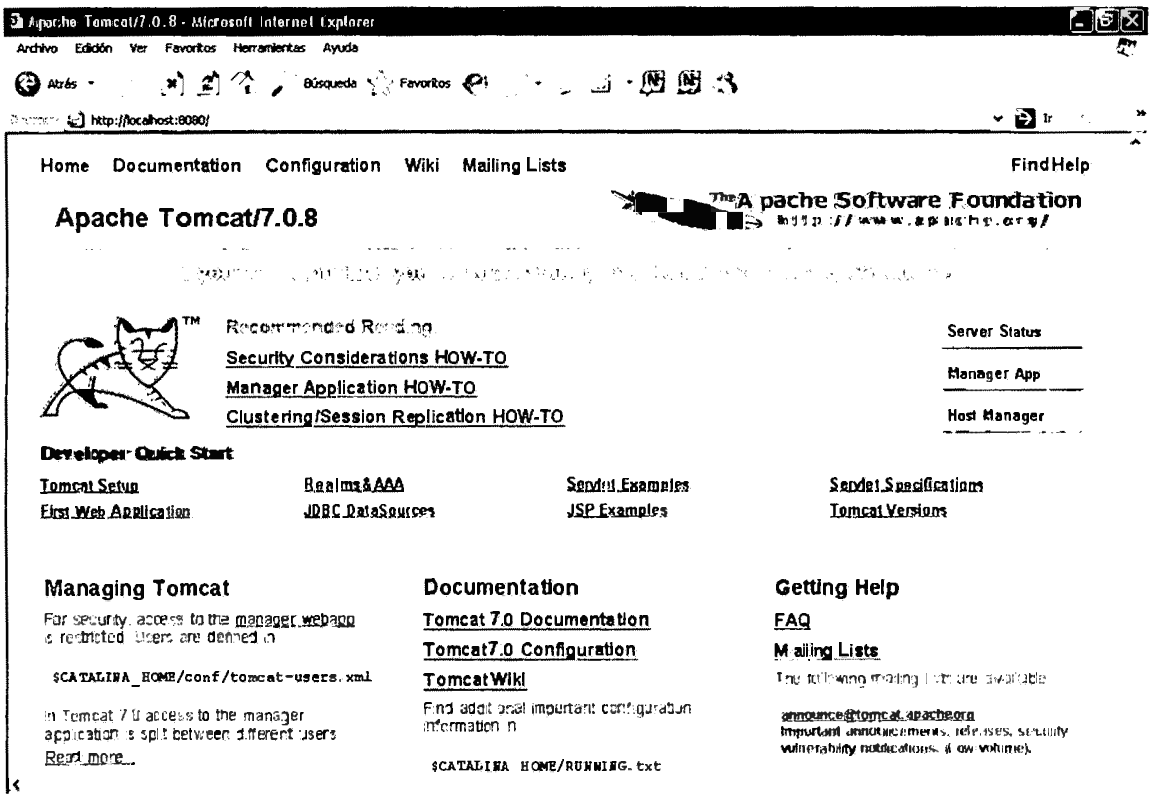


Figura N° 4.7: Localhost de apache tomcat (Elaboración propia)

b. CREAR UN CERTIFICADO SSL EN NUESTRO JAVA

Si se desarrolla una aplicación web en Java y si se va hacer uso de un recurso seguro (mediante un certificado).

Para generar el certificado SSL utilizaremos la herramienta incluida en el Java Development Kit, JDK llamada KeyTool, y para esto digitamos en la ruta del símbolo del sistema:

Keytool -genkeypair -alias MyCert -keyalg RSA -keystore "C:\MyCert.cert"

Con este comando y sus argumentos, estamos ordenando que generemos un nuevo certificado con el alias MyCert. Luego nos pedirá una serie de datos que son los que conformarán el almacén de claves. La clave que viene por defecto con el alias de MyCert es: changeit

Cuando termines de ingresar toda la información digita SI para que se valide. Para mantener la misma clave (es decir changeit) presionar Enter.

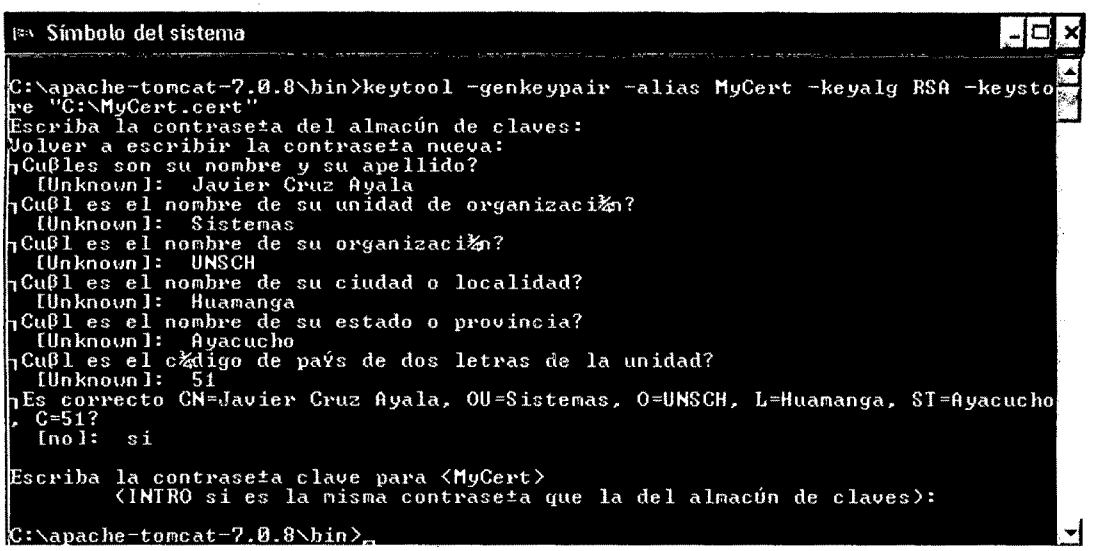


Figura N° 4.8: Keytool de java (Elaboración propia)

Al finalizar este paso el KeyTool genera un archivo que almaceno la llave, es decir un MyCert.cert.

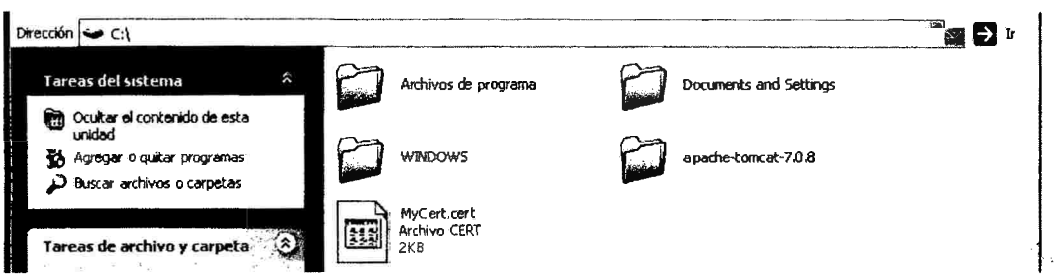


Figura N° 4.9: Fichero MyCert.cer (Elaboracion propia)

c. INSTALACION DE CERTIFICADO EN EL ALMACEN DE CERTIFICADOS DE JAVA

Con esta opción únicamente se añadirán los certificados a la instalación activa de JVM en Windows.

1. Se accederá al panel de control de java, seleccionar la pestaña Seguridad y pulsamos en el botón Certificados.

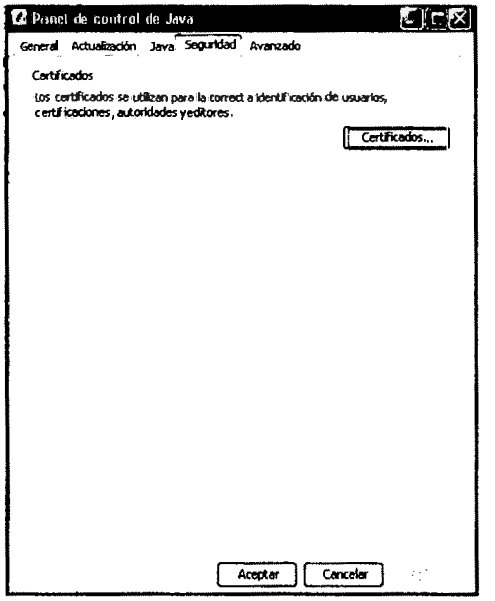


Figura N° 4.10: Pestaña de seguridad

2. En el apartado de certificados, se seleccionara la opción de importar y se localizara el archivo que contiene el certificado, y si nuestro certificado tiene la extensión .cert se tendrá que seleccionar la opción "todos los archivos".

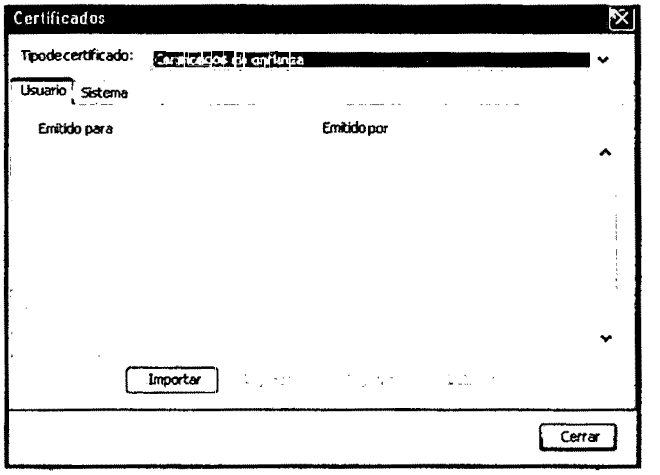


Figura N° 4.11: Ventana de certificados

3. Seleccionamos nuestro certificado, en formato PKCS#12 (extensión .p12), introducimos la clave del almacén de certificados y cerramos el panel de control.

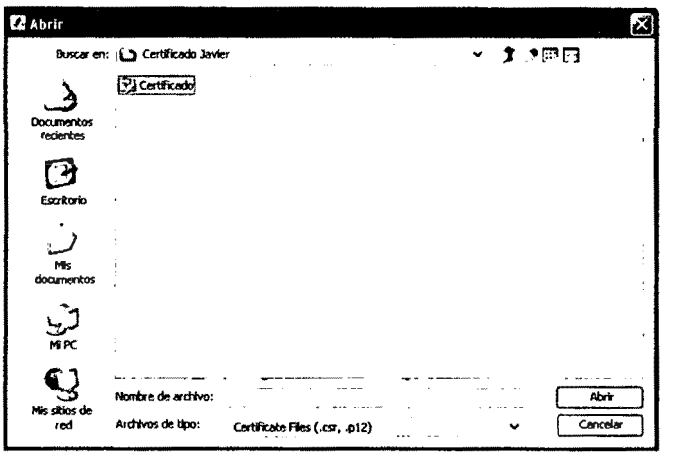


Figura N° 4.12: Ubicación de certificado de Carcert

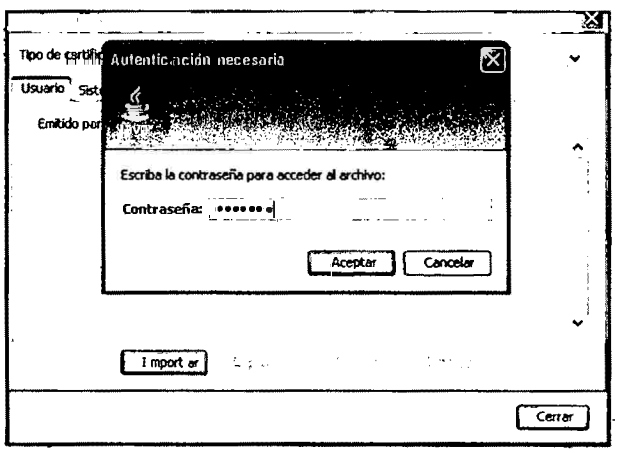


Figura N° 4.13: Autenticación necesaria (Cacert)

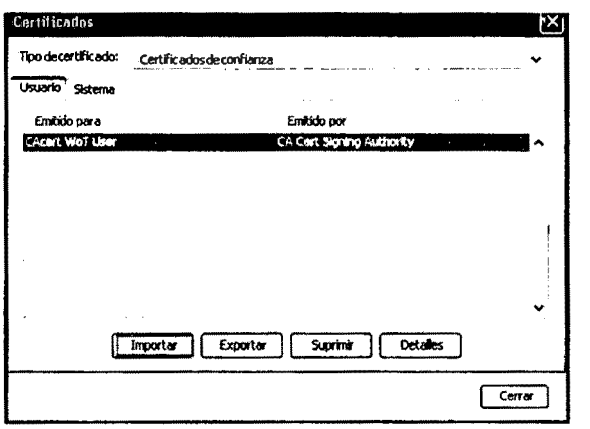


Figura N° 4.14: Certificado (Cacert)

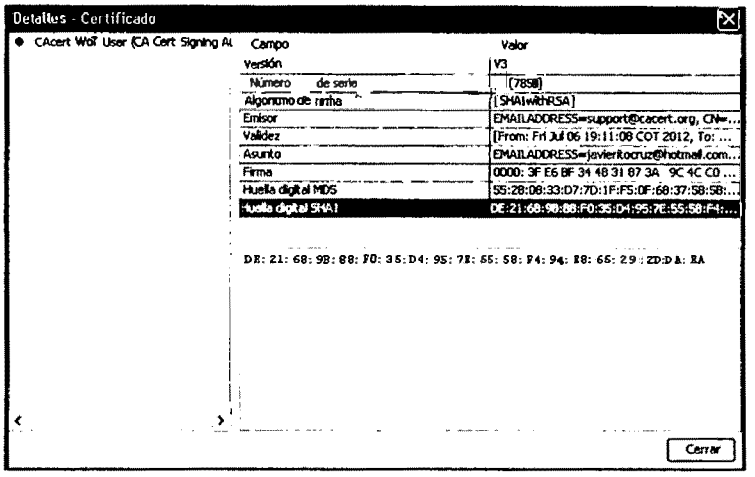


Figura N°: 4.15: Detalles del Certificado. (Cacert)

Ya estamos listos para utilizar nuestro certificado con la plataforma de firma y autenticación electrónica.

d. CONFIGURAR EL SERVIDOR APACHE TOMCAT CON EL CERTIFICADO SSL

1. configurar el server.xml del servidor apache Tomcat para cargar el protocolo https

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" SSLEnabled="true"
 maxThreads="150" scheme="https" secure="true"
 clientAuth="false" sslProtocol="TLS" keystoreFile="C:\MyCert.cert" keystorePass="changeit" />
```

2. Cargamos https://localhost:8443/ que contiene el tomcat con certificado.

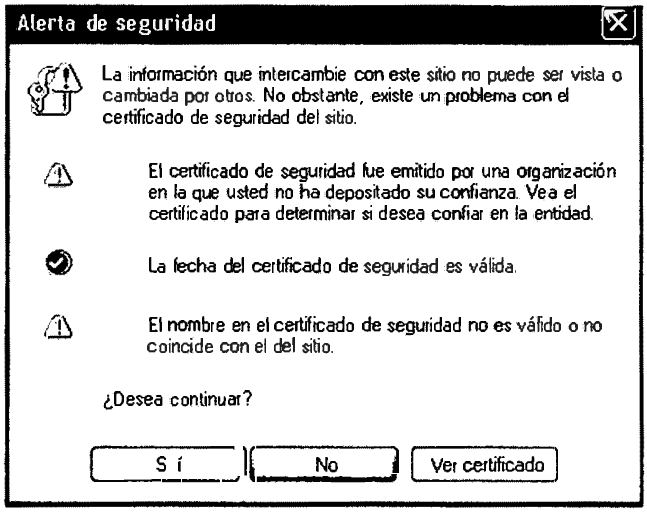


Figura N° 4.16: Alerta de seguridad

Clic en SI, e inmediatamente nos muestra la página de apache

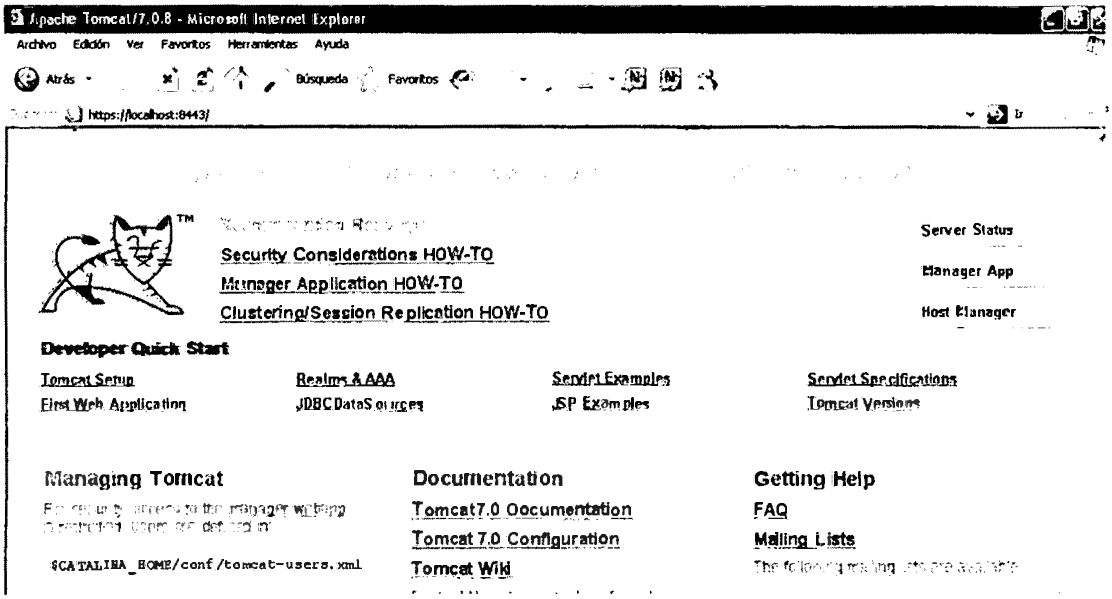


Figura N° 4.17: Localhost Apache Tomcat

3. Con el explorador Mozilla Firefox, seleccionar Entiendo los riesgos y clic en Añadir excepción.

Figura N° 4.18: Añadir conexión segura

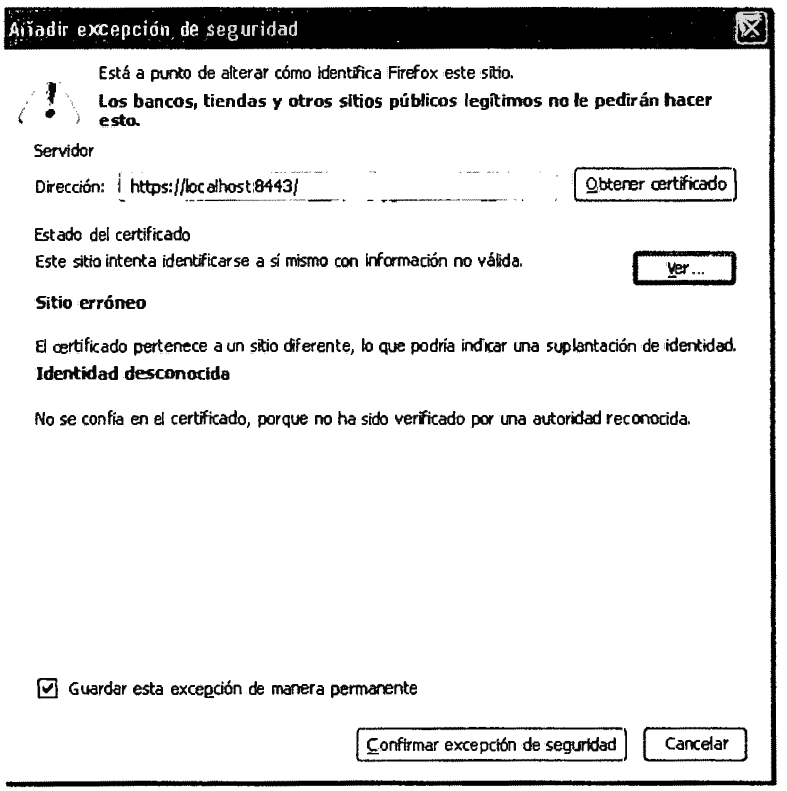


Figura N° 4.19: Excepción de seguridad

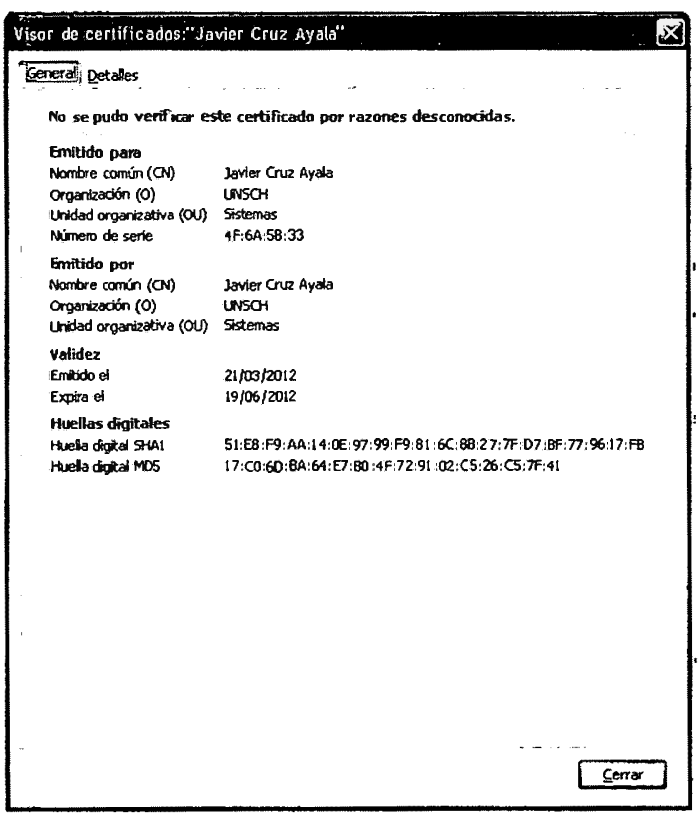


Figura N° 4.20: Detalles del certificada java

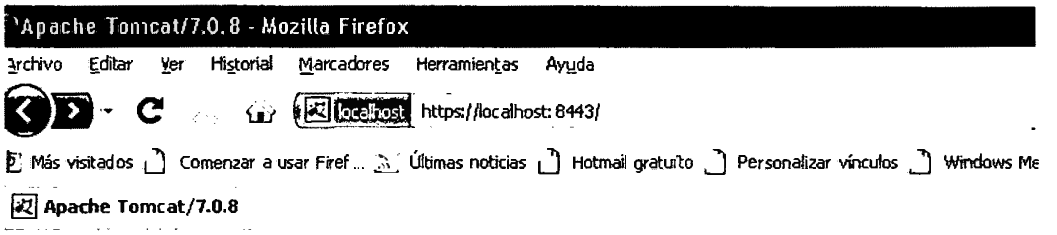


Figura N° 4.21: LocalHost modo seguro Apache Tomcat

e. AÑADIR SEGURIDAD EN NUESTRO PROYECTO BIBLIOTECA UNSCH DE JAVA PARA QUE SE CARGUE POR DEFECTO

1. Modificar el archivo Web XML de nuestro proyecto en java, para agregar las políticas de seguridad, indicar el nombre de nuestro proyecto en el nombre del recurso de la web.

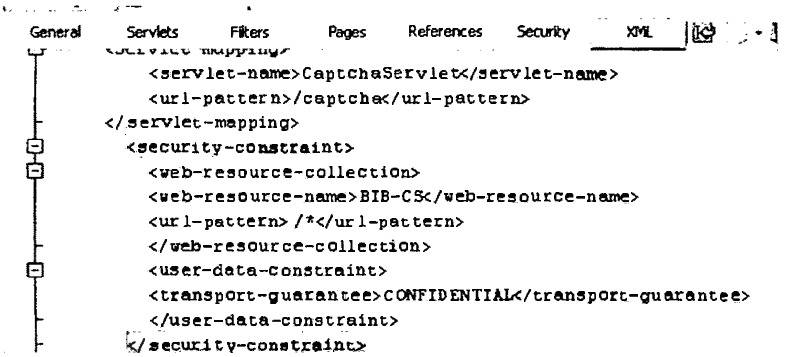


Figura N° 4.22: Configuración Web XML de java

2. Una vez cargado nuestro localhost con la seguridad, copiar nuestro proyecto BIB.war dentro de la carpeta de nuestro servidor en webapps.

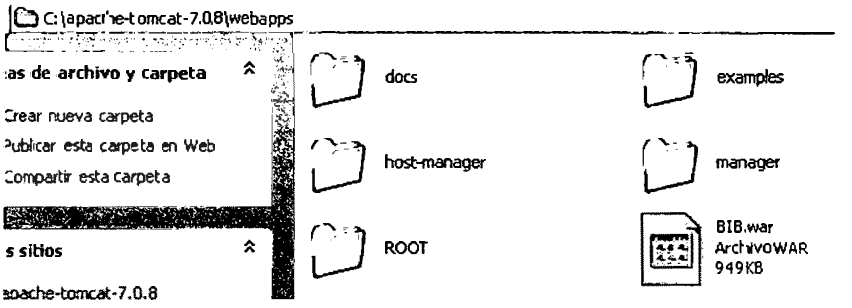


Figura N° 4.19: Configuración de LocalHost

3. Cargar nuestro https://localhost:8443/BIB/ en modo seguro

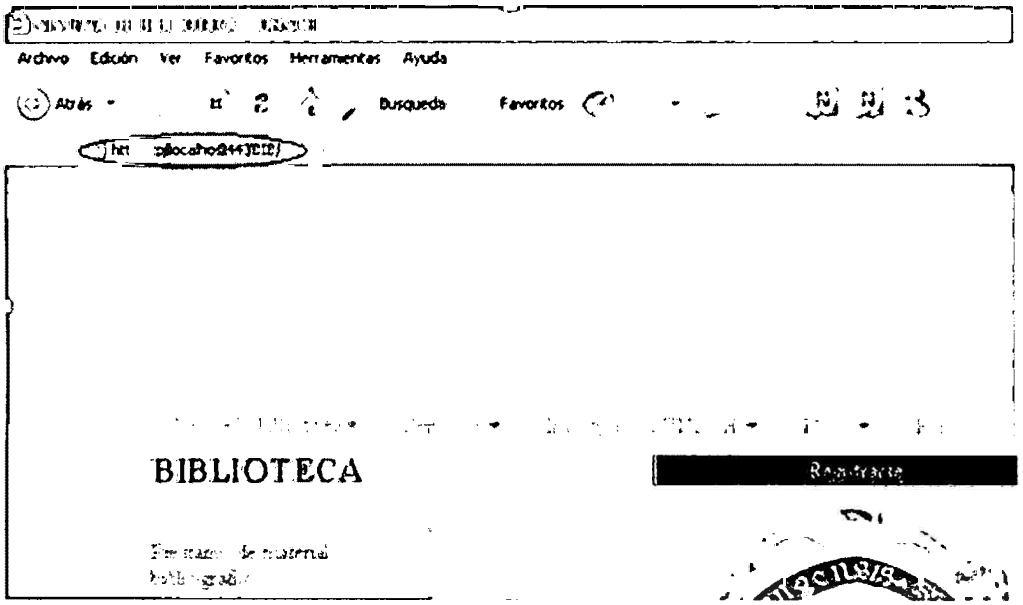


Figura N° 4.23: Pagina inicial en modo seguro. (Elaboración Propia)

4.1.3 TECNICAS DE AUTENTICACION PARA PROTEGER LA CONFIDENCIALIDAD

a. AUTENTICACION CAPTCHA CONTRA ROBOTS

Para iniciar la autenticación contra los robots se tendrá que utilizar el método de la librería SimpleCaptcha v1.1

Codigo en java CaptchaServlet.java

```
package Servlet;

import java.awt.Color;
import java.awt.Font;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import nl.captcha.Captcha;
import nl.captcha.backgrounds.GradiatedBackgroundProducer;
import nl.captcha.gimpy.DropShadowGimpyRenderer;
import nl.captcha.gimpy.GimpyRenderer;
import nl.captcha.servlet.CaptchaServletUtil;
import nl.captcha.text.renderer.ColoredEdgesWordRenderer;
import nl.captcha.text.renderer.WordRenderer;

public class CaptchaServlet extends HttpServlet
{
    private static final long serialVersionUID = 412786129726298907 8L;
```

```
    private int width = 200;
    private int height = 50;

    private final WordRenderer wordRenderer;
    private final GimpyRenderer gimpyRenderer;
    private final GradiatedBackgroundProducer bgProducer;

    public CaptchaServlet() {
        List<Font> fonts = new ArrayList<Font>(2);
        List<Color> colors = new ArrayList<Color>(3);

        fonts.add(new Font("Arial", Font.ITALIC, 48));
        fonts.add(new Font("Verdana", Font.ITALIC, 48));
        colors.add(Color.ORANGE);
        colors.add(Color.GREEN);
        colors.add(Color.RED);

        wordRenderer = new ColoredEdgesWordRenderer(colors, fonts);
        gimpyRenderer = new DropShadowGimpyRenderer();
        bgProducer = new GradiatedBackgroundProducer();
        bgProducer.setFromColor(Color.GRAY);
        bgProducer.setToColor(Color.DARK_GRAY);
    }
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);

        if (getInitParameter("width") != null) {
            width = Integer.parseInt(getInitParameter("width"));
        }
        if (getInitParameter("height") != null) {
            height = Integer.parseInt(getInitParameter("height"));
        }
    }

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse resp) {
        Captcha.Builder builder = new Captcha.Builder(width, height);
        builder.addText(wordRenderer);
        builder.addBackground(bgProducer);
        builder.gimp(gimpyRenderer);
        builder.gimp();
        builder.addNoise();
        builder.addBorder();

        Captcha captcha = builder.build();
        CaptchaServletUtil.writeImage(resp, captcha.getImage());

       req.getSession().setAttribute(Captcha.NAME, captcha);
    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {


        String respuesta = req.getParameter("respuesta");
        Captcha captcha = (Captcha) req.getSession().getAttribute(Captcha.NAME)

        if ((captcha != null) && captcha.isCorrect(respuesta)) {
            req.setAttribute("resultado","OK");
            getServletContext().getRequestDispatcher("/prestamo.jsp").forward(req, resp);
        }
        else{
        getServletContext().getRequestDispatcher("/index.jsp").forward(req, resp);
        }
    }
```

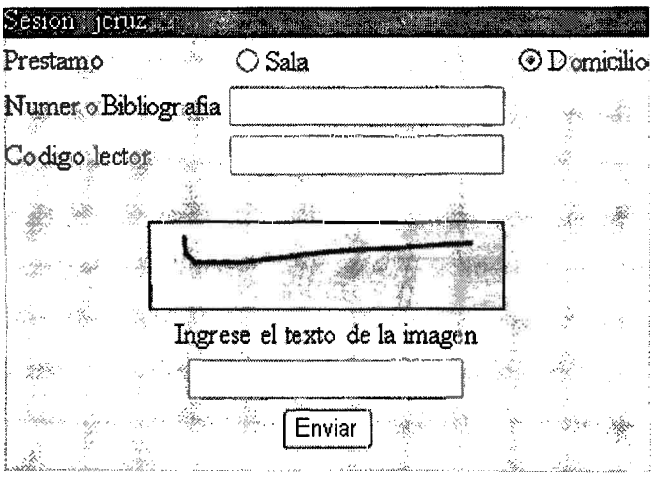


Figura N° 4.24: Imagen captcha. (Elaboración Propia)

b. AUTENTICACION TECLADO VIRTUAL

Crearemos un JavaScript, que tendrá la función de realizar el ingreso del teclado virtual en la que almacenara la clave de usuario.

Antes que nada crearemos una página para el inicio de sesión mediante un formulario, donde insertaremos un teclado virtual, y deshabilitaremos la caja de texto para el ingreso de la clave.

Indexiniciarsesion.jsp

```
<form action="<%=request.getContextPath()%>/SLoginBibliotecario" method="post">
    <center>
        <table border="0">
            <tr>
            <td colspan="2" bgcolor="#330000">
             <div align="center"><font color="#FFFFFF">Autenticacion de Bibliotecario</font></div></
            </tr>
            <tr>
                <td>Usuario: </td><td><input type="text" name="usuario" value=""></td>
            </tr>
            <tr>
                <td>Password: </td><td><input type="password" name="password" value="" id="campo_cla
            </tr>
            <tr>
                <td colspan="2" align="right"><center><input type="submit" value="Iniciar sesion"></
            </tr>
        </table>
        <table>
            <tr>
                    <td align="center">
                        <div id="contenedor"></div>
                    </td>
            </tr>
        </table>
```

Teclado.js

```
function AsignaValor(Nombre,Valor){
    var Campo= document.getElementById(Nombre);
    if(Valor==""){
        Campo.value="";
    }else{
        if(Campo.value!=""){
            Campo.value = Campo.value + Valor;
        }else{
            Campo.value = Valor;
        }
    }
}

function randOrd(){
    return (Math.round(Math.random())-0.5);
}

function marcador(Div,Nombre){
    var resultado = "";
    var num  = new Array('1','2','3','4','5','6','7','8','9','0');
    var key  = new Array('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T',
    num.sort(randOrd);
    key.sort(randOrd);
    resultado = "<table cellpadding='1' cellspacing='1' width='100'>";

var fin= 10;
resultado += "<tr bgcolor='#F0F7FD'>";
for ( var n=0; n<10; ++n ){
    resultado += "<td align='center'><input type='button' onclick=\"AsignaValor('"+Nombre+"','"+num[n
}
resultado += "</tr>";
for ( var i=0; i<3; ++i ){
    resultado += "<tr bgcolor='#F0F7FD'>";
    for ( var j=ini; j<fin; ++j ){
        resultado += "<td align='center'><input type='button' onclick=\"AsignaValor('"+Nombre+"','"+k
    }
    if(j<20){
        ini = j;
        fin= ini + 10;
    }else if(j==20){
        ini = j;
        fin = ini + 6;
    }else if(j==26){
        resultado += "<td bgcolor='#1C5280' colspan='4' align='center' style='cursor:pointer;' onclic
    }
    resultado += "</tr>";
}
resultado += "</table><br>";
document.getElementById(Div).innerHTML=resultado;
```

Aquí la muestra del teclado virtual

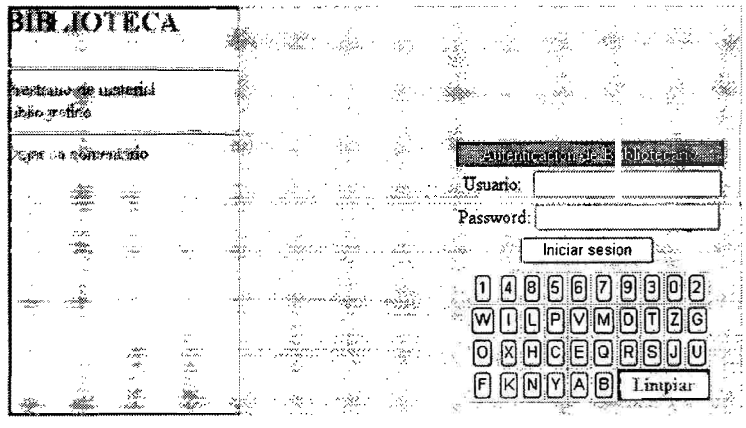


Figura N° 4.25: Teclado Virtual

4.1.3 CONFIGURACION DE ATAQUES DE INYECCION DE CÓDIGO

PROCEDIMIENTO ALMACENADO

Crear un procedimiento almacenado donde se podrá realizar un función buscar una cuenta de usuario, que se realizara en la tabla bibliotecario.

```
CREATE OR REPLACE FUNCTION spBuscarCuenta(IN character varying, IN character varying, OUT ResultadoOUT int) AS$$
DECLARE
id_btcIN ALIAS FOR $1;
pw_btcIN ALIAS FOR $2;
pw_btc character varying;
id_btc character varying;

BEGIN
SELECT bibliotecario.id_btc, bibliotecario.pw_btc
INTO id_btc, pw_btc
FROM bibliotecario
WHERE bibliotecario.estado = 1;
IF id_btc = id_btcIN AND pw_btc = pw_btcIN THEN
        ResultadoOUT:=1;
ELSE
        ResultadoOUT:=0;
END IF;
RETURN;
END;
$$ LANGUAGE 'plpgsql';
```

4.1.4 INSTALACION DEL CERTIFICADO RAIZ CLIENTE - SERVIDOR

Para la instalación del certificado raíz, debemos tener instalador el Explorador Mozilla Firefox, para su posterior configuración.

Ingresamos a la página web de CACERT http://www.cacert.org/

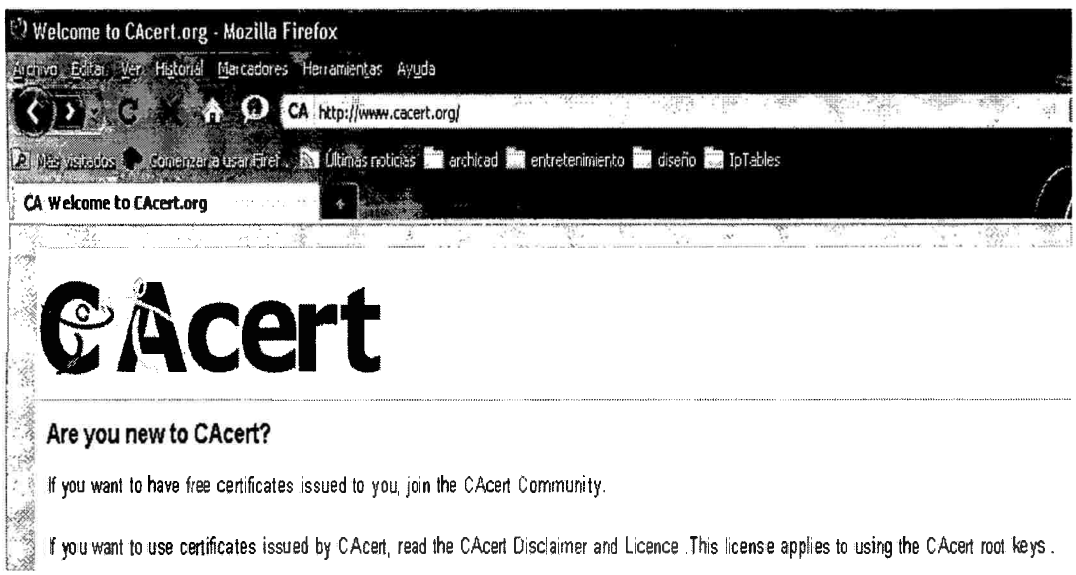


Figura N° 4.26: Pagina de CACERT

https://www.cacert.org/

supuesto un nuevo enfoque sobre las actividades a realizar por las empresas mineras en orden a los temas de seguridad y salud ocupacional, lo cual representa, en la mayoría de los casos, una reorganización e intensificación de la actividades preventivas.

En el D.S. 055-2010-EM, se establece que las empresas mineras deben cumplir las normas establecidas para:

a) Desarrollar una cultura preventiva de seguridad y salud combinando el comportamiento humano con la preparación teórico-práctica de los métodos de trabajo.
b) Practicar la explotación racional de los recursos minerales, cuidando la vida y la salud de los trabajadores y el medio ambiente.
c) Fomentar el liderazgo, compromiso, participación y trabajo en equipo en lo que respecta a la seguridad y salud.
d) Fomentar entre todos los trabajadores una cultura de seguridad y salud en el trabajo, que implica comprometerse con sus compañeros, el trabajo y la propia empresa.
e) Promover el conocimiento y fácil entendimiento de los estándares, procedimientos y reglas para realizar trabajos bien hechos mediante la capacitación.
f) Promover el cumplimiento de las normas de seguridad y salud ocupacional aplicables, disposiciones legales vigentes y los conocimientos técnicos profesionales de la prevención.

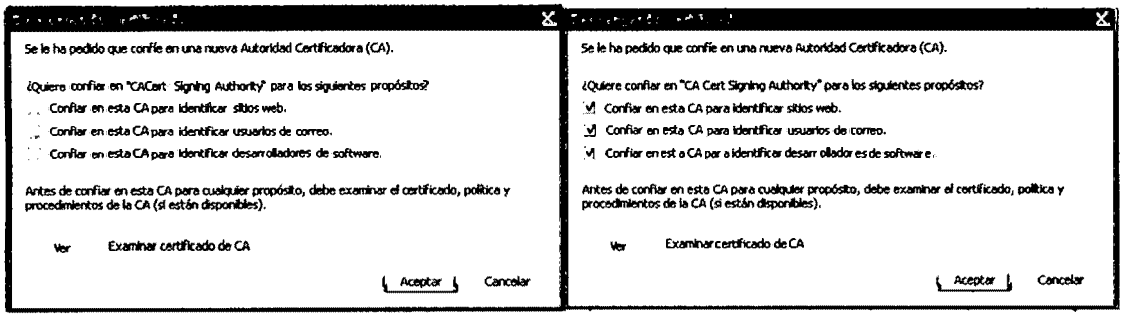


Figura N° 4.30: Pagina de descarga de CACERT

VERIFICACION DEL CERTIFICADO RAIZ

Figura N° 4.31: Configuración de la opciones de internet

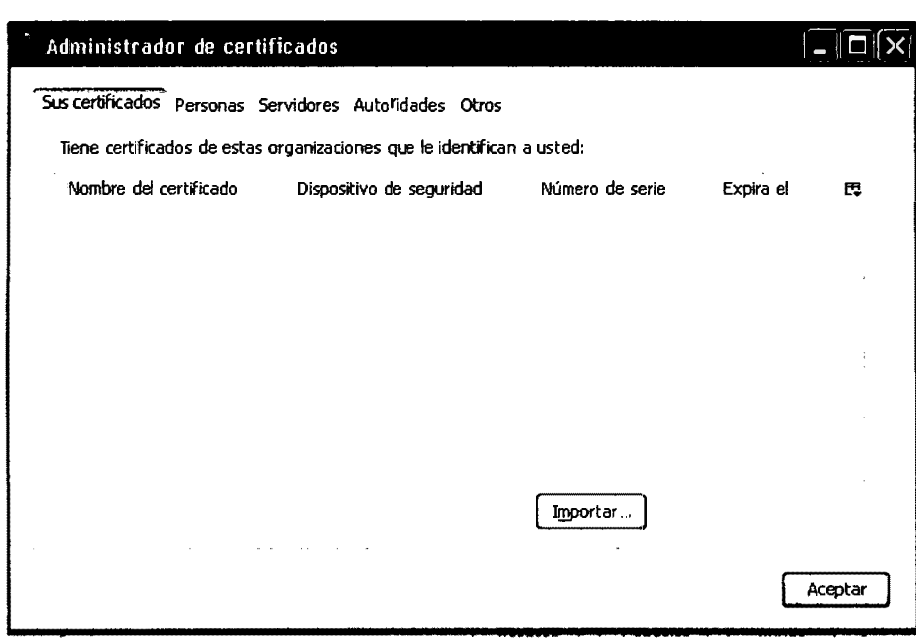


Figura N° 4.32: Certificado no existe

CREAREMOS UNA CUENTA EN EL CACERT

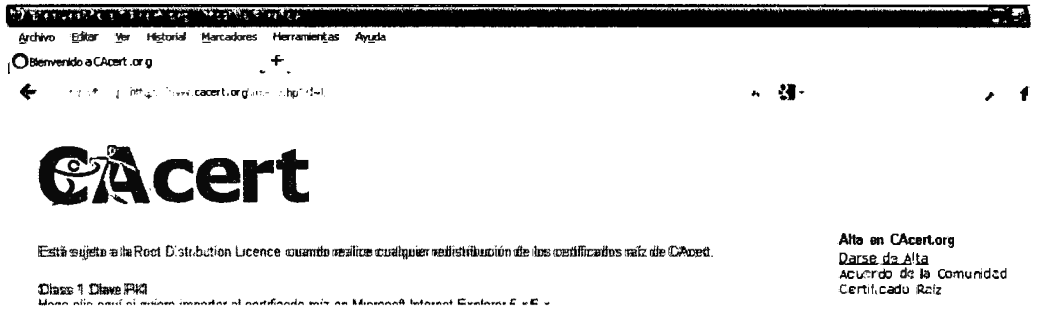


Figura Nº 4.33: Registro en CACERT

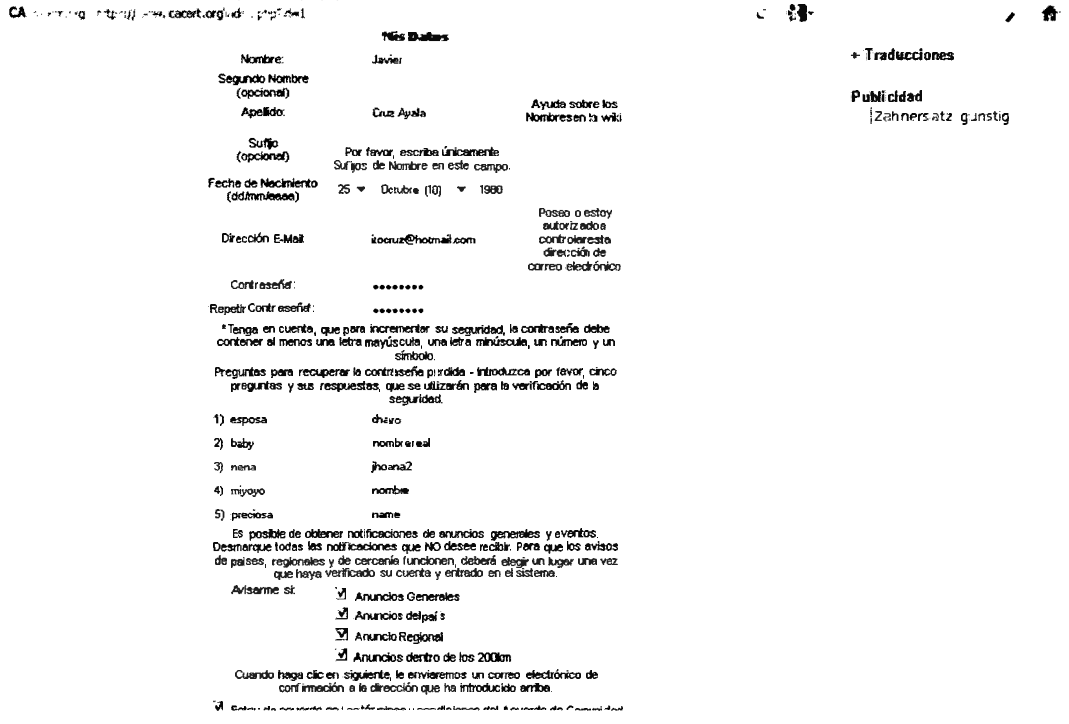


Figura Nº 4.34: Formulario de registro (CACERT)

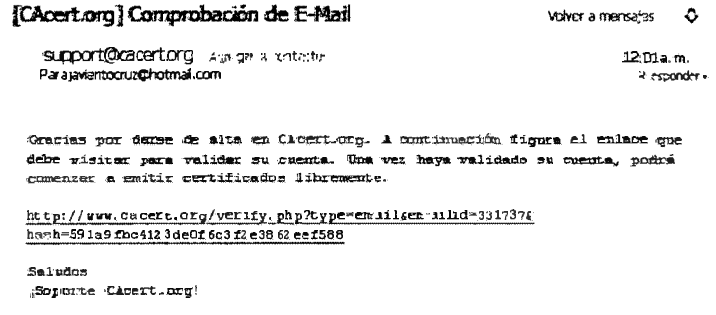


Figura N° 4.35: Confirmación de registro en el correo (CACERT)

Figura N° 4.36: Inicio de sesión CACERT

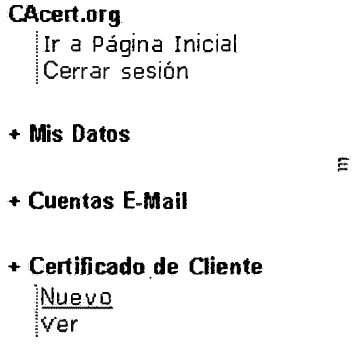


Figura N° 4.37 Certificado de Cliente (CACERT)

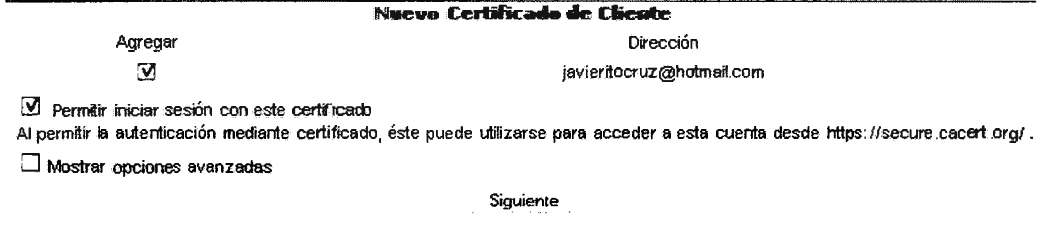


Figura N° 4.38: Nuevo certificado de CACERT

Figura N° 4.39: Crear requerimiento de certificado (CACERT)

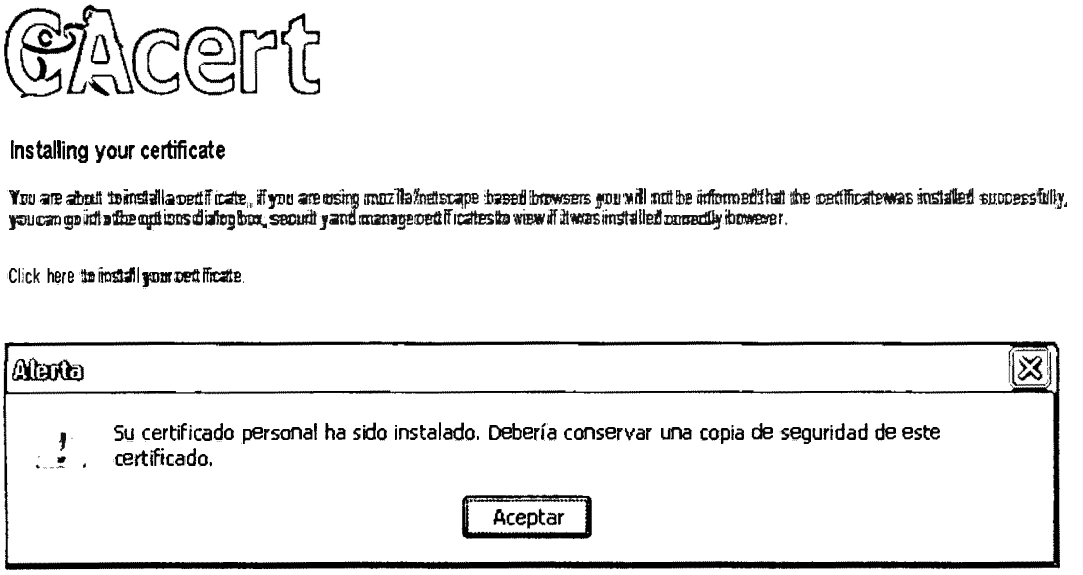


Figura N° 4.40: Instalación de certificado (CACERT)

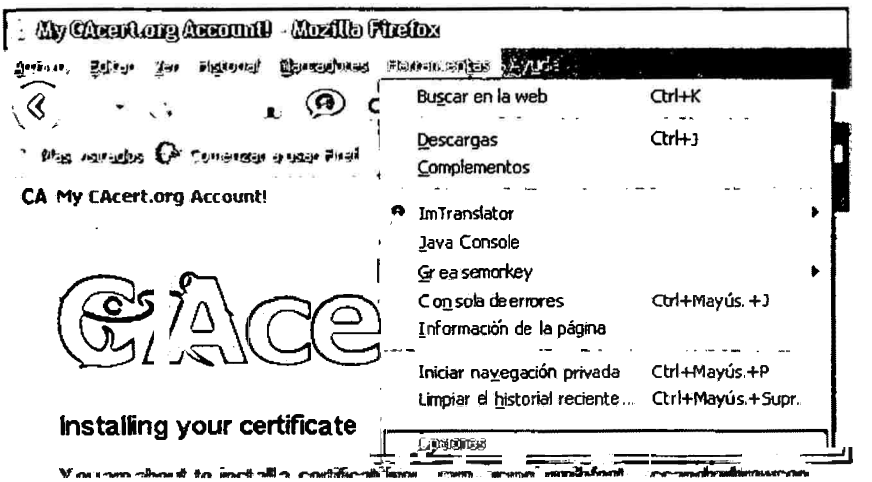


Figura N° 4.41 : Comprobación de certificado (CACERT)

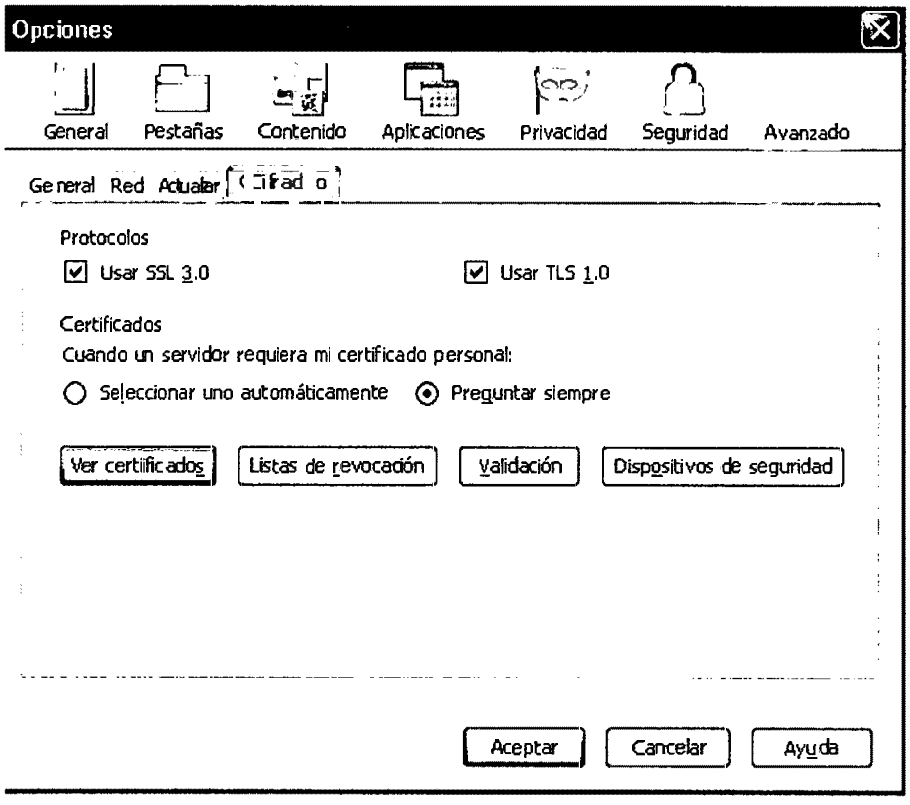


Figura N° 4.42: Ventana ver certificado (CACERT)

Revisión del administrador de certificado para su verificación.

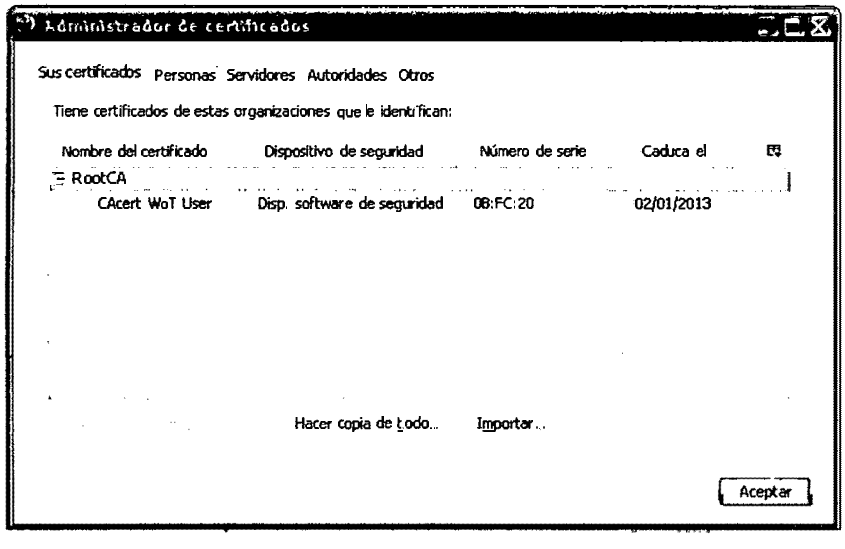


Figura N° 4.43: Pagina administrador de certificado (CACERT)

Generamos una copia de seguridad

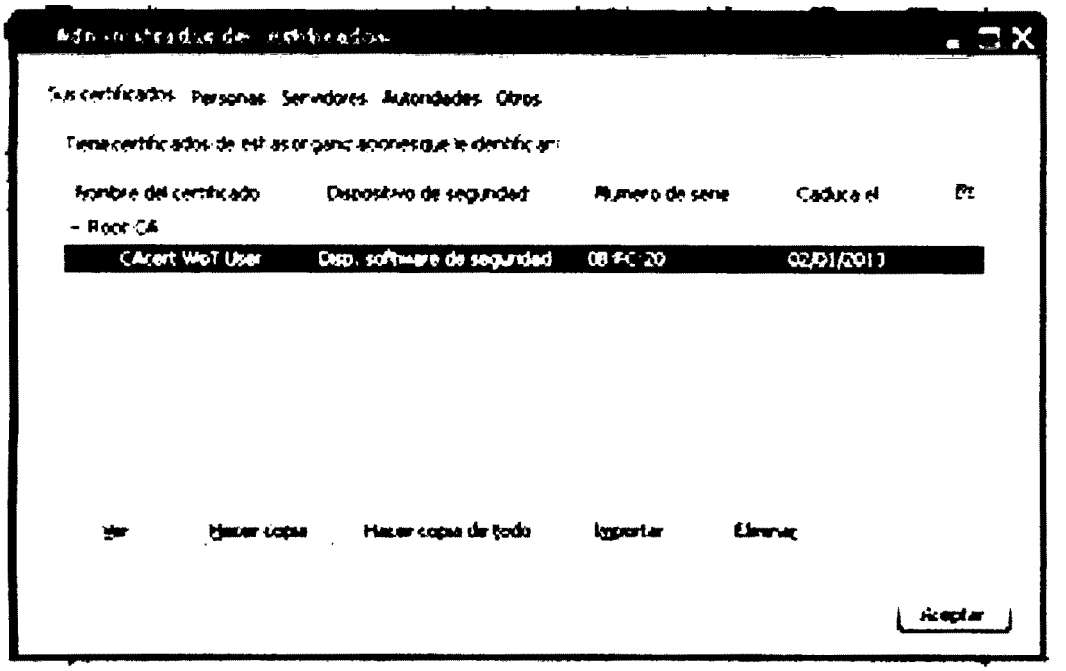


Figura N° 4.44: Realizar copia de certificado (CACERT)

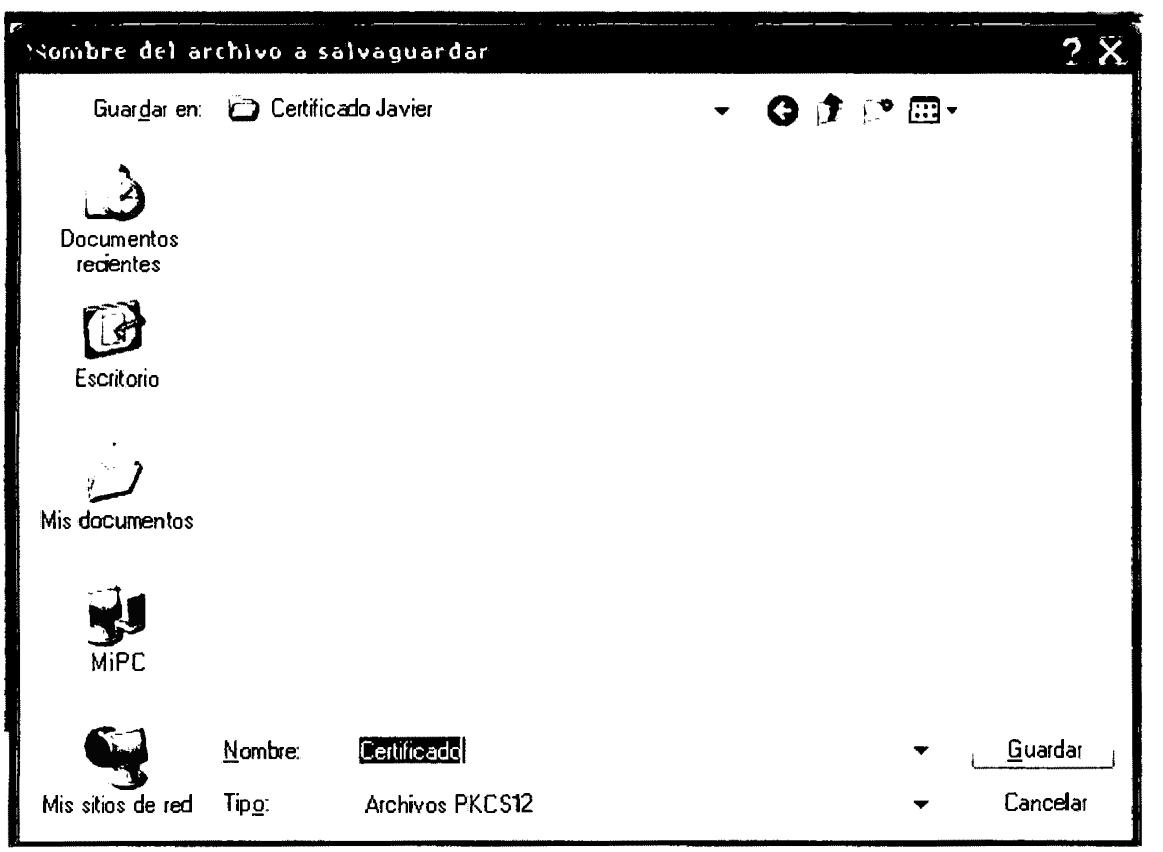


Figura N° 4.44: Ubicación de certificado (CACERT)

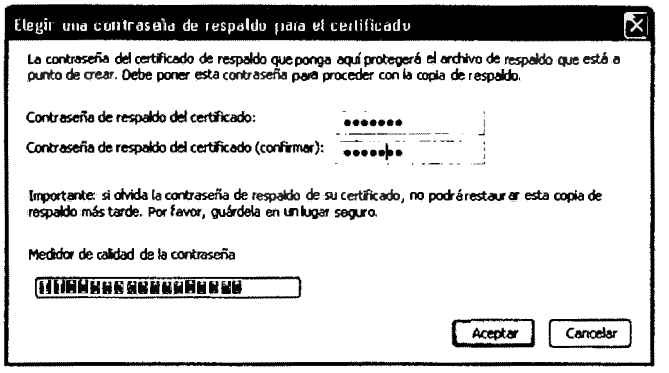


Figura N° 4.45: Contraseña de respaldo de certificado (CACERT)

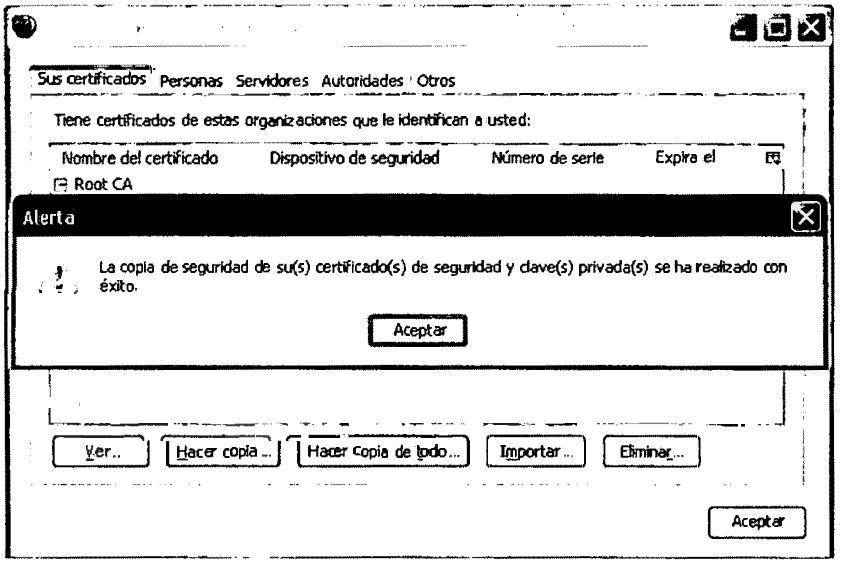


Figura N° 4.46: Confirmación de copia de seguridad de certificado (CACERT)

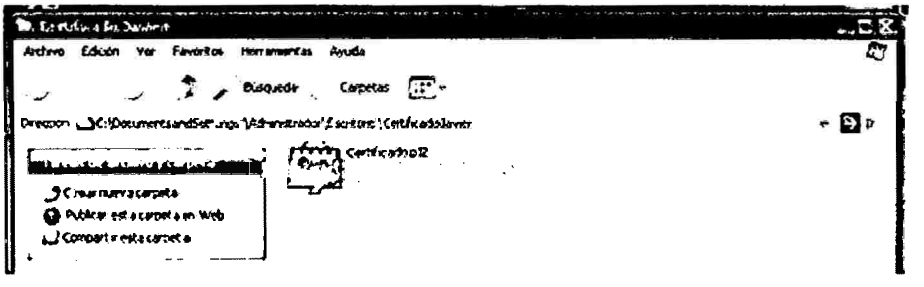


Figura N° 4.47: Certificado (CACERT)

Instalar el OPENSSL

Instalación del OpenSSL Y Configuración del Patch

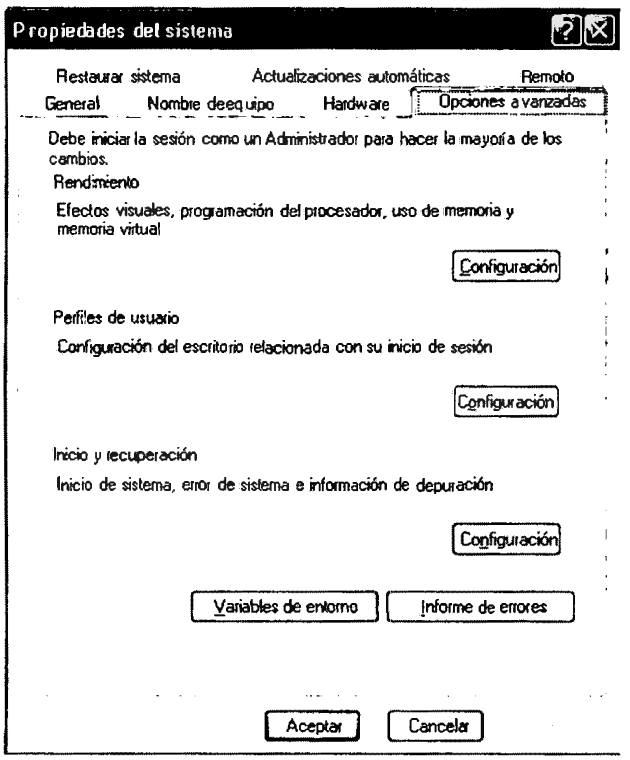


Figura N° 4.48: Opciones avanzadas (Windows)

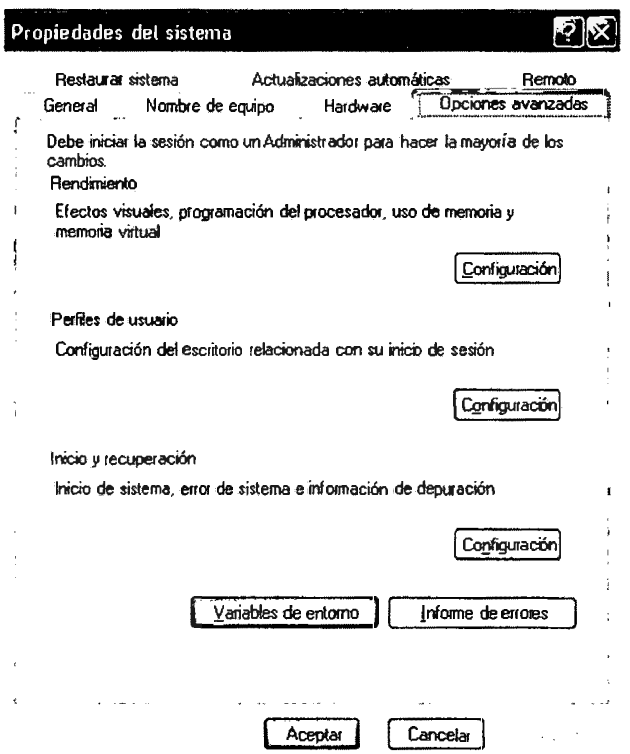


Figura Nº 4.49: Variables de entorno (Windows)

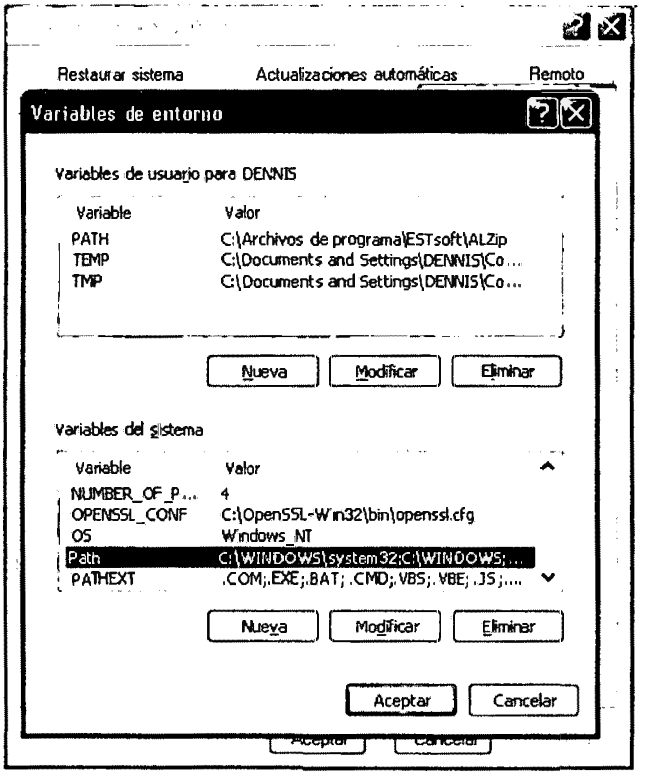


Figura Nº 4.50: Modificar variables de entorno (Windows)

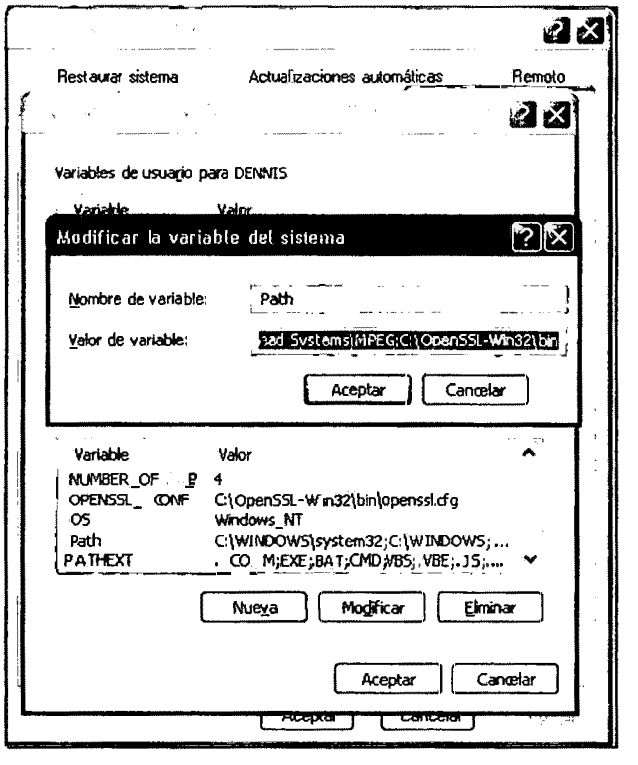


Figura N° 4.51: Modificar variables de sistema (Windows)

Ejecutando el cmd y cargar el openssl

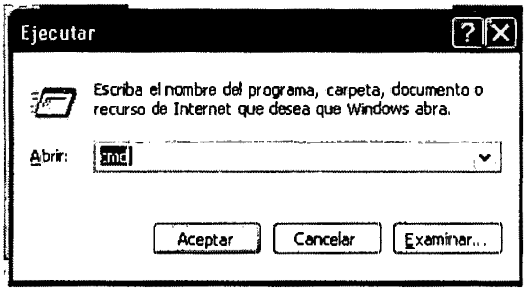


Figura N° 4.52: Ventana ejecutar (Windows)

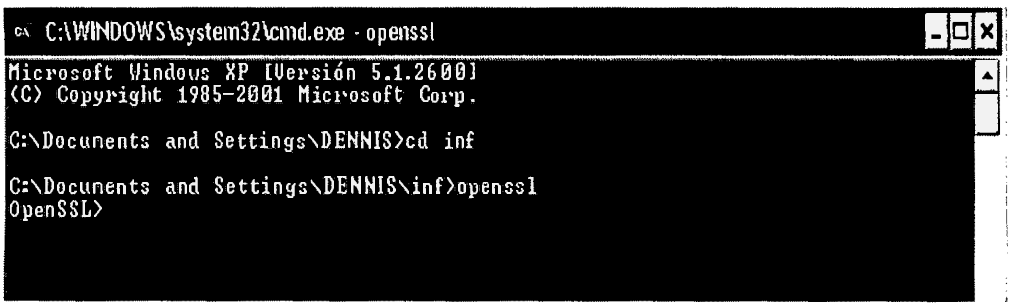


Figura N° 4.53: Ejecución de openssl (Windows)

Con el comando **pkcs12** de openssl permite **convertir** un fichero **.p12** en un fichero **.pem** que contiene tanto la clave pública como la privada.

Figura N° 4.54: Comando pkcs12 (Windows)

A continuación podemos utilizar el comando RSA para extraer en otro fichero .pem sólo la clave pública y sólo la clave privada.

Creando Clave publica

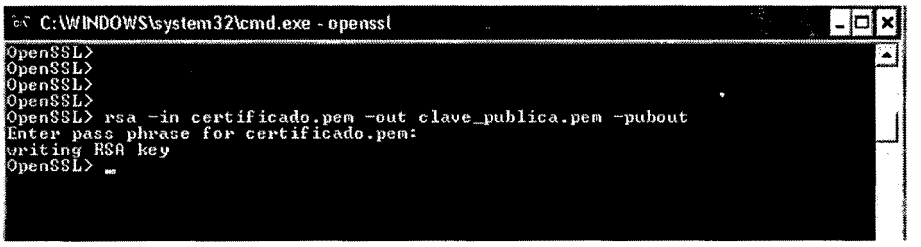


Figura Nº 4.55: Extracción de clave pública (Windows)

Creando Clave privada

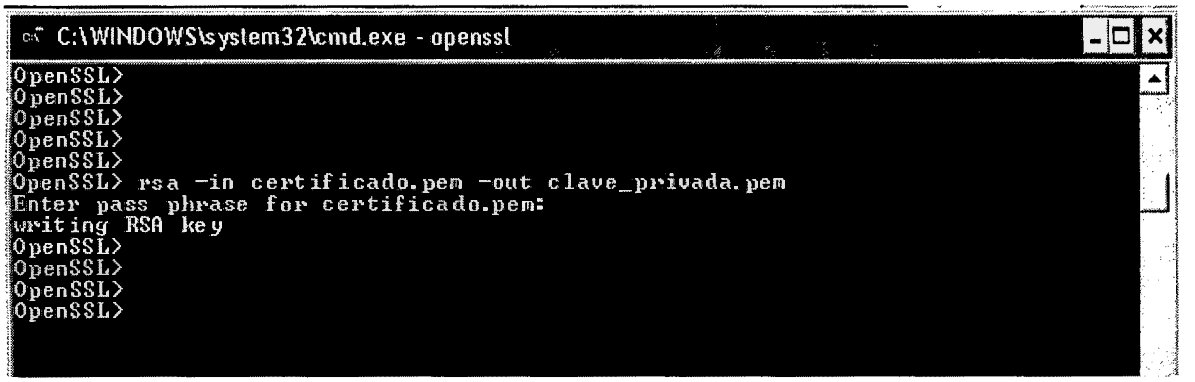


Figura Nº 4.56: Extracción de clave privada (Windows)

4.10. Cifrado Asimétrico

Para cifrar con la clave pública:

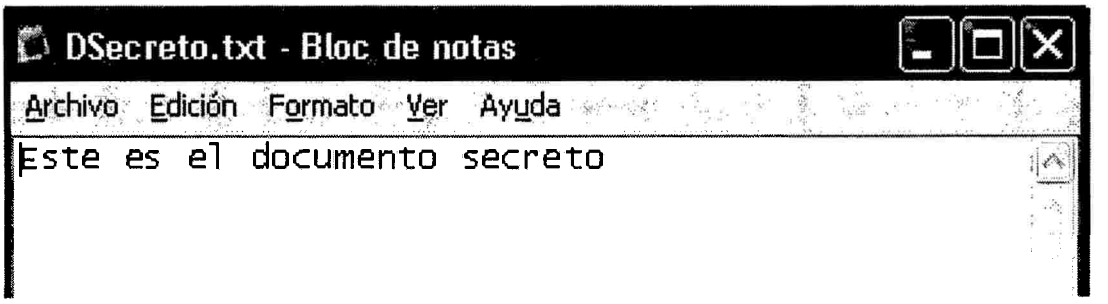


Figura Nº 4.57: Texto plano (Windows)

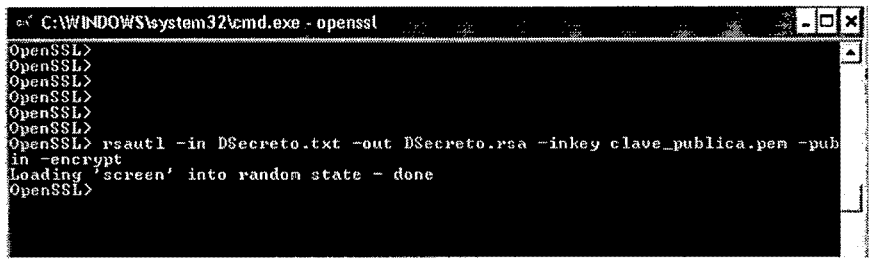


Figura Nº 4.58: Cifrado clave pública (Windows)

- ¿qué significa cada opción?

rsautil: orden puede ser usada para firmar, verificar, codificar, y descifre datos usando el algoritmo RSA

-in: Especifica el nombre de archivo de entrada para leer datos.

-out: especifica el nombre de archivo de salida.

inkey: especifica el archivo de llave de entrada que, en ausencia esto debería ser una llave privada RSA.

-pubin: Especifica que el archivo de entrada es una llave pública RSA

-sign: firma los datos de entrada y la salida el resultado firmado. Este requiere y RSA llave privada.

-verify: verifica los datos de entrada y la salida los datos recuperados.

-encrypt: codifica los datos de entrada usando una llave pública RSA

-decrypt: descifra los datos de entrada usando una llave privada RSA.

- ¿Pide el password? NO

Para descifrar con la clave privada:

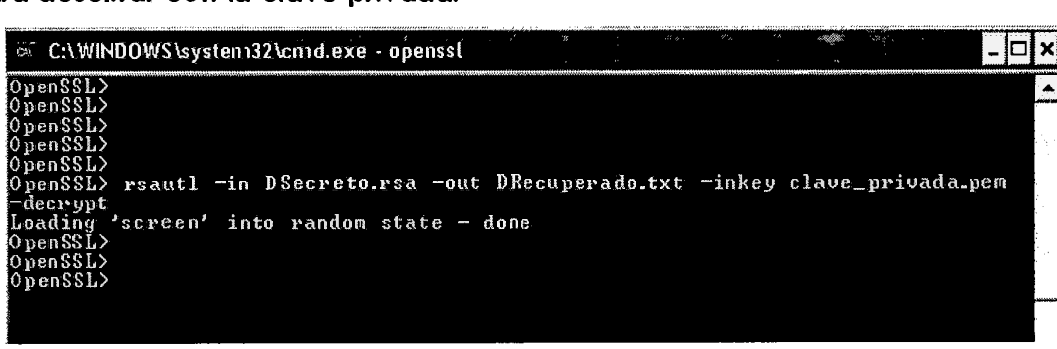
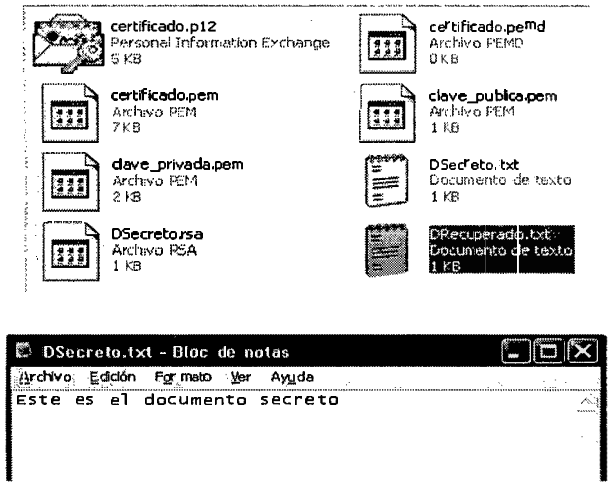


Figura N° 4.59: Cifrando clave privada (Windows)



Usando el Certificado

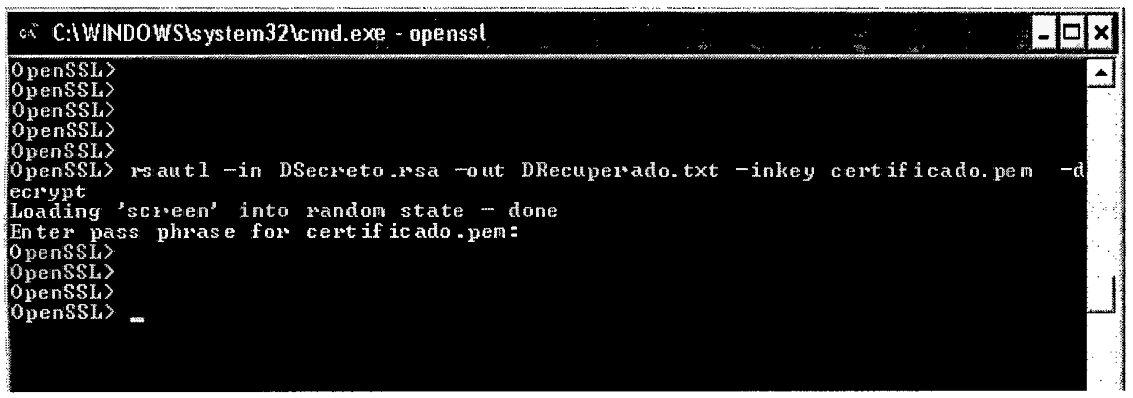


Figura N° 4.60: ingreso de la clave del nuestro certificado (Windows)

¿Qué significa cada opción?

rsautil: orden puede ser usada para firmar, verificar, codificar, y descifre datos usando el algoritmo RSA.

-in: Especifica el nombre de archivo de entrada para leer datos.

-out: especifica el nombre de archivo de salida.

inkey: especifica el archivo de llave de entrada que, en ausencia esto debería ser una llave privada RSA.

-pubin: Especifica que el archivo de entrada es una llave pública RSA

-sign: firma los datos de entrada y la salida el resultado firmado. Este requiere y RSA llave privada.

-verify: verifica los datos de entrada y la salida los datos recuperados.

-encrypt: codifica los datos de entrada usando una llave pública RSA

-decrypt: descifra los datos de entrada usando una llave privada RSA.

¿Pide el password? SI

4.11. Envío de un mensaje cifrado a un compañero

Archivo que será cifrado y enviado

Archivo Cifrado

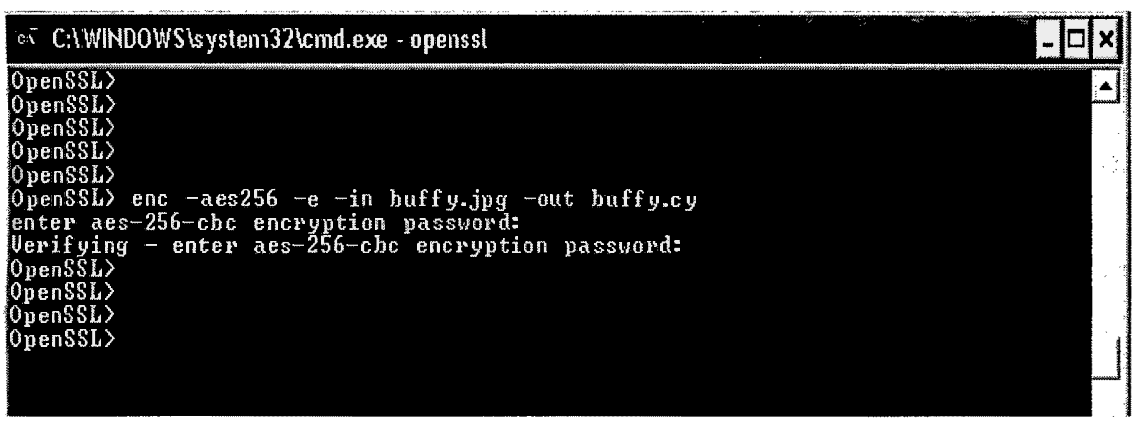


Figura N° 4.61: Envió de archivo cifrado (Windows)

Compartiendo la CLAVE PUBLICA con un amigo a través del correo electrónico.

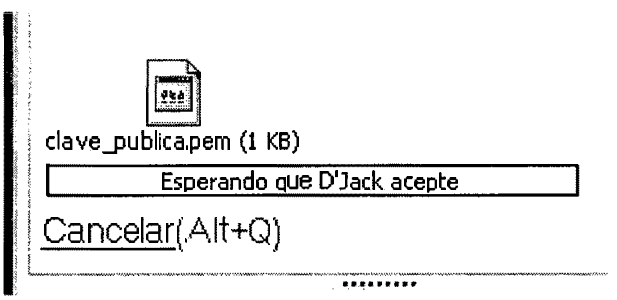


Figura N° 4.61: Transferencia de la clave pública (Windows)

Cifrando la clave utilizada para el cifrado simétrico

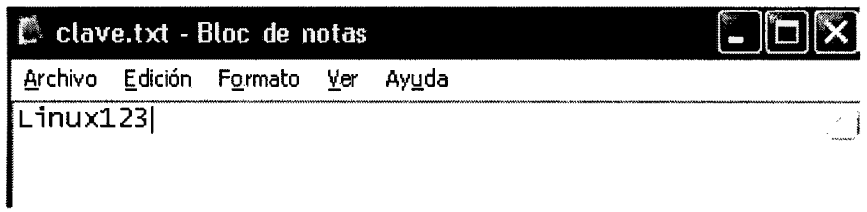


Figura N° 4.62: Cifrando clave (Windows)

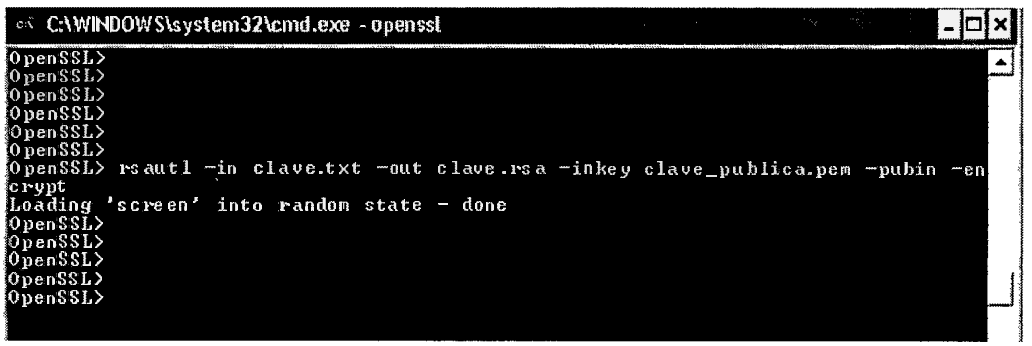


Figura N° 4.632: Cifrando clave (Windows)

Enviar al compañero el archivo cifrado (simétrico) junto con la clave cifrada (en Asimétrico).

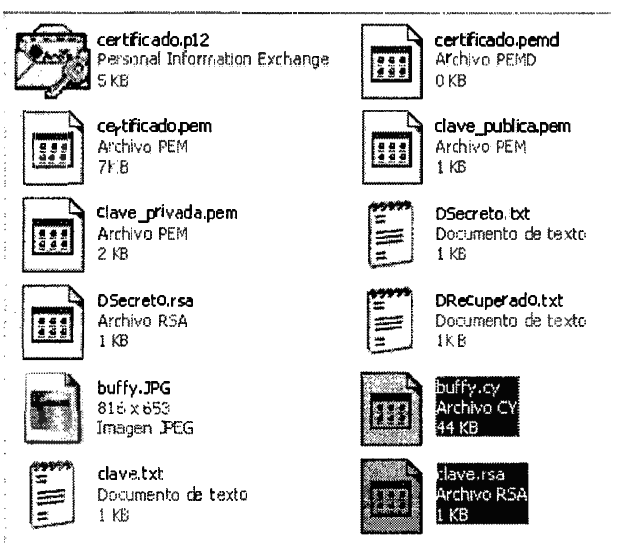


Figura N° 4.62: Fichero Cifrado (Windows)

Deshacer los cifrados para obtener el archivo original. Para obtener la clave simétrica, deberíamos utilizar nuestra clave privada.

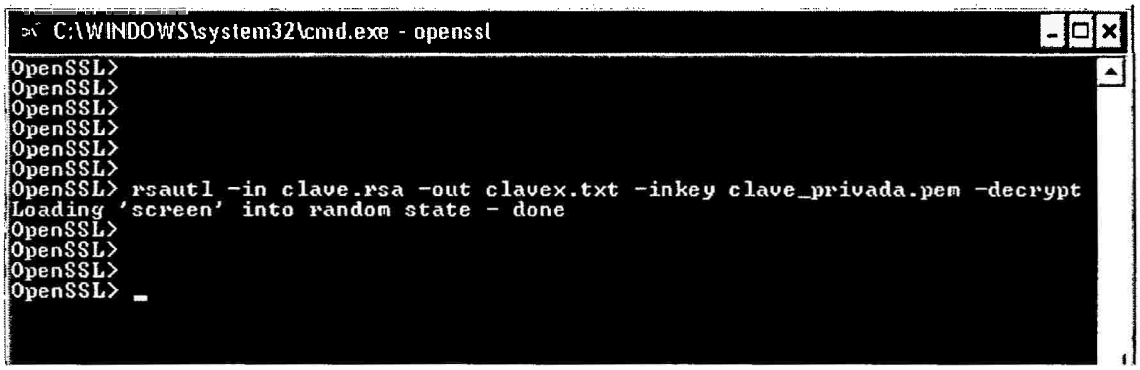


Figura N° 4.63: Obtención de la clave (Windows)

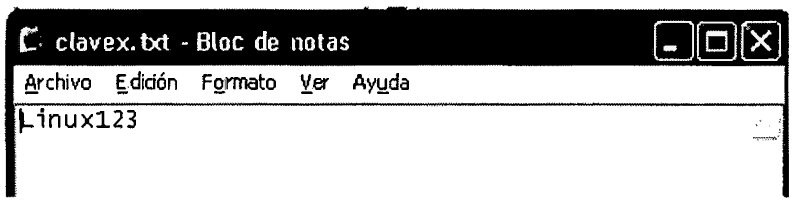


Figura N° 4.64: Clave (Windows)

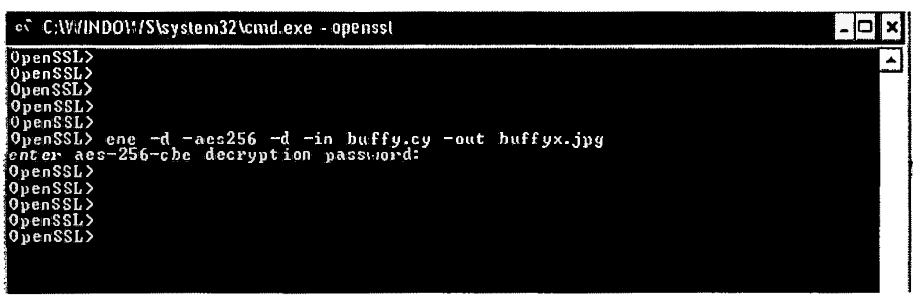


Figura N° 4.65: Cifrando clave (Windows)

4.12. Envío de un mensaje firmado

Obtenemos el resumen de un documento

Figura N° 4.66: Resumen de documento (Windows)

Obteniendo el Resumen del Documento

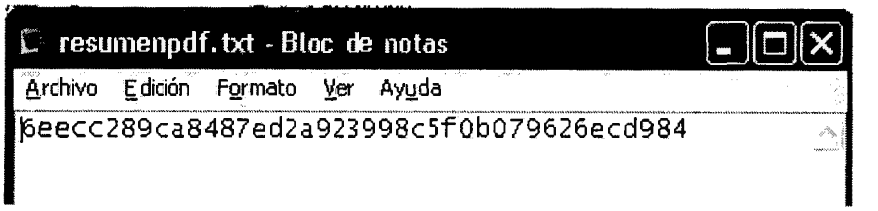


Figura N° 4.65: Resumen del documento en formato txt (Windows)

Cifrar el Resumen

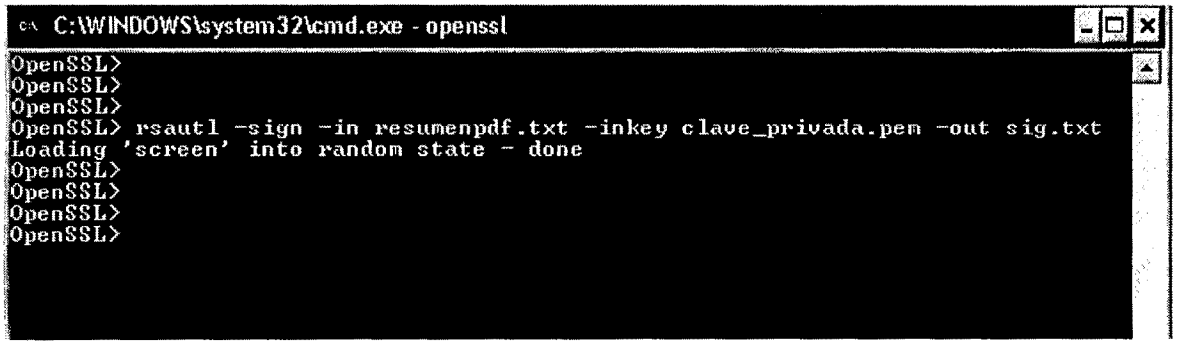


Figura N° 4.66: Cifrando el resumen (Windows)

Enviar por correo electrónico

Figura N° 4.67: Archivos por enviar (Windows)

Verificar la firma

Obteniendo el resumen del documento recibido.

OpenSSL>rsautl -verify -in sig.txt -inkeypubkey.pem –pubin –raw

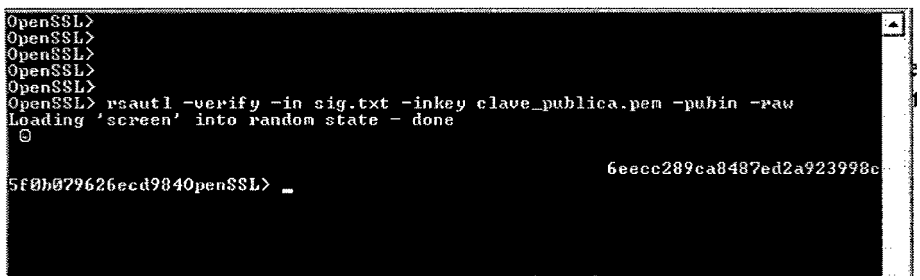


Figura N° 4.68: Obtención del documento recibido (Windows)

4.2 CONFIGURACION DE MANIPULACION DE URL PROTECCION A LA CONFIDENCIALIDAD.

Código fuente java – manipulación de url

Index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

Principal.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Front Controller</title>
    </head>
    <body>
        <h1>Bienvenidos a la pagina de Javier Cruz </h1>
        <a href="login.do">ingresar</a>
    </body>
</html>
```

Redirecciona.jsp

```
<% request.getRequestDispatcher("index.do").forward(request, response); %>
```

Welcome.jsp

```
<%
    String name = request.getParameter("name");
%>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Bienvenido <%=name %></h1>

    </body>
</html>
```

Accionurl.propertes

```
login.do=pe.emp.action.InicioAccion
pe.emp.action.InicioAccion=autenticar.jsp
index.do=pe.emp.action.IndexAccion
pe.emp.action.IndexAccion=principal.jsp
bienvenido.do=pe.emp.action.WelcomeAccion
pe.emp.action.WelcomeAccion=welcome.jsp
```

DaoActionUrl.java

```
package pe.emp.dao;

import java.util.ResourceBundle;

public class DAOAccionUrl {
    private ResourceBundle ListaAccionUrl;
    public DAOAccionUrl()
    {
        this.ListaAccionUrl = ResourceBundle.getBundle("accionurl");
    }

    public String getNombreAccion(String noetiqueta){
        String noaccion = this.ListaAccionUrl.getString(noetiqueta);
        return noaccion;
    }

     public String getNombreUrl(String noetiqueta){
     String nourl = this.ListaAccionUrl.getString(noetiqueta);
     return nourl;
```

FactoryAccion.java

```
package pe.emp.factory;

import pe.emp.action.IAccion;

public class FactoryAccion {
    public static IAccion getClassAccion(String noaccion){
        IAccion accion = null;
        try {
            accion = (IAccion) Class.forName(noaccion).newInstance();
        } catch (Exception e) {
            System.out.println("e="+e.getMessage());
        }
        return accion;
```

SControl.java

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    String uri = request.getRequestURI();
    System.out.println("la uri = " + uri);

    String vista = uri.substring(uri.lastIndexOf("/")+1, uri.lastIndexOf("."));
    //todo despues de esto
    if(vista.equals("principal"))
    {
        System.out.println("Procesa y verifica usuario: ");
        /* 01 para llamar al principal*/
        //String newurl = vista+ ".jsp";
        /* 02 para llamar al login pasando por principal*/
        String newurl = "Login.jsp";
        this.dispatch(request, response, newurl);
    }
    //
    if(vista.equals("bienvenido"))
    {
        System.out.println("Procesa y verifica usuario: ");
        /* 01 para llamar al principal*/
        //String newurl = vista + ".jsp";
        /* 02 para llamar al login pasando por principal*/
```

SController.java

```
protected void processRequest(HttpServletRequest request, HttpServletResponse resp
        throws ServletException, IOException {

    String url = request.getRequestURI();
    String pagina = url.substring(url.lastIndexOf("/")+1);
    //creo un dao que accede al archivo que contiene acciones y urls
    DAOAccionUrl daoaccionurl = new DAOAccionUrl();
    String noaccion = daoaccionurl.getNombreAccion(pagina);
    //crea objeto helper y precesa informacion
    IAccion accion = FactoryAccion.getClassAccion(noaccion);
    accion.verificarInformacion();

    //consigue vista destino y reenvia
    String nourl = daoaccionurl.getNombreUrl(noaccion);
    dispatch(request, response, nourl);

}

protected void dispatch(HttpServletRequest request, HttpServletResponse response,
        throws ServletException, IOException {
    request.getRequestDispatcher(url).forward(request, response);
    //response.sendRedirect(url);
}
```

En consecuencia se verá a continuación

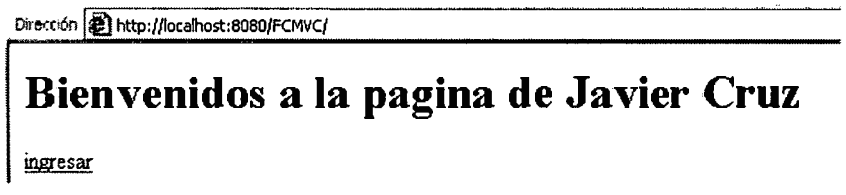


Figura N° 4.69: Página de inicio (Elaboración propia)

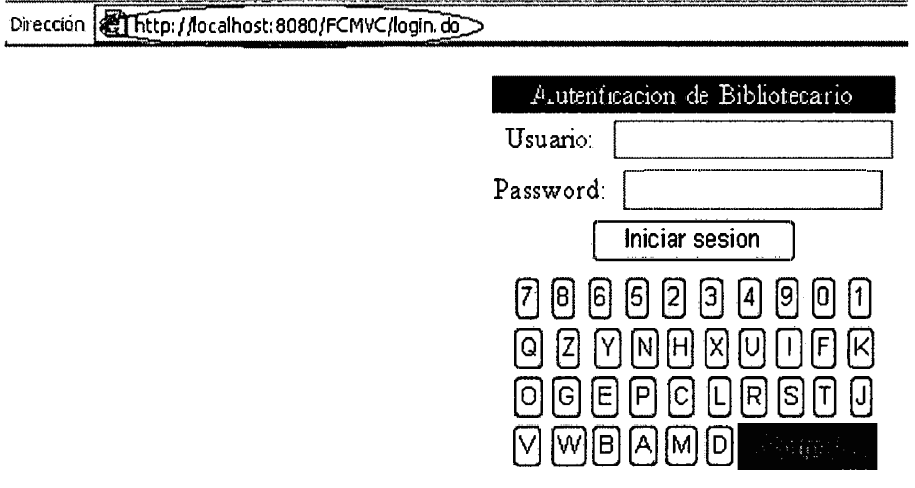


Figura N° 4.70: Autenticación de usuario, escondido (Elaboración propia)

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

a) Para realizar las técnicas de protección, se utilizó una aplicación web, de la biblioteca, implementado en java, en la cual se realizó todas las técnicas de protección, para un proyecto seguro.

b) Se implementó en el lenguaje de programación java, el algoritmo de encriptación de texto, como se muestra en el apartado A, como también el desencriptado de texto, para lo cual se implementó el generador de llaves públicas y privadas, que realiza el encriptado y desencriptado de nuestro texto que deseamos cifrar.

c) Se generó un certificado SSL (Secure Socket Layer) de Java, se utilizó la herramienta Keytool, que viene incluida en Java, y que se muestra en la figura N° 4.8, de manera que se va hacer uso de un recurso seguro, mediante certificado.

d) Se configuro el servidor de aplicaciones, como es el Apache Tomcat, con el certificado SSL, ya que se realizó activaciones del protocolo https, que viene desactivado por defecto, y para su reconocimiento inicial de nuestro certificado, como se muestra en la figura 4.16, que garantizan la comunicación interna.

e) Para realizar la técnica de autenticación de usuario, se implementó un Javascript, para el teclado virtual, y para el ingreso de la clave, se deshabilito el teclado físico, de esta manera, se ingresa de manera segura, la contraseña del usuario.

f) Se implementó capcha, para generar imágenes aleatorias, para confirmar su registro de usuario.

g) Para la protección de la URL, se implementó el Fron Controller, que nos da una dirección falsa y el bloqueo de la barra de direcciones.

5.2 RECOMENDACIONES

a) Debido al incremento de robo de información, se recomienda realizar,

b) un escaneo de vulnerabilidades con la herramienta Acunetix, de manera general a nuestra aplicación web, para ver la seguridad.

c) Desarrollar técnicas OWASP (Open Web Application Security Project), que se encuentra los ataques más comunes, en una web.

d) Adquirir un certificado digital, para nuestro servidor de aplicaciones, como es el caso del Apache Tomcat.

5.3 BIBLIOGRAFIA

a) Mattos, E. (2005). *Seguridad en el comercio electrónico.* UNMSM, Lima.

b) Sebastián, L. (2006). *Soluciones de seguridad informática sirviendo a Latinoamérica y Caribe. México.*

c) Gómez, A. (2007). *Enciclopedia de la seguridad informática:* Fundamentos de criptografía (p 279). Alfaomega Grupo Editor, México.

d) Arcert. (2008). *Seguridad en Servidores y Aplicaciones Web,* Coordinación de Emergencias en Redes Teleinformáticas en la APN.

e) Meiners, L. (2005). *Soluciones de seguridad informática.* Buenos Aires, Argentina.

f) Howard y LeBlanck. (2003). *Writing Secure Code*: Arquitec Secure (2da Edition), (p 69 – 124).

g) Magazine, Pc. (2005). *Web Application Definition.*

h) Presman, R. (2002). Ingeniería del Software, un enfoque práctico. (5ta Ed)., Madrid: McGraw-Hill.

i) Wikipedia, (2008). *Framework para aplicaciones WEB.*

j) Stallings, W. (2000). *Comunicaciones y redes de computadoras.* (6ta Edic.), Madrid: Pearson Educación.

k) Deitel, H., Harvey, M., Paul, J. (2005). Como *programar en Java, un enfoque práctico.* (5ta Ed), Mexico: McGraw-Hill.

l) Ceballos, J. (2006). Java 2: Curso de programación, (3ra Ed)., Madrid: AlfaOmega, 2006.

m) Cabrera, S., García, MC., y Salinas, J. (2009). *Modelo de seguridad en aplicaciones web desarrolladas por un tercero.* Instituto Politécnico Nacional, México.

n) Gonzalo, E. y Gene, S. (1999). *Seguridad y comercio en el web.* Mc Graw Hill, México.

WEBSITE–ARTICULOS DE UNA REVISTA

a) Yoskovitz, B. (1997, 14 de marzo). Computer Security Techniques: Niche Search Security and Guides. OWAS Web Magazine, 4, 15-18. Obtenido en la Red Mundial el 20 de marzo de 1997: www.seguridadredes.web/informatica.htm

WEBSITE – COMUNICACIÓN ELECTRONICA

b) Security Informatic. How to Cite Information From the World Wide Web. Obtenido en la Red Mundial el 1 de abril de 1997: http://www.trucoswindows.net/forowindows/off-topic/52799-revista-hakin9-seguridad-informatica.html

5.4 ANEXOA

IDENTIFICACION DE AMENAZAS COMUNES DE LA APLICACIÓN WEB

DATOS GENERALES DE LA ENTREVISTA	
NOMBRE DEL ENTREVISTADO:	**FECHA:**
CARGO:	**LUGAR:**
DISPOSICION:	**TIEMPO UTILIZADO:**
Nº DE INTERRUPCIONES:	**Nº DE OBSERVACION:**

PRESENTACION: La presente entrevista está dada sobre la seguridad en una aplicación web, donde se tendrá en la Ciudad de Ayacucho, como parte de la seguridad en su importancia, con el fin de poder determinar los puntos más vulnerables en una aplicación web, y así poder dar solución.

INSTRUCCIONES: Al leer el cuestionario Ud., tendrá que marcar con una aspa si cree conveniente la respuesta correcta en la preguntas con las casillas, y Ud. contestara si el cuestionario le realiza una Pregunta

DATOS ESPECIFICOS DE LA ENTREVISTA

1. ¿Ha realizado consultas a información confidencial y/o servicios que involucren transacciones de comercio electrónico (transacciones, pagos, etc.), sin necesidad de autenticarse?

   ☐ SI

   ☐ NO

   Porque?.....

   .....

2. Existen antecedentes en la que los documentos fueron modificados durante la transmisión de datos?

   ☐ SI

   ☐ NO

   Como?.....

   .....

3. ¿Algunas veces Ud. tuvo quejas que sus datos fueron modificados o suplantados en la web?

   ☐ SI

   ☐ NO

4. Existen antecedentes en la que el link de la URL de la página web no son los correctos.

   ☐ SI

   ☐ NO

5. Existen antecedentes en la que el usuario tuvo problemas al acceder a sesión.

   ☐ SI

   ☐ NO

   Otros motivos

   ¿Cuales?.....

Punto de vista de profesional de TI

¿Le preocupa la seguridad en las transacciones de datos y su inviolabilidad?

- ☐ SI
- ☐ NO

¿Intenta que la seguridad sea transmitida al cliente?

- ☐ SI
- ☐ NO

¿Cómo califica la relación entre la seguridad y el éxito comercial de una transacción de comercio electrónico?

- ☐ Muy alta
- ☐ Alta
- ☐ Neutra
- ☐ Baja
- ☐ Muy Baja

¿Conoce las posibilidades que tiene el usuario para determinar la seguridad de un sitio Web antes de relacionarse comercialmente con él?

- ☐ SI
- ☐ NO

¿Conoce los servicios de las autoridades certificadoras de e-commerce?

- ☐ SI
- ☐ NO

Considera a la encriptación como factor de la seguridad en las transacciones condición:

- ☐ Necesaria
- ☐ Suficiente
- ☐ Necesaria y suficiente

¿Qué importancia le da a las políticas de seguridad en sus proyectos de TI?

- ☐ Bastante
- ☐ Alguna
- ☐ Ninguna

¿Sigue buenas prácticas relacionadas con la seguridad (Auditar, utilizar seguridad física, restringir acceso)?

- ☐ SI
- ☐ NO

¿Le recomendaría a un cliente utilizar el e-commerce?

- ☐ SI
- ☐ NO

5.5 ANEXO B

DATOS PROCESADOS RESPECTO A LAS ENTREVISTAS REALIZADAS

a. Información confidencial

Transacciones de comercio electrónico sin necesidad de autenticarse?
Transmisión de datos
Problemas de Inicio de sesión
Datos modificados o suplantados en la web
Encriptación de datos

b. Integridad de datos

Manipulación de los datos
Manipulación de url

c. Autenticación

Autoridades certificadoras
Datos interceptados

5.6 ANEXO C