ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA LA INFRAESTRUCTURA TECNOLOGICA DE LOS SERVICIOS POR INTERNET DE LA BANCA ECUATORIANA

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

MARIA GABRIELA COELLO SALAS
gabbylf87@hotmail.com

DIRECTOR: PhD. LUIS ENRIQUE MAFLA GALLEGOS
mafla@epn.edu.ec

Quito, Noviembre 2012

DECLARACIÓN

Yo, María Gabriela Coello Salas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

María Gabriela Coello Salas

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por María Gabriela Coello Salas, bajo mi supervisión.

PhD. Enrique Mafla

DIRECTOR DE PROYECTO

DEDICATORIA

“Me gusta lo imposible porque hay menos competencia. "
Walt Disney

A mi mami Magdalena, por tu amor, ternura y dulzura. Gracias por enseñarme a luchar por mis sueños y acompañarme en cada segundo de mi vida. Espero que te sientas orgullosa de mí. Te amo.

A mi papi Carlos, por tu tenacidad y lucha insaciable a lo largo de mi vida, por brindarme ante todo tú sincero amor. Gracias por todo lo que has hecho para que este sueño se convierta en realidad.

A mi hermano Fernando, por tu amor incondicional y apoyo. Gracias ñaño por ayudarme a pelear mis batallas, por darme la fuerza para alcanzar mis sueños, olvidándote muchas veces de los tuyos. Gracias por ser un ejemplo, un amigo y una guía en mi vida que únicamente me impulsa a ser mejor. Gracias por ayudarme en cada momento de mi vida, todo te lo debo a ti y este sueño cumplido te pertenece.

A mi hermana Paulina, por tu amor, cariño, paciencia, sinceridad y por jamás dejarme decaer. Gracias ñaña porque en ti encontré una segunda madre, un ejemplo de mujer y profesional. Gracias por apoyarme y guiarme en cada paso de mi vida, quiero llegar algún día a ser como tú.

A mi hermano Vinicio, por tu amor, cariño y tu apoyo incondicional. Gracias por guiar mis pasos y ayudarme en los momentos más difíciles de mi vida. Gracias por ser mi amigo, por haber siempre velado por mi bienestar y ser una guía importante en mi vida.

A mis sobrinitas Andrea, Camila, María José, Emily y mi sobrinito Matías, por todo su amor, cariño, dulzura y ternura. Gracias por impulsarme a ser mejor, sin ustedes no hubiese podido alcanzar este sueño, su presencia en mi vida es el mejor regalo.

A mi novio Edison, solo puedo decirte gracias por todo tu amor y lucha constante, gracias por darme fuerzas y acompañarme en los momentos más difíciles y de cansancio. Gracias por querer compartir tu vida junto a la mía y llenarme de tantas alegrías. Te amo mucho.

De manera especial, agradezco al Dr. Enrique Mafla, por sus sabios consejos y guía que fueron determinantes para haber alcanzado este momento.

A mis amig@s por las malas noches y tantas batallas superadas, gracias a todos ustedes por brindarme su sincera amistad, su apoyo y su cariño. Gracias a todos por haberme ayudado incondicionalmente.

Gabby.

CONTENIDO

INDICE DE FIGURAS

RESUMEN

El presente proyecto de titulación desarrolla un procedimiento formal de ethical hacking para la infraestructura tecnológica de los servicios por Internet de la banca ecuatoriana, con el objetivo de que entidades de control de Gobierno o privadas puedan medir las vulnerabilidades de seguridad y el cumplimiento de la normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V.

Para el desarrollo del presente proyecto se utilizó el Manual de la Metodología Abierta de Testeo de Seguridad (OSSTMM), sección C Seguridad en las Tecnologías de Internet y los módulos de Búsqueda y Verificación de Vulnerabilidades, Identificación de los Servicios de Sistemas y Testeo de Aplicaciones de Internet.

Este procedimiento formal de ethical hacking servirá para un análisis de seguridad externo a las entidades financieras en los servicios de banca electrónica por Internet.

INTRODUCCIÓN

El presente proyecto de titulación tiene como objetivo elaborar un procedimiento formal de ethical hacking para la infraestructura tecnológica de los servicios por Internet de la banca ecuatoriana.

Este procedimiento formal de ethical hacking permitirá a las entidades financieras medir las vulnerabilidades existentes en la aplicación de banca electrónica por Internet.

Este documento está dividido en 3 capítulos:

CAPITULO 1 “Planteamiento del problema”, está sección comprende el planteamiento del problema actual de la seguridad de los servicios de banca electrónica por Internet, una propuesta de solución y la selección de la metodología y herramientas utilizadas para el cumplimiento de la normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V.

CAPITULO 2 “Definición del procedimiento formal de ethical hacking”, este capítulo contiene el desarrollo del procedimiento formal de ethical hacking para la banca ecuatoriana mediante el desarrollo de las pruebas de rastreo y de intrusión descritas en las secciones 2.3 y 2.4 respectivamente. Las pruebas de rastreo comprenden la identificación de vulnerabilidades por intrusión y configuración, mientras que las pruebas de rastreo comprenden la identificación de vulnerabilidades en la aplicación de banca electrónica, para el desarrollo de estas pruebas se realizó el análisis de los requerimientos expuestos en la normativa vigente de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V descrito en la sección 2.1, la identificación de vulnerabilidades en una infraestructura web esta descrita en la sección 2.2 y finalmente, se describen tareas que le permitirá al analista de seguridad eliminar resultados erróneos los cuales están descritos en la sección 2.5.

CAPITULO 3 “Conclusiones y Recomendaciones”, esta sección contiene las conclusiones y recomendaciones obtenidas durante el desarrollo de este proyecto.

CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA.

En este capítulo se definirá el problema, se propondrá una solución, y se seleccionará la metodología y herramientas a ser utilizadas en el desarrollo del presente proyecto. Se definirá el problema en la sección 1.1 del presente proyecto determinando la falta de procedimientos o políticas de seguridad especificados en la normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V, punto 4.3.4, con el objetivo de medir las vulnerabilidades de seguridad en las entidades financieras. Después, se propondrá una solución que satisfaga los requerimientos planteados en la normativa vigente de la Superintendencia de Bancos y Seguros.

La metodología define los pasos a ser utilizados para la realización del presente proyecto. Finalmente, se definirán las herramientas a utilizar en las tareas especificadas en los módulos de la metodología OSSTMM (Manual de la Metodología Abierta de Testeo de Seguridad) que permitan el cumplimiento de la normativa utilizada para la realización de este proyecto.

1.1 DEFINICIÓN DEL PROBLEMA.

La normativa de la Superintendencia de Bancos y Seguros del Ecuador sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V, punto 4.3.4 no cuenta con procedimientos o pruebas formales de seguridad. La normativa establece como requerimiento que las entidades financieras posean políticas y procedimientos de seguridad de la información internos, pero esta normativa no cuenta con procedimientos o pruebas que permitan a entidades financieras o de Gobierno medir las vulnerabilidades de seguridad existente en los servicios por Internet en la banca electrónica. Además, la normativa sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V de la Superintendencia de Bancos y Seguros, no ha sido actualizada desde el año 2005 para que esté acorde a las nuevas formas de delito informático. Debido a estos problemas, de enero a junio del 2011 se presentaron 1366 denuncias por delitos informáticos en la Fiscalía General del Estado.

1.2 PROPUESTAS DE SOLUCIÓN.

El presente proyecto propone la creación de un procedimiento formal de ethical hacking[1]. Este procedimiento permitirá realizar pruebas formales de seguridad informática en las capas de red, transporte y aplicación del modelo TCP/IP[2].

Los objetivos del presente proyecto de titulación son:

- Proponer un procedimiento formal de pruebas de ethical hacking para ser utilizado por la banca.
- Mejorar la seguridad de los servicios por Internet de la banca.
- Ayudar a la banca ecuatoriana a cumplir con la normativa sobre gestión de riesgos tecnológicos.
- Diseñar un procedimiento formal que permita a entidades de control de Gobierno y/o privadas medir las vulnerabilidades de la seguridad y cumplimiento de la normativa.

1.3 METODOLOGÍA Y HERRAMIENTAS

Esta sección se divide en dos fases, la selección de la metodología y la selección de las herramientas a ser utilizadas para el desarrollo del proyecto.

---

[1] Ethical Hacking.- Técnica en la evaluación del riesgo informático de una red, sistema o aplicación. (Fuente: http://www.slideshare.net/mfrayssinet/etical-hacking).

[2] Modelo TCP/IP.- Es una arquitectura de protocolos que describe guías generales de diseño e implementación de protocolos de red que permitan la comunicación en una red. (Fuente: http://es.wikipedia.org/wiki/Modelo_TCP/IP).

1.3.1 SELECCIÓN DE LA METODOLOGÍA.

Para realizar el presente proyecto se utilizará el manual OSSTMM (Manual de la Metodología Abierta de Testeo de Seguridad) versión 2.1. La metodología divide a un ambiente tecnológico para el análisis de seguridad en secciones. En la Figura 1-1 se muestran la división del ambiente tecnológico.

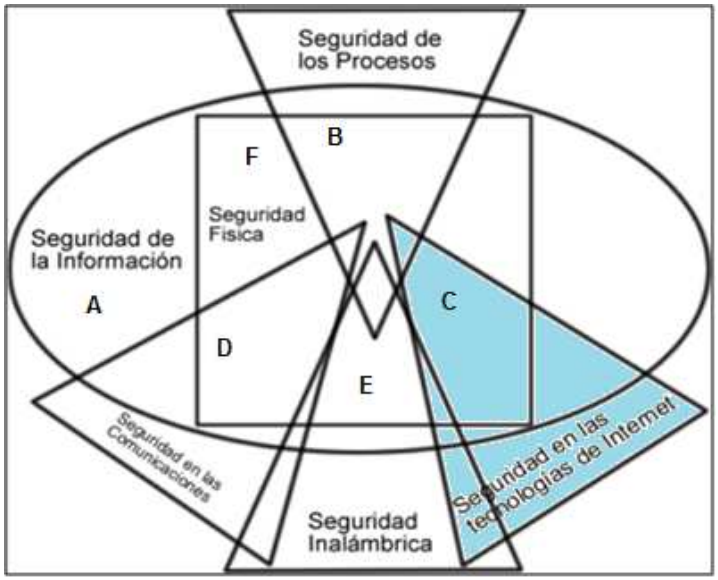

Figura 1- 1 Secciones del manual OSSTMM [3]

La sección C Seguridad en las Tecnologías de Internet del manual OSSTMM, será utilizada para la realización del presente proyecto. Esta sección será enfocada en la seguridad de los servicios de banca electrónica por Internet. Se desarrollarán dos etapas para el desarrollo del proyecto: Descripción y Análisis de requerimientos y Diseño del procedimiento formal.

---

[3] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

- **Descripción y Análisis de requerimientos**

El análisis y descripción de requerimientos recopilará las necesidades descritas en la normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V, punto 4.3.4. Los requerimientos expuestos en la normativa, son descritos a continuación:

Requerimiento 1: Requerimiento descrito en el punto 4.3.4.3 de la normativa establece que, se debe implementar:

Los controles necesarios para asegurar la integridad, disponibilidad y confidencialidad de la información administrada. [4]

Requerimiento 2: Requerimiento descrito en el punto 4.3.4.6 de la normativa establece que, se debe implementar:

Adecuados sistemas de control y autenticación para evitar accesos no autorizados, inclusive de terceros; y, ataques externos especialmente a la información crítica y a las instalaciones de procesamiento. [5]

Requerimiento 3: Requerimiento descrito en el punto 4.3.4.12 de la normativa establece que:

Las instituciones controladas que ofrezcan los servicios de transferencias y transacciones electrónicas deberán contar con políticas y procedimientos de seguridad de la información que garanticen que las operaciones sólo pueden ser realizadas por personas debidamente autorizadas; que el canal de comunicaciones utilizado sea seguro, mediante técnicas de encriptación de información; que existan mecanismos alternos que garanticen la continuidad del servicio ofrecido; y, que aseguren la existencia de pistas de auditoría. [6]

---

[4], [5], [6] FISCALIA GENERAL DEL ESTADO, LIBRO I.- Normas generales para la aplicación de la ley general de instituciones del sistema financiero. TÍTULO X.- De la gestión y administración de riesgos, CAPÍTULO V.- De la gestión del riesgo operativo, Octubre, 2005.

El análisis de requerimientos será realizado en la sección 2.1 del capítulo 2 del presente proyecto.

- **Diseño**

La sección C del manual OSSTMM está compuesta de 12 módulos. Los módulos que permiten cumplir con los requerimientos expuestos en la fase de descripción de requerimientos de la presente sección son:

1. **Búsqueda y Verificación de Vulnerabilidades**

El objetivo del módulo es identificar y verificar vulnerabilidades existentes en el servidor web. Para cumplir este objetivo se utilizarán herramientas que permitan identificar agujeros de seguridad y niveles de parcheado de las aplicaciones. Este módulo permite el cumplimiento del requerimiento 1.

2. **Identificación de los Servicios de Sistemas**

El objetivo del módulo es identificar los puertos abiertos, cerrados o filtrados en el servidor web de banca electrónica. Permitiendo determinar el servicio de Internet activos o accesibles detrás de aquellos puertos que se encuentren abiertos. Este módulo permite el cumplimiento del requerimiento 1.

3. **Testeo de Aplicaciones de Internet**

El objetivo de este módulo es encontrar fallas de seguridad en las aplicaciones cliente/servidor. Este módulo nos permitirá usar técnicas de testeo de software para determinar vulnerabilidades existentes en la aplicación web. Este módulo permite el cumplimiento de los requerimientos 1,2 y 3.

1.3.2 SELECCIÓN DE LAS HERRAMIENTAS.

Las herramientas a ser utilizadas en el proyecto tienen la finalidad de asegurar el cumplimiento de los requerimientos expuestos en la fase de descripción y análisis de requerimientos de la sección 1.3.1 del proyecto, que a su vez permite cumplir con las tareas descritas en los módulos de la metodología OSSTMM.

A continuación, se detallaran las herramientas utilizadas para el cumplimiento de cada requerimiento:

- **Requerimiento 1**

Herramienta Nmap

Es un escáner de vulnerabilidades utilizada para el análisis de seguridad informática. En el proyecto esta herramienta permitirá determinar los puertos[7] que se encuentran abiertos, cerrados o protegidos en el servidor e identificar los servicios que se están ejecutando. Además permite identificar el sistema operativo y su versión.

Herramienta Firewalk

Es un analizador de respuestas de los paquetes IP[8]. Esta herramienta permitirá determinar las técnicas de filtrado utilizadas en los cortafuegos dentro de la red.

Herramienta Netcat

Es un escáner de vulnerabilidades utilizado para el análisis y manipulación de servicios TCP[9]/UPD[10]. En el proyecto esta herramienta permitirá escanear puertos, abrir puertos TCP/UDP e iniciar servicios en el servidor a ser analizado.

---

[7] Puertos.- Un puerto se denomina a una interfaz por la que se puede enviar y recibir diferentes tipos de datos. (Fuente: http://es.wikipedia.org/wiki/Puerto_(informática)).

[8] IP (Protocolo de Internet).- Es un protocolo no orientado a conexión, utilizado por el origen y destino para la comunicación de datos. (Fuente: http://es.wikipedia.org/wiki/Internet_Protocol).

[9] TCP (Protocolo de Control de Transmisión).- Es un protocolo de comunicación orientado a conexión y fiable a nivel de la capa de transporte, utilizado en Internet. (Fuente: http://es.wikipedia.org/wiki/Transmission_Control_Protocol).

[10] UDP (Protocolo de Datagrama de Usuario).- Es un protocolo de la capa de transporte basado en el intercambio de datagramas. (Fuente: http://es.wikipedia.org/wiki/User_Datagram_Protocol).

Herramienta Ppscan.c

Es un proxy[11] web, que permite ocultar la dirección IP del atacante durante el análisis de seguridad. Esta herramienta permitirá el escaneo de puertos por sondeo FTP[12] y proxy.

Herramienta Nikto

Es un escáner de servidores web que analiza script potencialmente peligrosos que utilicen CGI[13]. Esta herramienta permitirá determinar errores de configuración en aplicaciones y servicios activos en el servidor web.

- **Requerimiento 2.**

Herramienta Brutus

Es un craqueador[14] de contraseñas. Esta herramienta permitirá craquear contraseñas por fuerza bruta (intentando descifrar la contraseña con combinaciones de caracteres) o utilizando diccionarios de palabras (archivos donde se referencian las contraseñas más usadas por los administradores de una red).

---

[11] Proxy.- Es un programa o dispositivo que realiza una acción en representación de otro. (Fuente: http://es.wikipedia.org/wiki/Proxy).

[12] FTP (Protocolo de Transferencia de Archivos).- Es un protocolo para la transferencia de archivos entre sistemas conectados en una red. (Fuente: http://es.wikipedia.org/wiki/File_Transfer_Protocol).

[13] CGI (Interfaz de Entrada Común).- Es un método para la transmisión de información del compilador instalado en el servidor. (Fuente: http://www.maestrosdelweb.com/editorial/cgiintro/).

[14] Craqueador.- Programas que sirven para eliminar protecciones en archivos o programas. (Fuente: http://www.alegsa.com.ar/Dic/crackeador.php).

Herramienta BackTrack

Es una colección de herramientas diseñadas para realizar auditoria[15] de seguridad a los sistemas. Esta herramienta permitirá analizar los puertos en el servidor web, identificar vulnerabilidades, cambiar privilegios y analizar las aplicaciones web.

Herramienta Cain & Abel

Es un snnifer[16] de red. En el proyecto esta herramienta permitirá capturar contraseñas por fuerza bruta y diccionario de palabras, enviadas en los paquetes de red.

Herramienta Saint

Es un escáner utilizado para el análisis de seguridades. En el proyecto permitirá analizar objetos protegidos por firewall, rastrear y reparar vulnerabilidades y simular ataques dentro de una red.

Herramienta Hunt

Es un snnifer de red. Esta herramienta permitirá observar el tráfico en la red y restablecer conexiones.

Herramienta WebScarab

Es un escáner de aplicaciones web utilizada para el análisis de seguridad. Esta herramienta permitirá observar el tráfico entre el navegador del cliente y el servidor y modificarlo, revelar campos ocultos y realizar cambios en parámetros del formulario.

---

[15] Auditoria.- Es un examen crítico que se realiza para evaluar la eficacia y eficiencia de un sistema de comunicación. (Fuente: http://www.monografias.com/trabajos/auditoinfo/auditoinfo.shtml).

[16] Snnifer.- Un snnifer es un programa que registra los datos enviados, así como la actividad realizada en un computador. (Fuente: http://es.wikipedia.org/wiki/Detección_de_sniffer).

Herramienta Ettercap

Es un snnifer de red. En este proyecto permitirá capturar el tráfico en la red inyectar código y ataques por envenenamiento ARP (Protocolo de Resolución de Direcciones)[17].

- **Requerimiento 3.**

Herramienta Wireshark

Es un snnifer de red que analiza los protocolos de red. En el proyecto esta herramienta permitirá capturar el tráfico generado entre cliente y servidor y analizar los paquetes generados.

Herramienta CookieDigger

Es un analizador de cookies. Esta herramienta nos permitirá recolectar y analizar las cookies.

Herramienta Spike Proxy

Es un escáner de vulnerabilidades en aplicaciones web utilizada para detectar fallas de seguridad. En este proyecto permitirá detectar desbordamientos de buffer y la detección de los directorios existentes en el servidor de banca electrónica.

Herramienta Nessus

Es un escáner de vulnerabilidades utilizada para el análisis de seguridad en aplicaciones web. En el proyecto nos permitirá escanear puertos y realizar pruebas de vulnerabilidades a nivel de la capa de aplicación del Modelo TCP/IP.

---

[17] ARP (Protocolo de Resolución de Direcciones).- Es un protocolo de capa de enlace de datos responsable de encontrar la dirección de hardware que corresponde a una determinada dirección IP. (Fuente: http://es.wikipedia.org/wiki/Address_Resolution_Protocol).

Herramienta N-Stealth

Es un escáner de seguridades en aplicaciones web utilizado para realizar análisis de vulnerabilidades. En el proyecto esta herramienta permitirá determinar los errores de configuración en la aplicación y detectar vulnerabilidades en las cookies[18].

Herramienta Achilles

Es un proxy que analiza la seguridad en aplicaciones. Esta herramienta permitirá realizar ataques man in the middle (Hombre en el medio)[19] y capturar y modificar el tráfico en la red.

Herramienta Wfuzz

Es un escáner que permite analizar la seguridad en aplicaciones y el servidor. Esta herramienta permitirá encontrar recursos, archivos y directorios por fuerza bruta.

Herramienta Web Intruder Burp Suite

Es un escáner de seguridades en aplicaciones web. En este proyecto permitirá obtener credenciales de usuarios validas y capturar el tráfico en la red.

Herramienta Historian

Es un analizador de navegadores web. En el proyecto permitirá leer ficheros, cookies y favoritos permitiendo exportar estos datos.

---

[18] Cookies.- Es la información que guarda un servidor sobre un usuario en su equipo. (Fuente: http://es.wikipedia.org/wiki/Cookie).

[19] Ataque Man in the middle.- Es un ataque en el cual el enemigo adquiere la capacidad de leer, modificar e insertar a voluntad los mensajes de las dos partes sin que ninguna conozca de su existencia. (Fuente: http://es.wikipedia.org/wiki/Ataque_Man-in-the-middle).

CAPÍTULO 2: DEFINICIÓN DEL PROCEDIMIENTO FORMAL DE ETHICAL HACKING

En este capítulo se definirá el procedimiento formal de ethical hacking. La definición del procedimiento se realiza en los sección 2.3 pruebas de rastreo y 2.4 pruebas de intrusión, utilizando como marco referencial las tareas descritas en la metodología OSSTMM. Para cumplir este objetivo, primero se realizará el análisis de requerimientos en la sección 2.1, utilizando los requerimientos descritos en la sección 1.3 del presente proyecto. Segundo, se analizará las vulnerabilidades existentes en ambientes web en la sección 2.2. Estas vulnerabilidades son especificadas en la metodología OSSTMM sección C Seguridad en las tecnologías de Internet utilizada para el desarrollo de este proyecto. Finalmente, se analizará los resultados en la sección 2.5.

2.1 ANÁLISIS DE LOS REQUERIMIENTOS DE SEGURIDAD BASADOS EN LA NORMATIVA VIGENTE.

Los servicios de banca electrónica serán evaluados bajo el modelo de seguridad, de la Figura 2-1. En el modelo de seguridad se representan los actores (cliente y personal del banco), el canal de comunicación (Internet) y equipos informáticos (computador del cliente e intranet del banco) que intervienen en una conexión de banca electrónica por Internet.

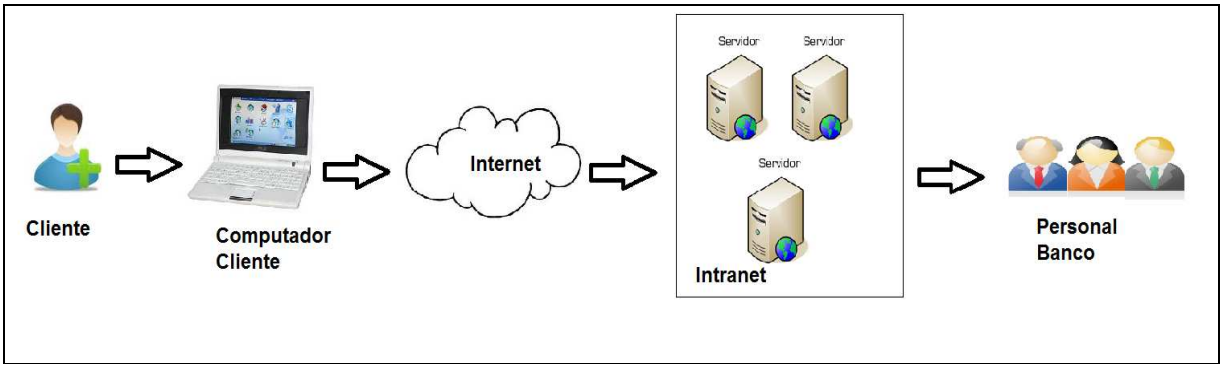

Figura 2- 1 Modelo de Seguridad

El análisis de requerimientos se encuentra basado en la normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V. Esta normativa trata sobre requisitos que intervienen en el canal de comunicación (Internet) y la infraestructura tecnológica del banco (intranet).

2.1.1 REQUERIMIENTO 1: EL SISTEMA DE BANCA ELECTRÓNICA POR INTERNET DEBE ASEGURAR LA INTEGRIDAD, DISPONIBILIDAD Y CONFIDENCIALIDAD DE LA INFORMACIÓN.

La normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V Artículo 2, define los 3 requisitos:

Integridad.- "Es la garantía de mantener la totalidad y exactitud de la información y de los métodos de procesamiento."

Disponibilidad.- "Es la garantía de que los usuarios autorizados tienen acceso a la información cada vez que lo requieran a través de los medios adecuados que satisfagan sus necesidades."

Confidencialidad.- "Es la garantía de que sólo el personal autorizado accede a la información preestablecida."

2.1.1.1 Integridad de la información

El cumplimiento del requerimiento de integridad de la información, se garantizará utilizando los siguientes módulos de la metodología OSSTMM: Búsqueda y verificación de vulnerabilidades, Identificación de los servicios de sistemas y Testeo de Aplicaciones por Internet.

Los módulos de búsqueda y verificación de vulnerabilidades e identificación de los servicios de sistemas, se utilizarán para el análisis de vulnerabilidades existentes en un ambiente web, con el objetivo de prevenir ataques que permitan modificar la información durante el ciclo de vida de la conexión del cliente.

El módulo de testeo de aplicaciones de Internet, se utilizará con el objetivo de identificar vulnerabilidades dentro de la aplicación de banca electrónica y evitar la modificación de la información de campos ocultos dentro de la aplicación o la inyección directa de código SQL.

En la figura 2-2 se describen los módulos utilizados para garantizar el cumplimiento del control de integridad.

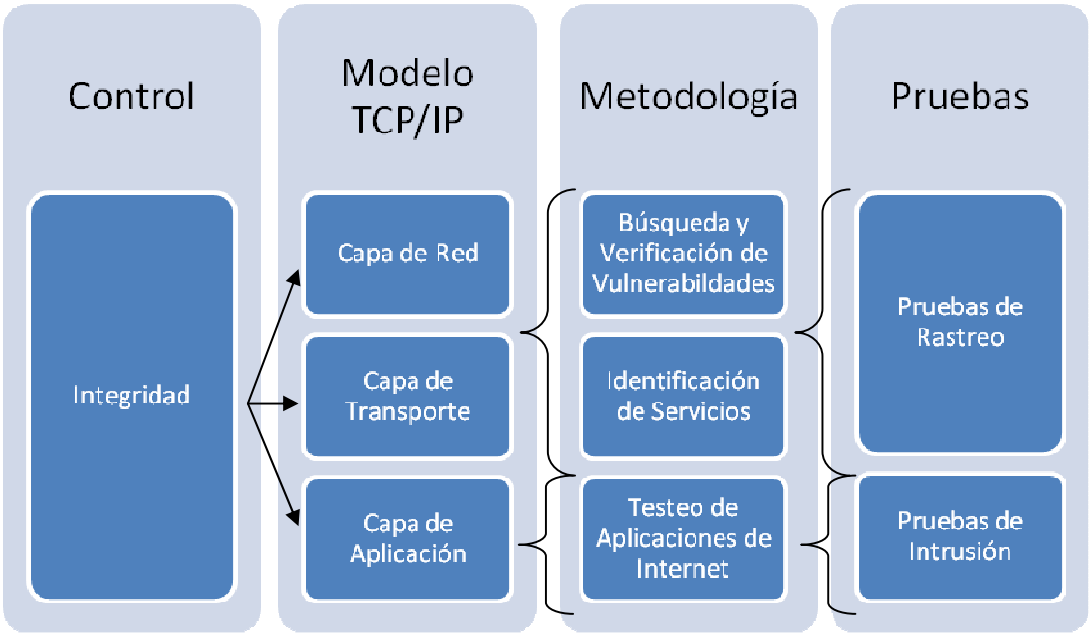

Figura 2- 2 Control de Integridad

2.1.1.2 Disponibilidad de la información

El cumplimiento del requerimiento de disponibilidad de la información, se garantizará utilizando el módulo de la metodología OSSTMM: Testeo de Aplicaciones por Internet.

El módulo de testeo de aplicaciones por Internet, será utilizado para realizar de pruebas de carga y stress que nos permitirá determinar el performance de la aplicación, ancho de banda entre otros.

En la figura 2-3 se describen los módulos utilizados para garantizar el cumplimiento del control de disponibilidad.

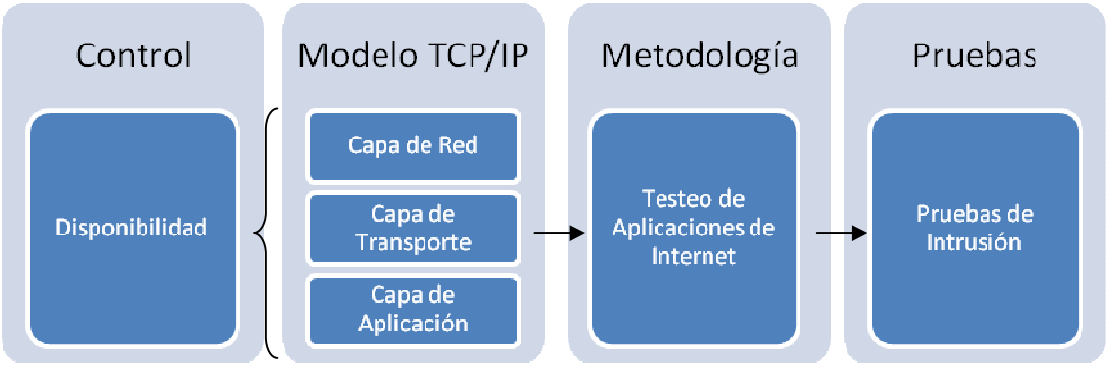

Figura 2- 3 Control de disponibilidad

2.1.1.3 Confidencialidad de la información

El cumplimiento del requerimiento de confidencialidad de la información, se garantizará utilizando el módulo de la metodología OSSTMM: Testeo de Aplicaciones por Internet.

El módulo de testeo de aplicaciones por Internet, será utilizado para realizar pruebas de autenticación de usuarios, y administración de sesiones que nos permitirá asegurar que la información pueda ser accedida por los usuarios autorizados.

En la figura 2-4 se describen los módulos utilizados para garantizar el control de confidencialidad.

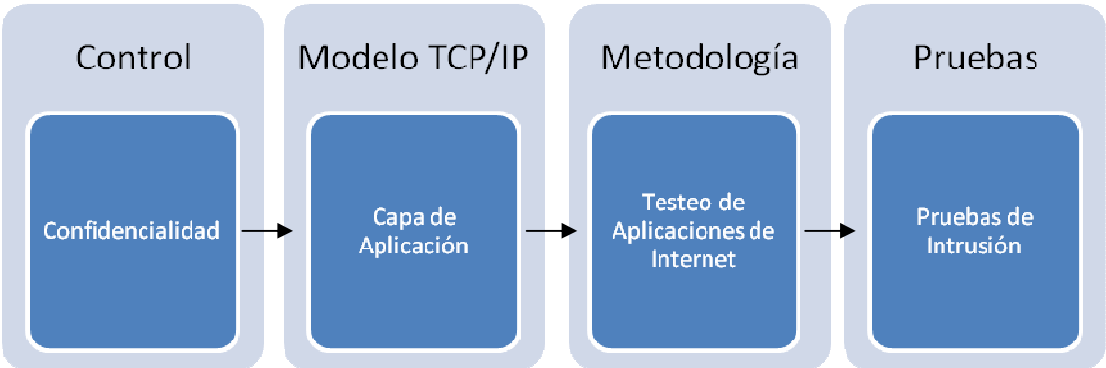

Figura 2- 4 Control de confidencialidad

2.1.2 REQUERIMIENTO 2: EL SISTEMA DE BANCA ELECTRÓNICA POR INTERNET DEBE CONTAR CON UN SISTEMA DE CONTROL Y AUTENTICACIÓN DE USUARIOS.

La implementación de este requerimiento, se garantizará utilizando el módulo de la metodología OSSTMM: Testeo de Aplicaciones de Internet.

El módulo de testeo de aplicaciones de Internet será utilizado para la creación de pruebas de autenticación y administración de sesiones, que permitirá realizar pruebas a los sistemas de autenticación.

En la figura 2-4 se describen los módulos utilizados para garantizar el correcto funcionamiento del sistema de control y de autenticación.

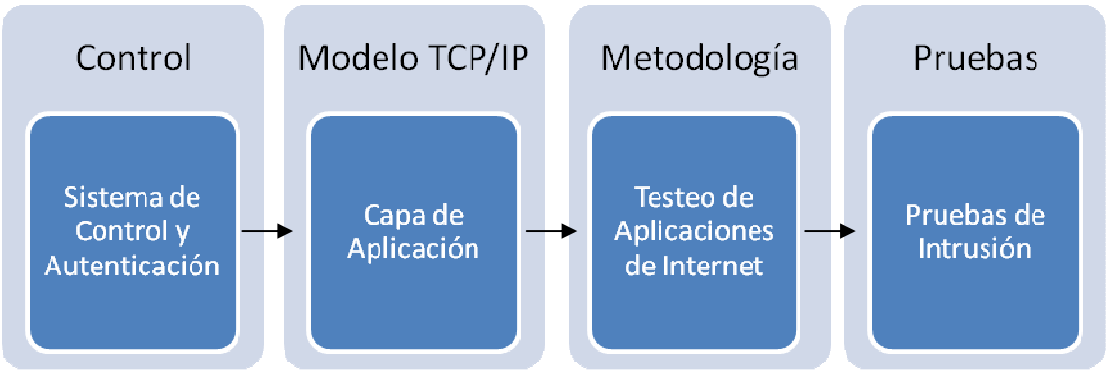

Figura 2- 5 Sistema de control y de autenticación

2.1.3 REQUERIMIENTO 3: LAS ENTIDADES FINANCIERAS QUE OFREZCAN SERVICIOS DE TRANSFERENCIAS Y TRANSACCIONES ELECTRÓNICAS DEBEN CONTAR CON PROCEDIMIENTOS DE SEGURIDAD PARA QUE LA INFORMACIÓN SEA ACCEDIDA ÚNICAMENTE POR PERSONAS AUTORIZADAS Y LA BANCA DEBE ASEGURAR QUE EL CANAL DE COMUNICACIÓN SEA SEGURO Y UTILICE TÉCNICAS DE ENCRIPTACIÓN.

El requerimiento 3 debe cumplir dos puntos importantes que son:

a) **Confidencialidad de la información:** Las entidades financieras que ofrezcan servicios de transferencia y transacciones electrónicas deben contar con

procedimientos de seguridad para que la información sea accedida únicamente por personas autorizadas.

b) **Canal de comunicación seguro utilizando técnicas de encriptación:** Las entidades financieras deben asegurar que el canal de comunicación sea seguro y utilice técnicas de encriptación.

2.1.3.1 Confidencialidad de la información

El cumplimiento del requerimiento se analizó en las secciones 2.1.1.3 y 2.1.2 de este capítulo.

2.1.3.2 Canal de comunicación seguro utilizando técnicas de encriptación

El cumplimiento de este requerimiento se garantizará utilizando el módulo de la metodología OSSTMM: Testeo de Aplicaciones de Internet.

El módulo de testeo de aplicaciones de Internet es utilizado para la verificación de la manipulación de datos de entrada y salida dentro de la aplicación, permitiendo determinar si el canal utilizado es seguro, realizando pruebas man in the middle, inyección de código y manipulación de datos.

En la figura 2-6 se describen los módulos utilizados para garantizar el control para determinar si el canal de comunicaciones es seguro.

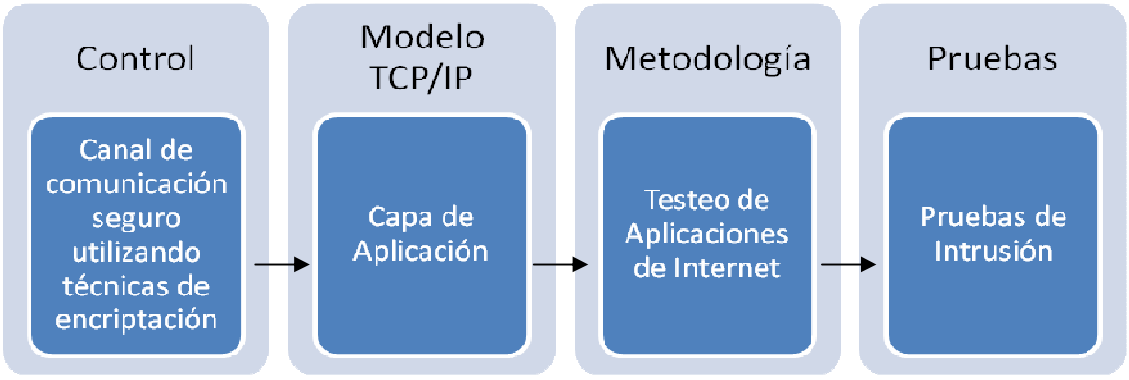

Figura 2- 6 Canal de comunicaciones

2.2 IDENTIFICACIÓN DE VULNERABILIDADES.

La identificación de vulnerabilidades del presente proyecto, está basado en el análisis de requerimientos de la sección 2.1. En el modelo de seguridad representado en la Figura 2-7, se identifica el área de análisis determinado por los requerimientos planteados en la normativa de la Superintendencia de Bancos y Seguros sobre Gestión y Administración de Riesgos Libro I Título X Capítulo V punto 4.3.4, descritos en la sección 2.1.

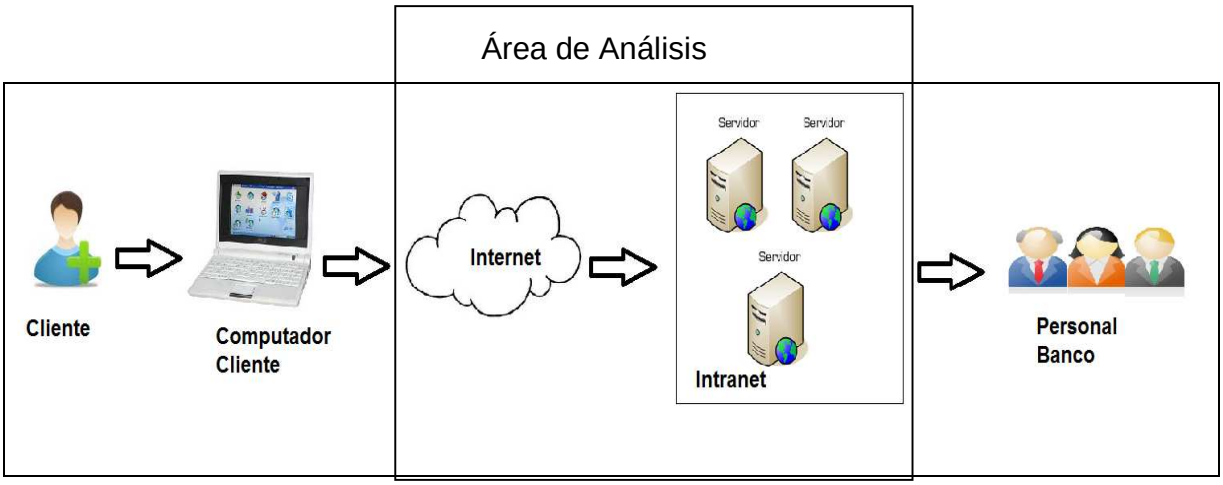

Figura 2- 7 Modelo de Seguridad (Área de análisis).

En esta sección se identificarán enemigos, vulnerabilidades y amenazas en los servicios de banca electrónica. Primero, para identificar enemigos se examinará el área de análisis del modelo de seguridad. Segundo, para identificar vulnerabilidades se analizarán aquellas que pueden presentarse en la infraestructura tecnológica de servicios por Internet de la banca. Finalmente, por cada vulnerabilidad existente se determinará una amenaza.

2.2.1 IDENTIFICAR ENEMIGOS

Un enemigo es una persona que haciendo uso de medios y conocimientos informáticos, robará información sensible para el cliente y su entidad financiera. El área de análisis del modelo de seguridad, determinado por el canal de comunicación (Internet) y la infraestructura tecnológica tiene los siguientes enemigos:

- Hacker: Un hacker es un delincuente informático, que ataca a la intranet del banco (ataques de denegación de servicios, inyección de código SQL, entre otros), o la computadora personal del cliente (ataques de virus, keyloggers, troyanos, phishing, entre otros) e intercepción de la comunicación (man in the middle).

- Personal del banco: El personal del banco tiene acceso a la información sensible de los clientes y de los procesos internos del banco, por lo que puede realizar ataques a la entidad financiera.

2.2.2 IDENTIFICAR VULNERABILIDADES

Una vulnerabilidad es “una debilidad en un sistema permitiendo a un atacante violar la confidencialidad, integridad, disponibilidad, control de acceso y consistencia del sistema o de sus datos y aplicaciones” [20].

Los tipos de vulnerabilidades en el sistema web de banca electrónica son: Identificación de vulnerabilidades por intrusión, de configuración y de la aplicación web.

2.2.2.1 Identificación de vulnerabilidades por intrusión.

Las vulnerabilidades por intrusión se presentan cuando un atacante consigue tener el control de un ordenador por medio de una puerta de enlace abierta (puerto de comunicaciones).

2.2.2.2 Identificación de vulnerabilidades por configuración.

Las vulnerabilidades por configuración ocurren cuando se realizan configuraciones por defecto en aplicaciones o servicios web. Las vulnerabilidades existentes son:

- Se habilitan servicios que no son ocupados.

- Sistemas de seguridad que facilitan el acceso no autorizado.

---

[20] http://www.alegsa.com.ar/Dic/vulnerabilidad.php, ALEGSA, Santa Fe, Argentina, 1998.

- Se utilizan protocolos de ruteo poco seguros.
- La mayoría de personas no modifican las contraseñas por defecto del fabricante.
- No realizar un adecuado mantenimiento de parches.

2.2.2.3 Identificación de vulnerabilidades de las aplicaciones web.

Una vulnerabilidad en la aplicación web permite que los hackers puedan manipular la aplicación con el objetivo de controlar el sistema.

Las vulnerabilidades de una aplicación web son:

- Gestión de Credenciales: Se basa en la gestión de usuarios, contraseñas y ficheros en los que se almacena dicha información, tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.
- Permisos, privilegios y control de acceso: Ocurre cuando la asignación de permisos o privilegios es defectuoso, permitiendo a todos los usuarios realizar funciones las cuales no están permitidas de acuerdo al rol de usuario, tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.
- Inyección de código: El código es introducido directamente al software permitiendo manipular un archivo, plantilla o biblioteca, con el objetivo de que el código se ejecute directamente en la aplicación y modifique su funcionamiento tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.
- Inyección SQL: Ocurre al nivel de base de datos de una aplicación, cuando se introduce código SQL a otro código intentando modificar su comportamiento, logrando ejecutar el código malicioso directamente en la base de datos, tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.

- Secuencias de comandos en sitios cruzados (XSS): Ocurre cuando un ataque puede ejecutar código como VBScript o javascript con el objetivo de pasar variables de una página HTML a otra sin iniciar sesión de usuario o para localizar puntos débiles, tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.

- Fallo de autenticación: Se produce cuando el sistema no es capaz de autenticar al usuario o proceso correctamente, tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.

- Falsificación de petición en sitios cruzados (CSRF): Ocurre cuando un atacante puede colocar en la página cualquier código con el objetivo de ejecutar acciones no planificadas, por ejemplo capturar cookies de sesión, tema tratado en el módulo de Testeo de Aplicaciones de Internet del manual OSSTMM.

2.2.3 IDENTIFICAR AMENAZAS

"Se puede definir como amenaza a todo elemento o acción capaz de atentar contra la seguridad de la información" [21]. Una amenaza puede ocurrir si existe una vulnerabilidad que pueda ser aprovechada en un sistema de información.

2.2.3.1 Identificar Amenazas de vulnerabilidades por intrusión y por configuración.

Las amenazas presentes cuando existen puertos abiertos en los servidores utilizados para proveer el servicio de banca de electrónica, y las configuraciones por defecto de aplicaciones o servicios de la entidad financiera son:

a) Ataques de denegación de servicio: Es un ataque realizado a un sistema de computadoras causando que un servicio o recurso sea inaccesible para los usuarios.

---

[21] http://www.seguridadinformatica.unlu.edu.ar/?q=node/12, Universidad Nacional de Luján, Buenos Aires, Argentina, 2012.

b) Ataques de envenenamiento ARP (Protocolo de resolución de direcciones): Es una técnica de ataque utilizada para infiltrase en una red, para leer paquetes LAN y modificar el tráfico e incluso detenerlo.

c) Inyección de código: La inyección de código permite al atacante manipular la información enviada en la aplicación.

d) Instalación de virus (Ataques de troyanos, backdoors, keyloggers, etc.): Permiten capturar información sensible del cliente, o alterar el funcionamiento normal de las aplicaciones y sistemas operativos.

2.2.3.2 Identificar Amenazas de vulnerabilidades de aplicaciones web.

Las amenazas presentes son aquellas existentes en el código fuente de la aplicación y estas son:

a) Suplantación de identidad: Ocurre cuando un hacker obtiene credenciales validas de un usuario, para realizar transacciones no autorizadas, capturando las credenciales por fallas de seguridad en la programación de la aplicación, por ejemplo credenciales almacenadas en campos ocultos o en cookies.

b) Robar información sensible sobre el estado almacenado en las cookies del cliente.

c) Saltarse los controles de acceso a la aplicación de banca electrónica.

d) Manipulación de datos de entrada y de salida: Ocurre cuando existen fallas de programación en los sistemas, ya que guardar información sobre estado o peticiones en campos ocultos de la aplicación, por lo que un hacker puede cambiar la información contenida en los mismos.

2.3 PRUEBAS DE RASTREO

En esta sección del proyecto se realizarán las pruebas de rastreo para analizar las vulnerabilidades por intrusión y de configuración. Estas pruebas permiten cumplir con

el análisis del modelo de seguridad en el canal de comunicación (Internet) y la intranet del banco.

Las pruebas de rastreo se encuentran basadas en las tareas descritas en los módulos de: búsqueda y verificación de vulnerabilidades e identificación de los servicios de sistemas, de la sección de seguridad en las tecnologías de Internet del manual OSSTMM. Estas pruebas comprenden la descripción teórica, la herramienta recomendada y el comando o módulo de cada herramienta que permite cumplir cada tarea, finalmente se presenta una lista de resultados esperados al finalizar cada prueba.

2.3.1 PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA LA DETECCIÓN DE VULNERABILIDADES POR INTRUSIÓN.

Paso 1.- Enumeración de sistemas

La enumeración de sistemas permitirá determinar los recursos y servicios disponibles en el área de red a analizar. Uno de los primeros pasos para un analista de seguridad es reducir el número de rango de direcciones IP en una lista de equipos activos. Para la enumeración de sistemas se realizaran tareas que permitan: recoger respuestas de broadcast de una red, técnicas para traspasar los cortafuegos, determinar el nombre de los servidores. Las tareas que se deben cumplir están descritas en el manual OSSTMM de enumeración de sistemas y son detalladas a continuación: [22]

---

[22] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

1. Recoger respuestas de broadcast desde la red.

Broadcast de una red o dominio de difusión es una forma de trasmisión de información, en donde cualquier host emisor puede transmitir mensajes directamente a una multitud de host receptores sin tener que retransmitir el mensaje host por host.[23] Recolectar las respuestas de broadcast de una red permite al analista de seguridad obtener información enviada a un segmento de red.

- Herramienta Wireshark.

Esta herramienta permite capturar el tráfico generado entre el cliente y el servidor mediante la opción capturar e iniciar. Para que Wireshark permita capturar el tráfico de broadcast de una red se debe utilizar el filtro basado en red conocido como "broadcast" Figura 2-8.

Figura 2- 8 Filtro de Wireshark

En la herramienta Wireshark, se desplegará la siguiente información.

a) **No.**: El número de paquetes que son enviados o recibidos.

b) **Tiempo:** Es el tiempo que tarda en transmitirse el paquete.

c) **Origen:** La IP de origen del paquete.

d) **Destino:** La IP destino.

---

[23] http://es.wikipedia.org/wiki/Broadcast_%28inform%C3%A1tica%29, Noviembre, 2012.

e) **Protocolo:** El protocolo utilizado para la transmisión

f) **Info:** Información sobre el paquete de datos.

Wireshark presentará primero la información de los paquetes capturados, segundo al seleccionar un paquete presentará información sobre el protocolo y los campos correspondientes del paquete. Finalmente se presentará el contenido del paquete en formato hexadecimal.

2. Intentar traspasar el cortafuego con valores estratégicos de TTLs (Tiempo de vida) (Firewalking) para todas las direcciones IP.

Los cortafuegos son herramientas de seguridad que ofrecen protección en una red. Un cortafuegos permite controlar el tráfico generado dentro de una red, evita la instalación de nuevos servicios. El objetivo del analista de seguridad es controlar el acceso y auditar servicios, permitiendo o denegando el flujo de paquetes por el cortafuegos.

Firewalking es una técnica utilizada por traceroute que permite determinar que técnicas de filtrado son utilizadas en equipos de transporte de paquetes como los cortafuegos.

- Herramienta Firewalk.

La herramienta Firewalk utiliza la técnica de expiración IP (Protocolo de Internet) que consiste en manipular el tiempo de vida (TTL) de la cabecera IP, para mapear los routers o saltos realizados para la entrega del paquete desde el host origen al destino.

Para la utilización de la herramienta Firewalk se utilizan dos tipos de escaneo: Con ACL (Listas de control de acceso) y Sin ACL.

a) Escáner sin ACL:

Opción -s: Permite especificar los puertos por donde se realizaran las pruebas.

b) Escáner con ACL:

Opción –p: Permite especificar el protocolo a ser escaneado siendo este TCP (Protocolo de control de transmisión) o UDP (Protocolo de Datagrama de Usuario).

3. Emplear ICMP (Protocolo de Mensajes de Control de Internet) y resolución inversa de nombres con el objetivo de determinar la existencia de todos los sistemas en la red.

El protocolo de mensajes de control de Internet (ICMP) se encarga de controlar y enviar mensajes de error del protocolo de Internet (IP), los mensajes de error indican por ejemplo que servicio no puede ser accedido, o que host no puede ser localizado, entre otros. Los mensajes ICMP son enviados en respuesta a un datagrama IP mal generado.

Enviar datagramas erróneos permitirá al analista de seguridad obtener los nombres de los host y los servicios que no se encuentran disponibles para determinar los servicios disponibles dentro de la infraestructura tecnológica del banco.

- Herramienta Nmap.

Opción -sP (Sondeo ping): Envía una solicitud de eco ICMP en un paquete TCP (Protocolo de control de transmisión) utilizando por defecto el puerto 80. Esta opción permite determinar los sistemas existentes en la red mediante un sondeo ping.

Opción -R (Realizar resolución de nombres con todos los objetivos): Se utiliza DNS (Sistemas de Dominio Inverso) inverso de las IP para la resolución de nombres.

4. Emplear paquetes TCP con puerto origen 80 y el bit ACK (Acuse de recibo) activo en los puertos de destino 3100-3150, 1001-10050, 33500-33550 y 50 puertos aleatorios por encima del 35000 para todos los sistemas de la red.

El protocolo de control de transmisión (TCP), es orientado a conexión permitiendo que las dos computadoras que se encuentren conectadas controlen el estado de la transmisión. En una comunicación TCP interviene una conexión cliente/servidor, y

esta se realiza en ambas direcciones. En la figura 2-9 se muestra la comunicación en ambas direcciones. El protocolo permite garantizar la transferencia de datos porque tiene un acuse de recibo (ACK).

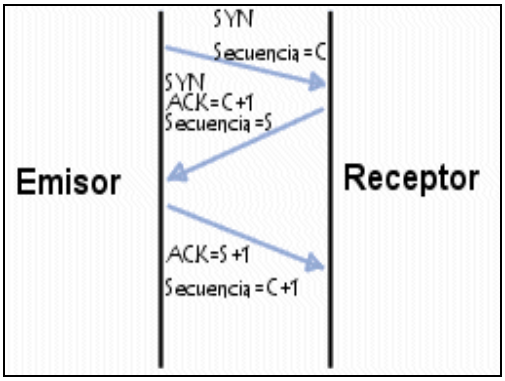

Figura 2- 9 Comunicación en línea [24]

El protocolo envía paquetes ACK que indica que se ha recibido una conexión TCP, pero al no enviar SYN (sincronizar) como primer paquete no se ha establecido una conexión TCP, por estos los sistemas responderán con un paquete RST que permitirá determinar si los puertos están disponibles.

- Herramienta Nmap.

Opción -PA [lista de puertos] (Ping TCP ACK): Envía un paquete TCP con la bandera ACK, puerto utilizado por defecto 80, para enviar a los diferentes puertos utilizar el comando -PA 3100-3150, 1001-10050, 33500-33550, 50.

Opción --source-port <número de puerto> o -g < número de puerto >: La herramienta Nmap obliga a que se utilice un puerto de origen fijo.

5. **Emplear paquetes TCP fragmentados en orden inverso mediante escaneos FIN , NULL y XMAS en los puertos destino 21, 22, 25, 80 y 443 para todos los servidores de la red.**

---

[24] http://es.kioskea.net/contents/Internet/tcp.php3, Mayo, 2012.

El paquete de conexión FIN es enviado por una de las partes de la conexión cliente/servidor, terminando la conexión desde cada lado independientemente. Si un puerto abierto recibe un paquete FIN ignora el paquete, mientras que los puertos cerrados envían un paquete RST.

En el paquete de conexión NULL, se envían las señales (URG, ACK, SYN, FIN, RST), pero mantienen desactivada la cabecera. Si el puerto está abierto no se recibe respuesta, si el puerto está cerrado se recibe RST|ACK.

En el paquete de conexión XMAS envía señales parecidas al paquete NULL, pero la cabecera TCP esta activada. Si el puerto está abierto no se recibe respuesta pero si está cerrado se recibe RST|ACK.

- Herramienta Nmap.

Opción -sF (FIN): Envía un paquete FIN vacio para la prueba.

Opción -sX (XMAS): Activa las banderas FIN, URG y PUSH.

Opción -sN (NULL): Desactiva todas las banderas.

6. Usar escaneos TCP SYN sobre los puertos 21, 22, 25, 80 y 443 para todos los servidores de la red.

Una señal de conexión SYN, indica un pedido para establecer una conexión entre cliente/servidor. Un escaneo TCP SYN no establece una conexión completa, por el contrario se envía una señal SYN y si se recibe la señal SYN|ACK indicando que un puerto determinado esta en escucha y finalmente se envía un paquete RST para cerrar la conexión.

- Herramienta Nmap

Opción -PS [lista de puertos] (Ping TCP SYN): Envía un paquete TCP con la bandera SYN, puerto utilizado por defecto 80, para enviar a los diferentes puertos utilizar el comando -PS 21,22,25,80,443.

7. Emplear intentos de conexión a DNS (Sistema de nombres de dominio) para todos los servidores de la red.

El sistema de nombres de dominio (DNS), intenta traducir nombres que son fáciles de entender para las personas en códigos binarios que pueden ser entendidos por las computadoras. El uso más común de DNS es asignarle nombres de dominio a las direcciones IP. En la figura 2-10 se muestra la resolución de nombres por DNS.

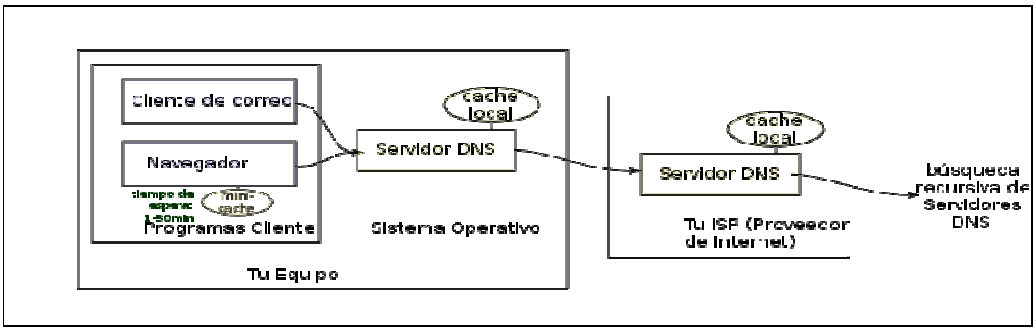

Figura 2- 10 Resolución de nombres DNS. [25]

- Herramienta Nmap.

Comando --system-dns (Utilizar resolución DNS del sistema): Envía consultas directamente a los nombres de servidores configurados y espera la respuesta.

8. Emplear FTP (Protocolo de Transferencia de Archivos) y Proxys para escanear al interior de la DMZ para los puertos 22, 81, 111,132, 137 y 161 para todos los servidores de la red.

Emplear el protocolo FTP para realizar escaneos de puertos es conocido como sondeo de rebote FTP, utilizando un servidor de archivos que permitirá enviar un fichero a cada puerto emitiendo el comando “port” para cada host de la intranet bancaria, de acuerdo al mensaje error recibido se conocerá si el puerto está abierto o cerrado.

---

[25] http://es.wikipedia.org/wiki/Domain_Name_System, Mayo, 2012.

El analista de seguridad envía el comando “port” y el número de puerto, si se recibe un paquete ACK se determina que el puerto se encuentra abierto, si se recibe el paquete RST el puerto se encuentra cerrado.

Emplear los proxy para realizar escaneos de puertos es conocido como sondeo de rebote con proxy. Un escaneo de proxy es utilizado para enmascarar la dirección IP utilizada para el análisis de seguridad.

- Herramienta Nmap.

Opción -b <sistema de rebote ftp> (sondeo de rebote FTP): Nmap permite usar conexiones FTP, para hacer un sondeo de puertos a otro sistema.

- Herramienta ppscan.c

Opción./ppscan: permite el sondeo de rebote con proxy.

Paso 2.-Sondeo de puertos.

El sondeo de puertos consiste en establecer conexiones con los puertos para determinar los servicios activos dentro de la intranet. El analista de seguridad debe conocer que puertos se encuentran abiertos, cerrados o filtrados en los equipos activos, utilizando el escaneo de puertos TCP y UDP. Las tareas que debe cumplir el analista de seguridad descritas en el manual OSSTMM para la enumeración de puertos son detalladas a continuación: [26]

1. **Usar escaneos SYN TCP (llamado también Half-Open) para enumerar puertos abiertos, cerrados o filtrados para aquellos puertos TCP utilizados en todos los servidores de la red.**

---

[26] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000

Este sondeo no llega a abrir una conexión TCP completa y determina que puertos están abiertos, cerrados y filtrados. Este sondeo es conocido como silencioso, así existen menos probabilidades que se lo registre en la intranet bancaria. Para realizar este tipo de sondeo se necesitan permisos de root.

- Herramienta Nmap.

Opción -sS (Sondeo TCP SYN): Nmap envía un paquete SYN si recibe un paquete SYN/ACK indica que el puerto está abierto, si recibe un paquete RST (Reset) indica que el puerto está cerrado, y si no se recibe ninguna respuesta luego de varios intentos o se recibe un error de tipo ICMP se lo clasifica como filtrado

- Herramienta Nessus.

Opción SYN Scan: Nessus determinará los puertos TCP abiertos. Nessus enviará un paquete SYN en espera de un paquete SYN|ACK definiendo el puerto como abierto, si no se recibe respuesta definirá el puerto como cerrado.

2. **Usar escaneos TCP full connect para escanear todos los puertos por encima del 1024 en todos los servidores de la red.**

Este tipo de sondeo es utilizado cuando no se puede utilizar el SYN TCP por falta de privilegios de usuarios. TCP full realiza una conexión completa realizando los pasos de la conexión (SYN, SYN|ACK, ACK).

- Herramienta Nmap.

Opción -sT (sondeo TCP connect()): Nmap solicita al sistema operativo que se realice una llamada del sistema connect(), que escriben los paquetes a alto nivel.

3. **Usar escaneos TCP fragmentados en orden inverso para enumerar puertos y servicios para el conjunto de puertos definidos en el *Anexo 1: Puertos de Análisis para los Servidores de Red* por defecto para todos los servidores de la red.**

Escaneo por fragmentación de cabecera TCP permite que el análisis de seguridad no sea fácilmente detectado, ya que divide la cabecera en partes.

- Herramienta Nmap.

Opción -f (fragmentar los paquetes); --mtu (utilizar el MTU (Unidad Máxima de Transferencia) especificado): Nmap fragmenta la cabecera TCP para que los filtros de paquetes, IDS, etc., no puedan detectar lo que se está haciendo.

4. **Usar escaneos UDP para enumerar puertos abiertos o cerrados para los puertos UDP por defecto si UDP no está siendo filtrado. [Recomendación: primero comprobar el sistema de filtrado para un subconjunto de puertos UDP.]**

EL protocolo UDP es un protocolo sin estado, que significa que la comunicación no se realiza con diálogos de protocolo de enlace por lo que el sondeo UDP envía una cabecera sin datos para cada puerto.

- Herramienta Nmap.

Opción -sU (Sondeo UDP): Nmap recibe una error ICMP (tipo3, codigo3) entonces se marca como cerrado, si se recibe un error ICMP (tipo3, códigos 1,2,9,10, o 13) se marca el puerto como filtrado, si se recibe una respuesta al paquete UDP significa que el puerto está abierto y si no se recibe ninguna respuesta después de varias transmisiones se cataloga al puerto como abierto | filtrado.

- Herramienta Netcat.

Opción -u: Permite determinar que puertos UDP se encuentran abiertos. Los puertos abiertos tendrán el mensaje “éxito”.

Opción -vz: Permite presentar información más descriptiva en la cual los puertos abiertos tendrán el mensaje “éxito” y los puertos cerrados tendrán el mensaje “fallido”.

Paso 3.- Identificación de servicios.

La identificación de servicios permite determinar qué servicios se encuentran habilitados para el uso de los clientes en la intranet del banco. Las tareas realizadas para la identificación de servicios relacionarán cada puerto abierto determinado en el sondeo de puertos con el servicio relacionado y se determinará los niveles de parchado del sistema y aplicación. Las principales tareas descritas en el manual OSSTMM para la identificación de servicios son detalladas a continuación: [27]

1. Relacionar cada puerto abierto con un servicio y protocolo.

Detectar cada servicio y posibles vulnerabilidades existentes de acuerdo al puerto que se encuentra abierto en el servidor ya que asocia el servicio por el puerto en que comúnmente es utilizado.

Por ejemplo:

Si el puerto abierto es el 21 comúnmente utilizado por FTP, se asocia que el servicio activo es FTP.

- Herramienta Nmap.

Opción -sV: Nmap determina el tipo de servicio y su versión mediante una consulta a su base de datos “nmap-service-probes” donde se puede consultar y reconocer los servicios, dependiendo exclusivamente del número de puerto que se encuentra abierto.

Opción -sO: Nmap trata de comprobar los protocolos que soporta el servidor o computador personal.

---

[27] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000

- Herramienta Nessus.

La opción de Nessus “probe services on every port”, intenta descubrir mediante los puertos abiertos los servicios que se ejecutan.

- Herramienta Netcat.

Opción -z: Permite determinar que puertos TCP se encuentran abiertos y el servicio al que pertenece ese puerto.

2. Identificar el nivel de parcheado del sistema a partir de su up-time.

En sistemas operativos GNU/Linux el nivel del parcheado del sistema operativo puede ser consultado desde consola con el comando “uname –a”, donde el resultado del comando describe:

a) El nombre y versión del sistema operativo

b) Nivel de parcheado

c) El procesador de la máquina.

3. Verificar la aplicación y su versión en el sistema.

Determinar la aplicación que se encuentra ejecutándose como servicio por el puerto y su versión ayuda al analista de seguridad ya que en cada versión de software existen vulnerabilidades propias y estas pueden ser explotadas.

- Herramienta Nmap.

Para determinar las versiones, en Nmap pueden ser utilizar las siguientes opciones:

Opción -sV (Detección de versiones): Activa la detección de versiones.

Opción --allports (No excluir ningún puerto de la detección de versiones): Nmap permite excluir puertos del análisis mediante la opción Exclude, con la opción allport se realiza la detección de versiones de todos los puertos aun cuando se encuentren excluidos.

Opción --version-light (Activar modo ligero): Nmap envía una serie de sondas cuando se desea determinar versiones con una intensidad del 1 a 9. El valor de intensidad es el número de sondas enviadas para determinar el servicio, esta opción permite determinar versiones de forma más rápida, mientras más alta es la intensidad hay mayor probabilidad de identificar el servicio.

Opción --version-all (Utilizar todas las sondas): Esta opción hace que se utilice todas las sondas en cada puerto con intensidad nivel 9.

Opción --version-trace (Trazar actividad de sondeo de versiones): Esta opción permite obtener información sobre lo que está ocurriendo en el sondeo de versiones.

Paso 4.- Detección del sistema operativo.

Si un analista de seguridad conoce el sistema operativo del computador a analizar puede determinar las vulnerabilidades propias de cada sistema y explotarlas.

Las principales tareas descritas en el manual OSSTMM para la identificación del sistema operativo son detalladas a continuación: [28]

1. **Examinar las respuestas de los sistemas para determinar el tipo de sistema operativo y su nivel de parcheado.**

Para determinar el sistema operativo, se utiliza una técnica conocida como “OS fingerprinting” que consiste en analizar las huellas dejadas en las conexiones de red. Estas huellas se basan en los tiempos de respuesta utilizados para la conexiones TCP/IP.

Existen dos tipos de escaneo utilizado por las herramientas que ejecutan esta técnica:

---

[28] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

a) Escaneo activo: La herramienta envía los paquetes esperando la respuesta para poder analizar los tiempos de respuesta.

b) Escaneo pasivo: La herramienta escucha el tráfico en la red para poder capturarlo y analizarlo.

- Herramienta Nmap.

Nmap permite determinar el sistema operativo mediante el envió de paquetes TCP y UDP al sistema remoto analizando prácticamente todos los bits de respuestas. Nmap compara resultados de una serie de pruebas tales como análisis ISN (Selección del número inicial de secuencia) de TCP, soporte de opciones TCP y su orden, entre otras, con la base de datos “nmap-os-fingerprints”. Si existe una coincidencia se presentan los detalles del sistema operativo.

Para la detección del sistema operativo se puede utilizar las siguientes opciones:

Opción -O Activa la detección del sistema operativo.

Opción -A Activa la detección del sistema operativo y versiones.

2. Verificar la predicción de secuencia TCP para todos los servidores de la red.

Cuando el protocolo de control de transferencia envía un segmento este es asociado a un número de secuencia, cuando el SYN está fijo en 0 el número de secuencia es la primera palabra del segmento actual, si el SYN está fijado en 1 el número de secuencia es igual al número de secuencia inicial. En la figura 2-11 se puede observar el cambio del número de secuencia por cada envió y recepción de mensajes.

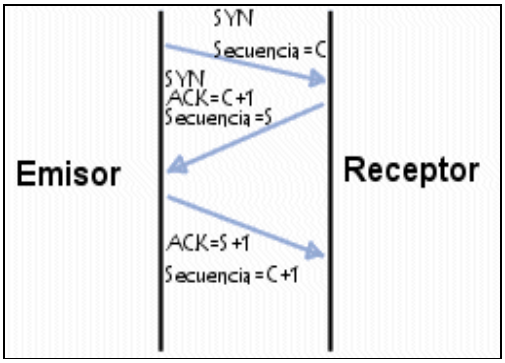

Figura 2- 11 Número de secuencia [29]

- Wireshark

Esta herramienta permite capturar el trafico generado entre el cliente y el servidor mediante la opción Capturar e Iniciar. Para que Wireshark permita capturar todos los segmentos TCP de una red se debe utilizar el filtro basado en red conocido como “ip proto\ tcp”.

Wireshark presentará primero la información de los paquetes capturados, segundo al seleccionar un paquete presentará información sobre el protocolo y los campos correspondientes del paquete. Finalmente se presentará el contenido del paquete en formato hexadecimal.

3. **Busque ofertas de trabajo donde obtener información sobre los servidores y aplicaciones del objetivo, con la finalidad de obtener información sobre la tecnología utilizada en la entidad financiera.**

---

[29] http://es.kioskea.net/contents/Internet/tcp.php3, Mayo,2012.

Para realizar las tareas 3 se puede buscar información sobre la empresa en Internet, en portales de empleo, o en la página web oficial de la empresa, si esta cuenta con la opción de “trabaja con nosotros”.

Paso 5.- Verificación de respuestas para varios protocolos

La verificación de respuestas de varios protocolos permitirá determinar que protocolos son utilizados en la infraestructura tecnológica de la banca. Se realizarán tareas para determinar protocolos de enrutamiento, cifrados y protocolos sobre IPv6. Las principales tareas descritas en el manual OSSTMM para la verificación de respuestas para varios protocolos son detalladas a continuación: [30]

1. Verificar y examinar el uso de tráfico y protocolos de enrutamiento.

Los protocolos de enrutamiento permiten que los routers determinen la ruta por la cual se enviarán los datos. Si existen varias rutas, el router seleccionará el camino más corto (de acuerdo a la métrica utilizada).

Los protocolos de enrutamiento pueden ser estáticos (son los generados por el administrador de la red) o dinámicos (el administrador solo configura el protocolo).

En la figura 2-12 se muestra un esquema de protocolos de enrutamiento.

---

[30] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

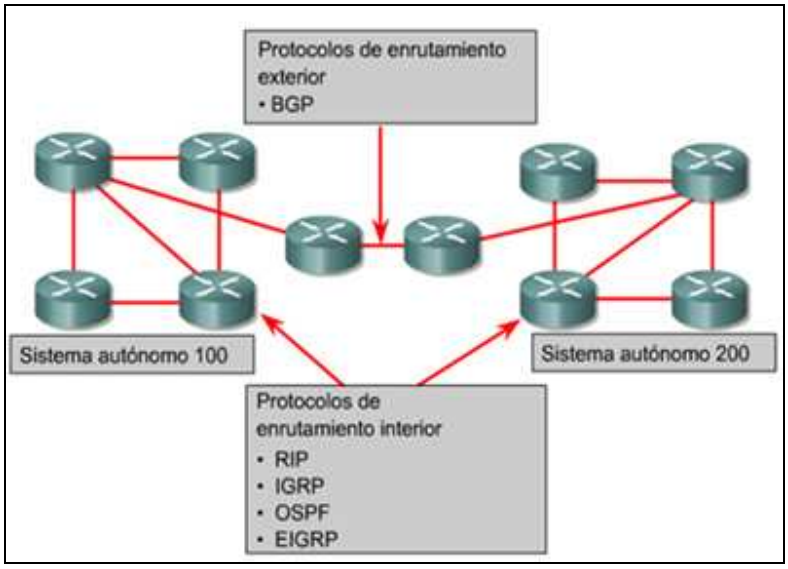

Figura 2- 12 Protocolos de enrutamiento [31]

- Herramienta Nmap.

Opción -sO (sondeo de protocolo IP): El sondeo de protocolo IP permite determinar que protocolos son soportados por el sistema. El sondeo de protocolos envía datagramas IP vacios (no contienen datos), si Nmap recibe un mensaje de error ICMP tipo inalcanzable (tipo 3, código 2) se marca el protocolo como cerrado, si se recibe un mensaje de error (tipo 3, códigos 1, 3, 9, 10, o 13) se marca el protocolo como filtrado, si no se recibe nada se marca como abierto | filtrado y si recibe respuestas se marca como abierto.

2. Verificar y examinar el uso de protocolos cifrados.

Un protocolo cifrado es aquel que realiza funciones relacionadas con seguridad mediante los métodos criptográficos.

---

[31] http://fortalezadigital08.wordpress.com/2008/09/23/protocolos-de-enrutamiento-parte-1/, Mayo, 2012.

Este tipo de protocolos son utilizados para enviar datos sensibles en la capa de aplicación. Estos protocolos incorporan por lo menos uno de los siguientes aspectos:

a) Cifrado simétrico y autenticación de mensajes: El emisor y el receptor utilizan una misma clave que intervienen en la comunicación donde se intercambian la clave simétrica, que les permita cifrar o descifrar.

b) Establecimiento de claves: Los protocolos de establecimiento de claves envían información secreta compartida conocida como clave, la cual es usada como clave de un algoritmo criptográfico.

c) Transporte de datos de forma segura: Existen protocolos en los cuales se puede enviar la información correspondiente de forma segura. En el caso de los servicios de banca electrónica la información secreta compartida viaja por HTTPS (Protocolo seguro de transferencia de hipertexto). [32]

- Herramienta Nmap.

Opción -sO (sondeo de protocolo IP): Opción descrita en la tarea 1 del paso 5.

3. Verificar y examinar el uso de TCP e ICMP sobre IPV6.

TCP es un protocolo de comunicación orientado a conexión y que garantiza que los datos serán entregados en su destino ya que implemente el acuse de recibo. El protocolo TCP corresponde a la capa de transporte, proporcionando un servicio de comunicación intermedio entre un programa de aplicación y el protocolo de Internet. Para realizar el cálculo de la suma de comprobación (Checksums) de TCP sobre IPv6 se debe incluir las direcciones IPv6 de 128 bits en lugar de 32-bit de las direcciones IPv4. [33]

---

[32] http://es.wikipedia.org/wiki/Protocolo_criptogr%C3%A1fico, Mayo, 2012.

[33] http://tools.ietf.org/html/rfc2460, Noviembre, 2012.

El protocolo ICMP sobre IPv6 combina las funciones de los protocolos ICMP, IGMP, ARP y elimina tipos de mensajes obsoletos. El objetivo principal de este protocolo es el envió de mensajes, tiene 2 tipos de mensajes: de error y de información. Los mensajes de error se dividen en 4 categorías: destino inaccesible, paquete demasiado grande, tiempo excedido y problemas de parámetros. Los mensajes de información se subdividen en tres grupos: mensajes de diagnóstico, mensajes para la administración de grupos multicast y mensajes descubrimiento de vecinos, de routers y de parámetros. [34]

Además, ICMP sobre IPv6 está diseñado para cumplir las siguientes funciones: detectar errores, realizar diagnósticos y detectar direcciones IPv6 multicast.

- Herramienta Nmap.

Opción -6 (Activa el sondeo IPv6): Nmap tiene soporte para escanear direcciones IPv6.
Para utilizar el escaneo en TCP se debe utilizar la opción –sT descrita en sondeo de puertos y para el escaneo ICMP se puede utilizar la opción –sP.

Paso 6.- Verificación de Respuestas a Nivel de Paquete

La verificación de las respuestas a nivel de paquete se realiza mediante el análisis de los paquetes capturados, este análisis permitirá al analista de seguridad determinar la predictibilidad de las secuencias TCP. Las principales tareas descritas en el manual OSSTMM para la verificación de respuestas a nivel de paquete son detalladas a continuación: [35]

1. **Identificar la predictibilidad de las secuencias TCP e identificar la predictibilidad de los números de secuencia TCP ISN.**

---

[34] http://es.wikipedia.org/wiki/ICMPv6, Noviembre, 2012.

[35] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

De acuerdo al análisis realizado en el paso 4 tarea 2 “Identificar la predictibilidad de la secuencia TCP”, se debe realizar un análisis de envió de paquete que permita determinar una regla común entre la asignación de secuencia.

- Wireshark

Esta herramienta permite capturar el trafico generado entre el cliente y el servidor mediante la opción Capturar e Iniciar. Para que Wireshark permita capturar el tráfico de los segmentos TCP de una red, se debe utilizar el filtro basado en red conocido como “ip proto \tcp”.

Wireshark presentará primero la información de los paquetes capturados, segundo al seleccionar un paquete presentará información sobre el protocolo y los campos correspondientes del paquete. Finalmente se presentará el contenido del paquete en formato hexadecimal.

2. Identificar el up-time del sistema.

En sistemas operativos GNU/Linux se puede conocer el tiempo que está funcionando sin interrupciones el sistema con el comando uptime.

En sistemas operativos Windows se puede conocer el tiempo que está funcionando sin interrupciones el sistema con el comando net statistics server.

Paso 7.- Resultados esperados y su análisis.

Los resultados esperados al realizar la detección de vulnerabilidades de puertos y servicios son detallados a continuación:

- Puertos abiertos, cerrados y filtrados: El analista de seguridad debe presentar un listado con los puertos abiertos, cerrados y filtrados de los servidores de la intranet bancaria analizados.
- Direcciones IP de los sistemas activos: El analista de seguridad debe presentar un listado de las direcciones IP de las computadoras activas en la intranet bancaria.

- Servicios activos: El analista de seguridad debe presentar un listado de los puertos abiertos en los servidores de la intranet bancaria asociados a un servicio. Por ejemplo: Si el puerto abierto es el 21 el servicio activo será FTP.
- Tipos de Servicios: El analista de seguridad debe clasificar los servicios activos en tipos de servicios. Ejemplo: Servicios de correo, servicios FTP, etc.
- Tipo y nivel de parcheado de las Aplicaciones de los Servicios: El analista de seguridad debe presentar un listado sobre el nivel de parcheado de los servicios activos encontrados en la intranet bancaria.
- Tipo de Sistema Operativo: El analista de seguridad debe presentar un listado de los resultados de las técnicas de detección de sistema operativo realizado a los servidores de la intranet bancaria.
- Mapa de la red: El analista de seguridad al finalizar las pruebas de rastreo de vulnerabilidades por intrusión debe contar con una mapa de red de la intranet del banco analizada.[36]

La plantilla de los informes requeridos para el procedimiento formal de ethical hacking para vulnerabilidades por intrusión se encuentra detallado en el ***Anexo 2: 2.1 Rastreo de Vulnerabilidades por intrusión***.

2.3.2 PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA LA DETECCIÓN DE VULNERABILIDADES POR CONFIGURACIÓN.

Paso 1.- Búsqueda de vulnerabilidades por configuración.

La búsqueda de vulnerabilidades por configuración permitirá al analista de seguridad determinar qué sistemas han sido instalados con configuración por defecto.

---

[36] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

Las tareas que debe cumplir el analista de seguridad descritas en el manual OSSTMM para la búsqueda de vulnerabilidades por configuración son detalladas a continuación: [37]

1. Identificar todas las vulnerabilidades relativas a los sistemas operativos.

Seguridad de Sistemas Operativos Linux.

La familia de sistemas operativos Linux, son compatibles con Unix, las principales características de Linux se describen a continuación:

- Para utilizar la familia de sistemas operativos Linux no se requiere pagar por la licencia a ninguna casa desarrolladora.
- Los sistemas operativos vienen con código fuente, permitiendo que sea mejorado por el usuario.

Las características de seguridad son descritas a continuación:

a) Linux es un sistema multiusuario

Linux reconoce al administrador del sistema como root o superusuario, dándole a este todos los permisos para realizar cualquier operación, mientras que un usuario común tienen permisos limitados.

b) No hay archivos ejecutables ni registro

El código malicioso que afecta a la familia de sistemas operativos Windows generalmente son archivos ejecutables que pueden guardar configuraciones en el registro, en Linux el usuario debería guardar el archivo, darle los derechos de ejecución y finalmente ejecutarlo.

---

[37] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

De igual manera Linux utiliza archivos de configuración en sustitución del registro que es utilizado por Windows. Mejor Configuración por defecto. La configuración por defecto hace que los usuarios tengan privilegios limitados, mientras que en Windows los usuarios generalmente tienen privilegios de administrador.

c) Sistema modular

El diseño de Linux hace posible eliminar un componente del sistema en caso de que sea necesario, porque algún virus ha infectado nuestro sistema.

Seguridad de Windows.

Windows es la familia de sistemas operativos gráficos desarrollados por Microsoft Corporation. Uno de los principales problemas encontrados en estos sistemas operativos es la debilidad en seguridad y el alto índice de vulnerabilidades existentes. Sin embargo esta familia de sistemas operativos es la más utilizada en computadoras personales y que hace pocos años fue superado en servidores por Linux. La familia de sistemas operativos Windows cuentan con la versión actual de Windows Server 2008 R2 que cuentan con las siguientes características de seguridad:

- Este sistema operativo caracteriza la función que se le da al servidor, permitiendo deshabilitar los servicios que no son utilizados por el servidor.
- Posee una función de encriptación de datos.
- Este sistema operativo cuenta con analizadores de procesos que permite auditar el servidor para verificar que este tenga los últimos parches de seguridad. [38]

---

[38] http://www.microsoft.com/en-us/server-cloud/windows-server/2008-r2-benefits.aspx, Enero, 2012.

- Herramienta Nikto.

Opción -h <host>: Esta opción permite a Nikto escanear un host para buscar vulnerabilidades de configuración.

- Herramienta N-Stealth.

N-Stealth es una herramienta de interfaz gráfica amigable para el usuario y de muy fácil usabilidad. Para que N-Stealth pueda realizar un análisis de vulnerabilidades por configuración, se debe ingresar en la pestaña técnicas hacking y luego de seleccionar opciones especiales se debe activar la opción de "ver problemas de configuración", además se debe especificar el host que se desea analizar.

2. Intentar ajustar vulnerabilidades a servicios.

De acuerdo a las vulnerabilidades encontradas se puede clasificar las que pertenecen al sistema operativo y las que son propias de los servicios.

- Herramienta Nikto.

Opción -h <host>: Esta opción permite a Nikto escanear un host para buscar vulnerabilidades de configuración.

Opción -p <puerto>: Es el número del puerto por donde corre el servicio.

Paso 2.- Verificación de vulnerabilidades por configuración encontradas.

1. Verificar todas las vulnerabilidades encontradas durante la fase de búsqueda de exploits[39] con el objetivo de determinar falsos positivos y falsos negativos.

---

[39] Exploits: Secuencias de comandos con el fin de causar un error.

Para la verificación de las vulnerabilidades encontradas se deben programar escaneos, documentando el proceso realizado con diferentes herramientas que permita determinar que vulnerabilidades encontradas son falsos positivos y falsos negativos.

2. **Verificar todos los positivos (Se debe tener en cuenta el contrato firmado con la organización objetivo en el caso de estar intentando penetrar o si se puede llegar a provocar un ataque de denegación de servicio).**

Si se establece en el contrato entre el cliente y el analista el permiso de realizar pruebas de intrusión, se debe programar pruebas de intrusión que permita determinar cuál es el riesgo latente ante cada vulnerabilidad encontrada.

Paso 3.- Resultados esperados y su análisis.

Los resultados esperados al realizar la detección de vulnerabilidades por configuración es detallada a continuación:

- Tipo de aplicación o servicio por vulnerabilidad: El analista de seguridad debe presentar un listado de las vulnerabilidades por configuración encontradas asociadas a la aplicación de banca electrónica analizada.

- Listado de vulnerabilidades por configuración de sistema operativo: El analista de seguridad debe presentar un listado de las vulnerabilidades encontradas en los sistemas operativos de los servidores de la intranet de banca electrónica analizada.

- Listado de vulnerabilidades por configuración de servicio: El analista de seguridad debe presentar un listado de las vulnerabilidades encontradas en los servicios activos de los servidores de la intranet de banca electrónica analizada. [40]

---

[40] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

La plantilla de los informes requeridos en este procedimiento formal de ethical hacking para vulnerabilidades por configuración se encuentra detallado en el ***Anexo 2: 2.2 Rastreo de vulnerabilidades por configuración***.

2.4 PRUEBAS DE INTRUSIÓN

En la presente sección del proyecto se realizarán las pruebas de intrusión para analizar las vulnerabilidades de la aplicación web de banca electrónica. Estas pruebas nos permiten cumplir con el análisis del modelo de seguridad en el canal de comunicación (Internet) y la intranet del banco.

Las pruebas de intrusión se encuentran basadas en las tareas descritas en el módulo de: testeo de aplicaciones de Internet, de la sección de seguridad en las tecnologías de Internet del manual OSSTMM. Estas pruebas comprenden la descripción teórica, la herramienta recomendada y el comando o modulo de cada herramienta que permite cumplir cada tarea, finalmente se presenta una lista de resultados esperados al finalizar cada prueba.

2.4.1 PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA EL TESTEO DE AUTENTICACIÓN.

Paso 1.- Autenticación

La autenticación en los sistemas trata de verificar la identidad de los usuarios, por medio del envió de credenciales de usuarios validas. Las principales tareas descritas en el manual OSSTMM para la autenticación son detalladas a continuación: [41]

1. **Buscar las posibles combinaciones de contraseñas por fuerza bruta en las aplicaciones.**

---

[41] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

Un ataque por fuerza bruta consiste en realizar procesos de prueba y error probando múltiples claves con diferentes combinaciones de caracteres, por tal razón este método es más lento en comparación cuando se utilizan diccionarios de palabras. Los diccionarios de palabras son un conjunto de contraseñas donde se encuentran las contraseñas más utilizadas por los usuarios.

En esta tarea se deben determinar contraseñas por fuerza bruta para lo cual se pueden considerar las siguientes herramientas:

- Herramienta Brutus

La opción “Brute Force (Fuerza Bruta)” permite encontrar contraseñas sin cargar ningún archivo con nombres de usuarios o contraseñas, aun cuando tarda más tiempo es la manera más efectiva de buscar contraseñas. Al analizar un servidor web se debe especificar la URL[42]. En la herramienta se especifica en el objetivo del ataque, la URL del sistema web de banca electrónica, el puerto y tipo de protocolo utilizado, finalmente se selecciona Iniciar, esta configuración se muestra en la Figura 2-13.

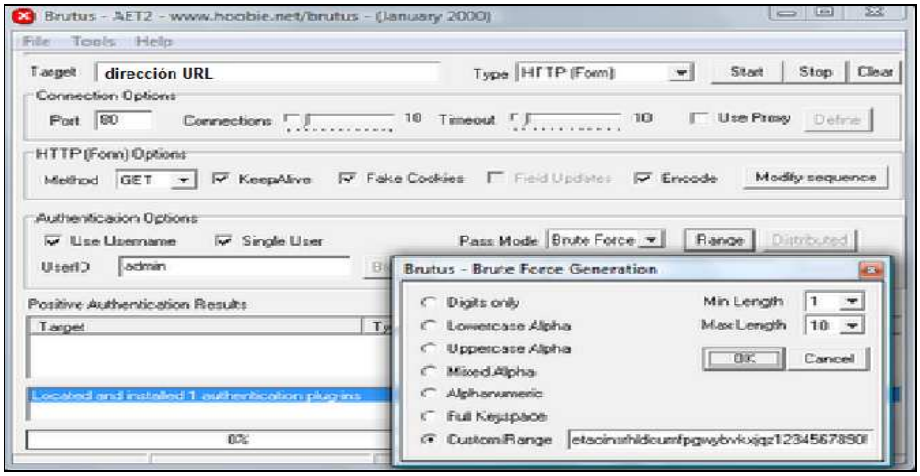

Figura 2- 13 Configuración de Brutus

---

[42] URL: Localizador de Recursos Uniforme.

- Herramienta Cain & Abel

La opción “Brute Force Attack (Ataque por Fuerza Bruta)”, prueba todas las combinaciones posibles de un cadena de caracteres predefinidos, cargados en un cuadro de dialogo de la interfaz de la herramienta Figura 2-14. El cuadro de dialogo permite que el analista de seguridad utilice un conjunto de contraseñas predefinidas o uno personalizado.

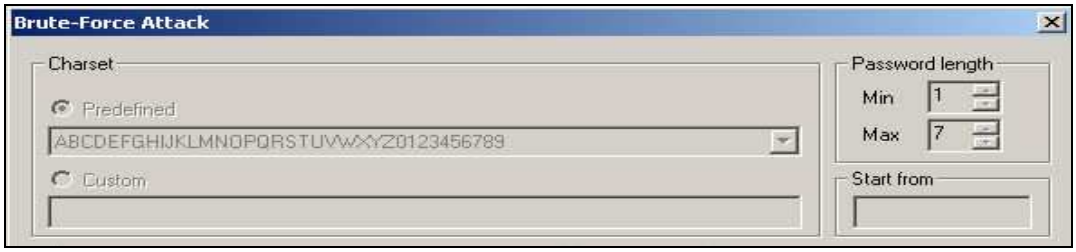

Figura 2- 14 Cuadro de Dialogo Cain & Abel

El ataque se ejecuta desde la pestaña cracker de la herramienta → seleccionando nombre de usuario → clic derecho → seleccionar Brute Force Atacck.

2. Buscar credenciales de cuentas válidas por fuerza bruta.

Los hackers intentan determinar credenciales validas de las cuentas de usuario que les permita autenticarse con dichas credenciales para cometer un delito informático. En esta tarea se deben determinar usuarios por fuerza bruta para lo cual se pueden considerar las siguientes herramientas:

- Herramienta BackTrack

BackTrack utiliza el phising para obtener credenciales válidas, para lo que se debe ejecutar: live cd, luego nos dirigimos a la barra de tareas → Aplicaciones → BackTrack → Herramientas de Ingeniería Social. De las opciones presentadas en el terminal seleccionamos Ataque de vectores website y finalmente Método de Ataque de credenciales.

BackTrack permite a un hacker clonar un sitio para que el usuario sea engañado y digite sus credenciales y luego sea dirigido al sitio original.

3. Saltarse el sistema de autenticación con una validación cambiada.

Las vulnerabilidades presentes en un sistema de autenticación insuficiente de una aplicación web permiten que los atacantes puedan acceder a las funciones proporcionadas por la aplicación sin encontrarse autenticados correctamente. Las vulnerabilidades presentes en el sistema de autenticación de una aplicación se deben a fallos en la etapa de desarrollo, convirtiéndolos en un problema de seguridad.

Las aplicaciones web de la banca requieren autenticación por parte del usuario para poder acceder a su información privada o para realizar tareas dentro de la misma, pero no todos los métodos de autenticación son seguros, ya que es posible saltarse los controles de autenticación modificando las peticiones y engañando a la aplicación para que crea que el usuario ya se autenticó.

En esta tarea se deben realizar pruebas de caja negra[43]. Para saltarse la autenticación se pueden considerar las siguientes pruebas: [44]

a) Petición directa de páginas

En una aplicación web que implementa el control de acceso solo por una página de registro, este control puede ser omitido en algunas ocasiones realizando una petición directa de una página diferente a la de login.

---

[43] Pruebas de caja negra.- Las pruebas de caja negra son las aplicables a la interfaz del usuario. Fuente: (http://www.slideshare.net/cliceduca/pruebas-de-software-2420588).

[44] http://es.scribd.com/doc/45203785/37/SALTARSE-EL-SISTEMA-DE-AUTENTICACION, Abril, 2012.

Ejemplo:

Los sistemas web de banca electrónica utilizan la autenticación por credenciales de usuario y contraseña, utilizando páginas de login en las cuales los usuarios se autentican, si los sistemas no implementan controles de autenticación los usuarios podrían saltarse la autenticación ingresando directamente la URL de páginas web de transferencias, o de consultas.

b) Modificación de parámetros

Si una aplicación verifica la autenticación del usuario por parámetros de valores fijos, estos pueden ser modificados por el usuario manualmente, indicándole a la aplicación que el usuario ya se autentico sin haber realizado esta tarea.

Ejemplo:

En algunos sistemas web se envían números de autenticación añadidos a la URL de las páginas web similares a “auth=cadena de caracteres/números”, el analista de seguridad puede saltarse la autenticación haciendo una petición directa a un pagina web del sistema distinta al login añadiendo el número de autenticación a la URL.

c) Inyección SQL

Una inyección SQL se basa en alterar la información existente en las bases de datos, ya que los datos son el principal recurso de una empresa, la manipulación de los mismos permitiría en este caso autenticarse mediante el ingreso de usuarios y contraseñas ingresados directamente a la base de datos. La inyección de código SQL se detalla en el procedimiento de ethical hacking para el testeo de manipulación de información de entrada y salida, donde se detallarán las herramientas recomendadas a utilizar.

4. Saltarse el sistema de autenticación reproduciendo información de la autenticación.

Los sistemas que no cierran adecuadamente las sesiones de usuario o permiten que un mismo usuario se autentique más de una vez al mismo tiempo, ocasionan vulnerabilidades reproduciendo la información de autenticación, ocasionada cuando el analista de seguridad puede recuperar los datos de credenciales validas de un usuario que se encuentra interactuando directamente con la aplicación y mediante este robo de información, el puede suplantar la identidad de un usuario válido.

- Herramienta Hunt

Hunt es un sniffer de paquetes que permite saltarse la autenticación reproduciendo la información mediante la opción intruso de conexiones con la que cuenta, la herramienta hunt es nativa de GNU/Linux y sirve para rastrear las conexiones de usuarios activos mediante el comando –v (pids impresión de hilos creados) –i (interfaz de red).

5. **Determinar las limitaciones de control de acceso en las aplicaciones, permisos de acceso, duración de las sesiones, tiempo inactivo.**

El analista de seguridad debe analizar las limitaciones de los sistemas de autenticación y controles de acceso de los usuarios, permisos de acceso, duración de las sesiones, tiempo inactivo que le permita determinar las vulnerabilidades y riesgos asociados.

- Herramienta Saint

La herramienta Saint permite a través de la opción Soporte para Firewalls, seleccionar escanear, los resultados presentados por la herramienta pueden ser presentados en reportes, el reporte contendrá gráficos de barras con los riesgos encontrados.

Paso 2.- Resultados esperados.

Los resultados esperados al realizar la detección de vulnerabilidades de autenticación son detallados a continuación:

- Lista de las vulnerabilidades de la autenticación de la aplicación de banca electrónica: El analista de seguridad debe presentar un listado de las vulnerabilidades de autenticación encontradas y asociarlas a la aplicación de banca electrónica analizada.

- Lista de contraseñas encriptadas o no encriptadas: El analista de seguridad debe presentar un listado de las contraseñas encriptadas y no encriptadas capturadas en la realización de las tareas de este procedimiento.

- Lista de cuentas, con usuario o contraseña de sistema: El analista de seguridad debe presentar un listado de credenciales validas capturadas en el análisis.

- Lista de archivos o documentos vulnerables a ataques de descifrado de contraseñas: El analista de seguridad debe presentar un listado de los archivos, documentos o sistemas de almacenamiento de credenciales de usuario en la intranet bancaria. [45]

La plantilla de los informes requeridos en este procedimiento formal de ethical hacking para el testeo de autenticación se encuentra detallado en el ***Anexo 3: 3.1 Autenticación***.

2.4.2 PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA EL TESTEO DE ADMINISTRACIÓN DE SESIONES.

Paso 1.- Administración de Sesiones

La administración de sesiones en aplicaciones accesibles por Internet, es un aspecto crítico para garantizar la seguridad, ya que el protocolo utilizado en Internet no maneja estados. Las principales tareas descritas en el manual OSSTMM para la administración de sesiones son detalladas a continuación: [46]

---

[45], [46] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

1. **Determinar la Información de Administración de Sesiones, número de sesiones concurrentes, autenticaciones basadas en IP, autenticación basada en roles, uso de Cookies, ID de sesión dentro de las secuencias de codificación de la URL, ID de sesión en campos HTML ocultos.**

Debido a la naturaleza del protocolo HTTP de no mantener estados, las organizaciones han recurrido a mantener el estado de los usuarios utilizando varios métodos que son:

a) ID de sesión por usuario: Los IDs de sesión permiten que el servidor identifique al usuario en la aplicación, mientras que al cliente le permite seguir navegando por la aplicación sin que tenga la necesidad de volver a autenticarse.

b) Autenticaciones basadas en IP: La autenticación basada en IP implica que el cliente debe enviar al servidor su dirección IP y este autenticarla para tener acceso a las funcionalidades de la aplicación web.

c) Autenticación basada en roles: La autenticación por roles es un método de programación muy utilizado ya que permite determinar el acceso a las funciones de la aplicación por usuario dependiendo de su perfil.

d) Uso Cookies: La utilización de las cookies para mantener los estados por usuario, significa que las cookies almacenaran información sensible tales como credenciales del usuario validas que serán enviadas al servidor al realizar una petición.

En la Figura 2-15 se muestra el método utilizado por los analistas de seguridad para obtener el ID de sesión, realizando una petición mediante el método GET desde un cliente al servidor.

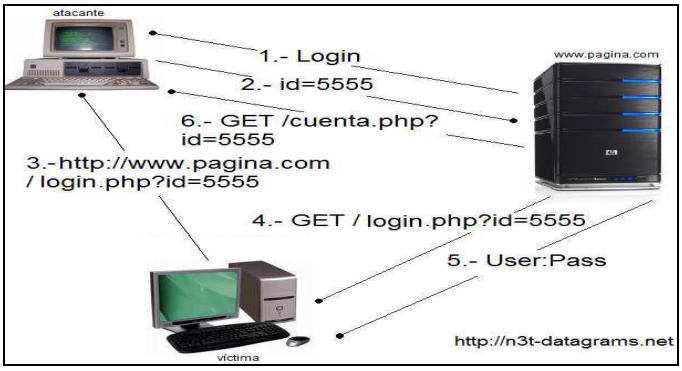

Figura 2- 15 ID de sesión capturada por un atacante [47]

- Herramienta WebScarab

Las opciones de esta herramienta que permiten cumplir con la tarea 1 son:

a) La herramienta permite capturar información sobre las URL en la aplicación web, mediante la opción Spider, con el objetivo de poder editarla y reenviarla. Mediante esta opción se puede analizar las secuencias de ID de sesión dentro de una URL.

b) La herramienta permite capturar parámetros de las cookies en la URL que se envían como IDs de sesión mediante la opción análisis de identificadores de sesión. Que le permitirá al analista de seguridad analizar las secuencias de IDs de sesión y determinar la estructura de la misma.

c) La herramienta le permite al analista de seguridad revelar campos ocultos para modificarlos antes de ser enviados al servidor.

d) Adivinar la secuencia y formato de los IDs de sesión: La herramienta permite en la opción de análisis de IDs de sesión determinar el grado de aleatoriedad y la predictibilidad de los mismos. Para la predicción del ID de sesión un analista de seguridad debe determinar los patrones de generación de IDs, que pueda utilizar el servidor.

---

[47] Inteco - Cert, Gestión de sesiones web: Ataque y medidas de seguridad, Marzo, 2012.

2. **Determinar si el ID de sesión está formada con información de direcciones IP; mirar si la misma información de sesión puede ser recuperada y reutilizada en otra máquina.**

Las vulnerabilidades existentes al reutilizar un ID de sesión valido de un usuario, ocurre cuando no se realizo el cierre de sesión de manera correcta, puede darse porque el usuario olvido cerrar sesión o el servidor reutiliza los ID de sesión. El analista de seguridad debe capturar el tráfico generado en la red.

- Herramienta WebScarab

Esta herramienta mediante la opción Proxy permite observar el tráfico de los protocolos HTTP y HTTPS, utilizando una conexión SSL entre la herramienta y el navegador. La herramienta mediante la utilización de plugins externos brinda al analista de seguridad la posibilidad de controlar las peticiones y respuestas que pasen por el proxy.

La opción Proxy de la herramienta permite interceptar solicitudes get, post, head, put, entre otras generadas entre el cliente y servidor Figura 2-16. En la pestaña Proxy de la herramienta se debe seleccionar Manual Edit → seleccionar la opción “Intercept request” y elegir el método de solicitud a interceptar.

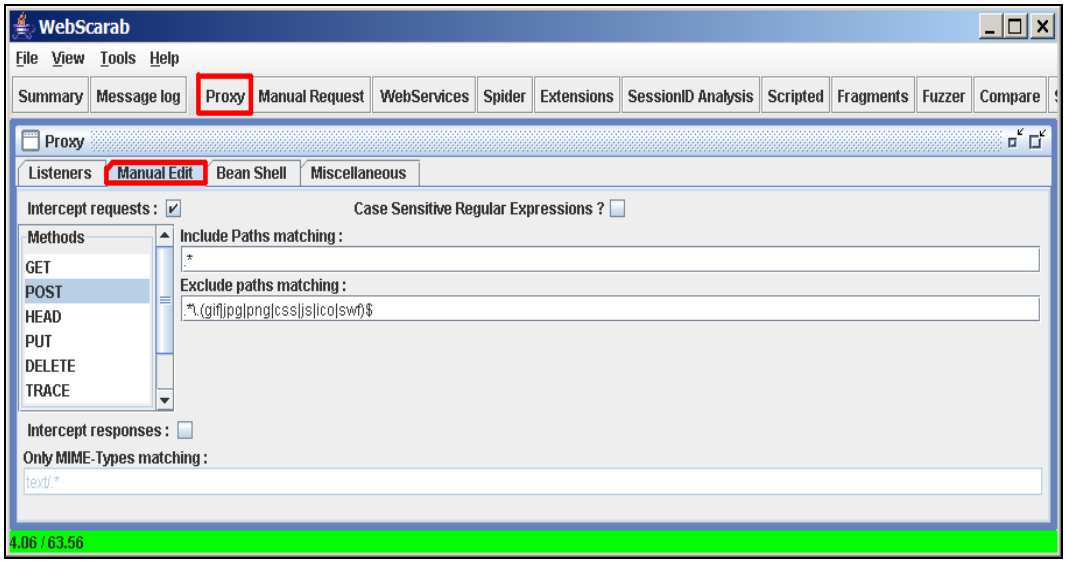

Figura 2- 16 Proxy de WebScarab [48]

3. Determinar las limitaciones de mantenimiento de sesión - uso del ancho de banda, limitaciones de bajadas/subidas de archivos, limitaciones en transacciones.

Las aplicaciones web utilizan identificadores de sesión que permiten dar un servicio personalizado por usuario, controlar el acceso, y mostrar elementos a los cuales el usuario tenga acceso. El protocolo HTTP es un protocolo sin estados, por lo que es incapaz de manejar estados de los usuarios que han hecho previamente una solicitud, para resolver este problema las aplicaciones web implementan administradores de sesiones. Las aplicaciones web o servidores envían un identificador de sesión por usuario al cliente y el cliente lo envía nuevamente al servidor para que este verifique la identidad del cliente y le proporciona los privilegios al que tiene acceso el usuario.

Las limitaciones de mantenimiento de sesión pueden causar las siguientes vulnerabilidades descritas a continuación:

---

[48] https://www.owasp.org/index.php/WebScarab_Getting_Started, Noviembre, 2012.

a) Ataques por fijación de sesión que permite que el analista de seguridad pueda crear una petición de ID de sesión valida.

b) Luego de la autenticación de un usuario se mantiene el ID de sesión.

c) Reutilizar los IDs de sesión.

d) Ataques XSS (Secuencias de comandos en sitios cruzados): Descrito en la sección 2.2.2.3

La seguridad de una aplicación web bancaria es importante por el tipo de información que se intercambia entre el cliente y servidor, y esto afecta al rendimiento de la aplicación ya que un nivel alto de seguridad en la web realiza un gran intercambio de datos con sistemas de validación y mantenimiento de sesiones, ya que por cada petición realizada por el cliente o respuesta del servidor se añadirá al paquete de datos los identificadores de sesión.

Para realizar pruebas de limitaciones de ancho de banda, subida o bajada de archivos y de transacciones, se recomienda realizar pruebas de carga y stress en la aplicación simulando usuarios conectados, y peticiones al servidor, que permita realizar un análisis definiendo el tiempo de respuesta aceptable para consultas, subida/bajada de archivos, actualizaciones o inserciones.

Las herramientas recomendadas para realizar pruebas de carga y stress de una aplicación web se detallan a continuación:

- Herramienta Apache JMeter

Esta herramienta mide el rendimiento de una aplicación simulando varios escenarios de pruebas en el servidor. Para realizar el análisis se debe crear un plan de pruebas, que está definido por varios elementos tales como: número de usuarios simulados, peticiones al servidor, verificación de datos enviados desde el servidor, entre otros. Para ejecutar el análisis se debe seleccionar el menú Lanzar finalmente clic en Arrancar el análisis. En la figura 2-17 se muestra como añadir elementos a un plan de pruebas creado.

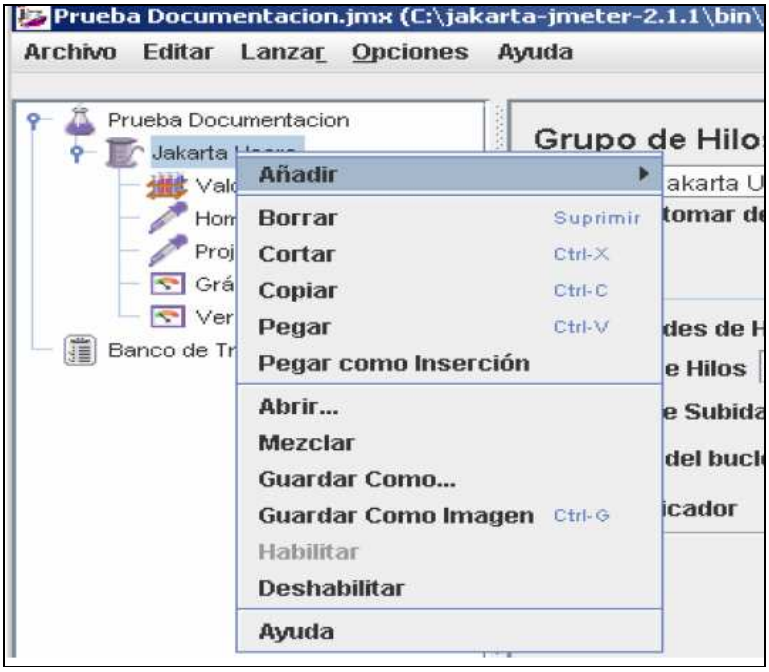

Figura 2- 17 Plan de pruebas en Apache JMeter[49]

- Herramienta Mercury LoadRunner

Mercury LoadRunner permite realizar pruebas de rendimiento como pruebas de carga y estrés a la aplicación web. Para realizar el análisis se debe seleccionar la aplicación y la opción Iniciar. La herramienta permite determinar y eliminar cuellos de botella.

4. Reunir bastante información con URL's exactas e instrucciones exactas.

En el análisis de la información recolectada de la aplicación web, se puede obtener URLs exactas, instrucciones get, post entre otras realizadas entre el cliente y servidor. Esta información le permitirá al analista de seguridad determinar la información manejada en capa de aplicación y canal de comunicación.

---

[49] EJIE S.A., Manual de usuario Apache JMeter.

- Herramienta WebScarab

La herramienta permite identificar y obtener URLs exactas de la aplicación analizada y obtener su contenido mediante la opción Spider (Araña) con la que cuenta. Para capturar las instrucciones get y post esta herramienta utiliza la opción proxy descrita en la tarea 2 del presente procedimiento.

5. Reunir información sensible a partir de ataques "Hombre en el Medio".

Un ataque hombre en el medio es aquel en el cual un hacker puede leer, interceptar o modificar las peticiones hechas entre el cliente y servidor. En la banca electrónica este ataque ocurre cuando el cliente establece un canal de comunicación (encriptado) con el servidor web del banco, mediante la instalación de código malicioso en el computador personal del cliente, se puede capturar información sensible tal como usuario, contraseña. O el analista de seguridad puede interceptar las peticiones realizadas entre el computador cliente y el servidor.

En la figura 2-18, se ilustra el esquema de un ataque "Hombre en el Medio" que utiliza un hacker para robar información a los usuarios.

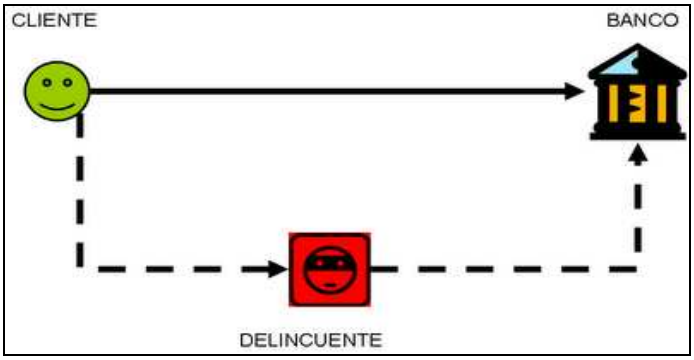

Figura 2- 18 Esquema ataque Hombre en el Medio [50]

[50] http://seguridadbe.blogspot.com/2010/10/el-problema-omar-herrera.html, Abril, 2012.

- Herramienta Wireshark.

Esta herramienta permite capturar el tráfico HTTP y HTTPS, mediante la opción capturar e iniciar, que permitirá reunir la siguiente información:

a) El puerto por el que se conecta la víctima.

b) El protocolo utilizado.

c) Paquetes intercambiados entre el servidor y cliente.

Para que Wireshark permita capturar el tráfico HTTP de una red se debe utilizar el filtro basado en red conocido como “http contains página URL”. Wireshark presentará primero la información de los paquetes capturados, segundo al seleccionar un paquete presentará información sobre el protocolo y los campos correspondientes del paquete. Finalmente se presentará el contenido del paquete en formato hexadecimal.

6. Inyectar falsa información con técnicas de Hacking.

Inyección de información falsa puede realizarse mediante ataque “Hombre en el Medio” descritos en la tarea 5 del presente procedimiento.

- Herramienta WebScarab

La herramienta permite mediante la opción Intercepción Manual modificar las peticiones HTTP y HTTPS que intercambia el cliente y el servidor. Descrito en el punto 2 del presente procedimiento como utilizar esta herramienta.

- Herramienta Ettercap

La herramienta permite establecer un ataque “Hombre en el Medio” y para ello el analista de seguridad debe realizar los siguientes pasos:

a) Crear un filtro para indicar a la herramienta la información que se quiere modificar. Para crear el filtro se puede acceder a la página de ayuda “Etterfilter”.

b) Compilar el filtro mediante la opción de consola –o.

c) Realizar el ataque “Hombre en el Medio” para ello se utilizará el envenenamiento ARP. En la herramienta mediante su interfaz gráfica se puede seleccionar la pestaña Mitm y seleccionamos ARP Poisoning. Realizando estos pasos el ataque hombre en el medio se encuentra funcionando para detenerlo se selecciona la opción Mitm →Stop Mitm Attack's.

En la Figura 2- 19 se muestra el inicio de un ataque de envenenamiento ARP.

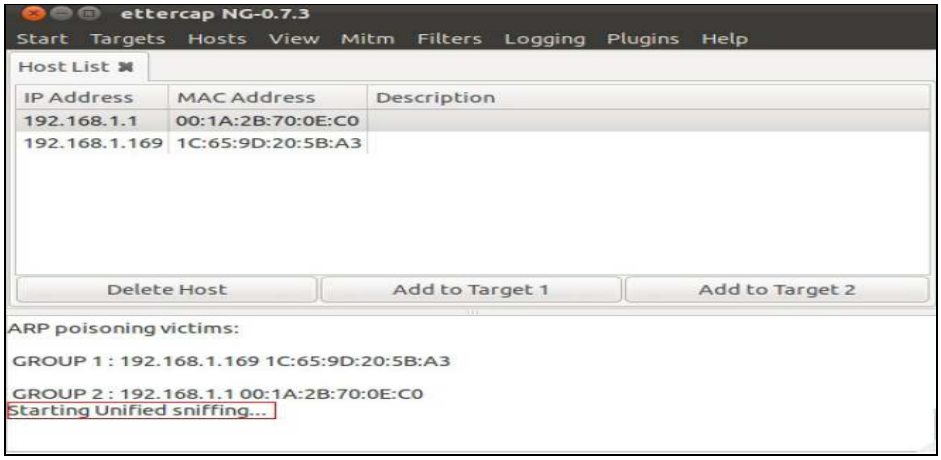

Figura 2- 19 Ataque envenenamiento ARP[51]

7. Reproducir la información reunida para engañar a las aplicaciones.

Para esta tarea se puede utilizar el método de fijación de sesión que consiste en generar un ID de sesión válido en el servidor web para que este sea asignado a un usuario en el momento que el mismo trate de autenticarse, con el objetivo de que el analista de seguridad pueda acceder a las funciones del usuario con un ID de sesión válido.

Los pasos para realizar un ataque de fijación de sesión son:

---

[51] SUBIRES, Para David, Seguridad y Alta Disponibilidad, 2011/2012.

a) El analista de seguridad realiza un petición HTTP o HTTPS para que el servidor le devuelva un ID de sesión válido.

b) El analista de seguridad le envía a la victima una petición que incluye el ID de sesión.

c) La víctima se autentica mediante la petición que le hizo el analista de seguridad.

d) Finalmente, el analista de seguridad tiene un identificador de sesión valido asociado a su víctima.

Herramientas para obtener un ID de sesión valido.

- Herramienta CookieDigger

Permite capturar las cookies generadas para el análisis manual del atacante y poder obtener los ID de sesión. La herramienta analiza y recopila la información generada por aplicaciones web a varios usuarios.

- Herramienta WebScarab

La herramienta mediante la opción “revelar campos ocultos” permite interceptar y modificar la información enviada. Esta herramienta cambia todos los campos ocultos en un formulario HTML en campos de texto haciéndolos visibles y editables.

Paso 2.- Resultados esperados.

Los resultados esperados al realizar el análisis sobre el manejo de sesiones protocolo HTTP y HTTPS son detallados a continuación:

- Lista de las vulnerabilidades de gestión de sesiones: El analista de seguridad debe presentar un listado de las vulnerabilidades de gestión de sesiones encontradas asociadas a la aplicación de banca electrónica analizada.

- Lista de ataques permitidos en gestión de sesiones: El analista debe presentar un listado con los ataques exitosos, fallidos y los resultados esperados al final de estos ataques. [52]

La plantilla de los informes requeridos en este procedimiento formal de ethical hacking para el testeo de administración de sesiones se encuentra detallado en el ***Anexo 3: 3.2 Administración de Sesiones***.

2.4.3 PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA EL TESTEO DE MANIPULACIÓN DE LA INFORMACIÓN DE ENTRADA Y SALIDA.

Paso 1.- Manipulación de la información de entrada.

La manipulación de la información de entrada permite que el analista de seguridad pueda encontrar vulnerabilidades en la aplicación y explotarlas. Las principales tareas descritas en el manual OSSTMM para manipulación de datos de entrada son detalladas a continuación: [53]

1. **Encontrar las limitaciones de las variables definidas y de los protocolos - longitud de datos, tipo de datos, formato de la estructura.**

Las vulnerabilidades existentes en variables definidas en la aplicación, información en campos ocultos, la longitud de los datos y el tipo de los mismos pueden llevar a problemas en la aplicación tales como desbordamiento de memoria, manipular la información contenida en variables, entre otras.

Las vulnerabilidades creadas en la etapa de desarrollo del sistema, puede ser analizado capturando el tráfico generado por el protocolo HTTP o HTTPS.

---

[52], [53] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

- Herramienta Wireshark

Wireshark permite capturar los paquetes que se transmiten en una red, para capturar paquetes se deben contar con permisos de administrador y conocer la interfaz de red en la que se desea capturar paquetes. Para que Wireshark permita capturar el tráfico HTTP de una red se debe utilizar el filtro basado en red conocido como “http contains página URL”. Wireshark presentará primero la información de los paquetes capturados, segundo al seleccionar un paquete presentará información sobre el protocolo y los campos correspondientes del paquete. Finalmente se presentará el contenido del paquete en formato hexadecimal.

2. Usar cadenas largas de caracteres para encontrar vulnerabilidades de desbordamientos de memoria en las aplicaciones.

El desbordamiento de memoria ocurre cuando un analista de seguridad puede introducir una importante cantidad de bytes en un área de la aplicación, si el área de la aplicación no puede contener los bytes introducidos la memoria se desborda y es cuando el atacante puede sobre escribir otras áreas de la memoria.

Los principales tipos de vulnerabilidades por desbordamiento de memoria son:

a) Desbordamiento de Memoria Heap (Heap overflow): La memoria heap se compone de etiquetas de limitación que contiene información sobre gestión de memoria que es utilizado para almacenar datos o variables globales asignados dinámicamente. Un desbordamiento de memoria heap sobreescribe estas etiquetas.

b) Desbordamiento de Pila (stack overflow): El desbordamiento de pila ocurre cuando se copia datos de tamaño variable sobre búfers de tamaño fijo, sin realizar una comprobación de tamaño. Las consecuencias de este tipo de desbordamiento permitirían ejecución de código o denegación del servicio.

En esta tarea se requiere suministrar cadenas largas de caracteres para desbordar la memoria, el tipo de vulnerabilidad utilizado en esta tarea es el desbordamiento de memoria heap.

Desbordamiento de Memoria Heap

El heap de etiquetas contiene información de memoria, si un analista de seguridad desborda la memoria heap, este puede por ejemplo sobre escribir la dirección de memoria que sirve para saltarse el control de acceso.

Las pruebas de caja negra para el desbordamiento de memoria heap se basa en usar grandes cadenas en comparación con la entrada especificada.

Las herramientas que permiten realizar un desbordamiento de memoria son conocidas como “remote buffer overflow exploit”, estas herramientas se ejecutan en la computadora del analista de seguridad permitiendo desbordar la memoria de la aplicación que se ejecuta en la computadora de la víctima. Se recomienda utilizar la siguiente herramienta:

- Herramienta SPIKE Proxy.

La herramienta permite el análisis automático de desbordamiento de memoria al iniciar el análisis.

3. Inyectar comandos SQL en las entradas de cadenas de caracteres de aplicaciones web basadas en bases de datos.

Las vulnerabilidades existentes al permitir inyecciones SQL, tratan de ejecutar una determinada consulta, inserciones o actualizaciones directamente a la base de datos de la aplicación web, sin que se realice la validación adecuada por la capa de la aplicación.

- Herramienta Nessus

Nessus posee la opción “Web Application Tests Settings” que permite buscar vulnerabilidades de una aplicación entre ellas la inyección de SQL. Para realizar este

análisis se debe activar esta opción presentada “Web Application Tests”. Este análisis se realiza probando los argumentos de Interfaces de puertas abiertas de enlaces comunes (CGI) indicando que fue afectado por una vulnerabilidad de inyección SQL. En la Figura 2-20 se muestra la configuración de para la ejecución de este análisis.

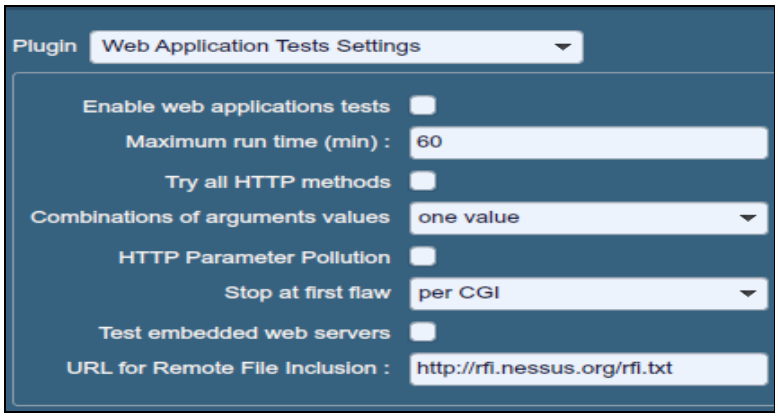

Figura 2- 20 Configuración Nessus para pruebas en aplicaciones web [54]

- Herramienta Web Burp Suite

Burp Intruder permite lanzar ataques que permiten inyecciones SQL, para el ataque se requiere configurar en la herramienta la dirección IP del servidor a ser atacado. Este ataque se ejecuta seleccionando en la herramienta la pestaña “Intruder”, en la cual se identificará la IP del servidor de banca electrónica y se seleccionará el ataque realizado Figura 2-21.

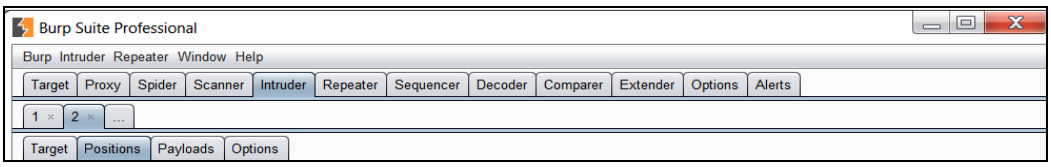

Figura 2- 21 Brup Intruder Suit [55]

---

[54] Tenable Network Security, Inc, Guía del usuario de Nessus 5.0, Abril, 2012.

[55] http://portswigger.net/burp/intruder.html, Noviembre, 2012.

4. **Examinar vulnerabilidades "Cross-Site Scripting" en las aplicaciones web del sistema.**

Las vulnerabilidades Cross-Site Scripting (XSS), ocurren cuando el analista de seguridad manipula los parámetros de entrada de una aplicación para que esta no valide los datos y genere los resultados esperados por el analista de seguridad.

- Herramienta Nessus

Nessus posee la opción “Web Application Tests Settings” que permite buscar vulnerabilidades de una aplicación entre ellas Cross-Site Scripting (XSS). Para realizar este análisis se debe activar esta opción presentada Web Application Tests. Este análisis se realiza probando los argumentos de Interfaces de puertas abiertas de enlaces comunes (CGI).

- Herramienta Web Burp Suite

Burp Intruder permite lanzar ataques que permiten inyecciones XSS, para el ataque se requiere configurar en la herramienta la dirección IP del servidor a ser atacado. La configuración fue descrita en la Tarea 3 del presente procedimiento, el tipo de ataque seleccionado en la herramienta es inyección XSS.

5. **Examinar accesos a directorios/ficheros no autorizados con directorios/rutas trasversales en las entradas de cadenas de caracteres de las aplicaciones.**

La mayoría de aplicaciones web utilizan control para el acceso de los usuarios a directorios o archivos. Si un analista de seguridad tiene acceso a los archivos o directorios que normalmente no tendría acceso puede modificar los mismos, atravesar directorios para el acceso de archivos adjuntos, acceder a datos fuera de la raíz.

- Herramienta N-Stealth.

N-Stealth permite el análisis de los directorios y ficheros, mediante la opción de manejo de archivos y directorios embebidos en la aplicación.

6. Ejecutar comandos remotos a través de "Server Side Include".

La inyección de comandos SSI (Server Side Include), permite al analista de seguridad introducir pequeñas piezas de código dinámico en páginas web estáticas que después serán ejecutadas por el servidor. Las pruebas necesarias para determinar las vulnerabilidades presentes al utilizar directivas SSI, se describen a continuación:

a) Identificar si el servidor web que el analista de seguridad quiere hackear utiliza dicha directiva.

Para identificar la directiva utilizada, generalmente un servidor web usa archivos de extensión .shtml, si no se encuentra estos archivos no quiere decir necesariamente que no se utilice esta directiva.

b) Si no se pudo encontrar o identificar un archivo .shtml se utilizará la técnica utilizada para evaluar todas las vulnerabilidades existentes tales como:

   - Identificar puntos de entrada, donde se permita que un usuario envié algún tipo de entrada o archivo.
   - Analizar el contenido de las cookies.

- Herramienta Wfuzz

La herramienta Wfuzz permite al usuario realizar análisis de inyección de código mediante bibliotecas, aun cuando esta herramienta no cuenta con una biblioteca para el análisis de inyección SSI, existe una biblioteca que puede ser descargada en la siguiente dirección electrónica: http://www.iniqua.com/2010/01/27/fuzzing-server-side-includes-ssi-3/, y puede ser ejecutada esta opción con el siguiente comando:

wfuzz.py -c -z file -f wordlists/SSI.txt –hc 404,503 –html

7. Manipular el estado de las cookies (session/persistent) para modificar la lógica dentro de las aplicaciones web "server-side".

Las cookies almacenan la información de sesión por usuario, esto se da en aplicaciones que necesitan almacenar el historial de las acciones realizadas, para poder identificar al usuario. Por lo tanto, el servidor envía las cookies a la máquina cliente y este envía de vuelta al servidor.

En la siguiente Figura 2-22 se muestra como setear una cookie.

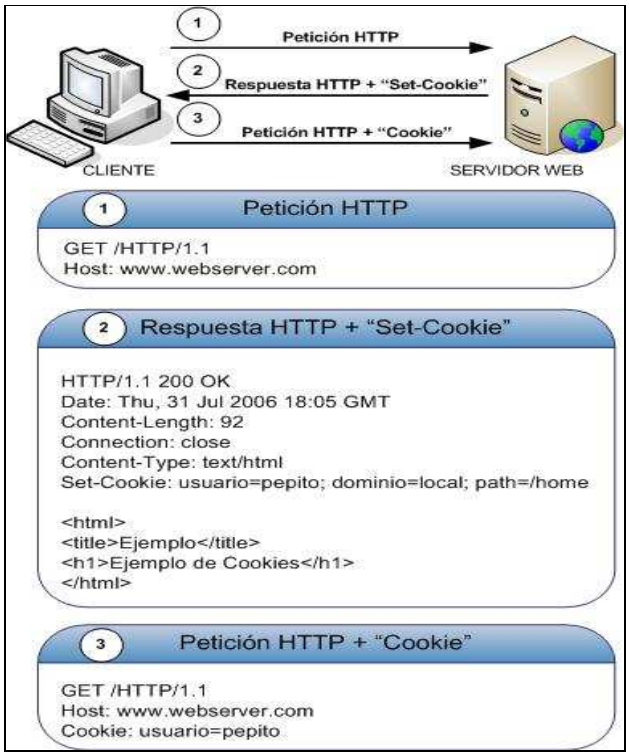

Figura 2- 22 Setear una Cookie [56]

Existen varias técnicas para robar una cookie las cuales son descritas a continuación:

---

[56] http://www.hackxcrack.es/forum/index.php?topic=6231.0 Abril, 2012.

a) Inyección de código XSS para aprovechar la vulnerabilidad existente.

b) Inyección de cookies, esta vulnerabilidad consiste en permitir que el analista de seguridad inyecte cookies en el navegador del cliente.

Las herramientas recomendadas para esta tarea se detallan a continuación:

- Herramienta CookieDigger

La herramienta CookieDigger permite recolectar las cookies mediante la utilización del sistema de login de un usuario, el analista de seguridad luego de obtener credenciales validas de un usuario ingresa a la aplicación en la interfaz de la herramienta, CookieDigger recolecta las cookies automáticamente permitiendo después realizar un análisis manual, para después manipularla.

8. Manipular los campos variables (ocultos) en los formularios HTML para modificar la lógica en las aplicaciones web "server inside".

Los programadores de aplicaciones utilizan los campos ocultos, para almacenar información sensible para la aplicación o el estatus de algún componente de la misma. La manipulación de las variables ocultas puede causar que la aplicación cambie de acuerdo al valor de dichas variables presentes en el formulario HTML.

- Herramienta WebScarab

La herramienta mediante la opción “revelar campos ocultos” permite interceptar y modificar la información enviada. Esta herramienta cambia todos los campos ocultos en un formulario HTML en campos de texto haciéndolos visibles y editables.

- Herramienta Achilles

La herramienta Achilles permite el análisis del protocolo HTTP y HTTPS, mediante la opción Navegador y Server Web, la herramienta permite interceptar todos los datos de la aplicación web incluyendo los campos de variables ocultos con la opción de modificarlos. Para poder interceptar los datos de una aplicación web se debe modificar la configuración de nuestro navegador, utilizando la configuración manual

de proxy, colocando localhost en PROXY HTTP y el número de puerto 5000 utilizado por la herramienta por defecto si se quiere cambiar este puerto se debe cambiarlo en la herramienta y en el navegador. Finalmente en la herramienta se selecciona Intercept mode ON, Intercept Client Data e Intercept Server Data (txt) Figura 2-23.

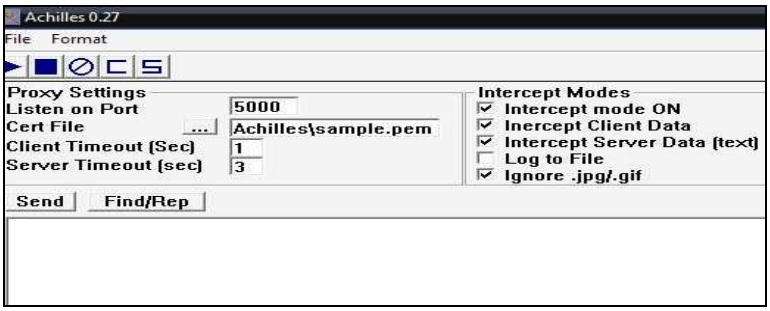

Figura 2- 23 Configuración Achilles [57]

- Herramienta Burp Proxy

Burp Proxy permite interceptar y modificar el tráfico del protocolo HTTP de las peticiones del cliente al servidor y del servidor al cliente. La herramienta mediante la opción modificación de HTML→ coincidir y reemplazar permite modificar el valor los campos ocultos de los formularios HTML. Esa herramienta es accesible desde la herramienta Burp Suite en la pestaña "Proxy" Figura 2-24.

Figura 2- 24 Burp Proxy [58]

---

[57] http://hispasystem.wordpress.com/2008/05/12/manual-de-achilles/, Noviembre, 2012.

[58] http://portswigger.net/burp/proxy.html, Noviembre, 2012.

9. **Manipular las variables "Referrer", "Host", etc. del protocolo HTTP para modificar la lógica en las aplicaciones web "server inside".**

Las variables Referrer de los formularios HTML son aquellas que permiten re direccionar al usuario a otra URL. La manipulación de estas variables por ejemplo Referrer permite al analista de seguridad direccionar al usuario a otras páginas web, en las cuales el usuario puede ser víctima de robo de información tales como credenciales de autenticación, entre otras.

Las herramientas recomendadas se encuentran detalladas en la tarea 2 y 8.

10. **Usar información de entrada ilógica/ilegal para testear las rutinas de error de la aplicación y encontrar mensajes de error/depuración que sean útiles.**

Testear la aplicación con ingreso de información ilógica para la aplicación web, con el objetivo de generar un error y mediante esto poder realizar el análisis de los mismos mediante la captura del error del protocolo HTTP.

Las herramientas recomendadas y como utilizarlas son descritas en la tarea 8. El análisis de estos códigos de error debe ser manual.

Paso 2.- Manipulación de la Información de salida

La manipulación de información de salida permite recuperar información almacenada en la computadora cliente, para obtener credenciales de usuario, claves de transferencias electrónicas e información personal del usuario. Las principales tareas descritas en el manual OSSTMM para manipulación de información de salida son detalladas a continuación: [59]

---

[59] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

1. **Recuperar información importante/comprometedora guardada en las cookies.**

Una cookie puede almacenar información sensible tal como usuarios, contraseñas, claves electrónicas, respuestas a preguntas secretas que sirva al atacante para realizar robos cibernéticos.

Las herramientas recomendadas se encuentran descritas en la tarea 7 del paso 1 de manipulación de información de entrada del presente procedimiento.

2. **Recuperar información importante/comprometedora en la caché de la aplicación cliente.**

Aun después del cierre de sesión de usuarios en las aplicaciones web, se almacena información del usuario en la caché del navegador. Este tipo de información puede ser:

a) Usuarios y contraseñas

b) Claves Electrónicas

c) Series de Tarjetas de Crédito.

d) Respuestas secretas.

e) Entre otras.

- Herramienta Historian

La herramienta permite analizar el navegador del computador y almacenar los ficheros, historial entre otros sitios visitados por la victima para después exportarlos para el análisis de los mismos. En la Figura 2-25 se muestra el análisis realizado al navegador.

Figura 2- 25 Herramienta Historian análisis de navegador [60]

Paso 3.- Resultados esperados y su análisis.

Los resultados esperados al realizar la detección de vulnerabilidades de configuración deben generar la siguiente información:

- Tipo de información recuperada: El analista debe presentar los resultados capturados y clasificarlo por tipos distinguiendo si los resultados esperados son de entrada o de salida.

- Lista de las vulnerabilidades de las aplicaciones: El analista de seguridad debe presentar un listado de las vulnerabilidades para la manipulación de información de entrada y de salida encontradas asociadas a la aplicación de banca electrónica analizada.[61]

---

[60] http://www.taringa.net/posts/info/2048400/10-Herramientas-que-usa-el-FBI-para-el-analisis-de-la-PC_.html, Noviembre, 2012.

[61] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

La plantilla de los informes requeridos en este procedimiento formal de ethical hacking para manipulación de información de entrada y salida se encuentra detallado en el ***Anexo 3: 3.3 Vulnerabilidades de manipulación de información de entrada y salida.***

2.4.4 PROCEDIMIENTO FORMAL DE ETHICAL HACKING PARA EL TESTEO DE FILTRACIÓN DE INFORMACIÓN.

Paso 1.- Filtración de información

La filtración de información permite buscar datos utilizables en campos ocultos utilizados por los programadores para enviar información como códigos. Las principales tareas descritas en el manual OSSTMM para filtrar información son detalladas a continuación: [62]

1. **Buscar información utilizable en campos ocultos de variables en formularios HTML y comentarios en los documentos HTML.**

Los programadores al desarrollar aplicaciones almacenan en los campos ocultos información sensible para la aplicación tal como, estatus de la misma, autenticación o inclusive ID de sesión. Además, los programadores comentan secciones de código donde especifican los resultados que se deben obtener luego de realizar alguna acción.

Las herramientas recomendadas para esta tarea se detallan a continuación:

---

[62] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

- Herramienta WebScarab

La herramienta descrita en el procedimiento de ethical hacking para el manejo de sesiones y manipulación de información de entrada, mediante las opciones revelar campos ocultos y fragmentos permite divisar los campos ocultos de los formularios HTML y extrae los script y comentarios respectivamente.

Paso 2.- Resultados esperados y su análisis.

Los resultados esperados al realizar la detección de vulnerabilidades de configuración deben generar la siguiente información:

- Tipo de información filtrada: El analista de seguridad debe clasificar el tipo de información capturada.
- Lista de las Vulnerabilidades de las Aplicaciones: El analista de seguridad debe presentar un listado de las vulnerabilidades de filtración de información encontradas asociadas a la aplicación de banca electrónica analizada. [63]

La plantilla de los informes requeridos en este procedimiento formal de ethical hacking para filtración de información se encuentra detallado en el ***Anexo 3: 3.4 Filtración de Información.***

2.5 ANÁLISIS DE RESULTADOS

Al finalizar el desarrollo de las pruebas descritas en este proyecto, se realizará el análisis de resultados que permitan al analista de seguridad reducir los falsos positivos, falsos negativos y errores humanos de las pruebas descritas en el presente proyecto.

---

[63] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

Primero.- Con el objetivo de reducir falsos positivos y negativos se recomienda tareas de comprobaciones de error y enrutamiento. Estas tareas se encuentran descritas en el modulo 1 del manual OSSTMM sección C seguridad en las tecnologías de Internet.

Segundo.- Para reducir errores humanos se recomienda que el personal encargado de realizar el análisis sea experto en la utilización de herramientas recomendadas en este proyecto.

2.5.1 ERRORES EN EL PROCESO.

Las pruebas realizadas examinarán el estado de un sistema dinámico de la banca en periodos de tiempo corto, ya que recolectar resultados en un periodo más amplio, no es práctico por que se obtendría una gran cantidad de información que debería ser procesada y la mayoría de esta no reflejaría datos correctos.

El analista de seguridad intenta probar el sistema de una manera distinta a la implementación del programador, poniéndola al límite, y simulando situaciones poco comunes que le permita comprobar las funciones que el sistema debería realizar. Los errores más comunes presentes en el análisis de seguridad son:

- Falsos positivos: Es cuando se detecta un error que no existe.
- Falso negativo: Es lo contrario al falso positivo, quiere decir que no se detecta errores cuando existen errores.
- Error humano: Son cometidos por la falta de experiencia del analista.

2.5.2 REDUCIR FALSOS POSITIVOS / NEGATIVOS.

Las tareas que permiten reducir falsos positivos / negativos realizan ajustes en las herramientas de análisis, tales como: comprobaciones de error y de enrutamiento. Las siguientes tareas descritas en modulo 1 sección C del manual OSSTMM permiten cumplir este objetivo:

2.5.2.1 Comprobaciones de Error

1. **Examinar la ruta a la red objetivo en busca de paquetes TCP perdidos y de paquetes UDP perdidos.**

El protocolo TCP utiliza un sistema de reconocimiento de paquetes recibidos y es establecido entre el host origen y host destino conocido como verificación extremo a extremo. La verificación extremo a extremo de TCP utiliza acuses de recibo, es decir, el host emisor mantiene un registro de cada paquete enviado mientras que el host destino mantiene un registro de cada paquete recibido. En el protocolo TCP se utiliza en número de secuencia y el número de acuse de recibo de la cabecera del paquete enviado, permite confirmar la recepción del mismo. Si un paquete TCP se ha perdido, este protocolo permite retransmitir los paquetes perdidos.

El protocolo UDP no garantiza la entrega de los paquetes enviados, ya que este protocolo es utilizado generalmente para el envió de voz. Si se requiere garantizar el envió de los paquetes se lo debe hacer en capas superiores a la capa de transporte del modelo OSI.

2. **Medir el tiempo utilizado en el recorrido TCP de los paquetes.**

El tiempo utilizado en la entrega de paquetes TCP depende de la ubicación del origen y de destino del paquete. El protocolo TCP registra el tiempo en el que se envía el paquete y el tiempo en el que se recibe el paquete, dando como resultado el tiempo de transmisión.

3. **Medir el porcentaje de paquetes aceptados y respondidos por la red objetivo.**

El porcentaje de paquetes aceptados y respondidos por la red, depende del número total de paquetes correspondientes a la información enviada. El porcentaje será calculado como:

$$\% \text{ paquetes aceptados} = \frac{\#\ \text{Paquetes aceptados} * 100}{\#\ \text{Total de los paquetes que deben ser enviados}}$$

4. **Medir la cantidad de paquetes perdidos o rechazados de conexión en la red objetivo.**

La cantidad de paquetes perdidos en la red, depende del número total de paquetes correspondientes a la información enviada. La cantidad es determinada por el número de paquetes perdidos.

Enrutamiento

1. **Examinar el camino de enrutamiento al objetivo desde los sistemas de ataque.**

El camino de enrutamiento indica el acceso de una red a otra. Este camino recibe información del protocolo de enrutamiento, el acceso puede tener varios caminos alternativos. Examinar los caminos de enrutamiento permitirá que el analista determine problemas de entrega de paquetes, fallos o pérdidas que pueden ser causadas por la mala configuración de las tablas de ruteo.

2.5.3 REDUCIR ERRORES HUMANOS.

Para reducir errores humanos se recomienda que el personal encargado del análisis de seguridad sea experto en las herramientas que van a ser utilizadas para minimizar el riesgo de este tipo de errores.

2.5.4 RESULTADOS ESPERADOS

Los resultados esperados para minimizar el riesgo de falsos positivos / negativos y errores humanos son descritos a continuación:

- Discrepancias por el ancho de banda usado en el testeo: El analista de seguridad debe presentar un listado de la utilización del ancho de banda

dentro de la intranet del banco a analizar, junto con los procesos y paquetes que consumen el ancho de banda.

- Paquetes TCP perdidos: El analista de seguridad debe contar con un listado de los paquetes TCP perdidos, el porcentaje de los paquetes perdidos y en lo posible la razón por la cual se perdieron los paquetes.
- Paquetes UDP perdidos: El analista de seguridad debe contar con un listado de los paquetes UDP perdidos, el porcentaje de los paquetes perdidos y en lo posible la razón por la cual se perdieron los paquetes.
- Problemas de enrutamiento: El analista de seguridad debe contar con el análisis desarrollado a los caminos de enrutamiento en la intranet del banco, y los problemas encontrados.[64]

---

[64] HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.

CAPÍTULO 3: CONCLUSIONES Y RECOMENDACIONES

En la presente sección se presentarán las conclusiones y recomendaciones derivadas del desarrollo de los capítulos del presente proyecto. Las conclusiones se desarrollarán en la sección 3.1 y las recomendaciones en la sección 3.2.

3.1 CONCLUSIONES.

- Las pruebas de rastreo y de intrusión planteadas en este proyecto deben ser realizadas por expertos en análisis de seguridad. Los expertos implementan procedimientos de seguridad y buenas prácticas, garantizando que los resultados obtenidos sean auténticos y confiables sobre la actividad web de los usuarios, permitiendo mitigar los riesgos descritos en la sección 2.5.1 y problemas en sistemas on-line que hagan inaccesible el sistema de banca electrónica a los usuarios. Los resultados obtenidos permiten la toma de decisiones que afecten a las políticas internas o externas de seguridad en la entidad financiera.

- El presente procedimiento de ethical hacking propone una guía para entidades de control privadas o de gobierno para medir el nivel de las vulnerabilidades de seguridad en los servicios de banca electrónica. Las pruebas de rastreo y de intrusión realizadas en los subcapítulos 2.3 y 2.4 respectivamente, establecen el análisis de vulnerabilidades por intrusión, de configuración y de la aplicación de banca electrónica del área de análisis (canal de comunicación e Intranet bancaria) del modelo de seguridad utilizado en este proyecto.

- El diseño del procedimiento formal de ethical hacking realizado en este proyecto aseguran el cumplimiento de los controles establecidos en la normativa de la Superintendencia de Bancos y Seguros sobre gestión y administración de riesgos a las entidades financieras. En la sección 2.1 de análisis de los requerimientos de la normativa se diseñó el esquema de las pruebas de intrusión y rastreo planteadas en las secciones 2.3 y 2.4 respectivamente del presente proyecto, fueron basadas en la metodología

OSSTMM sección C seguridad en las tecnologías de Internet y los módulos de la sección C fueron seleccionados de acuerdo a los requerimientos expuestos en la normativa de la Superintendencia de Bancos y Seguros.

- Los servicios de banca electrónica por Internet deben implementar técnicas de manejo de sesiones seguras. En la sección 2.4.2 del presente proyecto se analizaron técnicas de manejo de sesiones, la utilización de IDs de sesión, información almacenada en campos ocultos de la aplicación, en la caché o en cookies. El manejo de sesiones no seguras permiten al atacante infectar el computador cliente para acceder a la información del usuario de banca electrónica, porque la información del cliente es guardada en el computador del cliente y enviada al servidor de banca electrónica.

3.2 RECOMENDACIONES.

- Se recomienda tomar como referencia para el análisis de seguridad de la banca electrónica por Internet a la metodología OSSTMM Sección C seguridad en las tecnologías de Internet. OSSTMM agrupa en cada modulo de la sección C tareas que permiten determinar a los analistas de seguridad los riesgos existentes en la infraestructura tecnológica de la entidad financiera, además, de incorporar en el diseño de pruebas buenas prácticas como: ISO 17999-2000, entre otras.

- Se recomienda documentar el proceso de análisis de seguridad. El análisis de seguridad permite la toma de decisiones gerenciales para actualizar o mejorar las políticas y procedimientos internos de las entidades financieras, por lo que se requiere que las actividades realizadas durante el análisis sean documentadas, con el objetivo de sustentar los resultados obtenidos y brindar una guía de auditoría del proceso que permita ratificarlos.

- Se recomienda que la búsqueda y verificación de vulnerabilidades se debe realizar hasta el final del análisis de seguridad. Nuevas vulnerabilidades pueden ser descubiertas en cualquier periodo de tiempo durante el análisis de

seguridad, porque los atacantes buscan vulnerabilidades las cuales pueden deberse en ocasiones a limitaciones tecnológicas o fallos en el sistema, que les permita encontrar una puerta de entrada a la intranet bancaria.

- Finalmente, se recomienda capacitar a los clientes de banca electrónica y al personal de la entidad financiera, porque el eslabón más débil de un modelo de seguridad son las personas debido a que por desconocimiento, errores humanos, entre otros, no cumplen las políticas de seguridad recomendadas por la entidad financiera, causando perjuicio a la información de los clientes.

BIBLIOGRAFÍA

LIBROS Y MANUALES

1. HERZOG, Pete, OSSTMM 2.1 Manual de la Metodología Abierta de Testeo de Seguridad, Diciembre, 2000.
2. OWASP, Foundation, Guía de Pruebas OWASP, Versión2, 2007.
3. Normas Generales para la aplicación de la ley general de instituciones del sistema financiero, Libro 1, Título X, Capítulo V.

TESIS

4. GALLEGOS, Marco, Valoración del estado de seguridad de dos sistemas en red mediante el manual OSSTMM, Facultad de Ciencias de la computación, Benemérita Universidad autónoma de Puebla, México, 2003.
5. HERVALEJO, Sánchez Alberto, Auditorias de seguridad informática y la OSSTMM, Universidad Politécnica de Valencia, Julio, 2009.

DIRECCIONES ELECTRONICAS

6. INSECURE.ORG, **Las 75 herramientas de seguridad más usadas,** http://insecure.org/tools/tools-es.html, Mayo,2003.
7. ZONAGRATUITA.COM, **50 herramientas top de seguridad,** http://www.zonagratuita.com/a-cursos/utilidades/50_herramientas_top.htm, Mayo, 2012.
8. CEREAL_KILLER, **10 herramientas que usa el FBI para el análisis de la PC**, http://www.taringa.net/posts/info/2048400/10-Herramientas-que-usa-el-FBI-para-el-analisis-de-la-PC_.html, Enero, 2009.
9. PROGRAMAS DE HACK, **Herramienta brutus**, http://www.programas-hack.com/manuales/todo-sobre-brutus/, Octubre, 2007.

10. FUNDACION WIKIPEDIA, INC, **Herramienta Cain & Abel**, http://es.wikipedia.org/wiki/Ca%C3%ADn_y_Abel_(software), Abril, 2012.

11. FUNDACION WIKIPEDIA, INC, **Herramienta Backtrack**, http://es.wikipedia.org/wiki/BackTrack, Mayo, 2012.

12. DRAGONJAR, **Herramienta Saint**, http://labs.dragonjar.org/laboratorios-hacking-tecnicas-y-contramedidas-escaneo-de-vulnerabilidades-i, Agosto, 2008.

13. HACKHISPANO, **Herramienta Hunt**, http://foro.hackhispano.com/downloads.php?do=file&id=149, Enero, 2008.

14. BLACKPLOIT, **Manual hping**, http://www.blackploit.com/2009/11/manual-hping.html, Noviembre, 2009.

15. THE APACHE SOFTWARE FOUNDATION, **Apache JMeter,** http://jmeter.apache.org/, Noviembre, 2011.

16. TSOFT, **Manual Mercury LoadRunner,** http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CGEQFjAA&url=http%3A%2F%2Fwww.tsoft.com.ar%2Fpapers%2Fproductos%2FMercury%2520LoadRunner.pdf&ei=eD7ET5jdMMvPgAeqzdiqBQ&usg=AFQjCNEb5wrEkA2TzWKPDpsPPdheU4Eodg,

17. OWASP, **WebScarab Getting Started/es**, https://www.owasp.org/index.php/WebScarab_Getting_Started/es, Agosto, 2008.

18. ARENAS, José Luis, **Herramienta spike proxy**, http://cbtis194joseluisarenas.wikidot.com/spike-proxy, Mayo, 2010.

19. CARZEL.COM, **Password Sniffing con Ettercap**, http://carzel.wordpress.com/2006/11/28/password-sniffing-con-ettercap/, Noviembre, 2006.

20. PORTELA, Beatriz, **Uso de Burp Intruder para ataques de diccionario y fuerza bruta**, http://www.securitybydefault.com/2011/11/uso-de-burp-intruder-para-ataques-de.html, Noviembre, 2011.

21. BLACKPLOIT, **Wfuzz Enumeración de archives y directorios en aplicaciones web**, http://www.blackploit.com/2010/05/wfuzz-enumeracion-de-archivos-y.html, Mayo, 2010.

22. [IN] SEGURIDAD INFORMATICA**,** **cookieDigger**, http://mousehack.blogspot.com/2005/12/cookiedigger-v1.html, Diciembre, 2005.

23. PORT SWIGGER WEB SECURITY, **Burp Proxy Web**, http://portswigger.net/burp/help/proxy.html, 2010.

24. VLADIMIR, Erik, **Análisis de vulnerabilidades y auditorias de seguridad bajo demanda**, http://www.nobosti.com/spip.php?article39, Marzo, 2008.

25. INF-315, **Vulnerabilidades de los sistemas informáticos**: http://inf-tres-quince.blogspot.com/2011/04/vulnerabilidades-de-los-sistemas.html, Abril. 2011.

26. RODMEN82, **Herramienta Nikto**, http://www.linuxparatodos.net/portal/article.php?story=nikto, 2012.

27. ALEX, **Herramientas de hacking**: http://bookalexa.blogspot.com/, Junio, 2008.

28. RAMOS, David, **Experto en gestión de seguridad de la información**, http://www.slideshare.net/ramos866/seguridad-web-3112487, Febrero, 2010.

29. INTECO CERT, **Tipos de vulnerabilidades,** http://cert.inteco.es/Formacion/Amenazas/Vulnerabilidades/Tipos_Vulnerabilidades/, 2012.

30. TENABLE NETWORK SECURITY, **Documentación nessus**, http://www.tenable.com/products/nessus/documentation#spanish, 2012

31. NMAP.ORG, **Manual Nmap,** http://nmap.org/man/es/index.html, 2012.

32. SECLIST.ORG, Spike proxy, http://seclists.org/fulldisclosure/2002/Sep/638, 2012.

33. VLAN138, **Dominios de difusión y dominios de colisión** (BROADCAST), http://blogsdelagente.com/networking/2008/07/18/dominios-colision-y-dominios-difusion-broadcas/, Julio, 2008.

34. **Scanners**, http://www.vilecha.com/Autodidactas/scanners.html.

GLOSARIO

A

ACCESO REMOTO: Acceso remoto es acceder desde un computador a un recurso en otra computadora, de una red local o Internet.

ANALISTA: Analista es un profesional que crea modelos informáticos.

ARCHIVOS EJECUTABLES: Es un archivo binario, que la computadora interpreta como un programa.

APLICACIÓN WEB: Es una aplicación que los usuarios pueden acceder a través de un navegador conectándose a un servidor web.

APLICACIONES: Una aplicación es un programa informático, para permitir que un usuario o usuarios puedan realizar una tarea.

ARP: Protocolo de resolución de direcciones, es el encargado de encontrar la dirección de hardware (MAC de una computadora), referente a una dirección IP especifica.

ASCII: American Standard Code for Information Interchange, es un código de caracteres que utiliza 7 bits.

AUDITORIA: Es un examine critico que realiza a un sistema con el objetivo de mejorara su funcionamiento, seguridad, entre otros.

AUTENTICADO: Autorizado.

B

BANCA ELECTRONICA: En un sistema en línea que permite a los clientes realizar transacciones.

BASE DE DATOS: Es una conjunto de información relaciona que se encuentra estructurada.

BROADCAST: Es un dominio de difusión dentro de una red.

BUENAS PRÁCTICAS: Es un conjunto de acciones que han rendido excelentes resultados.

BÚFER: O Buffer es un espacio de memoria donde se almacenan datos.

BUGS: Son errores o defectos en software.

C

CGI: Common Gateway Interface, es una tecnología utilizada por un servidor.

CICLO DE VIDA: Describe el desarrollo de software, desde el inicio hasta el fin.

COMANDO: Es una instrucción u orden realizada a un sistema informático.

COMPAÑÍA: Es una organización, institución dedicada a una actividad comercial.

COMPUTADOR: Es una máquina que recibe y procesa datos que permite convertir dichos datos en información.

CONEXIONES: Comunicación establecida entre dos computadoras.

CONFIGURACION: Es un conjunto de variables que controlan la operación de un programa.

CONTRASEÑAS: Una clave es una forma de autenticación personalizada que utiliza información secreta para acceder a un recurso determinado.

COOKIES: Información guardada en un servidor referente a un usuario.

CORTAFUEGOS: Es un sistema diseñado para impedir el acceso desautorizado a un red o recurso.

CRIMEN CIBERNETICO: Es referente a las operaciones ilícitas realizadas por medio de Internet.

CUANTIFICABLE: Es convertir un objeto en un grupo de valores discretos.

D

DATAGRAMA: Es la estructura interna de un paquete de datos.

DESCENTRALIZAR: Traspasar funciones de centro único, a varios organismos.

DENEGACION DE SERVICIOS: Es un ataque que imposibilita el acceso a servicios o recursos dentro de una red.

DISTRIBUCIÓN: Es una distribución de software basada en el núcleo de Linux.

E

ENCRIPTADO: Es el proceso que permite volver ilegible una información considerada importante.

ETHICAL HACKING: Es una disciplina de seguridad informática que respalda las mejores prácticas de realizar la evaluación de un sistema.

ENRUTAMIENTO: Es buscar un camino dentro de todos los posibles en una red.

ESCANER: Es una aplicación que permite realizar un análisis de seguridad.

ESTANDAR: Es un tipo, modelo, o patrón que sirve como referencia.

EXPLOITS: Es un software que tiene el objetivo de causar algún error o fallo en una aplicación o sistema.

F

FIREWALL: Es un filtro que permite determinadas conexiones y transmisiones de datos.

FIREWALKING: Es una técnica que emplea el estilo de traceroute.

FTP: Es un protocolo de red para la transferencia de archivos.

G

GATEWAY: Es un dispositivo que permite conectar redes con protocolos y arquitecturas.

H

HACKERS: Es una persona que utiliza sus habilidades para invadir sistemas ajenos.

HARDWARE: Son todos los dispositivos físicos de un sistema informático.

HOST: Se refiere a las computadoras conectadas en una red.

I

INFORMACION: Es un grupo de datos ordenados.

INFRAESTRUCTURA TECNOLOGICA: Es el conjunto de todos los elementos tecnológicos que integran una organización.

INGENIERIA FORENSE: Es la investigación que se encarga de recopilar y analizar datos.

INTERNET: Es una red de comunicación interconectada que utiliza la familia de protocolos TCP/IP.

INTERFAZ: Es la conexión entre dos computadoras dando una comunicación de varios niveles.

J

JAVA: Es un lenguaje de programación orientado a objetos.

JAVASCRIPT: Es un lenguaje de programación interpretado.

L

LATEX: Es un sistema de composición de textos.

LINUX: Es el núcleo de sistema operativos basados en Unix.

M

METODO: Proceso o camino sistemático establecido para realizar una tarea o trabajo con el fin de alcanzar un objetivo predeterminado.

METODOLOGIA: Una metodología es una guía que se sigue para realizar las acciones propias de una investigación.

MODULO: Es una porción de un programa.

MULTICAST: Es un método para transferir datagramas a un grupo de receptores.

MULTIUSUARIO: Permite soportar varios usuarios.

N

NAVEGADOR: Aplicación que opera a través de Internet., que muestra y localiza sitios web.

NORMATIVA: Es el establecimiento de normas o leyes.

O

OVERFLOWS: Es un error de software conocido como desbordamiento.

P

POP3: Es un protocolo para recibir mensajes electrónicos.

PROCESO: Es un conjunto de actividades que se realizan con un fin determinado.

PROCEDIMIENTO: Es un conjunto de acciones que tienen que realizarse de la misma forma para obtener el mismo resultado.

PROXY: Es un programa que realiza la tarea de acceso a Internet.

PLUGLINS: Es una aplicación que permite añadirle nueva funcionalidad a un programa.

PRUEBAS DE SEGURIDAD: Simulan un ataque informático desde cualquier punto de acceso a la red.

PRUEBAS DE INTRUSIÓN: Permiten probar los métodos de protección del sistema.

PRUEBAS DE RASTREO: Permite identificar vulnerabilidades existentes dentro un sistema.

PUERTOS: Es un interfaz que permite la comunicación con un programa.

PROTOCOLOS DE RED: Implementa la organización y controles de los datos que van a ser transmitidos.

R

RENDIMIENTO: Se refiere al trabajo de las máquinas.

S

SCRIPT: Es un conjunto de instrucciones que son almacenados en un archivos de texto y que deben ser interpretadas de línea en línea.

SEGURIDAD DE LA INFORMACION: Son todas la medidas preventivas y correctivas que permitan proteger la información.

SERVIDOR: Es un computador dentro de una red que brinda servicios a otras computadoras denominadas clientes.

SISTEMA DE DETECCION DE INTRUSOS: Es un programa de un computador que permite identificar el acceso no autorizado al mismo.

SISTEMA OPERATIVO: Es un programa que se encarga de la gestión y control de recursos de hardware y provee servicios.

SISTEMAS: Es un conjunto de partes relacionas que interactúan entre sí para lograr un objetivo.

SNIFFER: Es un programa que registra la información enviada en una red de comunicación, o en una computadora.

SSH: Secure Shell es un protocolo que permite acceder a maquinas remotamente.

SOFTWARE: Son los datos almacenados y programas en un ordenador.

T

TELNET: Telecommunication Network es un protocolo que permite manejar remotamente una computadora.

TRACEROUTE: Es una consola que permite seguir la pista a los paquetes enviados en una red.

U

UNIX: Sistema operativo portable, multitarea y multiusuario.

V

VULNERABILIDADES: Es una debilidad en un sistema permitiendo a un atacante causar daños en el mismo.

W

WEB: Red informática mundial en un sistema distribuido basado en hipertexto, entre otros.

WEB PROXY: Es un proxy para una aplicación especifica el acceso a la web (HTTP y HTTPS).

X

XML: Extensible markup language, es un lenguaje de marcas desarrollado por World Wide Web Consortium (W3C).

ACRÓNIMOS

ACK: Acuse de recibo

ARP: Protocolo de resolución de direcciones

CGI: Interfaz de entrada común

CSRF: Falsificación de petición en sitios cruzados

DNS: Sistema de Nombres de Dominio

FTP: Protocolo de Transferencia de Archivos

HTML: Lenguaje de Marcado de Hipertexto

HTTP: Protocolo de Transferencia de Hipertexto

HTTPS: Protocolo Seguro de Transferencia de Hipertexto

ICMP: Protocolo de mensajes de control de Internet

IDS: Sistema de detección de intrusos

IGMP: Protocolo de administración de grupos de Internet.

IP: Protocolo de Internet.

ISN: Número de Secuencia Inicial

MTU: Unidad máxima de transferencia

OSSTMM: Manual de la Metodología Abierta de Testeo de Seguridad

OWASP: Proyecto de Seguridad de Aplicaciones Web

POP3: Protocolo de oficina de correo

SSH: Shell Seguro

SQL: Lenguaje de consulta estructurado

SYN: sincronizar

TCP: Protocolo de Control de transmisión

TTLs: Tiempo de vida

UDP: Protocolo de datagramas de usuario

URL: Localizador de recursos uniforme

XML: Lenguaje de marcas extensible

XSS: Secuencias de comandos en sitios cruzados

ANEXOS

Anexo 1:

Puertos de Análisis para los Servidores de Red.

1. PUERTOS SERVIDORES DE RED

Puerto	Nombre	Comentario
1	tcpmux	Multiplexador de servicios de puertos TCP
5	rje	Entrada de trabajo remota
7	echo	Servicio echo
9	discard	Servicio nulo para la evaluación de conexiones
11	systat	Servicio de estado del sistema para listar los puertos conectados
13	daytime	Envía la fecha y la hora al puerto solicitante
17	qotd	Envía la cita del día al host conectado
18	msp	Protocolo de envío de mensajes
19	chargen	Servicio de generación de caracteres; envía flujos infinitos de caracteres
20	ftp-data	Puerto de datos FTP
21	ftp	Puerto del Protocolo de transferencia de archivos (FTP); algunas veces utilizado por el Protocolo de servicio de archivos (FSP).
22	Ssh	Servicio de shell seguro (SSH)
23	telnet	El servicio Telnet
25	smtp	Protocolo simple de transferencia de correo (SMTP)

37	time	Protocolo de hora (Time Protocol)
39	Rlp	Protocolo de ubicación de recursos
42	nameserver	Servicio de nombres de Internet
43	nicname	Servicio de directorio WHOIS
49	tacacs	Terminal Access Controller Access Control System para el acceso y autenticación basado en TCP/IP
50	re-mail-ck	Protocolo de verificación de correo remoto
53	domain	Servicios de nombres de dominio (tales como BIND)
63	whois++ WHOIS++,	Servicios extendidos WHOIS
67	bootps	Servicios del Protocolo Bootstrap o de inicio (BOOTP); también usado por los servicios del protocolo de configuración dinámica de host (DHCP).
68	bootpc	Cliente bootstrap (BOOTP); también usado por el protocolo de configuración dinámica de host (DHCP)
69	Tftp	Protocolo de transferencia de archivos triviales (TFTP)
70	gopher	Búsqueda y recuperación de documentos de Internet Gopher
71	netrjs-1	Servicio de trabajos remotos

72	netrjs-2	Servicio de trabajos remotos
73	netrjs-3	Servicio de trabajos remotos
79	finger	Servicio Finger para información de contacto de usuarios
80	http	Protocolo de transferencia de hipertexto (HTTP) para los servicios del World Wide Web (WWW)
88	kerberos	Sistema de autenticación de redes Kerberos
95	supdup	Extensión del protocolo Telnet
101	hostname	Servicios de nombres de host en máquinas SRI-NIC
102/tcp	iso-tsap	Aplicaciones de redes del Entorno de desarrollo ISO (ISODE)
105	csnet-ns	Servidor de nombres de mailbox; también usado por el servidor de nombres CSO
107	rtelnet	Telnet remoto
109	pop2	Protocolo Post Office versión 2
110	pop3	Protocolo Post Office versión 3
111	sunrpc	Protocolo de Llamadas de procedimientos remotos (RPC) para la ejecución de comandos remotos, utilizado por Sistemas de archivos de red (Network Filesystem, NFS)
113	auth	Protocolos de autenticación y Ident

115	Sftp	Servicios del protocolo de transferencia de archivos seguros (SFTP)
117	uucp-path	Servicios de rutas de Unix-to-Unix Copy Protocol (UUCP)
119	nntp	Protocolo de transferencia para los grupos de noticias de red (NNTP) para el sistema de discusiones USENET
123	Ntp	Protocolo de tiempo de red (NTP)
137	netbios-ns	Servicios de nombres NETBIOS utilizados en Red Hat Enterprise Linux por Samba
138	netbios-dgm	Servicios de datagramas NETBIOS utilizados en Red Hat Enterprise Linux por Samba
139	netbios-ssn	Servicios de sesión NETBIOS utilizados en Red Hat Enterprise Linux por Samba
143	imap	Protocolo de acceso a mensajes de Internet (IMAP)
161	snmp	Protocolo simple de administración de redes (SNMP)
162	snmptrap	Traps para SNMP
163	cmip-man	Protocolo común de administración de la información (CMIP)
164	cmip-agent	Protocolo común de administración de la información (CMIP)

174	mailq	Cola de transporte de correos electrónicos MAILQ
177	xdmcp	Protocolo de control del gestor de pantallas X (XDMCP)
178	nextstep	Servidor de ventanas NeXTStep
179	bgp Border	Gateway Protocol
191	prospero	Servicios de sistemas de archivos distribuidos Prospero
194	irc	Internet Relay Chat (IRC)
199	smux	SNMP UNIX Multiplexer
201	at-rtmp	Enrutamiento AppleTalk
202	at-nbp	Enlace de nombres AppleTalk
204	at-echo	Echo AppleTalk
206	at-zis	Zona de información AppleTalk
209	Qmtp	Protocolo de transferencia rápida de correo (QMTP)
210	z39.50	Base de datos NISO Z39.50
213	Ipx	El protocolo de intercambio de paquetes entre redes (IPX), es un protocolo de datagramas usado comúnmente en ambientes Novell Netware
220	Imap3	Protocolo de acceso a mensajes de Internet versión 3
245	link	Servicio LINK / 3-DNS iQuery

347	Fatserv	Servicio de administración de cintas y archivos FATMEN
363	rsvp_tunnel	Túnel RSVP
369	rpc2portmap	Portmapper del sistema de archivos Coda
370	codaauth2	Servicios de autenticación del sistema de archivos Coda
443	https	Protocolo de transferencia de hipertexto seguro (HTTP)

Anexo 2:

Plantilla Informe Pruebas de Rastreo.

2. PLANTILLA INFORME PRUEBAS DE RASTREO

<Nombre de la Organización Analizada>	
Nombre: <Nombre del analista seguridad>.	**Fecha inicio:** <Inicio del análisis>.
Empresa: <Empresa del analista seguridad>.	**Fecha Fin:** <Fin del análisis>
Objetivos: <Objetivos del análisis de seguridad>.	
Alcance: <Alcance determinado con el cliente en la etapa de contrato del análisis de seguridad>.	

2.1 RASTREO DE VULNERABILIDADES POR INTRUSIÓN.

a) Servidores analizados

- **LISTA DE SERVIDORES**

Listar los servidores activos analizados en la intranet de banca electrónica.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Dirección IP	Nombre (s) de Dominio	Sistema Operativo

- **DATOS DEL SERVIDOR**

Detallar por cada Servidor los puertos abiertos, el protocolo utilizado, los servicios asociados y un detalle general.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Puerto	Protocolo	Servicio	Detalles de Servicio

b) Listado de direcciones IP analizados y dominios

Listar los rangos de direcciones IP analizados por cada segmento de red y detallar si se obtuvo información de la configuración incluirla en este punto.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Rangos de direcciones IP que fueron testeados y detalle de dichos rangos
Información de los dominios y su configuración

c) Listado de Puertos.

Listar los puertos por servidor activo analizado y clasificarlos por los estados abiertos, cerrados o filtrados.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

- **PUERTOS**

Protocolo	Puerto	Servidor / IP	Estado Puerto (abierto, cerrado, filtrado)

d) Servicios y Tipos de Servicios activos

Listar los servicios activos en los servidores analizados y clasificarlos por tipo de servicios Ejemplo: Correo, transferencia de archivos, etc.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Servidor / IP	Servicio	Tipo de Servicio	Nivel de Parcheado

e) Identificación de Sistemas Operativo

Listar los sistemas operativos de los servidores activos analizados, explicando la técnica utilizada y su nivel de parcheado.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Servidor / IP	Sistema Operativo	Técnica utilizada para la identificación	Nivel de Parcheado

f) Secuencias TCP

Determinar la predicción de número de secuencias TCP.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Predicción de secuencia TCP:
Números de secuencia ISN TCP:
Tiempo operacional

g) Análisis de cortafuegos

- **Identificación**

Este test permite determinar el éxito de las respuestas a los paquetes con métodos de identificación a través del cortafuegos.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Método	Resultado

- **Sigilo**

Este test determina la viabilidad de realizar un escaneo SYN sigiloso a través del cortafuegos y obtener una enumeración de servicios

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Resultado

- **Control de puerto origen**

Este test mide el uso de escaneo de puertos usando puertos de origen específicos a través del cortafuegos para enumerar los servicios.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Protocolo	Origen	Resultado
UDP	53	
UDP	161	
TCP	53	

TCP	69	

- **Fragmentos de paquetes**

Este test determina la habilidad del cortafuegos para manipular pequeños paquetes fragmentados.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	**Resultado**

- **Inundación Syn**

Este test mide la capacidad del cortafuegos de gestionar un flujo constante de paquetes SYN entrantes.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	**Resultado**

- **Bandera rst**

Este test exige respuesta del cortafuegos a paquetes enviados con la opción RST activada.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	Resultado

- **UDP**

Este test mide la capacidad de gestión de paquetes UDP estándar.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	Resultado

- **ACK**

Este test descubre la capacidad del cortafuegos para bloquear técnicas de enumeración que utilizan paquetes ACK.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	Resultado

- **FIN**

Este test descubre la capacidad del cortafuegos para bloquear técnicas de enumeración que utilizan paquetes FIN.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	Resultado

- **NULL**

Este test descubre la capacidad del cortafuegos para bloquear técnicas de enumeración que utilizan paquetes NULL.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	Resultado

- **XMAS**

Este test descubre la capacidad del cortafuegos para bloquear técnicas de enumeración que utilizan paquetes que tienen todas las opciones del protocolo habilitadas.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

IP	Resultado

h) Mapa de Red

Esquematizar el mapa de red con las convenciones para cada uno de los equipos computacionales.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

2.2 RASTREO DE VULNERABILIDADES POR CONFIGURACIÓN

a) Vulnerabilidades de configuración por servicios

Clasificar las vulnerabilidades por configuración de los servicios activos en los servidores analizados.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Servidor	IP Servidor	Servicio	Vulnerabilidad

b) Vulnerabilidades de configuración por aplicación

Clasificar las vulnerabilidades por configuración de la aplicación de banca electrónica de los servidores analizados.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Servidor	IP Servidor	Aplicación	Vulnerabilidad

c) Vulnerabilidades de configuración por sistema operativo

Clasificar las vulnerabilidades por configuración del sistema operativo de los servidores analizados.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Servidor	IP Servidor	Sistema Operativo	Vulnerabilidad

2.3 PREOCUPACIONES Y VULNERABILIDADES

Clasificar las vulnerabilidades y preocupaciones encontradas en la intranet bancaria. Esquematizar un ejemplo y proponer una solución.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Preocupación o Vulnerabilidad
Ejemplo
Solución

2.4 IDENTIFICACIÓN DE AMENAZAS

Determinar las amenazas de las vulnerabilidades 1.3 encontradas en la intranet bancaria. Esquematizar un ejemplo y proponer una solución.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Amenaza
Ejemplo
Solución

2.5 ANÁLISIS DE RIESGOS

Realizar el análisis de riesgos determinadas las amenazas 1.4 y las vulnerabilidades encontradas 1.3 en la intranet bancaria.

Riesgo = Amenaza * Vulnerabilidad

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Análisis de Riesgos

2.6 CONCLUSIONES Y RECOMENDACIONES

Conclusiones
<Conclusiones basadas en los problemas encontrados, sin juicios de valor y debidamente sustentadas>.
Recomendaciones
<Recomendaciones que permitan mejorar los inconvenientes encontrados>.

NOTA.- Adjuntar los resultados de las pruebas obtenidas que permitan sustentar la información ingresada en este informe.

_____	_____
Firma Analista de Seguridad	Firma Cliente
<Nombre>	<Nombre>
<CI>	<CI>

Anexo 3:

Plantilla Informe Pruebas de Intrusión.

3. PLANTILLA INFORME PRUEBAS DE INTRUSIÓN

<Nombre de la Organización Analizada>	
Nombre: <Nombre del analista seguridad>.	**Fecha inicio:** <Inicio del análisis>.
Empresa: <Empresa del analista seguridad>.	**Fecha Fin:** <Fin del análisis>
Objetivos: <Objetivos del análisis de seguridad>.	
Alcance: <Alcance determinado con el cliente en la etapa de contrato del análisis de seguridad>.	

3.1 AUTENTICACIÓN

a) Vulnerabilidades de autenticación

Determinar las vulnerabilidades de autenticación en la intranet bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Vulnerabilidad	**Detalle**

b) Lista de Contraseñas

Listar las contraseñas capturadas por fuerza bruta en la intranet bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Usuario	**Contraseña (Encriptada)**	**Contraseña (Desencriptada)**

c) Lista de cuentas de usuarios

Listar las cuentas de usuario recuperadas con las contraseñas capturadas por fuerza bruta en la intranet bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Cuenta	**Usuario**	**Contraseña (Encriptada)**	**Contraseña (Desencriptada)**

d) Lista archivos o documentos vulnerables

Listar los archivos vulnerables con las contraseñas capturadas en la intranet bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Archivo Protegido

Nombre de archivo	
Tipo de archivo	
Tiempo de duración del ataque	
Nombre del usuario	
Contraseña	

Servicios en línea protegido

Dirección IP	
Puerto de servicio	
Tipo de servicio	
Protocolo	
Nombres de usuario	
Contraseña	

3.2 ADMINISTRACIÓN DE SESIONES

a) Vulnerabilidades de administración de sesión

Listar las vulnerabilidades de administración de sesión la intranet bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Vulnerabilidad	Detalle

b) Lista de ataques exitosos

Listar la información capturada con los ataques realizados a la intranet bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Ataque	Detalle	Información capturada

c) Secuencias de IDs de Sesión

Determinar la predictibilidad de los IDs de sesión en la aplicación bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Predicción de secuencia IDs de Sesión:
Tiempo operacional

3.3 VULNERABILIDADES DE MANIPULACIÓN DE INFORMACIÓN DE ENTRADA Y SALIDA

Determinar las vulnerabilidades encontradas en la manipulación de información de entrada y salida en la aplicación bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Vulnerabilidad	**Detalle**

a) Información Recuperada

Clasificar la información recuperada en la manipulación de información de entrada y salida en la aplicación bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Información recuperada:

3.4 FILTRACIÓN DE INFORMACIÓN

a) Vulnerabilidades de filtración de información

Determinar las vulnerabilidades de la filtración de la información en la aplicación bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Vulnerabilidad	Detalle

b) Tipo de información recuperada

Clasificar el tipo de información recuperada en la filtración de la información en la aplicación bancaria.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Tipo Información recuperada

3.5 PREOCUPACIONES Y VULNERABILIDADES

Clasificar las vulnerabilidades y preocupaciones encontradas en la intranet bancaria. Esquematizar un ejemplo y proponer una solución.

Descripción: <Descripción general de los resultados obtenidos y técnicas utilizadas>

Preocupación o Vulnerabilidad
Ejemplo

Solución

3.6 CONCLUSIONES Y RECOMENDACIONES

Conclusiones
<Conclusiones basadas en los problemas encontrados, sin juicios de valor y debidamente sustentadas>.
Recomendaciones
<Recomendaciones que permitan mejorar los inconvenientes encontrados>.

NOTA.- Adjuntar los resultados de las pruebas obtenidas que permitan sustentar la información ingresada en este informe.

Firma Analista de Seguridad	Firma Cliente
<Nombre>	<Nombre>
<CI>	<CI>