UNIVERSIDAD DE CUENCA

FACULTAD DE INGENIERÍA

MAESTRIA DE GESTIÓN ESTRATÉGICA DE TECNOLOGÍAS DE LA INFORMACIÓN

“Metodología de Hacking Ético para Instituciones Financieras, aplicación de un caso práctico”

Trabajo de Titulación, previo a la obtención de título de Magister en Gestión Estratégica de Tecnologías de Información

Autor: **Ing. Andrés Santiago Mora Ortega**
C.I. 0104032495

Director: **Ing. Diego Arturo Ponce Vásquez, PhD**
C.I. 0101822609

Cuenca – Ecuador

2017

Resumen

Las metodologías de hacking ético, representan una fotografía al estado de la ciberseguridad de una organización en un determinado tiempo. En este documento se estudia las más utilizadas como Open Web Application Security Project Testing Guide, Open Source Security Testing Methodology Manual y la certificación Certified Ethical Hacking. Dentro de las normativas establecidas en los entes de control en el Ecuador se establece mediante la resolución de la Junta Bancaria JB-3066 del Ecuador, que se deberá realizar por lo menos una vez al año un escaneo de vulnerabilidades a los diferentes sistemas que utilizan como: el Núcleo (Core) Bancario, Banca en Línea, Banca Móvil, Sistemas de ATM's, Punto de Venta (Point of Sale) y Tarjetas de Crédito entre sus principales. Enfocados en los sistemas prioritarios utilizados en las entidades financieras, se propone una metodología aplicable de forma práctica, en la que se definen 4 fases principales que son: i) Reconocimiento, ii) Descubrimiento, iii) Obtención de Acceso y iv) Mantener Acceso, haciendo hincapié con las pruebas referentes a autorización, identificación, autenticación, criptografía, manejo de sesiones y validaciones de entrada. De esta manera se efectúa un examen de seguridad óptimo brindando una objetiva visión de la vulnerabilidad de los sistemas.

Palabras Claves: Hacking Ético, OWASP, OSSTMM, CEH, Instituciones Financieras.

Abstract

Ethical hacking methodologies represent a photography of the state of cybersecurity of an organization at a given time. This paper examines the most used current methodologies such as Open Web Application Security Project Testing Guide, Open Source Security Testing Methodology Manual and Certified Ethical Hacking certification. Within the regulations established in the control entities in Ecuador by resolution of the Banking Board JB-3066 that a vulnerability scan should be carried out at least once a year on different systems like: Core Banking, Online Banking, Mobile Banking, Automated Teller Machines, Points of Sales and Credit Cards as main systems. Focus in their primary systems on the financial institutions, a practical methodology is proposed, in which four main phases are defined: i) Recognition, ii) Discovery, iii) Access, and iv) Maintain Access, emphasizing tests regarding authorization, identification, authentication, cryptography, session management and input validations. In this way, an optimal security examination is carried out, providing an objective view of the vulnerability of the systems.

Keywords: Ethical hacking, OWASP, OSSTMM, CEH, Financial Institutions.

Introducción

La ciberseguridad hoy en día es un tema muy importante para todas las organizaciones, es por eso que es necesario determinar el nivel de seguridad de las instituciones utilizando un tipo de metodología que represente el estado actual de la ciberseguridad en una organización. En el presente documento se van a observar diferentes metodologías existentes de entre las más utilizadas como Open Web Application Security Project (OWASP) Testing Guide, Open Source Security Testing Methodology Manual (OSSTMM) y la utilizada por la certificación Certified Ethical Hacking (CEH) para construir una metodología acorde a las necesidades de las instituciones financieras y que sean aplicables en el Ecuador cumpliendo las normativas vigentes.

Para esto es necesario conocer los conceptos básicos sobre la información, sus características, además de los diferentes tipos de riesgos y amenazas a las que se encuentra expuesto; conocer el concepto de hacker y sus diferentes tipos para entender la metodología que se utiliza para obtener información importante de las organizaciones, lo que se verá a detalle en los siguientes capítulos.

Tabla de contenido

Índice de Tablas y Figuras.

Tablas

Figuras

Universidad de Cuenca

Clausula de propiedad intelectual

Andrés Santiago Mora Ortega, autor del Trabajo de Titulación "Metodología de Hacking Ético para Instituciones Financieras, aplicación de un caso práctico", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 22 de noviembre de 2017

Andrés Santiago Mora Ortega

C.I: 0104032495

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Andrés Santiago Mora Ortega en calidad de autor/a y titular de los derechos morales y patrimoniales del trabajo de titulación "Metodología de Hacking Ético para Instituciones Financieras, aplicación de caso practico", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 22 de noviembre de 2017

Andrés Santiago Mora Ortega

C.I: 0104032495

Capítulo I

Conceptos sobre Hacking Ético y Ciberseguridad

Este capítulo se enfoca en brindar los conocimientos necesarios sobre la seguridad de la información, la ciberseguridad y el hacking ético, para entender lo que realizan los hackers según su clasificación y poder describir la metodología que utilizan para obtener la información de los diferentes sistemas informáticos que existen tanto pública como privadamente.

Concepto de Información

La información es un conjunto de datos, los cuales pueden ser manipulados, distribuidos y almacenados, y representan un conocimiento sobre alguna temática; estos disponen de características fundamentales que son la confidencialidad, integridad y autenticidad. Otras características relacionadas con las transacciones en línea son la verificabilidad o no repudio y la disponibilidad.

1.1 Características de la información

Estas características permiten que la información disponible se pueda interpretar de una manera correcta, coherente y represente la verdad de su temática, a continuación, veremos más a fondo las características antes descritas sobre la información.

1.1.1 Integridad

Esta característica se enfoca en garantizar que la información obtenida y transmitida no sea cambiada o alterada. Por ejemplo; si envío un correo electrónico a un compañero de trabajo, se debe recibir exactamente lo que se envía, pues si en el camino ese correo es interceptado, alterado y luego reenviado, la información entregada no corresponde a su integridad.

1.1.2 Confidencialidad

La información debe ser compartida o distribuida para las personas a las que fue remitida única y específicamente, pues si alguien recibe esta, puede aprovecharse de ella para su beneficio o el de otros.

1.1.3 Disponibilidad

Esta característica de la información se trata de que la información se encuentre siempre disponible para que se pueda utilizar, es decir, que si se dispone de un sistema de facturación el cual es indispensable para las operaciones de una empresa de minoristas (retail), el sistema siempre debe encontrarse disponible para que esta pueda operar con normalidad.

1.1.4 Autenticación

Esta característica, permite identificar al remitente o al destinatario de la información, de esta forma se valida la autenticidad del usuario para la apertura o generación del archivo, como parte de la integridad.

1.1.5 No repudio

Esta característica permite que los remitentes o receptores de la información no puedan negar sus acciones que realicen a la misma y les quiten responsabilidad sobre ella.

Luego de ver las características principales de la información podemos definir que es la seguridad de la información y su importancia en el siguiente apartado.

1.2 Seguridad de la información.

Se define como seguridad de la información al conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permiten

resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de datos y de la misma (Toro, 2017).

El concepto anterior hace referencia a todo tipo de información sea está representada de forma física (papel, documentos, actas, etc.) como en medios electrónicos como discos duros, dispositivos mp3, memorias USB, teléfonos celulares, y otros medios de almacenamiento de información.

Muchas empresas en la actualidad disponen de mucha información, sea esta física como en sistemas de información que representa el activo principal de la institución; por lo tanto, existe la necesidad de protegerla utilizando diferentes políticas, procesos y procedimientos que ayuden a garantizar las características antes mencionadas.

El objetivo de la protección es minimizar el riesgo de ocurrencia de cualquier evento de que pueda comprometer la disponibilidad, integridad y confidencialidad de la información por lo que es necesario comprender los conceptos de riesgo, vulnerabilidad y amenaza para saber a lo que se está expuesto.

1.2.1 Vulnerabilidad

La vulnerabilidad se puede entender como el hecho de estar expuesto a que se presente alguna acción que cause un impacto. Por ejemplo; no tener instalado un antivirus en una computadora.

1.2.2 Amenaza

Representa la probabilidad de ocurrencia de un evento u acción en un tiempo determinado. Por ejemplo; los varios virus y malware que existen.

1.2.3 Riesgo

El riesgo es la probabilidad latente de que la vulnerabilidad se explote y produzca consecuencias del hecho. Por ejemplo; ser infectado por un virus por no utilizar un antivirus.

Una de las principales razones por las que hoy se debe resguardar la información se debe a que se ha incrementado considerablemente el número de ataques cibernéticos que buscan robar información crítica institucional, que va a repercutir en la imagen corporativa y en pérdidas irrecuperables que pueden llevar al cierre de operaciones.

Ante estos actos maliciosos elaborados por hackers y crackers, se deben establecer principalmente conciencia de los ciberataques a los que está expuesta los sistemas informáticos para tomar las acciones correctivas. Ya casi todos los sistemas se encuentran en ambientes web y muchos de ellos están expuestos al Internet sin mayores niveles de seguridad y son muy vulnerables, es así que existe el manejo de la seguridad vinculada a los sistemas informáticos y se denomina Ciberseguridad, que se la explica a continuación (Toro, 2017).

1.3 Ciberseguridad o Seguridad Cibernética

La ciberseguridad se especializa en la protección de la información almacenada en medios digitales, para garantizar la confidencialidad, disponibilidad e integridad de los sistemas utilizados en las TIC.

Esta protección y su consiguiente administración se definen mediante procedimientos, manuales y procesos a cumplir en las organizaciones, además se debe difundir como una cultura que se aplique día a día.

Una de las formas de trabajar para mejorar la seguridad y la conciencia sobre seguridad en la empresa se trata en los antipatrones y se los aborda en el siguiente punto.

1.4 Los Antipatrones de Seguridad

Se denominan antipatrones a las malas prácticas ampliamente evidenciadas o comportamientos claramente percibidas de actividades que están realizadas de mala manera (Mowbray, 2014).

El objetivo de utilizar los antipatrones es el de mejorar las prácticas que se realizan día a día en las TIC's para que mejore la ciberseguridad dentro de las empresas u organizaciones.

Los antipatrones se componen de fuerzas de entrada que siempre están vinculadas a la confidencialidad, disponibilidad e integridad de la información en los sistemas informáticos como tareas o acciones que se realizan y que generan consecuencias de impacto negativo como podemos observar en la Figura 1. Siempre se producen dos soluciones una representa la solución de antipatrón y la otra a una solución refactorizada o reconstruida.

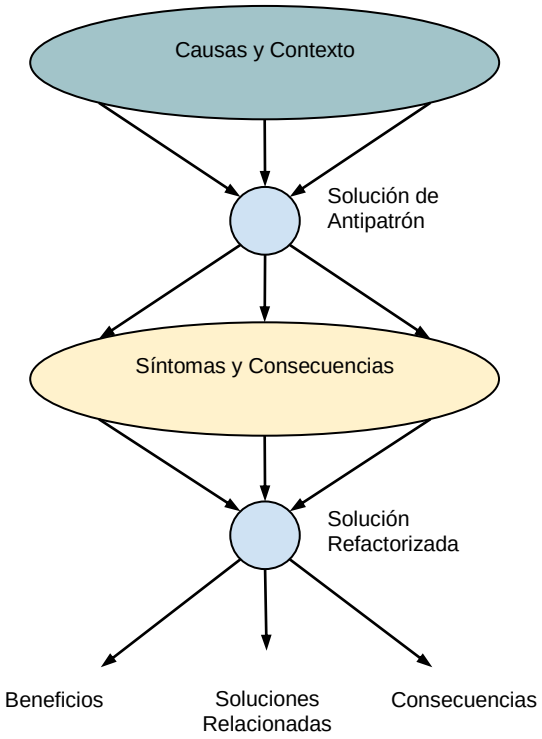

Figura 1. Concepto de Antipatrón. Fuente: Autor

La solución de antipatrón representa una situación o configuración disfuncional común y genera beneficios y consecuencias.

La solución refactorizada representa una solución más efectiva en la que se reconsideran los factores de diseño, y se generan más beneficios que consecuencias.

Los antipatrones se establecen como plantillas que se componen principalmente de los siguientes componentes:

Nombre del Antipatrón: Se especifica un nombre común fácilmente reconocible en una sola frase.

También conocido como: Muchos antipatrones son conocidos por varios nombres en diferentes organizaciones, estos se especifican como un listado.

Nombres de Soluciones Refactorizadas: Se especifica uno o varios nombres de soluciones alternativas que se pueden dar sobre el antipatrón.

Fuerzas Primarias desequilibradas: Este campo hace referencia a las fuerzas que son pobremente resueltas por el patrón, entendiendo por fuerzas a las que atenten a las características de la información.

Pruebas Anecdóticas: Son algunas burlas y anécdotas que se escuchan del antipatrón cuando se reconoce y cuando ya está presente.

El cuerpo de la plantilla de los antipatrones cibernéticos completos tiene:

Fondo: Este campo opcional proporciona explicaciones contextuales que son útiles o de interés general para el patrón y su solución refactorizada.

Solución de Antipatrón: Este espacio define la solución de antipatrón mediante el uso de diagramas, explicaciones, ejemplos y discusiones sobre las fuerzas de diseño. Es una situación comúnmente ocurrida o una configuración con implicaciones de seguridad significativas, como riesgos, amenazas y vulnerabilidades.

Causas, Síntomas y Consecuencias: Esta sección lista las típicas causas, los síntomas comunes y las consecuencias de la solución de antipatrón. Se intenta

facilitar el reconocimiento del antipatrón y entender el cómo y por qué es necesariamente reemplazada.

Excepciones Conocidas: Si hay situaciones donde la solución de antipatrón puede ser deseable se especifica en esta sección. Por ejemplo, si las consecuencias son aceptables en el contexto o si se reemplazan no se tomaría en cuenta.

Soluciones Refactorizadas y Ejemplos: Este campo define la solución refactorizada, se propone como una alternativa a la solución de patrón. La refactorización es un proceso de reemplazar o volver a trabajar la solución dada en una solución alternativa.

Soluciones Relacionadas: Si hay otras soluciones relacionadas al antipatrón se deben especificar en esta sección. A menudo hay diferentes enfoques para solventar el mismo problema que no necesariamente apuntan a la solución refactorizada escogida.

Algunos antipatrones pequeños no contiene el listado anterior completo de la plantilla por lo que se utilizan solo: El nombre del Antipatrón, Problema de antipatrón: resumiendo los síntomas, consecuencias y caracterizaciones, y finalmente la Solución Refactorizada.

A continuación, se muestra el antipatrón "*No se puede parchar lo tonto"*:

Nombre del Antipatrón: *No se puede parchar lo tonto.*

También conocido como: *Ingeniería Social, Phishing, Spam, Spyware, Manejado por Malware, Ransomware, Ataques auto ejecutados.*

Nombres de Soluciones Refactorizadas: *Conciencia de Seguridad.*

Fuerzas Primarias desequilibradas: *Confidencialidad (por ejemplo, divulgar información privada), Integridad (por ejemplo, rootkits).*

Solución al Antipatrón

La falta de conciencia de seguridad del usuario final pone en riesgo su información personal y la competitividad de la organización. La ingeniería social se puede explotar fácilmente con el simple hecho de ayudar a alguien, siendo engañados abriendo documentos maliciosos enviados por correo que pueden contener spam, y links de descarga de spyware. Estas también se encuentran en sitios web que espían los comportamientos y enviando a los beneficiarios para obtener beneficios. De igual forma para mitigar los problemas de virus existen a disposición alrededor de 9000 sitios maliciosos que brindan antivirus gratuitos, pero lo que en realidad se obtiene es una amenaza y una advertencia para pagar a un proveedor o de lo contrario el equipo se volverá inutilizable. Los malware son software que abre brechas de seguridad en los equipos que bien pueden ser instalados fácilmente insertando un dispositivo de almacenamiento USB en el equipo como un software autoejecutable que viene definido por Microsoft en su sistema operativo.

Causas, Síntomas y Consecuencias

Las causas y síntomas de este antipatrón son la falta de programas de entrenamiento de conciencia de seguridad de forma recurrente para todos los usuarios finales, incluyendo una prueba de evaluación.

Solución Refactorizada y Ejemplos

El programa de entrenamiento de conciencia sobre seguridad debe ser mandatorio para cada persona en la organización. El entrenamiento debe ser completo antes incluso de entregar el equipo al usuario final y debería refrescar los cursos anualmente. Los cursos deben incluir entrenamiento sobre destrezas de ingeniería social como también sobre seguridad en el Internet. El entrenamiento debe estar basado en las directivas de la organización sobre qué información puede ser divulgada y a quien.

Soluciones Relacionadas

Los usuarios finales deberían tener instalado herramientas de control de páginas web como parte de una suite de antivirus en sus equipos. Además, deberían tomar incluso mayores precauciones en las búsquedas en Google, ya que este da un mensaje de alerta en la página de búsqueda y presenta una página de desafío para que no se ejecuten scripts que puedan persuadir al usuario a evadir el sitio web. Soluciones como Firefox disponen de extensiones que no permiten disuadir comportamientos inseguros en la navegación. La extensión NoScript detiene los scripts web por defecto, solicitan permisos de usuario para habilitarlo en cada página web.

El listado de los antipatrones se los podrá observar en la sección de Anexos, del presente documento.

Estos antipatrones nos ayudan a identificar comportamientos que se puede corregir para que se tome en cuenta a veces las cosas más básicas que no se realizan o por el simple hecho de no tomar en cuenta a eventos que ocurren, se pueden evitar muchos incidentes de seguridad que son aprovechados por los denominados hackers.

Durante muchos años se han definido diferentes temas con respecto a los hackers y se los entiende como un grupo de personas con conocimientos avanzados en programación, redes y seguridad que buscan fallas en sistemas, para poder vulnerarlas y obtener información que les de algún rédito y por esto, se los asocia como ladrones; vamos a revisar las diferencias existentes entre los hacker y cracker para que se tenga claro la orientación.

1.5 Concepto de Hacker

Se le denomina de esta manera a las personas con conocimientos avanzados en Sistemas Operativos, desarrollo de aplicaciones y expertos en seguridad, se los

cataloga en 3 grupos que son sombrero blanco (white hat), sombrero gris (grey hat) y sombrero negro (black hat) (Miranda, 2017).

Clasificación de los Hackers.

Los hackers se los ha denominado por una relación de tipo de color de sombrero que usan, haciendo referencia del color al actuar de estos. Se clasifican en:

1.5.1 White Hat (Sombrero Blanco)

Se los denomina como los ethical hackers, ya que trabajan para empresas con el fin de asegurar sus sistemas ante cualquier tipo de ataque informático que puedan ejecutar en ellos otras personas.

1.5.2 Black Hat (Sombrero Negro)

Se encargan de romper las seguridades, colapsar sistemas, producir virus o malware para acceder información que les dé un rédito económico.

1.5.3 Grey Hat (Sombrero Gris)

Conocen las mismas técnicas que los black hat sin embargo buscan los problemas, notifican a los propietarios y cobran una cantidad de dinero para repararlos.

1.6 Concepto de Cracker

Son personas de igual forma con conocimientos avanzados, sin embargo, utilizan estos conocimientos para robar información, que pueda ser vendida para obtener algún beneficio, dañan o comprometen sistemas para sabotear eventos, empresas, personas, organizaciones y demás. Crean diferentes aplicaciones como virus, spyware, malware, cracks, etc. De la clasificación anterior al hacker black hat se lo denomina cracker.

1.7 Metodología utilizada por los Hackers

Al procedimiento utilizado por hackers éticos, se le denomina pen test o prueba de penetración, el cual se realiza en varias fases para tratar de contrarrestar la seguridad de los sistemas.

Los métodos que utilizan están definidos en varias fases para efectuar un ataque los cuales se las muestra en Figura 2:

Figura 2. Fases del Hacking Fuente: CEH v8.

1.7.1.- Reconocimiento

Fase en la que se identifica información pública que se encuentre en el Internet, como página web, servidores de correo, servidores DNS, cuentas de correo, etc. Esta fase de reconocimiento se puede clasificar entre activa y pasiva, siendo la primera en la que se utiliza diferentes herramientas para obtener la información y la pasiva sin interacción con el mismo utilizando por ejemplo ingeniería social.

Algunas de las técnicas y herramientas utilizadas en esta fase son:

- Google Hacking: Obtener la información sobre el objetivo utilizando el buscador google y sus operadores de búsqueda. Ej: site: www.amazon.com, intext:hack, filetype:mp3, info, etc.
- Dumpster Diving: traducido como Buscar en la Basura hace referencia al obtener información de códigos, claves, diagramas, contraseñas que se puedan obtener de la basura de la empresa.
- WhoIS: es una herramienta que se utiliza para conocer las direcciones IP públicas correspondientes a la empresa, sus servidores dns, proveedores de servicio, el dominio corporativo, etc.
- NMAP: esta herramienta se utiliza para realizar una exploración de la red, obtener los servidores activos, los puertos utilizados, sistema operativo, etc.
- Ingeniería Social: Mecanismo para obtener información de la empresa mediante la perspicacia del atacante, como por ejemplo fingir el ser un colaborador, o un amigo de un empleado para obtener la información necesaria.

1.7.2.- Escaneo

Esta fase determina mediante herramientas como OpenVas, Nessus entre otras, las vulnerabilidades existentes en los servidores encontrados, para determinar las fallas de seguridad existentes por malas configuraciones aplicadas, por versiones desactualizadas, etc.

Las herramientas utilizadas para esta fase realizan las siguientes operaciones:

a. Identificación de equipos en la red.
b. Verificación de puertos abiertos.
c. Identificación de sistemas operativos, aplicativos, software y parches instalados en el equipo.
d. En base a estándares de software o firmas determinar las vulnerabilidades existentes y clasificarlas según su nivel de criticidad.

1.7.3.- Obtener acceso

Se explotan las vulnerabilidades encontradas para obtener acceso a los sistemas y equipos a fin de obtener la información. Se utiliza la información recolectada de la fase anterior para conocer las vulnerabilidades que se pueden explotar para obtener el acceso, como configuraciones por defecto o mal configuradas.

Algunas de las técnicas utilizadas son:

- Desbordamiento de buffer.
- Denegación de Servicio.
- Denegación de Servicio Distribuida.
- Secuestro de sesión.
- Romper claves, mediante ataques de fuerza bruta o diccionario de datos.
- Ataques de hombre en el medio.

1.7.4.- Mantener acceso

Se debe mantener conectado a los diferentes servidores para lograr determinar la información relevante que pueda ser obtenida. Esta fase se puede utilizar el sistema vulnerado como una plataforma de lanzamiento hacia otros sistemas internos o externos. Se puede utilizar sniffers para detectar información en el tráfico de red que permita ingresar a otros sistemas.

Algunas herramientas utilizadas son:

- Puertas traseras
- Troyanos
- Shell
- Obtener los privilegios de cuenta de administrador o system administrator.
- Rootkits, spyware, Keyloggers, etc.

1.7.5.- Esconder rastro

Luego de obtener la información se procede con esconder cualquier tipo de rastro del ingreso y los ataques realizados a los diferentes sistemas. Se pueden eliminar los registros del sistema que demuestran las acciones realizadas en el equipo, alterar los logs para que no registre los eventos siguientes luego del acceso, eliminar las cuentas creadas y los archivos utilizados.

1.8 Seguridad en Profundidad

Un aspecto importante a tomar en cuenta dentro de la seguridad es que no solo intervienen los equipos de seguridad como firewall, routers, IPS, IDS, antivirus, sino que esta se debe globalizar. Esto quiere decir que para asegurar la información que brindan los sistemas de información se debe generar una cultura de seguridad en la que intervienen todos, desde las políticas que se definan para cultura de la gente, mecanismos de seguridad física, equipos electrónicos de seguridad, controles de acceso a la información, cifrado, etc.

A este procedimiento se le denomina seguridad en profundidad y consiste en separar en varias capas, que brinden seguridad cada una y permitan el identificar o detectar las intrusiones de atacantes. Cada uno de estos mecanismos sirve para que, si unos de los aspectos en una capa son vulnerados, en los otros se pueda compensar de forma que no se pueda acceder a la información que se busca como se observa en la Figura 3.

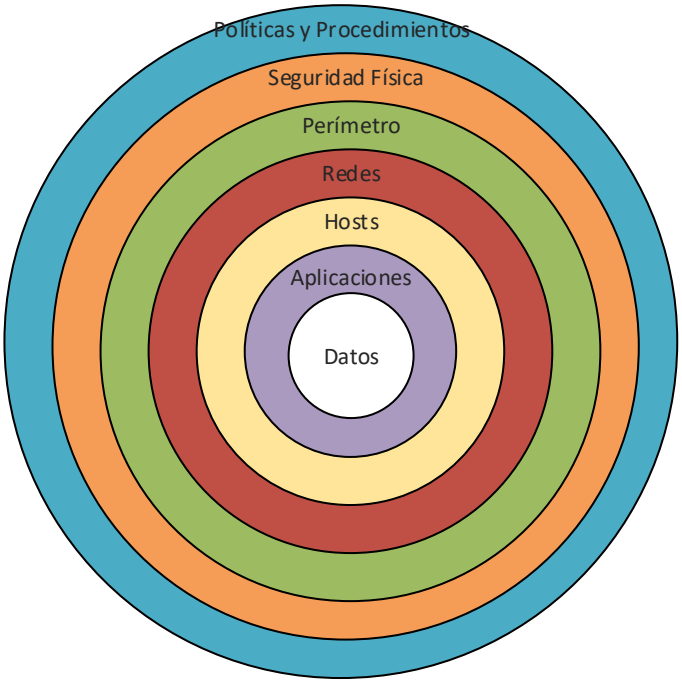

Figura 3. Modelo de Seguridad en Profundidad. Fuente: Autor

1.8.1 Políticas y Procedimientos

Representan las políticas o lineamientos definidos a nivel de Estado, Ciudad, Empresa, Departamento que generen un comportamiento de seguridad en los funcionarios o actores que la corresponden. Ej: Capacitación en seguridad a los usuarios.

1.8.2 Seguridad Física

Son los controles que se brinda al acceso físico, a Datacenters, Oficinas, Edificios, Bóvedas, etc: Guardias de seguridad, cámaras de seguridad, sensores y demás.

1.8.3 Perímetro

Equipos de seguridad de software y hardware que bloqueen el acceso a los equipos desde el exterior.

1.8.4 Redes

Equipos que permitan la segmentación de redes entre equipos, servidores, aplicativos, dispositivos, etc. Ej: VLAN, ACL, IDS.

1.8.5 Host

Seguridad aplicada a los diferentes equipos de usuario final y servidores. Mecanismos de autenticación, protocolos seguros de administración, hardening, partches de seguridad, IDS a nivel de host.

1.8.6. Aplicaciones

Manejo de antivirus dentro de las aplicaciones, metodologías de desarrollo de aplicaciones seguras.

1.8.7 Datos

Controlar mediante privilegios de acceso a la información, cifrado de datos, accesos de lectura y escritura.

Estos principios evidencian una cultura de seguridad organizacional que permitirá disponer de varios controles para que la información no pueda ser fácilmente accedida por intrusos.

Existen diferentes metodologías utilizadas para hacking ético, las cuales las observaremos en los siguientes capítulos.

Capítulo II

Metodología CEH (Certified Ethical Hacker)

Uno de los métodos más utilizados para realizar un hackeo ético es el utilizado por la certificación CEH, que es emitida por EC-Council (Consejo Internacional de Consultoría de Comercio Electrónico). Esta certificación es una de las más conocidas para cualquier especialista de seguridad, inclusive en instituciones de Defensa del gobierno de los Estados Unidos, uno de los requisitos es disponer de esta certificación para la acreditación.

La versión vigente es la 9, disponible con el código de examen 312-50. Para obtener la certificación es necesario tomar el curso y dar el examen; en caso de no querer tomar el curso se debe enviar información validada que justifique el conocimiento de por lo menos 2 años para que se pueda tomar el examen de certificación.

La certificación está compuesta de 18 módulos que tratan desde cero, el explicar los conceptos de seguridad, la metodología utilizada, las herramientas y los laboratorios para que se pueda aprobar las misma (EC-Council, 2015).

Valor Hack: Es la notación que se utiliza para decir que algo tiene un valor de información para el hacker.

Vulnerabilidad: Existencia de debilidad, diseño o errores de implementación que pueden guiar para que se comprometa la seguridad de un sistema.

Exploit: Software que aprovecha una o varias vulnerabilidades para obtener acceso o información de los sistemas.

Payload: Es una carga de datos que se agrega al exploit para vulnerar una aplicación o servicio de un sistema para obtener acceso o información.

Ataque Zero-Day: Representa un ataque de las vulnerabilidades encontradas al sistema antes de que se desarrolle el parche para remediar dicha vulnerabilidad. Ante este tipo de ataque, los antivirus resultan ineficaces.

Daisy Chainig: Representa obtener acceso a una red o a una computadora y utilizar la información para acceder a múltiples redes y computadoras que tienen información deseable.

Doxing: Divulgar la información personal de un individuo recogida de los medios.

Bot: Es una aplicación desarrollada que permite controlar remotamente la ejecución o automatizar tareas determinadas en el equipo comprometido.

Dentro de los conceptos de seguridad de la información se observaba que la información debe tener 3 características principales que son la confidencialidad, integridad y disponibilidad. Adicionalmente se debe manejar la autenticidad, que representa que un usuario pueda validar que durante la comunicación, su información es genuina. El no repudio que significa que se garantice que no se pueda rechazar la información enviada al destinatario borrando la información del envío o de la recepción de igual forma.

Esta metodología define al hacking o hackeo como la explotación de las vulnerabilidades para comprometer la seguridad y obtener acceso no autprizado a los recursos del sistema.

Al hacker se lo conoce como un individuo con excelentes habilidades, con la capacidad de crear y explorar el software y hardware de las computadoras, viéndolo como un mérito el hackear la mayor cantidad de equipos, para realizar cosas ilegales.

A diferencia de lo que se vio en el capítulo anterior se especificaron únicamente 3 tipos de hackers, en la versión 9 de CEH se establecen 8, los cuales se muestran a continuación.

Black-hats: Individuos con habilidades extraordinarias en computación, recurriendo a actividades maliciosas y destructivas y son conocidos como crackers.

White-hats: Individuos realizando tareas de hacker y usándolas para propósitos de defensa y que son conocidos como analistas de seguridad.

Grey-hats: Individuos que trabajan tanto para defensa y ataque en varios momentos.

Suicide Hackers: Individuos quienes derriban la infraestructura crítica por una causa y no se preocupan de ir a la cárcel o recibir otro tipo de castigo.

Script Kiddies: Un hacker no calificado que compromete el sistema mediante la ejecución de scripts, herramientas y software desarrollado por verdaderos hackers.

Cyber Terrorists: Individuos que tiene una amplia gama de habilidades, son motivados por la religión o creencias políticas para crear miedo a gran escala interrumpir las redes de computadoras.

State Sponsored Hackers: Individuos contratados por el gobierno para penetrar y obtener información secreta y dañar la información de otros gobiernos.

Hacktivist: Individuos que promueven una agenda política mediante el hacking para dar de baja o deshabilitar sitios web.

Las fases que se utilizan para el hacking son:

1. Reconocimiento.
2. Descubrimiento.
3. Obtención de acceso.
4. Mantener el acceso.
5. Limpiar el rastro.

A continuación, veremos con mayor detalle cada una de estas.

2.1 Reconocimiento

Es la fase en la que un atacante busca obtener información esencial sobre el objetivo para lanzar el ataque, puede ser el futuro punto de retorno de alguna de las fases, ya que se puede encontrar más información que necesite ser identificada. Esta

etapa incluye de objetivo a los equipos de clientes, empleados, operaciones, red y sistemas.

Existen dos tipos de reconocimiento:

- Reconocimiento Pasivo: Involucra la adquisición de información sin realizar ninguna tarea todavía contra el objetivo, sino obteniendo la misma de lugares y búsquedas públicas.
- Reconocimiento Activo: Implica interactuar directamente contra el objetivo de cualquier manera.

Footprintig: es el proceso de recolectar la mayor cantidad de información posible acerca de la red del objetivo, para identificar varias maneras de ingresar en la red de los sistemas de la organización.

Las características principales del footprinting son:

- Conocer la postura de seguridad.
- Reducir el área de búsqueda.
- Identificar las vulnerabilidades
- Dibujar el mapa de la red

Los objetivos principales del footprinting son:

- **Recolectar la información de la red:** nombre del dominio, dominio interno, direccionamiento de los sistemas, ver los servicios que se encuentran ejecutando, verificar la existencia de IDS, ver los mecanismos de autenticación utilizados, etc.
- **Recolectar la información del sistema:** Nombres de usuarios y grupos, ver las tablas de enrutamiento, la información SNMP, la arquitectura de los sistemas, contraseñas, etc.
- **Recolectar la información de la organización:** Datos de los empleados, directorio corporativo, las direcciones y números de teléfonos, políticas de

seguridad aplicadas, artículos de prensa sobre la organización, comentarios en el código sobre el desarrollo, etc.

La metodología usada para la recolección de información de footprinting utiliza los siguientes pasos:

1. Recolectar huellas en los buscadores.
2. Recolectar huellas utilizando google hacking (atributos especiales utilizados en la búsqueda para obtener información más explícita).
3. Recolectar huellas a través de las redes sociales.
4. Recolectar huellas en el o los websites.
5. Recolectar los correos electrónicos válidos.
6. Inteligencia Competitiva (Páginas de búsqueda de trabajo, rankings, reputación de la página, etc).
7. Recolectar información con WHOIS (Administradores de websites, administradores del dominio empresarial público).
8. Recolectar información usando el DNS (registros tipo A, CNAME, MX, NS, PTR, TXT, SRV).
9. Recolectar información de la red (rangos de direcciones IP públicas, y privadas usadas, mapa de red).
10. Recolectar información utilizando la ingeniería social (aprovecharse de conocidos, suplantar entidades de control, espionaje, buscar en la basura).

Las herramientas más usadas según cada fase de reconocimiento son las siguientes:

- **Buscadores y Redes Sociales:** Google, Facebook, Linked-In, Twitter, Google+, Pinterest,
- **Google Hacking:** Google Hacking Database, MetaGoofil, SiteDigger,
- **Sitios Web:** HTTrack Web Site Copier, Black Widow, Webripper,

- **Correo Electrónico:** eMailTrackerPro, PoliteMail, Email Lookup,
- **Inteligencia Competitiva:** Hoovers, LexisNexis, Business Wire,
- **WHOIS:** SmartWhois, Domain Dossier,
- **DNS:** DNSstuff, DNS Records,
- **Comunicaciones:** Path Analyzer Pro, VisualRoute, Network Pinger.

2.2 Descubrimiento

Este proceso implica el realizar un escaneo de la red en base a la información recolectada de la fase anterior, obteniendo equipos que se encuentran conectados, información del sistema operativo, los puertos que están abiertos, etc para poder estructurar el ataque.

2.2.1 Verificar equipos activos

Este mecanismo de descubrimiento se realiza enviando solicitudes de ping a los diferentes host o redes para identificar qué dispositivos responden, se puede determinar la máscara de red de los equipos utilizando una calculadora de red y se puede generar un inventario de los equipos habilitados en la red.

Para esto hay varias herramientas como: Advanced Ip Scanner, Visual Ping Tester, Ping Sweep, etc.

2.2.2 Escanear detrás del IDS

Luego de la etapa de descubrimiento de equipos activos, es necesario conocer que puertos y servicios están levantados en los equipos y conocer por donde se puede atacar. Algunas de las herramientas utilizadas pueden ser configuradas para que los IDS (Intrusion Detection System) no los detecten cambiando parámetros como el envío de las peticiones cada cierto tiempo o con cierta carga, para que puedan saltarse esa seguridad.

El uso de paquetes fragmentados también es de gran ayuda para evadir los controles del IDS, el cual consiste en dividir la cabecera del paquete en múltiples paquetes, así no son detectados los propósitos del paquete principal.

2.2.3 Banner Grabbing

Otro método para detectar información es el uso del Banner Grabbing que se trata de obtener la información del sistema operativo que está corriendo en el objetivo, el cual puede ser de forma activa cuando se envía información de paquetes específicos y la respuesta evidencia el tipo de SO que utiliza; y la pasiva que se puede obtener de los mensajes de error del servidor, también mediante la escucha del tráfico de la red y el ver las extensiones o plugins de las aplicaciones que están funcionando.

Algunas herramientas para realizar Banner Grabbing son:

- ID Serve (identificar SO de los servidores web)
- Netcraft (recupera la información del banner)
- Netcat (snnifer)

2.2.4 Escaneo de vulnerabilidades

El escaneo permite identificar vulnerabilidades y debilidades de un sistema o una red, para determinar cómo puede ser explotado.

Una de las principales herramientas que se utilizan para esto son Nessus, Qualys FreeScan, OpenVAS.

Estas herramientas muestran en un reporte las vulnerabilidades encontradas de cada servidor o equipo de red, con una clasificación por nivel de criticidad de las mismas, siendo estas high, medium y low principalmente.

2.2.5 Dibujar Diagramas de Red

El realizar un esquema del diagrama de red nos permite identificar cómo está estructurada la red y observar los diferentes puntos en los que se puede atacar.

Existen herramientas que nos permiten obtener el diagrama de red realizando un escaneo y determinando servidores, equipos de red, firewall, etc y poder exportarlos a un diagrama de Visio para poder manipular más fácilmente, tanto de forma física como lógica. Algunas de estas son: OpManager, NetMapper, NetworkView, etc.

2.2.6 Preparar el Proxy

Se denomina proxy a un servidor intermediario que se puede utilizar para tener acceso a un servicio o puerto de otro. Por ejemplo, para el uso común de internet compartido en una empresa se utiliza este tipo de herramientas que permiten mediante este acceder a la Internet. La idea de utilizar el proxy es el buscar por donde se puede acceder a otros equipos y servidores que no se pueden alcanzar fácilmente con los equipos clientes; además nos permiten esconder la dirección IP de nuestro equipo origen, ya que la conexión directa al objetivo la realiza el proxy.

Una técnica para esconder los equipos desde donde se origina es utilizar una cadena de proxys, lo que enmascara cada conexión anidada hasta llegar al destino. De esta forma se puede realizar un ataque desde un país origen, pero se puede detectar como si se lo hiciera desde otro.

Otro uso que tiene los proxys es el acceder a páginas que no están permitidas por restricción de país como Netflix, y otros servicios de streaming que solo funcionan para ciertos países.

2.2.7 Enumeración

La enumeración es un proceso interesante que nos permite extraer información como usuarios, nombres de equipos, archivos compartidos y servicios de un sistema, a partir de enumeraciones de ldap, SNMP, SMTP, NTP.

2.3 Obtener Acceso

A partir de esta fase viene lo más importante debido a que la obtención del acceso permite ingresar al equipo, escalar privilegios para llegar a ser administrador del equipo, del dominio o de la base de datos y sacar la mayor información posible.

Esta fase se logra con estas dos técnicas que son las siguientes: Romper Passwords (Cracking Password) y consecuentemente a eso la elevación de privilegios (Escalating Privileges).

2.3.1 Romper los Passwords

Son técnicas utilizadas para recuperar contraseñas de computadoras y sistemas, muchas de las cuales se las realiza mediante un diccionario con contraseñas fáciles de adivinar o algunas que vienen por defecto en servidores sin configuraciones seguras.

Los tipos de ataques para romper las contraseñas son:

- Ataques no electrónicos: Husmear sobre el hombro, Ingeniería social y Buscar en la Basura.
- Ataques activos en línea: Directamente conectado. Ataques de fuerza bruta y de diccionario. Inyección de hash y phishing, uso de keyloggers, troyanos y spyware.
- Ataques pasivos en línea: Sin conexión directa con el equipo objetivo. Escucha de tráfico (sniffing) ataques de hombre en el medio.
- Ataques fuera de línea: atacante copia el archivo de contraseñas y lo prueba en su sistema desde otra ubicación o equipo. Rainbow Tables.

El mecanismo de búsqueda comprende encontrar un usuario válido, obtenido de direcciones de correo, o de las consultas al servidor LDAP, para luego crear una lista de posibles contraseñas, ponerles con prioridad las que son más probables de las que no y probar cada una hasta encontrar la contraseña.

El uso de los passwords por defecto hace que sea muy fácil de ingresar a sistemas y aplicaciones según cada fabricante, algunos de estos utilizan varias e inclusive se

manejan passwords generales para el mantenimiento que la mayoría de las veces no se desactiva y permiten el acceso.

Una forma de obtener las credenciales se basa en el uso de keyloggers, que son dispositivos USB que se deben colocar en el equipo y comienzan a almacenar todo lo que se digita. Esta tarea se puede realizar mediante ingeniería social para conectar y retirar los mismos. Existen también keyloggers por software que puede obtener la información del texto digitado y del audio del micrófono y de igual forma se puede almacenar en un repositorio de archivos.

Otra forma de conseguir acceso es mediante la copia de los hashes de cada usuario; en lugar de buscar la contraseña se utiliza el hash correspondiente a cada usuario para reemplazarlo en la sesión nueva y tener el acceso al sistema.

Algunas de las herramientas utilizadas son:

- Offline NT Password & Registry Editor
- Password Unlocked Bundle
- Jhon the Ripper
- Password Cracker
- Hash Suite
- InsidePRO

2.3.2 Elevación de Privilegios

Los atacantes pueden obtener el acceso a la red sin utilizar un usuario administrador para luego conseguirlo. Se utilizan diferentes formas como errores de programación, bugs del sistema y supervisión de la configuración.

Existen 2 tipos de elevación de privilegios los cuales son: la vertical, que se basa en obtener privilegios más altos que el existente; y el horizontal que busca obtener privilegios del mismo nivel que fueron asignados a otro usuario.

La forma de escalar privilegios es instalar una DLL maliciosa en el directorio de aplicaciones reemplazando a la DLL real. Cuando se ejecuta la dll se obtiene el acceso remoto al equipo mediante un shell.

Luego de haber obtenido la sesión se vuelve sencillo en Windows cambiar la contraseña utilizando el comando net user <nombre de usuario> que nos permitirá disponer la contraseña para levantar servicios, instalar aplicaciones o removerlas sin ningún inconveniente.

Existen varias herramientas para escalar privilegios como Active@ Password Changer que detecta y obtiene la Base de Datos de Seguridad de Microsoft del equipo (SAM) y muestra toda la información de cualquier usuario del equipo.

2.4 Mantener Acceso

Esta fase consiste en mantener el acceso para efectuar las tareas posteriores como instalar spyware, keyloggers, y otras herramientas que le permitan a los hackers a controlar el equipo remotamente o vigilar las interacciones que realizan en la máquina.

Ya obtenido el acceso al sistema o aplicaciones en el equipo se pretende desplegar un software malicioso, que permite administrar de forma remota el equipo como el software denominado DameWare.

Los Sypware también son muy comunes luego del ingreso al equipo y graban todas las interacciones del usuario en el equipo sin que el usuario tenga conocimiento.

Algunos de estos spyware son:

- NetVizor
- Remote Desktop Spy
- Net Nanny Home Suite

Existen varios spyware según su objetivo, como por ejemplo Celulares, WebCam, Voz y sonido, USB, aplicaciones de mensajería como Facebook, Skype, Google Talk y demás, y GPS.

2.5. Limpiar Rastro

Se encarga de borrar la evidencia dejada en los equipos para que no se pueda verificar lo que hicieron, instalaron, borraron, etc. Se utiliza herramientas como rootkits que esconden la presencia del atacante mientras realiza sus actividades maliciosas.

Existen diferentes tipos de Rootkits los cuales que son:

2.5.1 Rootkit a Nivel de Hipervisor:

Actúan como un hipervisor y modifican la secuencia de arranque del equipo para que cargue el sistema operativo como una máquina virtual.

2.5.2 Rootkit de Hardware/Firmware:

Esconde en los dispositivos de hardware o en el firmware de los mismos su software. A estos no se les realiza una validación de integridad del código dentro del Sistema Operativo por lo que son susceptibles a que se ejecuten sin verificar.

2.5.3 Rootkit a nivel de Kernel:

Agrega código malicioso o reemplaza el código del kernel del sistema operativo drivers de dispositivos.

Adicionalmente a los rootkits existen técnicas para esconder información maliciosa en archivos, que se denomina esteganografía estos archivos pueden ser archivos de texto, pdf, archivos de audio, video e imágenes.

2.6 Equipos de Respuesta a Incidentes Informáticos.

Esta certificación no garantiza la experiencia para poder explotar las vulnerabilidades en una institución, sin embargo, muchas de las veces el conocimiento o los títulos que se obtienen, no reflejan el día a día de lo que acontece en el ciberespacio y la seguridad, esto se vería como un anti patrón como se indicaba en el capítulo 1. Este antipatrón estaría demostrado como la falta de experiencia en el campo, ya que no se podría ejecutar de forma correcta o a un nivel muy básico, los ataques que no representaría el verdadero nivel de seguridad que tiene el objetivo a ser evaluado.

El trabajo día a día brinda un mayor nivel de conocimiento que no está siempre plasmado en documentos como manuales de procedimientos si no en el conocimiento adquirido por el atacante.

Algunas de las empresas que brindan este tipo de servicio de compartir la información sobre los ataques, prácticas y formas de actuar generalmente, están vinculadas con la seguridad nacional, tratando de prevenir ataques de terceros a instituciones públicas o a organismos.

Existe un grupo de entidades que se denomina FIRST (Forum of Interactive Response and Security Teams o Foro de Respuesta a Incidentes y Equipos de Seguridad) fundando desde 1988 y se ha extendido en todo el mundo siendo un mecanismo de ayuda y respuesta a incidentes de todos los países en las que se comparte todos los conocimientos, para tomar medidas y prevenir los nuevos ataques que ocurren día a día en el ciberespacio.

2.6.1 EcuCert

En nuestro país, existe la entidad denominada EcuCERT, o el Centro de Respuesta de Incidentes Informáticos del Ecuador, que está vinculada directamente con la ARCOTEL, para la regulación y control de las telecomunicaciones del país.

Esta entidad es parte de la red FIRST antes descrita y su misión principal se basa en proporcionar servicios de respuesta a incidentes a nivel de las instituciones del

Estado de forma prioritaria, sin embargo, también se brinda los servicios a cualquier empresa que lo requiera. Estos servicios son:

- Diseñar y ejecutar procedimientos que se utilizan para la identificación de los componentes de infraestructura informática de alto riesgo.
- Realizar actividades proactivas y reactivas para proteger y asegurar la información de las TIC.
- Determinar el impacto, alcance y naturaleza de los eventos o incidentes y apoyar en la implementación de soluciones.
- Elaborar registros de los incidentes e investigaciones realizadas para tener la información necesaria para tomarlos como referencia de estos eventos.
- Cooperar con los centros de respuestas a incidentes de seguridad informática para tener y brindar el apoyo necesario para la solución de los mismos en el Ecuador.
- Realizar convenios con organizaciones nacionales e internacionales que permitirán desarrollar las investigaciones de casos de fraude a las TIC.
- Coordinar y colaborar con empresas públicas y privadas, como proveedores de Internet, prestadoras de servicios de seguridad, CSIRT de otros países, entes reguladores y otros con el objetivo de cumplir la reglamentación correspondiente.

Algo que impulsa a un hacker ético a sobresalir es el hecho de que tiene la pasión para hacerlo, no se trata únicamente de cumplir una tarea que le asigna, sino poner todo el empeño de sí mismo para alcanzar cosas nuevas y aprender de ellas. De esta forma es que siempre se encuentran nuevas formas de vulnerar los sistemas y de asegurarlas, se podría decir que el comportamiento que lo hace hacker a alguien, es el buscar más allá de lo conocido y siempre dedicar el máximo para ello. Siempre se

tratan de retos nuevos que hacen que la inventiva esta tan lúcida para idear nuevas formas de romper la seguridad.

En el siguiente capítulo se observará metodologías de hacking ético o Pentest que tiene otra forma de realizar de las definidas en este capítulo.

Capítulo III

Metodología OSSTMM

Esta metodología fue desarrollada por Pete Herzog, en 2001 desde el ISECOM (Instituto para la seguridad y Metodologías Abiertas), que se denomina Manual de Metodología Abierta de Testeo de Seguridad, en el que intervinieron alrededor de 150 expertos en seguridad y se enfoca en definir una metodología abierta que sea el estándar para el uso de hacking ético y una auditoría completa a la seguridad.

Esta metodología nació en el año 2000, y poco a poco fue tomando fuerza, la última versión publicada es la 3, que se dispone únicamente en inglés, sin embargo, está en fase de pruebas la versión 4 que se encuentra disponible para miembros de ISECOM, (Herzog, 2010). La versión vigente se emitió en el año 2010.

Esta metodología se basa en las mejores prácticas de ITIL, ISO 27001 y 27002, y aplica el estándar SP 800-30 sobre la evaluación y gestión de riesgos.

Adicional a la parte técnica y de operación que se define en esta metodología, también se manejan otros aspectos que no se toman en cuenta, como las credenciales del o de los profesionales que realizan el test, la distribución y comercialización de la prueba, la presentación de resultados, normativas y legislaciones que se deben tomar en cuenta antes durante y después de la realización, así como la gestión de las tareas que se realicen. En este sentido es una metodología muy enfocada que pretende no solo evaluar las seguridades que debe tener una organización, sino que se administra también la ejecución del test en cada una de sus etapas, para que se lo lleve de una manera correcta (Toth, 2014).

Un aspecto muy importante y relevante en la versión 3 es el uso de los RAV (Valores de Evaluación de Riesgo), que son considerados como las degradaciones de

la seguridad sobre un ciclo de vida específico. Entiéndase como los problemas de seguridad encontrados en una fase del test realizado, si bien se define una fórmula para el cálculo de cada una de estas se especifican muchas métricas para tomar en cuenta que no se verán a detalle en el presente documento.

Existen términos básicos que se toman en cuenta para el manejo de la metodología que son:

- Activo: Se denomina sistema o producto al cual se le realiza el análisis, Ej: computadora, servidor, aplicación, dispositivos de red, etc.
- Vulnerabilidad: Presencia de una debilidad que puede ser explotada.
- Control: Medida que se toma para que evitar que se produzca un evento.
- Separación: Se considera como el mecanismo para que no se lleve a cabo la amenaza sobre un activo, o no exista la amenaza.
- Porosidad: Representa los puntos de interacción y reducen la separación entre amenazas y activos.

3.1 Seguridad y Protección

Un concepto muy importante que se maneja es la seguridad operacional, la cual es el uso de la separación y los controles para que una amenaza no se vincule con el activo. En caso de que no se pueda dar una separación entre la amenaza y el activo se pueden establecer diferentes controles para se llegue a proteger el activo.

La porosidad se puede catalogar en las siguientes categorías:

- Visibilidad: es la forma en la que es visible un objetivo dentro del alcance.
- Accesos: Son cada punto desde donde se puede presentar una interacción.
- Confianza: Representa una interacción entre dos elementos que no requiere autenticación alguna.

Los tres factores definen la porosidad, de tal forma que mientras este valor sea más alto la seguridad del objetivo es menor, y si la porosidad es menor el nivel de seguridad es mayor para el activo.

Cuando existen interacciones es necesario tomar en cuenta los controles que permitirán que las amenazas no se exploten contra los activos.

3.2 Controles

En esta metodología se definen 10 tipos controles principales, y las limitaciones que tienen estos controles. Estos controles se evalúan con un valor, para determinar el estado de seguridad en un diferente estado de tiempo.

Los controles se dividen entre controles de interacción y de proceso que se muestran en las siguientes Tablas 1 y 2.

Tabla 1. Controles de Interacción. Fuente: Autor

Controles de Interacción	
Autenticación	Validación e intercambio de credenciales.
Indemnización	Compromiso entre las partes que intervienen de protección ante falla de uno de los involucrados.
Resistencia	Protección a los activos ante las fallas de las interacciones
Subyugación	Condiciones en la que se presentarán las interacciones.
Continuidad	Mantiene la interacción con los activos aún en caso de fallas.

Tabla 2. Controles de Proceso. Fuente: Autor

Controles de Proceso	
No repudio	Garantiza que los participantes que interactúan no puedan negar su interacción
Confidencialidad	Permite que la no divulgación de información entre otros que no intervienen y no están autorizados.
Privacidad	Evita que se conozca cómo se accede, muestra o intercambia un activo.

Integridad	Detecta cuando un activo se modificó por desconocidos en la interacción.
Alarma	Notifica que ha existido una interacción o que se encuentra en proceso.

Para la ejecución de los controles estos se establecen sobre las características que debe tener la información CIA (Confidencialidad, Integridad y Disponibilidad por sus siglas en inglés), cada una de estas se le asocian controles como se muestra en la

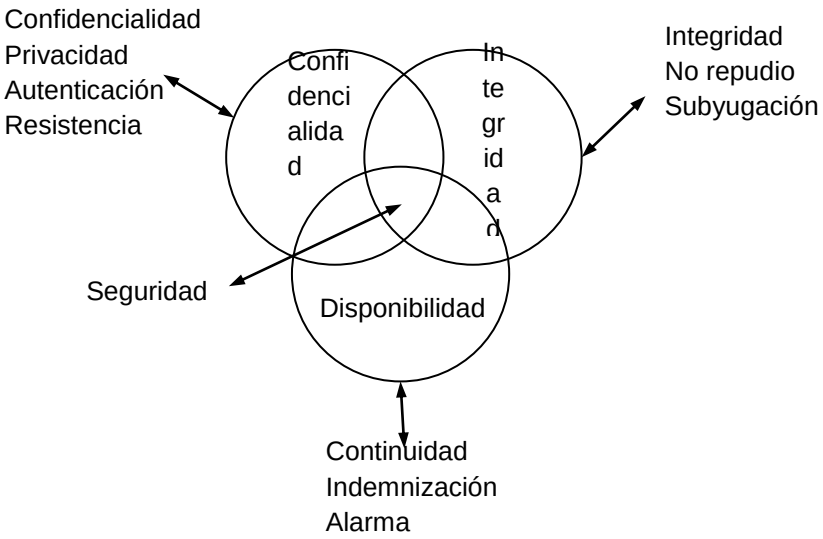

Figura 4.

Estas relaciones se pueden describir de la siguiente forma:

Confidencialidad

- Confidencialidad: Confidencialidad respecto a los activos.
- Privacidad: Confidencialidad sobre la forma de la interacción.
- Autenticación: Escoger entre quien puede acceder a la información confidencial y quién no.
- Resistencia: Confidencialidad sobre los activos en caso de falla.

Integridad

Figura 4. Relación entre CIA y controles operacionales, Fuente: Autor

- Integridad: Integridad sobre los activos.
- No repudio: Integridad de los participantes.
- Subyugación: Integridad en la forma de comunicación.

Disponibilidad

- Continuidad: Disponibilidad del servicio pese a fallas y desperfectos.
- Indemnización: Disponibilidad luego del a pérdida.
- Alarma: Notificación de pérdida de disponibilidad para tomar correcciones.

3.3 Limitaciones

Las limitaciones se denominan como la incapacidad de que un mecanismo de protección funcione como se espera.

Estas limitaciones se las clasifica dentro de 5 categorías las cuales son:

- Vulnerabilidad: Falla que puede permitir el acceso no autorizado a un activo.
- Debilidad: Falla que anula los efectos de los controles.
- Preocupación: Falla que reduce los efectos de los controles de proceso.
- Exposición: Dejar visible de forma directa o indirecta a un activo injustificadamente.
- Anomalía: Falla que no se comprende.

Definición del alcance

Para definir el alcance se establecen los activos que se quieren proteger, además se determina la zona de compromiso que está comprendida como un área alrededor de los activos. Las interacciones que existen desde y hacia adentro, desde afuera hacia adentro y desde adentro hacia afuera se denominan vectores.

3.4 Canales

Las interacciones se pueden representar sobre varios canales y se clasifican en:

Tabla 3. Clases y Canales utilizados en OSSTMM. Fuente: Autor

Clase	**Canal**	**Descripción**
Seguridad Física	Humanos	Interacción entre personas
	Físico	Todo elemento tangible que intervenga en el análisis
Seguridad Inalámbrica	Medios Inalámbricos	Señales emitidas y recibidas dentro del espectro electromagnético
Seguridad en las comunicaciones	Telecomunicaciones	Redes de comunicación sobre el cableado telefónico
	Redes de Datos	Redes de datos cableadas

Dependiendo del conocimiento que se disponga del entorno se pueden realizar diferentes pruebas. OSSTMM maneja 6 tipos de pruebas que son:

- Caja Blanca: Se tiene previo conocimiento integral del entorno y elementos de la prueba.
- Caja Gris: Conocimiento parcial del entorno y los elementos de la prueba.
- Caja Negra: No se tiene ningún conocimiento previo de ninguno de los dos factores.
- Sombrero Negro: denominado cracker, principalmente infringen la ley para obtener réditos.
- Secuencial: Auditor y objetivo conocen los avances y detalles a realizar antes durante y después de la auditoría. Se realizan pruebas de protección del objetivo y se puede observar.
- Inverso: Se está consciente tanto el auditor como el objetivo de las pruebas, sin embargo, el objetivo no tiene conocimiento de qué, cómo y cuándo se realizará las pruebas. También se denomina pruebas de Ejercicio de Equipo Rojo.

OSSTMM maneja un conjunto de reglas sobre cuándo, qué y cuáles eventos van a ser probados en la evaluación de seguridad, este documento se denomina STAR (Security Test Audit Reporting) y comprende el siguiente contenido.

1. Resultado Final
2. Fecha y hora del test
3. Duración del test
4. Tipo de Test
5. Índice del test
6. Canales probados y verificados
7. Vectores
8. Verificaciones y cálculo de métricas de los niveles de protección operacional, controles perdidos y limitaciones de seguridad.
9. Conocimiento de que pruebas pueden ser completadas, no completadas o parcialmente completadas.
10. Publicación del test y responsabilidad de los resultados.
11. El margen de error
12. Procesos que interfieran en la limitación del test.
13. Cualquier conocimiento de anomalías.

Algo que destacar de esta metodología es la utilización de los denominados dashboards o cuadros de mando para plasmar los resultados obtenidos de los RAVs y mostrarlos de una forma más amigable y entendible.

3.5 Proceso de cuatro puntos:

Dentro del proceso de evaluación, es necesario determinar que la información provenga de todas las fuentes posibles, estas posibles fuentes se denominan los cuatro puntos, que cuales son: el entorno, la interacción directa, las emanaciones del objetivo y la modificación del ambiente como se observa en la Figura 5.

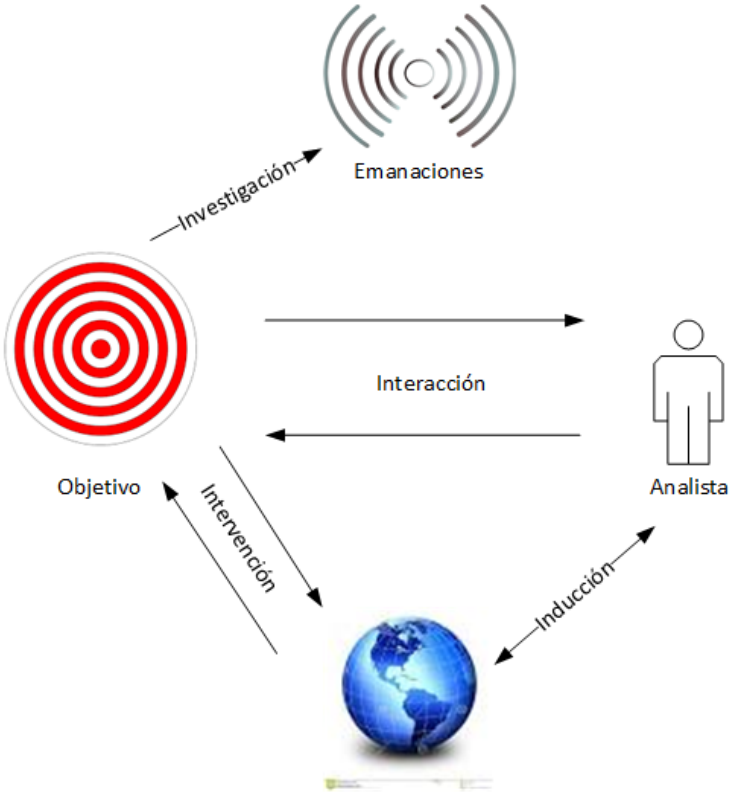

Figura 5. Interacciones en el proceso de 4 puntos. Fuente: Autor

Inducción: Conocer el entorno en donde reside el objetivo.

Interacción: Interactuar con el objetivo y observar los resultados obtenidos.

Investigación: Analizar las emanaciones que provienen del objetivo.

Intervención: Modificar los recursos del entorno que son necesarios por el objetivo para ver cómo se comporta.

Dentro de los puntos antes mencionados tiene varias etapas para dar un mayor sentido de profundidad del análisis.

Inducción: Revisión del Entorno, Logística, Verificación de detección activa.

Interacción: Auditoría de visibilidad, Verificación de accesos, Verificación de confianza y Verificación de controles.

Investigación: Verificación de procesos, Verificación de la configuración, Validación de la propiedad, Revisión de segregación, Verificación de exposición y exploración de inteligencia de negocios.

Intervención: Verificación de cuarentena, Auditoría de privilegios, Continuidad de negocio y Alerta y revisión de logs.

Representando esto de una forma de flujo nos da como resultado una metodología aplicable para cualquier tipo de test y sobre cualquier canal, como se observa en la Figura 6.

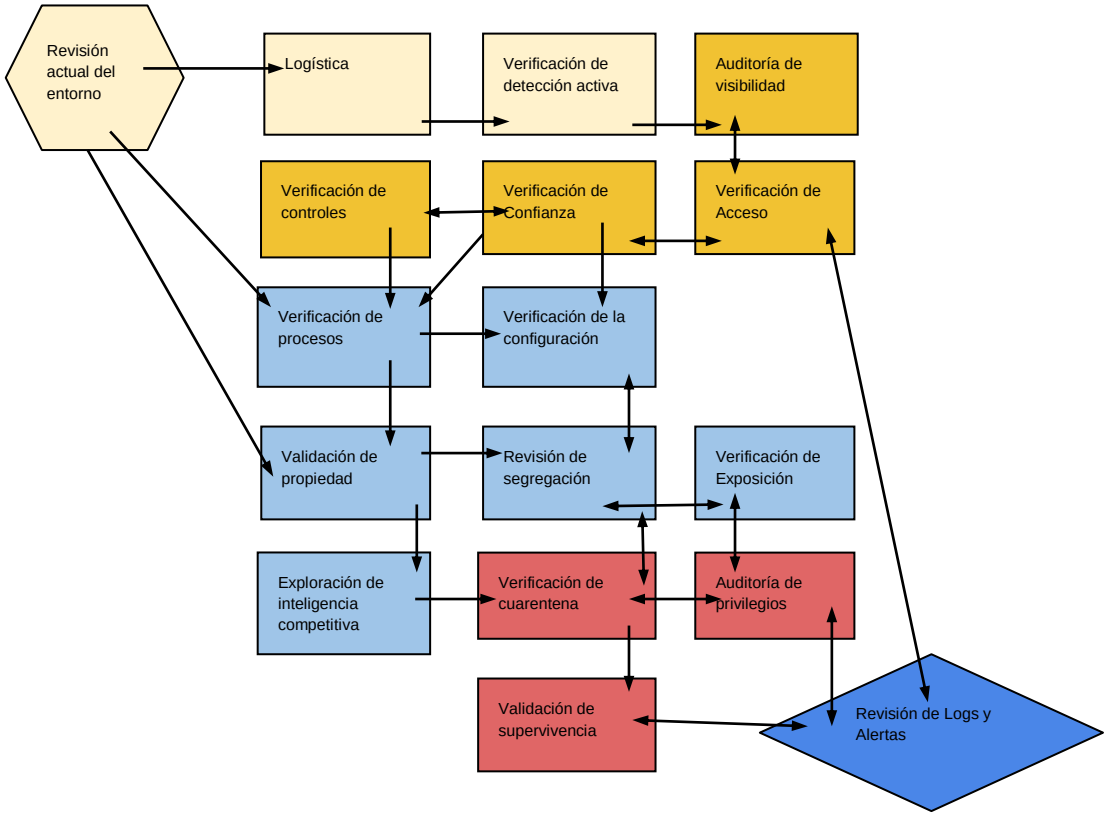

Figura 6. Diagrama de Flujo de la metodología OSSTMM. Fuente: Autor

3.6 Fases de análisis

Esta metodología está compuesta de las siguientes fases de análisis:

- **Seguridad de la información:** Se refiere al manejo de la información dentro de la organización. Consiste en que se asigne las responsabilidades y roles

para el acceso a la información necesaria y no más de esta. Revisión de Inteligencia Competitiva, Privacidad y Recolección de Documentos.

- **Seguridad de los procesos:** Se puede determinar qué personas intervienen en el uso de la información. Testeo de Solicitud, Testeo de Sugerencia Dirigida, Testeo de Personas Confiables.
- **Seguridad en las Tecnologías de Internet:** Logística y Controles, Exploración de Red, Identificación de los Servicios del Sistema
- **Seguridad en las comunicaciones:** Se maneja los diferentes medios de comunicación como: PBX, Mailbox, Fax, y módem.
- **Seguridad inalámbrica:** Se analizan los sistemas que en los intervengan sistemas inalámbricos, como redes inalámbricas, bluetooth, radio enlace, sistemas de control de acceso por proximidad, etc.
- **Seguridad Física:** Revisión del perímetro, monitoreo, evaluación de controles de acceso, respuestas ante alarmas,

Para cada una de las fases expresadas anteriormente es necesario contemplar los siguientes aspectos para realizar las pruebas

- Revisión de postura
- Logística
- Detección y verificación activa
- Visibilidad auditable
- Verificación de acceso
- Verificación de confianza
- Verificación de controles
- Verificación de procesos
- Verificación de configuraciones
- Validación de la propiedad

- Revisión de segregación
- Verificación de exposición
- Competitividad scout inteligencia
- Garantizar la verificación
- Privilegios auditados
- Validación de supervivencia
- Revisión de alertas y logs

3.7 Seguridad Operacional

Para la medición de la superficie de ataque se requiere la valoración de los accesos, visibilidad y confianza, considerando los siguientes aspectos:

- Visibilidad: Contar el número de objetivos dentro del alcance.
- Accesos: Contar todos los puntos de acceso para cada interacción.
- Confianza: Contar cada punto de confianza para cada lugar de interacción.

Dentro de los controles aplicables de igual forma se deben contar los controles utilizados con los siguientes aspectos:

- Autenticación: Contar cada instancia de autenticación requerida para obtener acceso.
- Indemnización: Contar cada instancia que cubre las pérdidas referidas hacia los activos.
- Resistencia: Contar cada instancia de acceso en donde la falla de seguridad no proporcione un nuevo acceso.
- Subyugación: Contar cada punto de acceso con de la interacción deba cumplir las condiciones preestablecidas.
- Continuidad: Contar todos los puntos de acceso donde una falla no cause interrupción en la interacción.
- No repudio: Contar cada acceso que provea algún mecanismo que permita identificar que la interacción se realizó en un tiempo determinado.

- Confidencialidad: Contar cada instancia de acceso que provea mecanismos para evitar revelar información a terceros.
- Privacidad: Contar cada acceso en donde el método de interacción sea ocultado.
- Integridad: Contar cada acceso en donde durante interacción se pueda determinar si existió algún cambio en la información.
- Alarma: Contar cada acceso que genere un registro de un evento no autorizado o erróneo.

Como se observó en los apartados anteriores también existen limitaciones que no pueden ser controladas, para esto se realiza una evaluación de las limitaciones con los siguientes aspectos:

- Vulnerabilidad: Contar cada falla que pueda provocar un acceso no autorizado o denegar un acceso.
- Debilidad: Contar todas las fallas en los controles de interacción.
- Preocupación: Contar todas las fallas en los controles de proceso.
- Exposición: Contar cada acción no justificada que se derive de la visibilidad de los activos.
- Anomalía: Contar cada elemento desconocido que no pueda ser clasificado dentro de las operaciones normales.

Como podemos observar la seguridad operacional nos permite evaluar cada punto y llevar a ponderar estos valores para determinar el nivel de riesgo posible que pueda llevar a efectuarse en un activo.

Capítulo IV

OWASP Testing Guide

OWASP es una organización sin fines de lucro orientada a proporcionar recursos gratuitos a la comunidad para promover el desarrollo seguro de aplicaciones. Entre los recursos o materiales que proporciona OWASP están Proyectos Documentales (publicaciones, artículos, guías y normas) y Proyectos de Software (herramientas de pruebas y software); las cuales han sido desarrolladas durante varios años y están dirigidas a Desarrolladores, Testers de software, especialistas de seguridad de la información, auditores, estudiantes, entre otros, a fin de incentivar la gestión de riesgos en aplicaciones y servicios web, durante su desarrollo, adquisición o mantenimiento (Valencia, 2013).

4.1 Guía de Pruebas OWASP v4.0

El objetivo de esta guía es permitir a las instituciones probar sus aplicaciones web a fin de crear software confiable y seguro.

Si bien OWASP proporciona el consenso de expertos para ejecutar las pruebas con eficiencia y permite identificar las vulnerabilidades de aplicaciones web, estas guías deben ser adaptadas a las necesidades de seguridad de la empresa; tomando en consideración su cultura, tecnología y tamaño, no por prescripción (Vela, 2015).

4.2 Metodología de pruebas OWASP

La metodología OWASP de pruebas de intrusión se basa en un enfoque de caja negra es decir se conoce información limitada de la aplicación a ser evaluada, el modelo de pruebas consiste en:

- Evaluador: Persona que ejecutará las pruebas

- Herramientas y metodología: guía de pruebas OWASP
- Aplicación: la Caja Negra sobre la cual se ejecutarán las pruebas

Las pruebas de intrusión se dividen en dos fases que se encuentran en el la Figura 7, representados como Modo Activo y Pasivo, en las que cada una maneja diferentes pruebas que se encuentran sobre las fases especificadas.

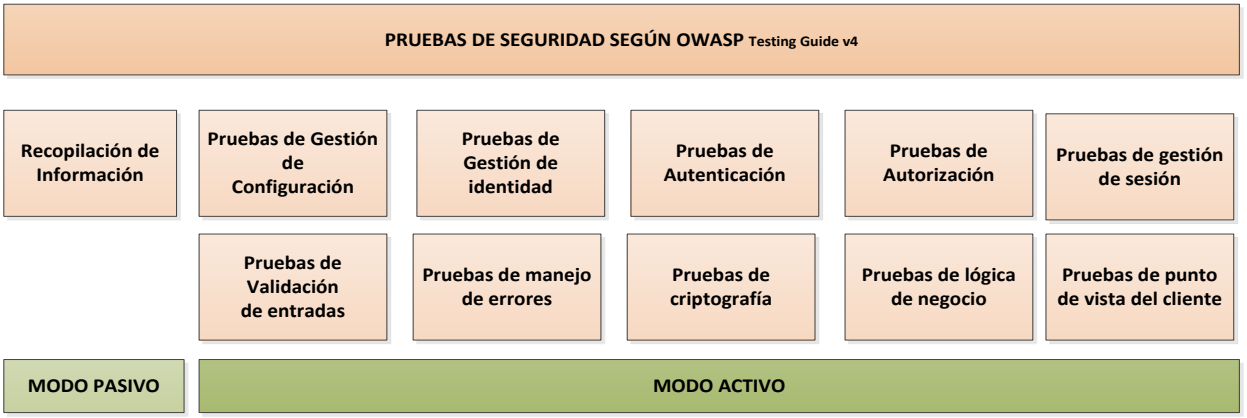

Figura 7. Fases de las Pruebas de Seguridad OWASP. Fuente: Autor

A continuación, se describen cada una de las fases en unas tablas descriptivas, con sus distintas Pruebas, Actividades, Herramientas y Remediaciones que se realiza a cada una.

4.2.1 Modo Pasivo

Acercamiento a la aplicación a través del cual se pretende conocer la funcionalidad y lógica de la aplicación a fin de recopilar la mayor cantidad de información (puntos de acceso, parámetros, etc.) que posteriormente deberán ser comprobados.

4.2.1.1 Recopilación de información

4.2.1.1.1 OTG-INFO-001

Nombre de la Prueba: "*Conducir el motor de búsqueda para el descubrimiento y reconocimiento de fugas de información".*
Descripción: Navegar y capturar recursos relacionados con la aplicación o la institución, información sensible del diseño o la configuración.

Pruebas: Emplear un motor de búsqueda mediante el cual se pueda obtener información referente a diagramas de red, configuraciones, mensajes archivados, procedimientos de inicio de sesión, formatos de nombres de usuario, contenidos de mensajes de error, entre otros.
Herramientas: Emplear buscadores como: Baidu, binsearch.info, Bing, Duck Duck Go, ixquick, Startpage, Google, Shoda, PunkSpider, entre otros.
Remediación: Previo a la publicación de información de diseño y configuración se deberá validar la sensibilidad de la misma, así como también efectuar revisiones periódicas de la información que se encuentra publicada.

4.2.1.1.2 OTG-INFO-002

Nombre de la Prueba: *"Huellas digitales servidor WEB"*
Descripción: Obtener información referente a la versión y el tipo de servidor a fin de determinar vulnerabilidades conocidas o cómo explotar las mismas.
Pruebas: Se envía al servidor WEB comandos específicos y se analiza la respuesta para compararla con la base de datos de firmas conocidas.
Herramientas: Emplear httprint, net-square.com, httprecon, computec.ch, Netcraft, netcraft.com, Desenmascarame, desenmascara.me, entre otras.
Remediación: Proteger la capa de presentación de los encabezados del servidor web.

4.2.1.1.3 OTG-INFO-003

Nombre de la Prueba: *"Revisión de los meta archivos del servidor web en busca de fugas de información"*
Descripción: Búsquedas de fuga de información de la aplicación web, mediante el uso de rastreadores web, se prueba el archivo robots.txt (recomendaciones de indexación).
Pruebas: Se emplean arañas, robots o rastreadores web a fin de recuperar una página web y atravesar sus hipervínculos para obtener otros contenidos.
Herramientas: Buscadores google, Yahoo, (emplear la función ver origen), curl wget, rockspider y similares.

4.2.1.1.4 OTG-INFO-004

Nombre de la Prueba: *"Enumere las aplicaciones en el servidor WEB"*
Descripción: Investigar que aplicaciones están alojadas en el servidor web a fin de detectar vulnerabilidades y explotarlas, debido a que las aplicaciones pueden estar desactualizadas o mal configuradas.
Pruebas: Identificar todas las aplicaciones accesibles asociadas a una dirección IP o nombre de dominio:
1. Navegar por el directorio, escaneo de vulnerabilidades para identificar URL's no estándar.
2. Escaneo de puertos (nmap) para identificar puertos no estándar.
3. Transferencia de zonas DNS, Consultas Inversas de DNS, Búsquedas DNS basadas en la web a fin de identificar Host virtuales.
Herramientas: nslookup, motores de búsqueda, nessus, nmap, entre otros.

4.2.1.1.5 OTG-INFO-005

Nombre de la Prueba: *"Revisión de los comentarios del sitio web y los metadatos en busca de fuga de información".*
Descripción: Búsqueda de comentarios y metadatos que hayan sido incluidos en el código HTML y puedan revelar información interna y que permita conocer más de cerca la aplicación. (Direcciones IP, usuarios, contraseñas, código, etc.)
Pruebas: Buscar comentarios en código HTML que comienzan con "" o meta-etiquetas. **Herramientas:** Wget, Browser función view source, Eyeballs.
Remediación: Revisar los comentarios y metadatos a fin de evitar fugas de información.

4.2.1.1.6 OTG-INFO-006

Nombre de la Prueba*: "Identificar puntos de entrada de la aplicación."*
Descripción: Comprender la aplicación, solicitudes HTTP, peticiones y respuestas, a fin de identificar vulnerabilidades.
Pruebas: Centrarse en los métodos GET y POST; Identificar los encabezados, parámetros y campos que exponen vulnerabilidades y anotarlos.
Herramientas: Utilizar interceptadores de proxy como: OWASP: Zed Attack Proxy (ZAP), OWASP: WebScarab, Burp Suite, CAT.

4.2.1.1.7 OTG-INFO-007

Nombre de la Prueba: *"Creación de mapas de rutas de ejecución a través de la aplicación."*
Descripción: Entender la estructura de la aplicación, comprender los flujos de trabajo y descripciones de las rutas descubiertas.
Pruebas: Documentar todos los enlaces descubiertos (manual o automáticamente).
Herramientas: ZAP Zed Attack Proxy, Spreadsheet software, Diagramming software

4.2.1.1.8 OTG-INFO-008

Nombre de la Prueba: *"Marco referencial para el uso de huellas digitales en aplicaciones WEB."*
Descripción: Identificar el tipo de Frameworks web utilizado (WAF, CMS) estructura del archivo, vulnerabilidades conocidas, errores de configuración y marcar con huellas digitales para las pruebas posteriores.
Pruebas: Identificar el framework en: código fuente, encabezados http, cookies, entre otros.
Herramientas: Se pueden utilizar netcat, WhatWeb, BlindElephant (checksum), Wappalyzer.
Remediación: Realizar cambios para ocultar las rutas del framework y que las mismas no puedan ser detectadas por análisis automático: eliminar etiquetas, comentarios innecesarios, retirar marcadores visuales, etc.

4.2.1.1.9 OTG-INFO-009

Nombre de la Prueba: "Huellas digitales aplicaciones WEB"
Descripción: Conocer los componentes de la aplicación web a fin de facilitar el proceso de las pruebas.
Pruebas: Encontrar información al revisar cookies, patrones en código fuente, estructura específica de archivos y carpetas en el servidor (propios para cada aplicación).
Herramientas: WhatWeb, BlindElephant, Wappalyzer.
Remediación: Realizar cambios para ocultar las rutas del framework y que las mismas no puedan ser detectadas por análisis automático: eliminar etiquetas, comentarios innecesarios, retirar marcadores visuales, etc.

4.2.1.1.10 OTG-INFO-010

Nombre de la Prueba: "Mapa de arquitectura de la aplicación".
Descripción: Revisión de la configuración, problemas de seguridad conocidos y crear un mapa de la red y de la arquitectura de la aplicación. Identificar los elementos de la infraestructura para entender cómo interactúan con una aplicación web y cómo ellos afectan a la seguridad.
Pruebas: Crear el mapa de la arquitectura de la aplicación, lo cual puede resultar difícil en caso de una prueba de penetración ciega, por lo tanto, se debe efectuar con base en supuestos de que existe una configuración simple (un servidor) y derivar otros elementos para ampliar el mapa de arquitectura
Herramientas: Pruebas de paquetes de red, ICMP, TRACE

4.2.2 Fase 2: Modo Activo

Ejecución de las pruebas de acuerdo a la metodología descrita, divididas en subcategorías de pruebas.

4.2.2.1 Pruebas para gestionar la configuración y la implementación

4.2.2.1.1 OTG-CONFIG-001

Nombre de la Prueba: ***"****Prueba de configuración Red/Infraestructura"*
Descripción: Una vez que se cuenta con un mapa de los elementos que conforman la infraestructura, se deberá revisar la configuración de los mismos y probarlos para identificar vulnerabilidades.
Pruebas: Las pruebas de penetración de caja negra para revisión de vulnerabilidades puede realizarse empleando herramientas automatizadas, pero no siempre será confiable, la mejor forma de realizar es cuando al evaluador se le proporciona información interna del software utilizado incluyendo versiones, actualizaciones y parches aplicados a fin de analizar el riesgo asociado al uso de estas versiones.
Remediación: Se debe comprender como interactúan los elementos de infraestructura, revisar los elementos para asegurarse que no contienen

vulnerabilidades, revisar sistemas de autenticación, puertos. Dar mantenimiento a herramientas administrativas.

4.2.2.1.2 OTG-CONFIG-002

Nombre de la Prueba: ***"Pruebas de la configuración de la plataforma de aplicaciones".***

Descripción: Identificar errores o configuraciones genéricas en elementos de la arquitectura de la aplicación.

Pruebas:

Caja negra

Revisión de archivos y directorios.

Revisión de comentarios que revelan información interna.

Caja gris.

Detectar errores comunes de configuración.

Revisión de la información sensible que contengan los registros.

Ubicación y almacenamiento de registros.

Protección del acceso a registros.

Remediación:

- Dar mantenimiento a la arquitectura de la aplicación.
- Evitar instalaciones y configuraciones por defecto.
- Habilitar únicamente módulos necesarios.

4.2.2.1.3 OTG-CONFIG-003

Nombre de la Prueba: ***"Prueba manejo de archivos de extensiones en busca información sensible".***

Descripción: Determinar a través de extensiones de archivo si se puede obtener información referente a diseño y tecnología de la aplicación web, manejo de peticiones, comportamiento del servidor web.

Pruebas:

- Navegación forzada, envío de solicitudes https de diferentes extensiones.
- Identificar archivos con una extensión en particular.

Herramientas: Escáneres de vulnerabilidades (Nessus y Nikto), Robots araña (spidering) y de reflejo, motores de búsqueda.

4.2.2.1.4 OTG-CONFIG-004

Nombre de la Prueba: ***"Revisión archivos viejos, copias de seguridad y archivos no referenciados para Información sensible".***

Descripción: Obtener información acerca de la infraestructura o las credenciales del servidor en archivos no referenciados u olvidados, archivos de copias de seguridad.

Pruebas:

- Deducir nombres y ubicación de las páginas no referenciadas (nombres comunes).
- Revisar manualmente el código fuente de archivos HTML para identificar pistas sobre otras páginas y su funcionalidad.

- Identificar extensiones de archivo en uso.

Herramientas: Herramienta de spidering, Nessus, Nikto para realizar verificaciones a directorios web que tienen nombres estándar o comunes.
Remediación: No exponer el código del lado del servidor, ya que esta puede exponer la lógica del negocio, información de la aplicación, e incluso usuarios y contraseñas. Evitar archivos viejos, copias de seguridad y archivos no referenciados.

4.2.2.1.5 OTG-CONFIG-005

Nombre de la Prueba: ***"Enumeración Infraestructura y de Interfaces de administración de aplicaciones".***
Descripción: Pruebas para descubrir si existen interfaces de administrador (actividades privilegiadas) que permiten el acceso a usuarios no autorizados.
Pruebas:

- Descubrir una interfaz administrativa.
- Pruebas para eludir la autenticación o intentar un ataque forzado.
- Forzar el contenido del servidor.

Herramientas: Dirbuster (forzar directorios y archivos en el servidor), THC-HYDRA (forzar interfaces), Diccionario de netsparker.
Remediación: Controles para proteger las interfaces administrativas de accesos no autorizados.

4.2.2.1.6 OTG-CONFIG-006

Nombre de la Prueba: ***"Prueba métodos HTTP".***
Descripción: Aprovechar los riesgos de seguridad de los métodos HTTP para modificar los archivos almacenados en el servidor web y robar las credenciales de usuarios legítimos
Pruebas:

- Descubrir los métodos HTTP soportados
- Aprovechar el método TRACE para robar credenciales de los usuarios.
- Pruebas de métodos HTTP.
- Pruebas para omitir el control de acceso HEAD.

Herramientas: NetCat, cURL.
Remediación: Verificar que el uso de métodos y servicios Web estén debidamente limitados a usuarios de confianza y condiciones de seguridad.

4.2.2.1.7 OTG-CONFIG-007

Nombre de la Prueba: ***"Prueba HTTP Strict Transport Security".***
Descripción: Verificar si el sitio web utiliza el encabezado HTTP, a fin de verificar si los datos viajan encriptados desde el navegador hacia el servidor.
Pruebas: Comprobar el encabezado HSTS en la respuesta del servidor en un proxy de intercepción a fin de identificar si la información es transferida a través de un canal sin codificar. Se puede explotar un ataque de man in the middle.

Remediación: El tráfico a ser intercambiado con un dominio debe siempre ser enviado mediante https, para proteger la información de que se envíe mediante peticiones no cifradas.

4.2.2.1.8 OTG-CONFIG-008

Nombre de la Prueba: ***"Prueba política de dominio cruzado RIA (Aplicaciones enriquecidas de internet)".***
Descripción: Acceso controlado de dominio cruzado para consumo de datos y servicios, se puede aprovechar deficiencias en la configuración de archivos de directivas que permitan acceder a datos sensibles del usuario o facilitar ataques.
Pruebas: Probar la debilidad del archivo de políticas RIA, recuperando los archivos de las políticas crossdomain.xml y clientaccesspolicy.xml., Aprovechar el uso de políticas débiles.
Herramientas: Nikto, OWASP Zed Attack Proxy Project, W3af.
Remediación: Evitar el uso de políticas de dominios cruzados excesivamente permisivas; las solicitudes únicamente deben permitir los dominios, puertos o protocolos necesarios.

4.2.2.2 Pruebas de Gestión de Identidad

4.2.2.2.1 OTG-IDENT-001

Nombre de la Prueba: ***"Prueba de Definición de Roles"***
Descripción: Probar la gestión de accesos a los objetos del sistema y los roles definidos dentro de la aplicación.
Pruebas: Elaborar una matriz de definición de roles, con permisos y restricciones.
Herramientas: Se realiza manualmente y puede emplearse como ayuda herramientas de spidering.
Remediación: Se debe gestionar un acceso adecuado a la información y funcionalidad del sistema.

4.2.2.2.2 OTG-IDENT-002

Nombre de la Prueba: ***"Prueba de Proceso de Registro del Usuario"***
Descripción: Probar la creación de acceso al sistema y registro de usuarios
Pruebas:

- Verifique que los requisitos de identidad de usuario (registro, roles, privilegios, etc.)
- Validar el proceso de registro.

Herramientas: Proxy HTTP
Remediación: Validar que el registro de usuarios y el proceso de registro esté alineado a las políticas de seguridad de la institución o mejores prácticas vigentes.

4.2.2.2.3 OTG-IDENT-003

Nombre de la Prueba: ***"Pruebas del Proceso de Creación de Cuentas"***

Descripción: Pruebas para crear cuentas válidas omitiendo el proceso de identificación y autorización.
Pruebas: Determinar los roles que están a disposición de los usuarios y el tipo de cuentas que se pueden crear. (Usuarios que permiten crear cuentas, autorizaciones para creación, eliminación).
Herramientas: Se puede efectuar manualmente y con apoyo de herramientas Proxy HTTP.
Remediación: Emplear procesos de autorización para crear cuentas.

4.2.2.2.4 OTG-IDENT-004

Nombre de la Prueba: ***"Pruebas de enumeración de cuentas y descubrimiento de cuentas de usuario"***
Descripción: Obtener nombres de usuarios válidos, a través de las respuestas del aplicativo para posteriormente obtener una contraseña a través de pruebas de fuerza bruta.
Pruebas: Análisis de mensajes de respuesta que revelan directa o indirectamente información útil para obtener usuarios. (Pruebas de búsqueda de contraseñas y usuarios válidos, pruebas con usuario válido y contraseña incorrecta, etc.)
Herramientas: Web Scarab, OWASP_WebScarab_Project, CURL: curl.haxx.se, PERL.
Remediación:

- Emplear mensajes que no revelen existencia de usuarios, mensajes de error genéricos.
- Eliminar cuentas de pruebas del sistema y cuentas por defecto.

4.2.2.2.5 OTG-IDENT-005

Nombre de la Prueba: ***"Pruebe las políticas de nombre de usuario débiles"***
Descripción: Determinar la estructura de nombres de cuenta de usuario para validar si es vulnerable a la enumeración de cuenta.
Pruebas: Evaluar:
Respuestas de la aplicación a nombres de cuentas válidos y no válidos.
Respuestas a nombres de cuentas válidos y no válidos.
Emplear diccionarios de nombre de cuenta para enumerar los nombres de cuenta válidos.
Remediación: Emplear mensajes de error genéricos al ingresar usuarios y contraseñas inválidos.

4.2.2.3 Pruebas de Autenticación

4.2.2.3.1 OTG-AUTHN-001

Nombre de la Prueba: ***"Pruebas del transporte de credenciales en un canal encriptado"***

Descripción: Se debe comprobar que los datos de autenticación del usuario se transfieren a través de un canal seguro o encriptado desde el navegador web hacia el servidor.
Pruebas: Emplear un proxy web para capturar los encabezados de los paquetes e inspeccionarlos para determinar cómo se envían y transmiten los paquetes.
Herramientas: WebScarab, OWASP Zed Attack Proxy (ZAP)
Remediación: Tráfico cifrado empleando algoritmo y claves para la aplicación robustas.

4.2.2.3.2 OTG-AUTHN-002

Nombre de la Prueba: ***"****Pruebas de las credenciales por defecto****"***
Descripción: Acceso a varios tipos de aplicaciones debido a configuraciones incorrectas, usuarios y contraseñas predefinidos.
Pruebas:

- Identificar interfaces de aplicaciones y probar en ellas usuarios y contraseñas por defecto (de fabricantes).
- Fijarse en los mensajes de error detallados.

Herramientas: Burp Intruder, THC Hydra, Brutus, Nikto 2.
Remediación:

- Evitar configuraciones por defecto en servidores, personalizar la administración del servidor.
- Personalizar las credenciales de fábrica de usuario.

4.2.2.3.3 OTG-AUTHN-003

Nombre de la Prueba: ***"****Pruebas para determinar un mecanismo de bloqueo débil****".***
Descripción:
Evaluar la capacidad de:

- Mecanismos de bloqueo de cuentas resistentes a ataques de fuerza bruta.
- Mecanismos de desbloqueo o liberación de cuentas.

Pruebas:

- Evaluar la capacidad del mecanismo de bloqueo de cuentas para mitigar forzado de contraseñas.
- Evaluar mecanismos de desbloqueo y sus debilidades.

Herramientas: Artículo OWASP sobre Ataques Forzosos.
Remediación: De acuerdo al nivel de riesgo emplear mecanismos de bloqueo fuertes.

4.2.2.3.4 OTG-AUTHN-004

Nombre de la Prueba: ***"****Pruebas para eludir el esquema de autenticación****"***
Descripción: Evaluar las seguridades adecuadas en métodos de autenticación y solicitudes.
Pruebas:

- Solicitud de página directa o navegación forzada (eludir esquema de autenticación)
- Modificación de parámetros (modificar parámetros de valor fijo)
- Predicción de sesión ID (indicadores de sesión previsibles)
- Inyección de SQL

Herramientas: WebScarab, WebGoat, OWASP Zed Attack Proxy (ZAP).
Remediación:

- Aplicar protocolos de encriptación fuertes.
- Validaciones de entrada de acuerdo a las mejores prácticas.
- Instalación de la aplicación sin configuraciones por defecto.

4.2.2.3.5 OTG-AUTHN-005

Nombre de la Prueba: ***"Pruebas de recordatorios de contraseñas vulnerables".***
Descripción: Contraseñas almacenadas en el navegador, las cuales un atacante puede obtener a través de ataque de Cross Site Scripting.
Pruebas:

- Examinar las cookies almacenadas por la aplicación en búsqueda de contraseñas.
- Evaluar la fuerza del mecanismo de hashing.
- Validar que las credenciales sean enviadas únicamente durante la fase de registro y no con cada solicitud a la aplicación.

Remediación: Evitar que las credenciales sean almacenadas en texto claro.

4.2.2.3.6 OTG-AUTHN-006

Nombre de la Prueba: ***"Pruebas para buscar la debilidad de memoria caché del navegador"***
Descripción: Evaluar si la aplicación ordena al navegador no almacenar datos sensibles previamente mostrados.
Pruebas: Validar que la historia y caché de navegador no pueda recuperarse o no tenga fugas de datos sensibles.
Herramientas: OWASP Zed Attack Proxy, Firefox add-on CacheViewer2.
Remediación:

- El servidor debe instruir al navegador para que cada página que contenga información confidencial, no almacene los datos en caché.
- Las páginas deben cumplir la directiva de no-cache.

4.2.2.3.7 OTG-AUTHN-007

Nombre de la Prueba: ***"Pruebas para determinar las políticas de contraseñas débiles"***
Descripción: Evaluar la resistencia de la aplicación a ataques de fuerza bruta o diccionarios de contraseñas.
Pruebas: Aplicación de políticas de gestión de contraseñas (caracteres, historial de contraseñas, caducidad de contraseña, entre otros).

Remediación: Controles de autenticación adicionales (dos factores) o emplear política de contraseñas fuertes.

4.2.2.3.8 OTG-AUTHN-008

Nombre de la Prueba: ***"****Pruebas para determinar la seguridad débil de pregunta/respuesta****"***
Descripción: Preguntas y respuestas secretas para restablecimiento de contraseñas o como seguridad adicional son de naturaleza simple y pueden llevar a respuestas inseguras.
Pruebas: Probar las debilidades de las preguntas previamente generadas o preguntas creadas por los usuarios, si estas pueden ser descubiertas, forzadas mediante ataque de fuerza bruta, disponible en las redes sociales, entre otros.
Remediación: De acuerdo a la criticidad de la aplicación: Determinar el número de preguntas que necesitan ser contestadas. Número de intentos para descubrir.

4.2.2.3.9 OTG-AUTHN-009

Nombre de la Prueba: ***"****Pruebas para determinar un cambio débil de contraseña o funciones de restablecimiento****".***
Descripción: Determinar las seguridades aplicadas para cambio de contraseña y restablecer la contraseña.
Pruebas: Pruebas de reinicio de contraseña, información necesaria para restablecer la contraseña, forma de comunicación de contraseñas restablecidas.
Remediación:

- Forma segura de presentar la contraseña después de restablecerla (correo electrónico).
- Obligar al usuario a cambiar inmediatamente su contraseña después de restablecerla.
- Contraseñas generadas aleatoriamente con algoritmos seguros.

4.2.2.3.10 OTG-AUTHN-010

Nombre de la Prueba: ***"****Pruebas para determinar la autenticación más débil en un canal alternativo****".***
Descripción: Identificar canales de autenticación alternos u otro tipo de canales que deberán ser evaluados y determinar si existen vulnerabilidades en ellos.
Pruebas: Entender canales de autenticación primarios y búsqueda de canales de autenticación alternos, determinar si en ambos canales se comparten cuentas de usuario.
Herramientas: Motores de búsqueda
Remediación: Emplear la misma política de autenticación en todos los canales, a fin de que sean igualmente seguros.

4.2.2.4 Pruebas de Autorización

4.2.2.4.1 OTG-AUTHZ-001

Nombre de la Prueba: ***"Probar la inclusión de archivos de directorio de circulación".***
Descripción: Emplear métodos de validación de entrada para leer o escribir archivos y ejecutar código o comandos del sistema.
Pruebas: Identificar en la aplicación campos donde permite agregar contenido por parte del usuario, codificación de rutas y ubicación de archivos.
Herramientas: Motores de búsqueda de código en línea. DotDotPwn, Path Traversal Fuzz Strings.
Remediación:

- Emplear mecanismos de autenticación para controlar el acceso a archivos y recursos.
- Emplear Listas de control de acceso para limitar el acceso a archivos sensibles.

4.2.2.4.2 OTG-AUTHZ-002

Nombre de la Prueba: ***"Pruebas para eludir el esquema de autorización"***
Descripción: Probar cómo funciona el esquema de autorización para acceder a funciones y recursos reservados.
Pruebas:

- Pruebas de acceso a funciones administrativas.
- Pruebas para intentar acceder a los recursos asignados a un rol diferente.

Herramientas: OWASP WebScarab Project, OWASP Zed Attack Proxy (ZAP).
Remediación: Verificar y reforzar el acceso a recursos.

4.2.2.4.3 OTG-AUTHZ-003

Nombre de la Prueba: ***"Pruebas para determinar el escalamiento de privilegios"***
Descripción: Verificar que un usuario no pueda modificar sus privilegios o roles dentro de la aplicación para ejecutar un escalamiento de privilegios.
Pruebas: Pruebas de la manipulación del rol/privilegio (acceder a funciones o privilegios no asignados).
Herramientas: OWASP WebScarab Project, OWASP Zed Attack Proxy (ZAP)

4.2.2.4.4 OTG-AUTHZ-004

Nombre de la Prueba: ***"Pruebas de las referencias de objetos directos inseguros"*.**
Descripción: Obtener acceso directo a recursos en el sistema como: registros de la base de datos o archivos en el sistema, omitiendo la autorización.
La aplicación toma la información ingresada por el usuario y la utiliza para recuperar un objeto sin realizar las comprobaciones de autorización suficientes.
Pruebas: El evaluador debe:

- Identificar las ubicaciones donde se ingresa información de usuario.
- Intentar modificar los parámetros utilizados para referenciar los objetos y evaluar si es posible recuperar objetos pertenecientes a otros usuarios u omitir la autorización; valor de un parámetro para recuperar un registro de la base de

datos, una operación en el sistema, archivo de recursos del sistema, acceder a la funcionalidad de la aplicación.

Herramientas: OWASP Top 10 2013-A4-Insecure Direct Object References
Remediación: Efectuar comprobaciones de autorización en la aplicación cuando se toma información ingresada por el usuario para recuperar objetos, formularios, etc.

4.2.2.5 Pruebas de Administración de sesión

4.2.2.5.1 OTG-SESS-001

Nombre de la Prueba: ***"Pruebas del esquema de gestión de sesión"***
Descripción: Se valida que las cookies usadas para mantener la sesión y las credenciales de la aplicación almacenadas no se puedan alterar para el robo de sesión.
Pruebas:

- Recolección de cookies
- Ingeniería inversa de cookies
- Manipulación de cookies

Herramientas: OWASP Zed Attack Proxy Project, Burp Secuencer, Foundstone Cookie Digger.

4.2.2.5.2 OTG-SESS-002

Nombre de la Prueba: ***"Pruebas de los atributos de las cookies"***
Descripción: Validar que los atributos utilizados dentro de las cookies están bien configurados
Pruebas: Validar existencia de Atributos: Security, HttpOnly, domain, path, expires.
Herramientas: OWASP Zed Attack Proxy Project, TamperIE (IE), Adam Judson (Firefox).
Remediación: El tiempo de expiración de la cookie debe ser corto.

4.2.2.5.3 OTG-SESS-003

Nombre de la Prueba: ***"Pruebas de Fijación de Sesión"***
Descripción: Validar que luego de la autenticación se renueven las cookies.
Pruebas: Copiar el identificador de sesión de la cookie y colocarlo en una nueva sesión.
Herramientas: Hijack, OWASP WebScarab.
Remediación: Renovación de ficha de sesión.

4.2.2.5.4 OTG-SESS-004

Nombre de la Prueba: ***"Pruebas para determinar la Exposición de las Variables de Sesión".***
Descripción: Validar que no se expongan las variables Cookie, SessionID, Hidden Field.
Pruebas:

- Validar la configuración de los proxys y cache.

- Observar el trafico GET y POST.

Herramientas: Proxy interceptor
Remediación:
Utilizar SSL
Expires: 0
Cache-Control: max-age=0

4.2.2.5.5 OTG-SESS-005

Nombre de la Prueba: ***"Pruebas de un CSRF"***
Descripción: Probar que se pueda forzar a ejecutar acciones por el usuario
Pruebas: Construir página HTML con la solicitud http a la url con los parámetros relevantes, verificar que el usuario está ingresado en el sistema, induzca al usuario a ingresar en la url definida en la página y verificar.
Herramientas: OWASP ZAP, CSRF Tester, Cross Site Requester.
Remediación:

- Usuario: Cerrar sesión luego de terminar de utilizar la aplicación, no almacenar las contraseñas en el navegador, no usar el mismo navegador.
- Desarrollador: Utilizar POST, utilizar el cierre automático de sesión.

4.2.2.5.6 OTG-SESS-006

Nombre de la Prueba: ***"Pruebas de la funcionalidad del cierre de sesión".***
Descripción: Verificar que la sesión fue cerrada exitosamente.
Pruebas:

- Probar el cierre de sesión desde el servidor, validar las cookies de sesión.
- Validar el tiempo de caducidad de la sesión.
- Probar el cierre de sesión del SSO

Herramientas: Burp Suite - Repeater
Remediación: Utilizar SSO (Single Sign On), botón de cierre de sesión en todas las páginas.

4.2.2.5.7 OTG-SESS-007

Nombre de la Prueba: ***"Pruebas del tiempo de cierre de sesión".***
Descripción: Validar el tiempo de cierre de sesión automático sin reutilizar la misma sesión.
Pruebas:

- Verificar el tiempo de valides de la sesión
- Verificar el tiempo de sesión en el servidor
- Verificar que el Servidor impide nuevas sesiones luego del cierre con el mismo ID de sesión.

Remediación: Borrar cookies en el navegar.

4.2.2.5.8 OTG-SESS-008

Nombre de la Prueba: ***"Pruebas de Session puzzling".***
Descripción: Validar la sobrecarga de variables de sesión, para que no se pueda esquivar los mecanismos de autenticación, elevar privilegios, manipular valores del lado del servidor.
Pruebas:

- Enumerar las variables
- Revisar el código fuente

Remediación: Variables de sesión solo se utilizan con una finalidad.

4.2.2.6 Pruebas de validación de entradas

4.2.2.6.1 OTG-INPVAL-001

Nombre de la Prueba: ***"Pruebas para la reflexión de Cross Site scripting (XSS)"***
Descripción: El atacante inyecta un código ejecutable en el navegador, mediante una sola solicitud y respuesta a fin de convencer a la víctima (ingeniería social) que ejecute el código ofensivo, utilizando el navegador de la víctima para: obtener claves, robar cookies, realizar robos del portapapeles y cambiar el contenido de la página (enlaces de descarga).
Pruebas:
1. Detectar las variables definidas por el usuario y cómo ingresarlas.
2. Efectuar pruebas para detectar vulnerabilidades en cada vector de entrada.
3. Analiza el resultado de las pruebas efectuadas determinan si representa una vulnerabilidad significativa, identifica los caracteres especiales que no han sido codificados correctamente.

Herramientas: Proxys web, OWASP CAL9000, PHP Charset Encoder(PCE), HackVertor, WebScarabHTTPS, XSS-Proxy ratproxy, Burp Proxy, OWASP Zed Attack Proxy, OWASP Xenotix XSS
Remediación:

- Correcta codificación de caracteres.
- Validar datos de entrada de solicitudes al cliente.

4.2.2.6.2 OTG-INPVAL-002

Nombre de la Prueba: ***"Pruebas de Cross Site Scripting almacenados"***
Descripción: Es el tipo Cross Site Scripting más peligroso, emplea al menos dos peticiones a la aplicación: La aplicación web almacena la información ingresada por un usuario malicioso y la información almacenada no se filtra correctamente, los datos maliciosos aparecen como parte del sitio web y se ejecutan en el navegador del usuario con los privilegios de la aplicación web, logrando: Secuestro del navegador de otro usuario, captura de información confidencial, pseudo desfiguración de la aplicación, escaneo de puertos en hosts internos, explotación basada en el navegador de entrega dirigida, entre otros.
Pruebas:

- Realizar las pruebas reflexión de Cross Site scripting (anterior), adicionalmente el evaluador debe investigar los canales a través de los cuales la aplicación recibe y almacena los datos ingresados por los usuarios.
- Probar todas las áreas de la aplicación que son accesibles por el administrador a fin de identificar información enviada por los usuarios.
- Comprobar si es posible cargar contenido HTML.

Herramientas: OWASP CAL9000, PHP Charset Encoder(PCE), Hackvertor, BeEF, XSS-Prox, Backframe.
Remediación: No permitir a los usuarios almacenar datos.

4.2.2.6.3 OTG-INPVAL-003

Nombre de la Prueba: ***"Pruebas de manipulación de solicitudes en HTTP"***
Descripción: Manipulación de solicitudes como métodos HTTP distintos, a fin de obtener información referente del contenido o funcionamiento de la aplicación web.
Pruebas: Emplear herramientas para crear solicitudes HTTP personalizadas para probar los otros métodos, o emplear pruebas de manipulación manual de verbos en HTTP.
1. Elaborar solicitudes HTTP personalizadas
2. Enviar solicitudes HTTP (netcat o telnet).
3. Analizar las respuestas HTTP (como respuesta de prueba exitosa el servidor ignora la solicitud completamente o devuelve un error.)
Herramientas: netcat, telnet.
Remediación: En el servidor de aplicaciones web, se debe proceder a deshabilitar las funciones que no sean GET o POST. Remediar acciones sin una apropiada autenticación para evitar que se revele información sobre el contenido o funcionamiento de la aplicación web.

4.2.2.6.4 OTG-INPVAL-004

Nombre de la Prueba: ***"Pruebas de contaminación de parámetros HTTP"***
Descripción: Al tener múltiples parámetros HTTP con el mismo nombre (parámetros duplicados), puede causar comportamientos anómalos en la aplicación ocasionando que la misma interpreta los valores de maneras imprevistas, las cuales facilitan a un atacante esquivar la validación de entrada, provocar errores de aplicación o modificar valores de variables internas.
Pruebas: Efectuar pruebas de parámetros HPP en componentes tanto del lado del cliente como del servidor, las pruebas manuales son confiables.
HPP del lado del servidor: identificar formularios o acciones que permitan ingresos suministrados por el usuario, probar vulnerabilidades editando los datos GET o POST, interceptando la solicitud, o cambiando la cadena de consulta después de que la página de respuesta se cargue.
HPP del lado del cliente: detectar vulnerabilidades de contaminación de parámetro que afectan a los componentes del lado del cliente, identificar formularios o acciones que permiten el ingreso de datos del usuario, a fin de contaminar cada parámetro HTTP y buscar ocurrencias de decodificación de la url desde la carga suministrada por el usuario.

Herramientas: OWASP ZAP HPP Passive/Active Scanners, HPP Finder.

4.2.2.6.5 OTG-INPVAL-005

Nombre de la Prueba: ***"Pruebas de Inyecciones de SQL".***

Descripción: Consiste en la inserción de una consulta SQL parcial o completa a través de los datos de entrada o transmitidos desde el navegador del cliente hacia la aplicación web, a fin de afectar la ejecución de instrucciones SQL predefinidas con el objetivo de leer o modificar datos sensibles de la base de datos, ejecutar operaciones administrativas de la base de datos, entre otros.

Pruebas:

- Detección: interacción con el servidor de base de datos (campos de entrada para la elaboración de consultas SQL).
- Pruebas de inyección SQL estándar.
- Huella digital en la base de datos.
- Técnicas de explotación
- Técnica de explotación: booleana, basada en el error, fuera de banda, de retraso de tiempo.
- Inyección de procedimientos almacenados
- Cada DBMS tiene características particulares como comandos especiales, funciones para recuperar datos como nombres de usuarios y bases de datos, funciones, líneas de comentarios, etc. Razón por la cual se deben revisar las guías de pruebas de las diferentes DBMS.

Herramientas: SQL Injection Fuzz Strings OWASP SQLiX, Oracle (SQLInjector, Orascan), MySQL (Francois Larouche: Multiple DBMS SQL injection, - Bernardo Damele A. G.: sqlmap, automatic SQL injection tool. - Muhaimin Dzulfakar: MySqloit, MySql Injection, SQL Server (Francois Larouche: Multiple DBMS SQL Injection tool icesurfer: SQL Server sqlninja.sourceforge.net - Bernardo Damele A. G.: sqlmap, automatic SQL injection), PostgreSQL (OWASP : "Testing for SQL Injection" OWASP : SQL Injection Prevention Cheat Sheet, PostgreSQL : "Official documentation", - Bernardo Damele and Daniele Bellucci: sqlmap, a blind SQL injection too), MS Access, Pruebas de inyección NoSQL (Bryan Sullivan from Adobe: "Server-Side JavaScript Injection":
media.blackhat.com - Bryan Sullivan from Adobe: "NoSQL, But Even Less Security": blogs.adobe.com - Erlend from Bekk Consulting: "[Security] NOS L-injection": erlend.oftedal.no - Felipe Aragon from Syhunt: "NoS L/SSJS Injection": syhunt.com)

4.2.2.6.6 OTG-INPVAL-006

Nombre de la Prueba: ***"Pruebas de inyección LDAP (Protocolo Ligero de Acceso de Directorios)"***

Descripción: Es un ataque de lado del servidor, la cual inyecta meta caracteres de filtros de búsqueda LDAP en consultas que serán ejecutadas por la aplicación, mediante la manipulación de parámetros de ingreso luego los pasa a las funciones de búsqueda interna, agregar y modificar logrando así que información sensible acerca de usuarios y hosts en una estructura LDAP pueda divulgarse, modificarse o insertarse, evadir restricciones de las aplicaciones, etc.

Pruebas: Probar filtros de búsqueda para verificar si la aplicación es vulnerable a una inyección LDAP, al mostrar atributos de los usuarios, dependiendo del flujo de ejecución de la aplicación y los permisos del LDAP del usuario conectado. Probar si una aplicación al Inicio de sesión utiliza LDAP para comprobar las credenciales de usuario, de esta manera es vulnerable a una inyección LDAP.
Herramientas: Softerra LDAP Browser

4.2.2.6.7 OTG-INPVAL-007

Nombre de la Prueba: ***"Pruebas de inyección de ORM (Mapeo Relacional de Objetos)"***
Descripción: ORM es una herramienta de mapeo relacional de objetos empleada para acelerar el desarrollo orientado a objetos dentro de la capa de acceso de datos de las aplicaciones de software, sin embargo, puede ser vulnerable a ataques de inyección SQL si los métodos permiten parámetros de entrada sin desinfectar.
Pruebas: Idénticas a las pruebas de inyección de SQL.
Herramientas: Hibernate, NHibernate
Remediación: Validar correctamente los parámetros de ingreso.

4.2.2.6.8 OTG-INPVAL-008

Nombre de la Prueba: ***"Pruebas de inyección de XML"***
Descripción: Al existir fallas en la validación de datos, un atacante podría intentar inyectar un documento XML a la aplicación, obteniendo resultados positivos. Las pruebas deben iniciar con el método de descubrimiento, una vez que se conoce información sobre la estructura se debe intentar insertar meta caracteres, inyectar datos y etiquetas XML.
Pruebas:

- Intentar insertar meta caracteres XML (comilla simple, comilla doble, paréntesis angular, etiqueta de comentario, Ampersand, Delimitadores de sección, etc.)
- Inyección de etiqueta una vez que se conoce la estructura del documento XML, se debe intentar inyectar datos XML y etiquetas.

Herramientas: XML Injection Fuzz Strings

4.2.2.6.9 OTG-INPVAL-009

Nombre de la Prueba: ***"Pruebas de inyección SSI (Server-Side includes)"***
Descripción: Probar si es posible inyectar código en los datos de la aplicación que serán interpretados por mecanismos del SSI. Una explotación exitosa de esta vulnerabilidad permite a un atacante inyectar código en páginas HTML, obtener el contenido de los archivos o incluso realizar ejecución remota de códigos.
Pruebas:

- Identificar si el servidor web admite directivas SSI.
- Verificar si SSI es compatible (contiene archivos .shtml).
- Determinar si es posible un ataque de inyección SSI e identificar los puntos de entrada que podemos utilizar para inyectar el código malicioso.

- Probar vulnerabilidades de inyección de código detalladas en puntos anteriores.

Herramientas: Web Proxy Burp Suite: portswigger.net, Paros: parosproxy.org, OWASP WebScarab, String searcher: grep: gnu.org.

4.2.2.6.10 OTG-INPVAL-0010

Nombre de la Prueba: ***"Pruebas de inyección XPath"***

Descripción: Las bases de datos XML utilizan XPath como su lenguaje de consulta estándar. Se debe probar si es posible inyectar la sintaxis de XPath en una solicitud permitiendo a un atacante eludir mecanismos de autenticación o información sin la debida autorización.

Pruebas: Los ataques de inyección XPath siguen la misma lógica que los ataques de inyección SQL, la aplicación autenticará al usuario incluso si no ha proporcionado un nombre de usuario o una contraseña.

Herramientas: Amit Klein "Blind XPath Injection" packetstorm.foofus.com, XPath 1.0 specifications.

4.2.2.6.11 OTG-INPVAL-0011

Nombre de la Prueba: ***"Pruebas de inyección de IMAP/SMTP"***

Descripción:

- Afecta a las aplicaciones que se comunican con servidores de correo (IMAP/SMTP).
- Se ejecutan pruebas para inyectar comandos IMAP/SMTP arbitrarios en los servidores de correo a fin de: Explotar vulnerabilidades en el protocolo IMAP/SMTP, evadir restricciones de la aplicación, evadir el proceso anti-automatización, fugas de información y relevo/SPAM.

Pruebas: Los pasos a seguir son:

Identificación de los parámetros vulnerables (enviar solicitudes falsas o maliciosas al servidor y analizar las respuestas).

Entendimiento del flujo de información y estructura de despliegue del cliente (determinar el nivel de inyección que es posible y luego diseñar un plan de prueba para explotar la aplicación).

Inyección de comandos IMAP/SMTP al explotar la funcionalidad, existen dos posibles resultados:

1. La inyección es posible en un estado no autenticado (no requiere autenticar al usuario).
2. La inyección sólo es posible en un estado autenticado (requiere que el usuario esté plenamente autenticado antes de que la prueba pueda continuar.

Herramientas: Inyección arbitraria de comandos IMAP/SMTP.

4.2.2.6.12 OTG-INPVAL-0012

Nombre de la Prueba: ***"Pruebas de inyección de código"***

Descripción: Se debe probar enviando información que es procesada por el servidor web como código dinámico o como un archivo incluido, para validar si es posible introducir código en una página web y ejecutarlo con el servidor web.

Pruebas:

Pruebas de caja negra: Vulnerabilidades de inyección PHP, utilizar la cadena de consulta para inyectar códigos.
Pruebas de caja gris: Examinar el código ASP en búsqueda de información ingresada por el usuario en funciones de ejecución, con el objetivo de probar las vulnerabilidades de inyección ASP.
Pruebas para determinar la inclusión de documentos locales (permite al atacante incluir un documento en la aplicación objetivo).
Pruebas para la inclusión remota de archivos (incluir un archivo empleando un mecanismo de "inclusión dinámica de archivos").
Herramientas: Security Focus, Insecure.org, Reviewing Code for OS Injection.
Remediación: Se deben emplear validación de la entrada y prácticas de codificación seguras, validación de datos ingresados por el usuario.

4.2.2.6.13 OTG-INPVAL-0013

Nombre de la Prueba: ***"Pruebas de inyección de comandos"***
Descripción: Pruebas para inyectar en la aplicación un comando de sistema operativo a través de una solicitud HTTP. Se realiza a través de una interfaz web con el objetivo de subir programas maliciosos o incluso obtener contraseñas.
Pruebas: En la URL se muestra el nombre del archivo, se puede probar añadiendo el símbolo Pipe "|" en el final del nombre del archivo, o añadiendo un punto y coma al final de una URL; Si la aplicación no valida la solicitud se puede recuperar la documentación y efectuar con éxito un ataque de inyección de OS.
Herramientas: OWASP ZAP
Remediación: Aplicar seguridades en el diseño y desarrollo de las aplicaciones. Desinfectar los formularios de datos y la URL de caracteres no válidos.

4.2.2.6.14 OTG-INPVAL-0014

Nombre de la Prueba: ***"Pruebe la saturación del buffer (desbordamiento de memoria)"***
Descripción: Pruebas para exceder el uso de cantidad de memoria asignada por el sistema operativo, los cuales permiten ejecutar código arbitrario en un equipo a fin de tomar control del equipo víctima o ejecutar ataques.
Pruebas:
Pruebas de saturación de Heap (heap es un segmento de memoria): Comprueba si se puede provocar una saturación del heap que explota un segmento de memoria. Las pruebas se realizan introduciendo cadenas de entrada que sean más largas de lo esperado.
Probar la saturación de pila de datos: copiar datos de tamaño variable en búferes en la pila de datos del programa sin ninguna comprobación de los límites; Se prueba suministrando datos demasiado grandes en comparación con los esperados, así como también inspeccionar el flujo de ejecución de la aplicación y las respuestas para determinar si una saturación se ha disparado realmente o no.
Pruebas para la cadena de formato: utilización de datos ingresados por el usuario para bloquear un programa o ejecutar un código dañino; se prueba proporcionando especificadores del tipo de formato al ingresar a la aplicación.

Herramientas:
OWASP sobre vulnerabilidades y ataques de saturación de buffer.
Pruebas de saturación de Heap OllyDbg - Spike - Brute Force Binary Tester (BFB) - Metasploit.
Probar la saturación de pila de datos OllyDbg - Spike - Brute Force Binary Tester (BFB)- Metasploit.
Pruebas para la cadena de formato ITS4

4.2.2.6.15 OTG-INPVAL-0015

Nombre de la Prueba: ***"Pruebas de las vulnerabilidades incubadas"***
Descripción: Conocida como ataques persistentes, para su funcionamiento se requiere de una vulnerabilidad de validación de datos. Es un ataque basado en el lado del cliente que afecta a todos los usuarios que navegan por el sitio.
Pruebas: Cubriendo varios puntos de ataque como son:
- Componentes de carga de archivos
- Cross-site scripting
- Inyección SQL/XPATH
- Servidores mal configurados

Herramientas: XSS-proxy: sourceforge.net, Paros parosproxy.org, Burp Suite portswigger.net, Metasploit metasploit.com.

4.2.2.6.16 OTG-INPVAL-0016

Nombre de la Prueba: ***"Pruebas para verificar la separación/contrabando de HTTP"***
Descripción: Pruebas para aprovechar las características del protocolo HTTP; Explotación de las debilidades de la aplicación web, ataques dirigidos a encabezados HTTP. Usan parte de los datos ingresados por el usuario para generar los valores de algunos encabezados de sus respuestas.
Pruebas:
Separación HTTP (HTTP splitting)
Aprovecha que los datos ingresados no han sido desinfectados por lo cual permiten a un intruso insertar caracteres en los encabezados de la respuesta de la aplicación, así como también separar la respuesta en dos mensajes HTTP diferentes. Su objetivo de ataque es desde el envenenamiento del caché hasta un cross site scripting.
Contrabando HTTP (HTTP smuggling)
Se aprovecha que algunos mensajes HTTP especialmente diseñados pueden ser analizados e interpretados de diferentes maneras según el agente que los recibe (prueba de Caja Gris).

4.2.2.7 Pruebas de manejo de errores

4.2.2.7.1 OTG-ERR-001

Nombre de la Prueba: ***"Pruebas de errores de código"***
Descripción: Esta prueba se enfoca en encontrar y analizar los errores de código generados por aplicaciones o servidores web durante las pruebas de penetración, son

útiles ya que revelan información sobre bases de datos, errores y otros componentes tecnológicos directamente relacionados con aplicaciones web.
Pruebas: Evaluar los mensajes de error detallados a fin de encontrar en la misma información referente a errores de base de datos, de servidor de aplicaciones, de servidor web. Los mensajes de error a ser analizados pueden ser referentes a problemas de red, errores de autenticación, entre otros.
Herramientas: ErrorMint, ZAP Proxy.
Remediación:
Manejo de errores en IIS y ASP .net: Configurar el servidor web para suprimir los mensajes de error detallados para el usuario y crear mensajes personalizados.
Manejo de errores usando web.config: emplear mode=RemoteOnly para mostrar errores personalizados a los usuarios de la aplicación web remota.
Manejo de errores en Global.asax: escribir código para el re direccionamiento del manejo de las páginas de error.
Manejo de errores en Apache: Usando la Directiva ErrorDocument [2] se puede personalizar y configurar las respuestas a los errores.
Manejo de errores en Tomcat: se puede personalizar y configurar los errores en el documento de configuración web.xml.

4.2.2.7.2 OTG-ERR-002

Nombre de la Prueba: ***"Pruebas para determinar los rastros de pila de datos"***
Descripción: Los rastros de pila de datos revelan información que puede ser útil para los atacantes para la alteración del ingreso a la aplicación web con peticiones HTTP.
Pruebas:

- Entrada no válida (no concuerda con la lógica de la aplicación).
- Entradas que contienen caracteres no alfanuméricos.
- Entradas vacías.
- Entradas extensas.
- Acceso a páginas internas sin autenticación.
- Evitar el flujo de la aplicación.

Herramientas: ZAP Proxy
Remediación: Emplear herramientas como OWASP ZAP y Burp proxy para detectar excepciones en las respuestas.

4.2.2.8 Pruebas para criptografía débil

4.2.2.8.1 OTG-CRYPST-001

Nombre de la Prueba: ***"Pruebas de codificadores SSL/TLS débiles, protección de transporte de capas insuficiente".***
Descripción: Protección de los datos sensibles cuando estos se transmiten a través de la red.
Pruebas:
Prueba de transmisión de datos sensibles (verificar si la información se transmite a través de HTTP en lugar de HTTPS).

Prueba de vulnerabilidades de SSL/TLS Ciphers/Protocolos/Claves (No deben utilizarse cifrados débiles, los protocolos débiles deben desactivarse, la renegociación debe estar configurada correctamente, La longitud de claves de los certificados deben ser fuertes, entre otros.)
Prueba de la validez de los certificados SSL - de clientes y servidores (actualizar el navegador, validar los certificados utilizados por la aplicación).
Herramientas: Qualys SSL Labs - SSL Server Tes, Tenable - Nessus Vulnerability Scanner, TestSSLServer, sslyze: github.com, SSLAudit, SSLScan, nmap.
Remediación:

- Configurar correctamente los protocolos SSL/TLS.
- El certificado debe estar validado.
- Actualizar periódicamente el software a fin de minimizar vulnerabilidades.

4.2.2.8.2 OTG-CRYPST-002

Nombre de la Prueba: ***"****Prueba del Padding Oracle (Relleno de Oracle*)***"**
Descripción: Padding Oracle (decodifica datos encriptados que proporciona el cliente), permite a un atacante descifrar datos encriptados y cifrar datos arbitrarios sin conocer la clave utilizada para estas operaciones criptográficas.
Pruebas: Se debe identificar los puntos de entrada posibles para los padding Oracle: si los datos están codificados y si se utiliza un cifrado de bloque.
El código debe cumplir con las siguientes condiciones:
1. Verificar la integridad del texto cifrado empleando un mecanismo seguro como HMAC.
2. Manejo de los estados de error uniformemente durante la decodificación y el posterior procesamiento.
Herramientas: PadBuster: github.com, python-paddingoracle, Poracle: github.com, Padding Oracle, Exploitation Tool (POET).

4.2.2.8.3 OTG-CRYPST-003

Nombre de la Prueba: ***"****Pruebas para el envío de información sensible por canales sin encriptar****".***
Descripción: Los mecanismos de protección o cifrado utilizados para transmitir datos seguros no deben tener limitaciones o vulnerabilidades.
Pruebas: Verificar como se efectúa la transmisión de la información, si esta se transmite a través de HTTP en lugar de HTTPS o si emplean mecanismos de cifrado débil.
Herramientas: curl

4.2.2.9 Pruebas de lógica de negocio

4.2.2.9.1 OTG-BUSLOGIC-001

Nombre de la Prueba: ***"****Prueba de validación de datos de la lógica del negocio****"***
Descripción: Se debe verificar que se introduzcan datos válidos en los campos solicitados.

Pruebas: Validar las entradas de datos del sistema o entre aplicativos.
Herramientas: HP Business Process Testing Software, OWASP ZAP, Burp Proxy, Paros Proxy, Web Development Toolbar extention.
Remediación: Los datos que se ingresen deben estar previamente validados, así como también los que se interrelacionan con otros sistemas

4.2.2.9.2 OTG-BUSLOGIC-002

Nombre de la Prueba: ***"Pruebas de habilidad para manipular consultas".***
Descripción: Verificar que la aplicación no permita manipular los datos a los se tiene acceso.
Pruebas:

- Verificar si existen campos que puedan ser predecibles u ocultos
- Insertar los datos validos en el sistema

Herramientas: OWASP Zed Attack Proxy
Remediación: No permitir la depuración desde la aplicación.

4.2.2.9.3 OTG-BUSLOGIC-003

Nombre de la Prueba: ***"Pruebas de comprobación de integridad".***
Descripción: Verificar que no se puedan alterar los datos en cualquier parte del sistema.
Pruebas:

- Capturar el trafico http en busqueda de campos ocultos.
- Buscar lugar para insertar información que sea editable

Herramientas: OWASP Zed Attack Proxy.
Remediación:

- No permitir el ingreso directo a las bases de datos del sistema.
- No permitir el ingreso de información sin validar.

4.2.2.9.4 OTG-BUSLOGIC-004

Nombre de la Prueba: ***"Pruebas del tiempo de procesamiento".***
Descripción: Validar que el sistema no determine su comportamiento basado en los tiempos de procesamiento de entrada y salida.
Pruebas: Validar que la funcionalidad no se vea afectada por tiempos predecibles de ejecución.
Remediación: Agregar pasos adicionales que no permitan determinar el tiempo de ejecución o que escondan el tiempo de ejecución de alguna validación o proceso.

4.2.2.9.5 OTG-BUSLOGIC-005

Nombre de la Prueba: ***"Pruebas del número de veces que limita el uso de una función".***
Descripción: Validar que el sistema no permita utilizar aplicaciones o funciones, más veces de lo requerido.

Pruebas: Validar que la funcionalidad de las aplicaciones o funciones internas se las realice un número determinado veces.
Remediación: Especificar la cantidad de veces que se puede realizar un proceso o determinada tarea y que la misma no pueda exceder el número de veces permitidas.

4.2.2.9.6 OTG-BUSLOGIC-006

Nombre de la Prueba: ***"Pruebas para la evasión de los flujos de trabajo"***
Descripción: Validar que la aplicación no permita saltarse flujos de trabajo para otras funciones o tareas
Pruebas: Validar que no se pueda saltar procedimientos que deben seguir una secuencia definida o que puedan ser llamados desde otra función.
Remediación: Establecer controles a los flujos de la información de tal manera que no se puedan acceder en distinto orden.

4.2.2.9.7 OTG-BUSLOGIC-007

Nombre de la Prueba: ***"Pruebas de defensa contra el mal uso de la aplicación"***
Descripción: Validar que la aplicación no permita manipularse a sí misma de una forma no deseada.
Pruebas:

- Rechazar los ingresos que contiene ciertos caracteres
- Bloquear una cuenta luego de una serie de fallos
- Numero grande de uso de funcionalidad
- Recepción de datos estructurados sin formato válido

Remediación: La aplicación debe determinar comportamientos anómalos que no representan una actividad normal en la aplicación.

4.2.2.9.8 OTG-BUSLOGIC-008

Nombre de la Prueba: ***"Pruebas de posibilidad de carga de tipos de archivos inesperados"***
Descripción: Evaluar que la aplicación no permita el ingreso de archivos que el sistema no espera.
Pruebas:

- Verificar la carga de archivos no válidos.
- Verificar si se rechazan los archivos.
- Verificar que cuando se suben varios archivos cada uno se valide

Remediación:

- Se debe permitir en la aplicación la carga de formatos de archivos específicos
- Se deben validar cada uno de ellos en la carga
- Se debe rechazar la carga si no son validos

4.2.2.9.9 OTG-BUSLOGIC-009

Nombre de la Prueba: ***"Pruebas de la posibilidad de carga de archivos maliciosos"***

Descripción: Validar que la aplicación no permita la carga de archivos maliciosos como malware y exploits.
Pruebas:

- Validar que el archivo se subió correctamente
- Cargar archivos con extensión cambiada al sistema

Herramientas: Metasploit's payload, Intercepting proxy.
Remediación: Validar que existan en Content-Type la extensión del archivo a cargar y se especifique en listas negras los archivos que no se pueden cargar al sistema.

4.2.2.10 Pruebas en el lado del cliente

4.2.2.10.1 OTG-CLIENT-001

Nombre de la Prueba: ***"Prueba de Cross Site Scripting basado en DOM"***
Descripción: Validar que no se modifique el contenido activo de un script definido en la aplicación.
Pruebas:

- Rastrear las instancias de ejecución de Javascript
- Insertar el código XSS
- Modificar el código activo por un Javascript

4.2.2.10.2 OTG-CLIENT-002

Nombre de la Prueba: ***"Pruebas de ejecución de JavaScript"***
Descripción: Determinar que se pueda inyectar código Javascript dentro de la aplicación.
Pruebas: Validar las entradas y salidas del usuario.
Remediación: Validar adecuadamente las entradas y salidas existentes en los JavaScript.

4.2.2.10.3 OTG-CLIENT-003

Nombre de la Prueba: ***"Prueba de inyección HTML"***
Descripción: Validar que se pueda inyectar código HTML en la página web.
Pruebas: Validar que permita que una entrada no validada se utilice para crear HTML.
Remediación: Proteger con entradas debidamente validadas.

4.2.2.10.4 OTG-CLIENT-004

Nombre de la Prueba: ***"Prueba de re direccionamiento de la URL del lado del cliente"***
Descripción: Validar que no se puedan realizar re direccionamiento de URL desde la aplicación del lado del cliente.
Pruebas:

- Verificar que se pueda re direccionar de URL.
- Validar que no exista código de redirección en Javascript window.location

Herramientas: DOMinator
Remediación: Utilizar variables absolutas de URL.

4.2.2.10.5 OTG-CLIENT-005

Nombre de la Prueba: ***"Pruebas de inyección de XSS"***
Descripción: Validar que no se pueda inyectar código arbitrario para el robo de información.
Pruebas:

- Analizar el código Javascript
- Insertar el XSS
- Observar el comportamiento

Remediación: Utilizar la función jQuery, con la lista blanca de caracteres permitidos en el contexto

4.2.2.10.6 OTG-CLIENT-006

Nombre de la Prueba: ***"Pruebas de manipulación de recursos del lado del cliente"***
Descripción: Consiste en validar que la aplicación acepta las entradas controladas por el usuario hacia un recurso.
Pruebas:

- Evaluar que utilice las entradas no validadas.
- Controlar la dirección URL con una solicitud.

Herramientas: DOMinator
Remediación: Utilizar variables absolutas de URL

4.2.2.10.7 OTG-CLIENT-007

Nombre de la Prueba: ***"Prueba para el intercambio de recursos de origen cruzado"***
Descripción: Valida que un navegador web, permita realizar peticiones de dominio cruzado.
Pruebas:

- Verificar el encabezado origen y los dominios permitidos.
- Revisión del JavaScript.

Herramientas: OWASP Zed Attack
Remediación:

- Restringir los dominios que se utilizan para que no exista el origen cruzado
- Configurar bien el método Access-Control-Request y Access-Control-Allow

4.2.2.10.8 OTG-CLIENT-008

Nombre de la Prueba: ***"Prueba de Cross Site flashing"***
Descripción: Validar que estén incrustadas XSS en las aplicaciones Flash de la pagina
Pruebas:

- Encontrar un SWF defectuoso
- De compilar el SWF e insertar
- Identificar un punto de entrada dentro del SWF

Herramientas: Adobe SWF Investigator, SWFScan, SWFIntruder, Decompiler – Flare.

Remediación: Utilizar Frame Busting en el cliente y X-Frame-Options en el servidor.

4.2.2.10.9 OTG-CLIENT-009

Nombre de la Prueba: ***"Prueba de Clickjacking"***
Descripción: Validar que el usuario interactúe de forma diferente en la página.
Pruebas:

- Validar si en la página destino se puede cargar un iframe
- Validar si se puede realizar un enmarcado doble

Herramientas: Contexto de Seguridad de la información
Remediación:

- Utilizar encabezado X-Frame-Options
- Utilizar los filtros XSS en las máquinas de usuario final
- Definir el parámetro segurito="restricto" en el iframe del código

4.2.2.10.10 OTG-CLIENT-0010

Nombre de la Prueba: ***"Prueba de los WebSockets"***
Descripción: Validar que el usuario interactúe de forma diferente en la página Web.
Pruebas: Capturar el WebSocket y enviar la información para capturar las solicitudes y respuestas.
Herramientas: OWASP Zed Attack, Client WebSocket, Client WebSocket Google Chrome Simple.

4.2.2.10.11 OTG-CLIENT-0011

Nombre de la Prueba: ***"Prueba de mensajería Web"***
Descripción: Validar que las aplicaciones de varios dominios se ejecuten de manera segura.
Pruebas: Validar que el acceso permita que se realice por dominio.
Herramientas: OWASP Zed Attack
Remediación: Definir controles de seguridad para permitir control sobre los dominios de mensajería establecidos.

4.2.2.10.12 OTG-CLIENT-0012

Nombre de la Prueba: ***"***Prueba de almacenamiento Local***"***
Descripción: Validar la información almacenada en el almacenamiento local y en el almacenamiento web.
Pruebas:

- Validar que lo guardado dentro del almacenamiento local no sean datos sensibles
- Validar que los datos almacenados no sean susceptibles de ataques de XSS

Herramientas: Firebug, Google Chrome Developer Tools, OWASP Zed Attack Proxy.
Remediación:

- No almacenar la información sensible en el almacenamiento local.

- Cerrar las sesiones para que el almacenamiento local guardado ya no sea válido en la siguiente sesión.

Capítulo V

Características de Seguridad de las Instituciones Financieras

En el presente capítulo vamos a entender las características que tienen las instituciones financieras y sus diferencias con las demás organizaciones, los entes reguladores existentes y sus aspectos normativos en cuanto a la legislación en el Ecuador. Esto nos permitirá el conocer las características a las que debe enfocarse la metodología que se propone en el siguiente capítulo.

5.1 Definición de Institución Financiera.

Las instituciones financieras están definidas como empresas legalmente constituidas las cuales brindan servicios financieros, a sus clientes o miembros, dentro de estos servicios principalmente se encuentran los préstamos de dinero por el que reciben un valor de interés producto de la operación.

Dentro de estas instituciones están Bancos, Cooperativas de Ahorro y Crédito y Mutualistas.

- Banco: Institución financiera autorizada por la Superintendencia de Bancos y Seguros, que se encarga de recibir dinero (captación) y otorgar créditos (colocación), formada de un grupo de inversionistas o accionistas.

- Cooperativa de Ahorro y Crédito: Institución financiera autorizada por la Superintendencia de Economía Popular y Solidaria, que de igual forma se encarga de brindar servicios financieros, como créditos, inversiones, ahorros y demás en el cual, su patrimonio está dividido entre todos los socios de la cooperativa de forma igualitaria y no intervienen accionistas que tengan algún

beneficio por la cantidad de dinero aportado, siendo su pilar fundamental el beneficio social a la comunidad.

- Mutualista: Son Instituciones financieras privadas, controladas por las Superintendencia de Economía Popular y Solidaria, con el objetivo principal de captar recurso para el financiamiento de planes de vivienda, construcción y bienestar familiar, para sus clientes.

5.2 Característica de las Instituciones Financieras.

Las instituciones financieras están compuestas generalmente de los siguientes sistemas de información que no disponen otro tipo de empresas, algunas de las cuales tienen los siguientes sistemas principales (Junta Bancaria del Ecuador, 2014):

- Sistema de Core Bancario o Core Transaccional: Sistema en el cual se maneja la información de socios, cuentas, créditos, y demás productos que brinde la institución.
- Sistema de Cajeros Automáticos: Sistema gestor de ATM's, monitorea estado de subsistemas, transacciones, cuadres, sistemas anti skimming.
- Sistema de Banca en Línea: Sistema transaccional para realizar gestión de la cuenta, pagos a terceros, transferencias, inversiones, créditos, etc. mediante el uso del Internet.
- Sistema de Banca Móvil: Sistema que permite realizar las mismas tareas que el sistema de Banca en Línea, a través del celular o dispositivos móviles.
- Sistema de Puntos de Venta o POS: Sistema gestor de dispositivos de cobro mediante tarjetas inteligentes o con chip en establecimientos afiliados a la institución bancaria.

Adicionalmente existen otros sistemas internos que cada institución podría o no utilizar en lo que se encuentran los siguientes y se los denomina sistemas de apoyo y gestión:

- Sistemas de toma de decisiones.
- Sistemas de gestión de resultados,
- Sistemas de mesa de ayuda.
- Sistemas de control venta de cobranza.
- Sistemas de Mercadeo y publicidad.

Estos sistemas complementarios son considerados de apoyo al negocio y dependen del tamaño de la institución para el uso de cada uno de estos.

Como se observó en los conceptos de cada una de las principales instituciones financieras, están regidas por dos entes principales, según el mercado al que van dirigidos como:

- **Superintendencia de Bancos y Seguros del Ecuador:** Es un organismo técnico, con autonomía administrativa, económica y financiera, cuyo objetivo principal es vigilar y controlar con transparencia y eficacia a las instituciones del sistema financiero, de seguro privado y de seguro social, a fin de que las actividades económicas y los servicios que prestan se sujeten a la ley y atiendan al interés general.
- **Superintendencia de Economía Popular y Solidaria:** Es una entidad técnica de supervisión y control de las organizaciones de la economía popular y solidaria, con personalidad jurídica de derecho público y autonomía administrativa y financiera, que busca el desarrollo estabilidad, solidez y correcto funcionamiento del sector económico popular y solidario.

5.3 Regulaciones y Normativas

La Superintendencia de Bancos y Seguros, como ente regulador de las instituciones financieras (Bancos) dentro del país, tiene definido la normativa JB-2014-3066 en la que se hace referencia a los mecanismos de seguridad para el manejo del riesgo operativo que se aplican a canales electrónicos, cajeros automáticos, Banca

electrónica, Banca Móvil, POS y PIN Pad, Sistemas IVR y corresponsales no bancarios, etc., que se encuentra en el apartado 4.3 sobre Tecnología de la información, este marco fue de igual forma acogido por la SEPS en el cambio de administración de las cooperativas y mutualistas hacia esta última.

A continuación, se hace mención al apartado indicado en la normativa.

> *"4.3.5.1 Las instituciones del sistema financiero deberán adoptar e implementar los estándares y buenas prácticas internacionales de seguridad vigentes a nivel mundial para el uso y manejo de canales electrónicos y consumos con tarjetas, los cuales deben ser permanentemente monitoreados para asegurar su cumplimiento;"*

A continuación, veremos un breve resumen de los puntos citados en esta normativa vigente y de estricto cumplimiento de las instituciones regidas por esta entidad.

5.3.1 Cajeros Automáticos:

- Encriptar la información ingresada y no almacenar las claves.
- Los cajeros deben procesar tarjetas inteligentes o con chip.
- Deben ser instalados bajo las normas del fabricante y normas de seguridad de la institución...
- Permitir ejecutar procedimientos de auditoría de seguridad (Análisis de vulnerabilidad y PEN TEST o Hacking Ético) por lo menos una vez al año.
- Autenticación de dos factores como mínimo.
- Educar a los usuarios sobre medidas de seguridad para el uso adecuado de los mismos.
- Validar la autorización de los dispositivos para su funcionamiento en el sistema gestor de los mismos.

5.3.2 Puntos de Venta (POS y PIN Pad)

- Validar que el personal que realice los mantenimientos, instalación y desinstalación de estos dispositivos, en los locales afiliados esté debidamente autorizado.
- Las comunicaciones de estos con el sistema gestor debe ser de forma inalámbrica y a través de un canal seguro.
- Deben ser capaces de procesar tarjetas inteligentes o con chip.

5.3.3 Banca Electrónica

- Utilizar algoritmos de seguridad alta, certificados digitales, encriptación de la información.
- Realizar una vez al año una auditoría de seguridad aplicando estándares vigentes.
- Implementar mecanismos de seguridad anti phishing.
- Utilización de equipos de seguridad FW, IDS Antivirus, Antimalware, etc.
- Establecer tiempos de inactividad del sistema para cancelación de sesión.
- Notificaciones vía correo, mensaje de texto de ingreso al sistema.
- Las credenciales usadas no deben ser números de cédula y deben manejar contraseñas complejas para el acceso.
- Utilizar mecanismos de autenticación como mínimo de dos factores.
- Matricular los dispositivos desde los cuales se accede al sistema, Computadores y números de celular.
- Control de montos de transacciones permitidas.

5.3.4 Banca móvil

- Implementar mecanismos de seguridad ante phishing.
- Utilización de equipos de seguridad FW, IDS Antivirus, Antimalware, etc.
- Establecer tiempos de inactividad del sistema para cancelación de sesión.

- Notificaciones vía correo, mensaje de texto de ingreso al sistema.
- Las credenciales usadas no deben ser números de cedula y deben manejar contraseñas complejas para el acceso.
- Utilizar mecanismos de autenticación como mínimo de dos factores.

5.3.5 Sistemas de Interacción de respuesta de VOZ (IVR).

- Utilizar los mismos mecanismos de los puntos anteriores.

5.3.6 Canales Electrónicos

- Monitoreo periódico de los niveles de seguridad de todos los equipos, sistemas y demás que intervengan.
- Uso de canales de comunicación seguros (VPN, HTTPS, SSH, L2TP, etc)
- Enmascarar la información de los números de tarjetas que se muestre en los diferentes sistemas gestores.
- Usar encriptación de la información.
- Usar una solución antimalware, actualizada activa y funcional.
- Sistema de manejo, generación y validación de claves.
- Sistema de alertamiento de eventos de dudosa procedencia o fraudes.
- Renovar las claves por lo menos 1 vez al año de cada sistema y cada una diferente de la otra.
- Uso de sistemas que permitan determinar comportamientos inusuales en los usuarios para prevenir fraudes, robos, etc.
- Hacer uso de un servidor de tiempo (NTP) configurado en todos los equipos, servidores, dispositivos de red, etc.
- Almacenar mínimo 1 año la información de las transacciones realizadas en los diferentes sistemas.

- Disponer de un sistema de Call Center para realizar reclamos, bloqueos, consultas de emergencias bancarias, por perdidas, robos, clonaciones y demás.
- Mantener las grabaciones de llamadas telefónicas al Call Center como mínimo de seis meses.
- Notificar mediante mensajes de texto, correo electrónico u otros sobre el acceso y la ejecución de las transacciones en cualquier canal electrónico.
- Capacitar permanentemente a los clientes sobre las seguridades de los canales electrónicos.
- Aplicar seguridad en el desarrollo de los sistemas que impliquen canales electrónicos.
- Enmascarar la clave de acceso.

El conocimiento del presente capítulo nos servirá como guía para determinar los requerimientos necesarios de una institución financiera, para desarrollar la metodología que se la verá a continuación.

Capítulo VI

Definición de la Metodología de Hacking Ético para Instituciones Financieras

Luego de haber revisado los conocimientos sobre las metodologías de hacking ético, las diferentes características que tienen las instituciones financieras en el Ecuador, y sus respectivas normativas vigentes y de obligada ejecución, se va a definir la metodología aplicable de una forma práctica en el presente capítulo.

Observábamos que es de obligatoriedad la ejecución de un escaneo de vulnerabilidades por lo menos una vez al año, sin embargo, eso no asegura ante posibles ataques, y nos representaría una falsa sensación de seguridad (antipatrón), adicionalmente a este tipo de herramientas para asegurar la infraestructura tecnológica, es necesario la intervención de terceros, como las consultorías de seguridad, servicios gestionados que se los puede realizar para los siguientes puntos:

- Monitoreo y descubrimiento eventos de seguridad a través de Herramientas como SIEM (Correlación de Eventos).
- Monitoreo de páginas web para prevención de Phishing de sistemas de Banca en Línea o de Página Web Institucional u otros.
- Monitoreo de servicios de correo, para que no se pueda emitir correos a nivel institucional desde otras direcciones ips públicas y CNAME diferentes que no esté publicadas en el DNS.
- Servicios de respaldo de información y cifrado, entre otros.
- Antivirus, Malware

El uso de las herramientas antes mencionadas ayuda a estar prevenido ante ataques vigentes que se presente dentro de la infraestructura, pero sin embargo estas son de carácter reactivo.

6.1 Aspectos de generales

Para la definición de la metodología es importante basarnos en los puntos principales que se observaron en las metodologías anteriores, como:

CEH: Esta metodología está orientada 100% en la práctica y la obtención de la información de una forma paulatina vulnerando cada vez más cada nivel de seguridad, en la obtención de la información crítica para el negocio, las fases son:

- Reconocimiento,
- Descubrimiento,
- Obtención de Acceso,
- Mantener el acceso, y
- Limpiar el rastro.

OSSTMM: Esta metodología explica, cómo se debe evaluar cada uno de las denominadas dimensiones en las que se encuentra la información, determinar los niveles de seguridad o controles que tiene, partiendo de las políticas definidas para el manejo de las características que tiene la información ya especificadas en los capítulos anteriores y validar quien interviene en el manejo de la misma para determinar en cada interacción sobre los diferentes canales, sean estos Humanos, Físicos, Inalámbricos, Telecomunicaciones y las Redes de datos, los controles que existen y las limitaciones de esos controles que se pueden aplicar.

Como se pudo observar esta se encarga de auditar a nivel muy completo todos los accesos, exposiciones, propiedad y segregación, entre otras características del manejo, distribución de la información en una organización y cuantificar el nivel de riesgo al que se encuentra expuesta (Valencia, 2013).

OWASP: Esta metodología nos ayuda a determinar el nivel de seguridad existente en los sistemas web que actualmente abarcan casi todos los ambientes que se utilizan en una institución financiera, por lo que es de gran importancia principalmente para los

ambientes como: Core Bancario, Banca en Línea, Sistemas de Tarjetas de Crédito, Sistemas de Cajeros Automáticos, entre los principales y a los que la normativa está solicitando su manejo adecuado de seguridad para estos sistemas transaccionales críticos.

6.2 Justificación de la metodología

El propósito de la metodología es el buscar un mecanismo efectivo que permita contemplar los sistemas principales utilizados en las instituciones financieras del Ecuador, y su infraestructura tecnológica sobre la cual se encuentra implementada, de modo que permita determinar el grado de seguridad que existente en un determinado tiempo. La metodología presentada en el presente documento se denomina como **"Evaluación de Seguridad para Instituciones Financieras del Ecuador" ESIFE** por su abreviatura en la que se establecen los siguientes mecanismos a realizar para las pruebas generales.

6.2.1 Pruebas Generales

Basado en la exposición o la visibilidad de los diferentes sistemas hacia el público y al usuario interno se determina que se deben realizar las siguientes pruebas.

- Caja Negra: Sin conocimiento de ningún tipo de información con antelación llamado prueba a ciegas. En este tipo de prueba se utiliza la información que está públicamente expuesta como página web, sistemas de banca en línea, aplicación móvil, sistemas de correo electrónico, información de personal que labora en la institución, obtenida de diferentes medios como redes sociales o profesionales, como: Facebook, Linked-in, Google+, Twitter, entre otros y

también en páginas de bolsas de trabajo en donde se brindan contactos de la empresa para el envío de hojas de vida y otros documentos importantes.

- Caja Gris: Para esta prueba se tiene un conocimiento previo de los diferentes sistemas que se utilizan, sin embargo, no se dispone de información como contraseñas de acceso, puertos de comunicación, etc. Para este tipo de prueba ya se utilizan mecanismos de búsqueda de vulnerabilidades para encontrar los eslabones débiles por donde se obtendrá el acceso a los sistemas a vulnerar.
- Caja Blanca: En esta se dispone de un panorama conocido como credenciales y permisos de acceso a los diferentes sistemas, y el objetivo es el probar que las aplicaciones internamente se encuentren seguras, y determinar que en su funcionalidad no presentan vulnerabilidades que sean aprovechadas. Para estas pruebas se hace énfasis en las aplicaciones web, como el Core Bancario, Sistemas de Gestión de Tarjetas de Débito y Crédito, Gestión de Cajeros Automáticos y Banca Móvil.

6.3 Fases

Para lograr realizar estos análisis de cajas negras, caja gris y caja blanca se va a utilizar las fases siguientes dentro de la evaluación de seguridad.

Figura 8. Fases de la metodología (ESIFE). Fuente: Autor.

6.3.1 Fase 1. Reconocimiento:

Esta fase se encarga principalmente en obtener la mayor cantidad de información necesaria sobre direccionamiento, servidores, direcciones de correo, dominios y

plataformas disponibles para tener una idea clara sobre el ambiente que se va a atacar.

Se realizan pruebas de caja negra principalmente, ya que no se tiene ninguna información con la que se puede trabajar sobre el objetivo. Se realiza principalmente de forma externa. Algunas de las herramientas utilizadas para esto son:

- Utilizar Google Hacking para determinar la mayor cantidad de información disponible que se encuentre de forma pública. (página web, servicios que brinda, representantes de la empresa, direcciones de correo, etc).
- Utilizar la herramienta Web Data Extractor para obtener la mayor cantidad posible de información de contactos disponibles en la página web.
- Determinar los dominios utilizados por el objetivo, que se encuentran vigentes.
- Buscar a las personas utilizando la herramienta pipl.com
- Buscar personas que utilicen las direcciones de correo institucionales en redes sociales.
- Utilizar la Herramienta SmarWhois, para determinar la información disponible a nivel del DNS.

6.3.2 Fase 2. Descubrimiento:

Se trata de determinar en base a lo conocido en la fase I la información correspondiente a las redes, equipos de red, servidores, firewall, y poder esquematizar el diagrama de red que se dispone en la entidad financiera para conocer exactamente los equipos importantes por donde se puede comenzar el ataque y cuáles son los puntos de seguridad a los que se piensa evadir o enfrentar para acceder y tener un mayor conocimiento de lo existente. Estas pruebas utilizan técnicas de caja negra y gris determinando los equipos existentes dentro de la red.

Adicionalmente también se realiza un escaneo de vulnerabilidades, puertos y servicios en los equipos a fin de determinar la mayor cantidad de información correspondiente del objetivo.

- MegaPing: Herramienta que nos permite buscar los hosts activos de la red definida, ya sea por rango o por ips específicas.
- Nmap: Identificar también los equipos que se encuentran en la red y los puertos que se encuentran abiertos por cada uno.
- Network Topology Mapper: Realiza un escaneo de la red y lo esquematiza en un diagrama.
- Nessus: Escaneador de vulnerabilidades
- OpenVas: Escaneador de vulnerabilidades

6.3.3 Fase 3. Obtención de Acceso:

Aprovechando el conocimiento obtenido en la fase de Identificación se procede a explotar las vulnerabilidades para obtener el acceso a los diferentes sistemas, o equipos y poder escalar privilegios para obtener la mayor cantidad de información, en la cantidad de tiempo definida dentro del alcance.

Dentro de las principales herramientas se utiliza el Metasploit que es una suite de software que contiene bases de datos de exploits para obtener principalmente acceso o abrir una shell para ejecutar remotamente comandos y tomar control del sistema operativo. Esta fase utiliza pruebas de caja gris y blanca ya que se requiere abrir servicios vulnerables para la obtención de acceso y poder acceder con otras credenciales obtenidas dentro del sistema, hasta obtener la denominada cuenta Administrador del equipo, Administrador del dominio o root dependiendo del sistema operativo.

- Metasploit
- XSS

6.3.4 Fase 4. Mantener Acceso

Es importante mantener las sesiones establecidas a los diferentes sistemas y aplicaciones vulneradas para extender la búsqueda de información que sea sensible, en bases de datos, sistemas transaccionales y archivos para determinar.

Para esto se pueden crear nuevos usuarios con los que se pueda acceder sin ningún problema a los servidores y continuar con el procedimiento de extracción de la información, con roles de administrador.

Estas fases se pueden realizar mediante la Tabla 4 en donde se indican los pasos utilizados en cada una de ellas.

Tabla 4. Tareas de cada fase Metodología ESIFE. Fuente: Autor

Fase Reconocimiento
1. Utilizar herramientas de google hacking para encontrar información sobre la institución.
2. Buscar el dominio y las direcciones IP utilizadas por la institución publicadas en DNS.
3. Determinar registros que brinden mayor información en el DNS (PTR, TXT, SPF, etc.).
4. Buscar cuentas de correos (pippl.com)
5. Obtener información de la paginas web institucional.

Fase de Descubrimiento
1. Realizar escaneos de red a las direcciones IP públicas, y a redes internas para determinar los equipos activos en la red
2. Descubrir equipos de red como FW router y demás para esquematizar un diagrama de red.
3. Realizar escaneos de puertos y servicios en las redes activas
4. Realizar escaneo de vulnerabilidades a los equipos encontrados.

Fase Obtener Acceso
1. Utilizar contraseñas por defecto de aplicaciones, equipos de red, configuraciones para ingresar a los equipos.
2. Utilizar metasploit para vulnerar los sistemas y obtener acceso.
3. Utilizar puertos abiertos para aprovechar agujeros de seguridad y obtener acceso
4. Cambiar de usuarios, robar sesiones
5. Obtener el acceso de administrador o root en el sistema operativo
6. Buscar información importante en archivos de configuración de las aplicaciones

Fase de Mantener Acceso
1. Crear usuarios con privilegios

2. Otorgar los accesos necesarios
3. Continuar con la búsqueda de información sensible

6.4 Pruebas OWASP a los Sistemas Transaccionales

Se realizan estas pruebas específicas para los ambientes transaccionales, debido a su criticidad para el negocio, en cuanto a operaciones diarias y gestión de los diferentes recursos como ATM's, Banca en Línea y demás que están implementados sobre plataformas web.

Esta metodología abarca muchos aspectos para la evaluación de la seguridad, sin embargo, muchos de estos ya son considerados dentro de las 4 fases vistas en la sección anterior, por lo que se pretende con la presente metodología el determinar ciertos aspectos como autorización, identificación, autenticación, criptografía, manejo de sesiones y validaciones de entrada que se especifican a continuación.

- Pruebas de Gestión de Identidad: Permiten determinar los mecanismos de identificación utilizados en el sistema, definición de roles y privilegios, canales seguros, contraseñas seguras, etc.
- Pruebas de Autenticación: Se establecen las pruebas para controlar el proceso de autenticación al sistema, como cantidad de contraseñas recordadas, credenciales por defecto, etc.
- Pruebas de Autorización: Estas pruebas permiten controlar el nivel de escalamiento de privilegios, mecanismos para eludir el modo de autorización.
- Pruebas de administración de sesión: Permite determinar si la sesión establece de una forma correcta.
- Pruebas de validación de entrada: Esta prueba es fundamental para prevenir la inyección de código SQL o malicioso que permite obtener información del

sistema, como bases de usuarios, contraseñas, cuentas, tarjetas de crédito, etc. dentro de los campos de ingreso en el sistema.

- Pruebas de manejo de errores: Consiste en evaluar que los errores que muestran información sobre las aplicaciones y bases de datos que se utilizan.
- Pruebas de criptografía débil: Permiten evaluar si el sistema utiliza protocolos seguros para el envío y recepción de la información a través de la red y con el uso de certificados digitales.

Dentro de la metodología propuesta **no** se evalúa las siguientes pruebas:

- Prueba de lógica de negocio. No se utiliza debido a que la prueba se basa en observar el nivel de seguridad de la misma, y no consiste en determinar los procesos internos del funcionamiento del sistema, como apertura de cuentas, acreditaciones, cobros, créditos, pagos que entraría como un proceso de certificación del software.
- Pruebas de configuración y recopilación de la información. Estas se realizan en las fases 2 y 3 de la metodología.

Dentro de las pruebas que se van a realizar, se contemplan las siguientes:

Pruebas de Gestión de identidad:

- Pruebas de enumeración de cuentas y descubrimiento de usuarios.

Pruebas de Autenticación:

- Prueba del transporte de credenciales a través del canal encriptado.
- Prueba para determinar un mecanismo débil de autenticación.
- Pruebas para determinar las políticas de contraseñas débiles.
- Pruebas para determinar un cambio débil de contraseña y el mecanismo de restablecimiento.

Pruebas de Autorización.

- Pruebas para eludir el esquema de autorización.

Pruebas de Administración de sesión.

- Pruebas de funcionalidad de cierre de sesión.
- Pruebas del tiempo de cierre de sesión.

Pruebas de validación de entrada.

- Pruebas para la reflexión de Cross Site Scripting.
- Pruebas de inyección de SQL.
- Pruebas de inyección LDAP.
- Pruebas de inyección de XML.
- Pruebas de inyección de código.
- Pruebas de inyección de comandos.

Pruebas de manejo de errores.

- Pruebas de errores de código.
- Pruebas para determinar los rastros de pila de datos.

Pruebas de criptografía débil.

- Pruebas de codificadores SSL/TLS débiles.
- Prueba de Padding Oracle.
- Pruebas para el envío de información sensible por canales sin encriptar.

Cada una de estas pruebas nos ayudará a determinar efectivamente las validaciones a estos sistemas montados sobre plataformas web considerando los

puntos más vulnerables a los que se encuentran esos ambientes expuestos y que son sujetos de ataques tanto interna como externamente, correspondiente a su función.

6.5 Elaboración del informe

Toda prueba realizada, debe ser evidenciada mediante un informe preparado de dos formas. Principalmente uno el que pueda ser presentado al área de TI encargada de la administración de la infraestructura tecnológica y de la seguridad de la información con un detalle más técnico de los sistemas, ips, bases de datos, evidencias, etc.

El formato para la elaboración del informe para el área de TI debe contener lo siguiente:

1. Carátula.
2. Acuerdo de Confidencialidad y Descargo.
3. Alcance y Definición de las pruebas realizadas.
4. Descripción de las Herramientas utilizadas.
5. Pruebas de Reconocimiento.
6. Pruebas de Descubrimiento.
7. Pruebas Obtención de Acceso.
8. Pruebas para Mantener Acceso.
9. Evaluación de Sistemas Transaccionales.
10. Resultados.
11. Conclusiones.
12. Glosario de términos.

El segundo que sería un informe con información simplificada de forma simple para la presentación a la gerencia, que no necesariamente conocen sobre terminología

informática, y se pueda mostrar de una forma fácil de entender para los altos mandos los resultados obtenidos sobre el nivel de seguridad que existe en la institución financiera.

El informe para la alta gerencia debe contener las siguientes secciones:

1. Carátula.
2. Acuerdo de Confidencialidad y Descargo.
3. Alcance y Definición de las pruebas realizadas.
4. Presentación de resultados por fase.
5. Conclusiones.
6. Anexos (de ser necesario).
7. Glosario de términos.

6.6 Caso Práctico.

Luego de realizar la definición de la metodología que se va a realizar a las instituciones financieras, la prueba de cómo realizarla se lo muestra a continuación con la aplicación a un caso práctico, indicando las herramientas utilizadas y los métodos que se siguen en cada fase descrita en el apartado anterior.

La descripción del caso práctico se encuentra en el apartado de Anexos, del presente documento.

6.7 Resultados.

Dentro de los aspectos que se observaron en la elaboración del caso práctico de la institución financiera, se verifico que los pasos que se siguen en la metodología parten de lo desconocido a lo conocido siguiendo una secuencia para determinar los niveles de accesos que existen, para explotarlos e ingresar a los sistemas.

Esta permitió determinar el nivel de seguridad actual de la institución financiera desde diferentes perspectivas.

1. Perspectiva Externa. Determinando el nivel de exposición de la información, se determinaron direcciones de correos externos publicados en redes sociales como Linked-in, usando direcciones de correos institucionales, se determinaron encargados de la administración de los sitios web institucionales y del dominio. Adicionalmente se encontraron diferentes direcciones públicas utilizadas para otros servicios como correos electrónicos, servidores ftps, sistemas de facturación electrónica.
2. Perspectiva Interna. Al desarrollar el caso práctico sobre una institución puntual se tuvo acceso a los ambientes de desarrollo de sistemas como Banca en Línea, ATM's, Gestión de Tarjetas de Débito y Core Bancario. Estos sistemas se encuentran mayoritariamente desarrollados como ambientes WEB salvo el caso del sistema de Gestión de Cajeros Automáticos. Se determinaron versiones tanto de sistemas operativos y servidores de aplicaciones utilizados que se encontraban sin actualización de parches de seguridad requeridos o con medidas compensatorias como reglas de firewall establecidas. Los sistemas transaccionales no utilizan protocolos seguros de transmisión por lo que fueron susceptibles de captura de información, como el usuario y la contraseña. Si bien se tenían medidas compensatorias el sistema Core Bancario en el que se tiene definición de terminales específicas por direcciones IP por las que unicamente se puede acceder, equipo con el que se realizó las pruebas no tenía el acceso requerido para el ingreso en la prueba de autenticación.

Se determinaron además contraseñas por defecto de los servidores de aplicaciones con los que se pudo acceder mediante metasploit con un payload a un Shell con privilegios de root en los servidores Linux.

El sistema de Banca en línea utiliza tecnología obsoleta que no tiene parches de actualización por lo que es muy vulnerable a ataques.

Estas perspectivas nos demuestran que son necesarias para contemplar su situación actual en cuanto a seguridad y demuestran que la metodología utilizada abarca correctamente los niveles de exposición a los que se encuentran una institución financiera.

Los resultados obtenidos a detalle en cada fase de la evaluación se encuentran en el Anexo B dentro del informe técnico y de forma resumida en el informe gerencial del mismo apartado.

Podemos determinar varias ventajas existentes en el uso de la metodología ESIFE que son las siguientes:

1. Las fases permiten determinar el entorno completo de exposición.
2. Las pruebas específicas a los ambientes críticos permiten evaluar su nivel de aseguramiento de la información y transmisión de esta para el uso tanto de forma interna como externa.
3. Se determinan las tecnologías existentes y sus vulnerabilidades.
4. Permite determinar anomalías en configuraciones antes de la puesta en producción de diferentes sistemas a fin de que se realice validaciones que prevengan mayores inconvenientes.
5. Se realiza de forma progresiva identificando la información paso a paso en los diferentes niveles hasta alcanzar a lo requerido.

6. Cumple a cabalidad con la reglamentación establecida por las entidades de control de las instituciones financieras en las que se incluyen los sistemas críticos del negocio.
7. Aplica para cualquier tipo de institución financiera, sea esta un banco, cooperativa o mutualista.

Conclusiones

Dentro del mercado existen diferentes metodologías para realizar una evaluación de seguridad a diferentes instituciones, sin embargo, el universo de aplicaciones difiere para cada tipo de institución, sean estas del gobierno, educacionales, financieras, de retail, manufactura, etc.

En la investigación realizada, varias metodologías tienen diferentes puntos de vista en cuanto a la extensión de su proceso de evaluación, siendo unos muy completos evaluando todo lo vinculado a la información, el manejo, proceso, accesos, políticas, medios de transmisión, etc., muy específicos de su plataforma, y de acuerdo a las perspectivas del atacante.

Con este trabajo se presenta una metodología aplicable a instituciones financieras en el Ecuador lo cual orienta a este sector importante con alta demanda de ataques por hackers informáticos en los últimos años, ya que han sido el foco de atención por los malos controles en cuanto a ciberseguridad y cultura a nivel de usuario en el País.

La metodología definida ESIFE determina todos los lugares dentro de la infraestructura tecnológica en los que se expone la información tanto pública como privada enfocando principalmente a las aplicaciones críticas del negocio como Core Bancario, Banca en Línea, Sistema de Cajeros Automáticos, Sistemas de Tarjetas de Crédito para plasmar el nivel de seguridad al que se encuentra expuesto ante posibles ataques; siguiendo de forma evolutiva el proceso para acceder a la información sensible y partiendo de la perspectiva de un atacante.

Esta metodología cumple con las normativas establecidas de las entidades gubernamentales de control en el país y su aplicación dependerá del grado en el que se requiera determinar la seguridad de sus sistemas críticos operacionales o extenderlos a sistemas de gestión utilizados por la alta gerencia para optimizar sus procesos y toma de decisiones.

La metodología ofrece varias ventajas que aplican directamente a las instituciones financieras en las que se puede realizar el proceso en un tiempo menor al que se lo desarrolla con metodologías como OSSTMM o la aplicación para certificaciones como PCI-DSS en el caso de ser requerido por la institución.

Si bien en la metodología se especifican las fases a seguir, depende de los conocimientos y experiencia del evaluador para utilizar diferentes herramientas ya sean estas de paga o gratuitas y estas podrán ayudar a comparar y validar resultados que se puede catalogar como falsos positivos o confirmar los hallazgos encontrados durante el ejercicio.

Esta metodología no define un manual paso a paso en el uso de herramientas y equipos para desarrollar la evaluación, sino un marco de referencia de las fases y procesos que se deben desarrollar para obtener el estado en un tiempo determinado sobre la ciberseguridad de la institución financiera a la que se aplique.

ESIFE se puede aplicar durante un período 5 años a partir de la entrega de este documento, debido a que dentro de las normativas y políticas del país se actualizan de forma continua, pero se mantiene la estructura haciendo énfasis a los nuevos sistemas informáticos que vayan ingresando para su uso en el mercado financiero, si se mantiene la tendencia en crecimiento de la migración a aplicaciones web se mantendría el esquema definido.

La esencia de la aplicación de la metodología es el obtener y concientizar los niveles de seguridad que se deben tomar en cuenta en los diferentes sistemas de gestión utilizados, fomentando el establecimiento de normativas y políticas que fortalezcan los activos de información de forma que se pueda garantizar las características fundamentales de la información (confidencialidad, integridad y disponibilidad) y manejar niveles de madurez sobre ciberseguridad en la cultura financiera.

Trabajos Futuros:

Como buena práctica siempre se debe evaluar el comportamiento de la metodología a nivel de encuestas o entrevistas con las áreas técnicas de seguridad y tecnología de la información para validar el nivel de aceptación, de calidad, eficiencia y resultados que se obtuvieron en su aplicación para determinar las mejoras que se puede realizar dentro de su estructura definida en el presente documento, tratando así de gestionar las mejoras aplicables para que se extienda su uso de forma general en el sector financiero y en el país y extender su campo de acción a más servicios financieros existentes a nivel global que no se utilizan dentro del país.

Referencias Bibliográficas

Herzog, P, (2010), *"The Open Source Security Testing Methodology Manual 3",* ISECOM

Kohnke, A., Sigler K., Shoemaker D. (2017), *"Implementing Cybersecurity A Guide to the National Institute of Standards and Technology Risk Management Framework"*

Kosutic, D. (2012), *"Ciberseguridad en 9 Pasos: El manual sobre seguridad de la información para el gerente"*, EPPS Services Ltd, Zagreb

Meucci, M., Muller A. 2014. "OWASP Testing Guide v4.0"

Mowbray, T. (2014) *"Cybersecurity: Managing Systems, Conducting Testing, and Investigating Intrusions"*, John Wiley & Sons Inc., Indianapolis, Indiana, Estados Unidos.

Toth, G., (2014) *"Implementación de la guía NIST SP800-30 mediante la utilización de OSSTMM"*, Tesis de Grado, Facultad de Informática, Universidad Nacional del COMAHUE, Buenos Aires, Argentina.

Valencia, L. (2013). *"Metodologías Ethical Hacking"*, Universidad Mayor de San Andrés, La Paz, Bolivia.

Junta Bancaria del Ecuador, (2014). *"Resolución Nº. JB-2014- 3066",* Junta Bancaria del Ecuador, Quito, Ecuador

EC-Council, (2015). *"CEH v9"* Módulo 01-05.

Vela, Fernando; Andrade Roberto, (2014), *"Guía de Pruebas OWASP 4.0 Borrador"* versión en español, Escuela Politécnica Nacional, Quito, Ecuador, Versión en inglés, Varios, *"OWASP Testing Guide v4",* OWASP Foundation.

Mendoza, M. (2017). *¿Ciberseguridad o seguridad de la información? Aclarando la diferencia*. *WeLiveSecurity*. Recuperado 29 septiembre 2017, a partir de

https://www.welivesecurity.com/la-es/2015/06/16/ciberseguridad-seguridad-informacion-diferencia/

Miranda, V. (2017). *Hackers vs Crackers*. *Víctor Miranda*. Recuperado 29 septiembre 2017, a partir de http://www.victormiranda.com.mx/vmwp/hackers-vs-crackers/

Toro, R. (2017). *ISO 27001: ¿Qué significa la Seguridad de la Información? Pmg-ssi.com*. Recuperado 29 septiembre 2017, a partir de http://www.pmg-ssi.com/2015/05/iso-27001-que-significa-la-seguridad-de-la-informacion/

Toro, R. (2017). *ISO 27001: Los aspectos básicos en la ciberseguridad*. *Pmg-ssi.com*. Recuperado 29 septiembre 2017, a partir de http://www.pmg-ssi.com/2015/03/iso-27001-los-aspectos-basicos-en-la-ciberseguridad/

Anexos

Anexo A. Antipatrones.

A1. Aplicaciones No Parchadas

Nombre Antipatrón: Aplicaciones No Parchadas

También conocido como: Actualizaciones específicas del proveedor, Configuración por defecto.

Nombres de soluciones refactorizadas: Administración de parches.

Fuerzas primarias desequilibradas: Administración de la integridad.

Evidencia anecdótica: “La mayoría de los nuevos ataques van luego de las aplicaciones, no de los sistemas operativos”

Solución Antipatrón

Según la lista emitida por SANS Institute de 2010 una de las principales vulnerabilidades de seguridad, las aplicaciones sin parches son uno de los mayores riesgos de seguridad. Las aplicaciones complementarias como QuickTime para Windows, Acrobat, Chrome y muchos otros son fuentes frecuentes de advertencias de seguridad del Equipo de Preparación para Emergencias Informáticas de Estados Unidos (US-CERT). Los proveedores intentan liberar parches para los problemas al mismo tiempo que se anuncian los defectos.

El retraso entre el lanzamiento del parche y la actualización instalada crea una ventana de vulnerabilidad para los atacantes. El anuncio de la vulnerabilidad y del parche binario ofrece a los atacantes pistas sobre cómo explotar la debilidad. Eventualmente, los investigadores de seguridad pueden incluso lanzar la explotación públicamente.

SANS encontró que las empresas son muy eficaces para mantener actualizados los parches del sistema operativo, pero son ineficaces para mantener los parches de aplicación actualizados.

Causas, Síntomas y Consecuencias

Las causas, síntomas y consecuencias de este antipatrón incluyen:

- Actualización automática deshabilitada en cualquier aplicación donde esté disponible.
- Nunca visite los sitios web de proveedores para buscar actualizaciones.
- Sin inventario de aplicaciones y proveedores.
- Sin programa de mantenimiento de actualización.
- No revisar los boletines del US-CERT.
- No hay gestión de las versiones de las aplicaciones.

Solución Refactorizada y Ejemplos

Un primer paso para administrar sus parches es obtener un inventario de sistemas y paquetes de software instalados. En las redes pequeñas, puede activar actualizaciones automáticas en Windows y en aplicaciones como Acrobat y Firefox. Para otras aplicaciones, como los controladores de vídeo, es posible que tenga que actualizar desde el sitio web del proveedor.

Vigile los boletines del US-CERT. Si se anuncian vulnerabilidades graves para sus aplicaciones, siga los enlaces disponibles del parche e instale los mismos. En algunos entornos, los usuarios sofisticados pueden mantener sus propios sistemas de esta manera.

Para redes más grandes, las herramientas de administración de parches pueden mantener cientos o miles de máquinas con un esfuerzo mínimo. Algunos de los mejores proveedores de estas tecnologías son LANDesk, BMC, Altiris y HP.

Para mayor seguridad, muchas tiendas están adoptando herramientas de escaneo de vulnerabilidades, como Retina de eEye, Nessus de Tenable y NeXpose de Rapid7. A menudo se utilizan herramientas para pruebas de certificación de seguridad antes de la liberación del sistema. Algunos pueden configurarse para escaneos automáticos, como trimestrales o diarios. Las herramientas pueden comprobar parches, problemas de configuración de políticas y vulnerabilidades de la red.

A2. Nunca Lee los logs

Nombre Antipatrón: Nunca Lee los logs.

También conocido como: Muchachos observando las pantallas de monitoreo de red pierden todo, Inicio de Amenazas, Amenazas Persistentes Avanzadas, Centro de Operaciones de Red.

Nombres de soluciones refactorizadas: Análisis avanzado de Logs.

Fuerzas primarias desequilibradas: Administración de la confidencialidad.

Evidencia anecdótica: WikiLeaks, Aurora Cyber Intrusions.

Solución Antipatrón

Los centros operativos de red (NOC, por sus siglas en inglés) son instalaciones con grandes pantallas coloridas de sistema y estado de la red. Los dispositivos de sistema, red y seguridad envían mensajes sobre eventos (registros de auditoría) a aplicaciones de administración centralizada, que prueban las condiciones de alarma y generan las grandes pantallas.

Las reglas de alerta se establecen generalmente para eliminar las falsas alarmas positivas. Por ejemplo, las reglas del Sistema de Detección de Intrusos (IDS) y los Sistemas de Prevención de Intrusos (IPS) que causan falsas alarmas están deshabilitados. Los sucesos registrados con frecuencia, como los cambios de configuración en los sistemas de usuario final, no se alarman.

Todo esto está bien, suponiendo que realmente funciona, pero esas pantallas de colores dan una falsa sensación de seguridad. ¿Está funcionando realmente la solución? Las reglas de alerta IDS deshabilitadas introducen vulnerabilidades para que los atacantes evadan la detección. ¿Los dispositivos de seguridad funcionan realmente? Si hay una amenaza interna o una amenaza persistente avanzada (APT) haciendo transferencias masivas de datos en momentos inusuales con propósitos no autorizados, ¿alguien lo notaría?

Causas, Síntomas y Consecuencias

Las causas y síntomas de este antipatrón incluyen:

- Nadie es responsable de leer los registros de red, del sistema y de seguridad.
- No hay supervisión de estado y estado de sucesos de syslog.
- No hay reglas de alarma para las configuraciones de Windows.
- El nuevo IDS genera numerosas alertas.
- Muchas reglas IDS deshabilitadas.

Solución Refactorizada y Ejemplos

La lectura de los logs es una actividad periódica esencial; sin ella, pierde mucha actividad inusual, sospechosa y errónea en sus redes. Dependiendo de la criticidad de las aplicaciones, puede ser necesario revisar los registros diarios o múltiples veces durante el día.

Revise periódicamente los registros de eventos de seguridad del sistema, los registros del sistema, los registros de dispositivos de red y los registros de IDS / IPS. No dependa siempre de las versiones en el gestor de registros centralizado, sino que periódicamente audite los registros locales y asegúrese de que se reflejan exactamente en los registros centrales.

A3. Las Redes Siempre juegan según las reglas.

Nombre Antipatrón: Las Redes siempre juegan según las reglas.

También conocido como: Confiar en todos los servidores, confiar en todos los clientes, ¿crees en la magia?

Nombres de soluciones refactorizadas: Hardening de Sistemas, Estado del arte de los protocolos de redes inalámbricas.

Fuerzas primarias desequilibradas: Administración de la confidencialidad e integridad.

Evidencia anecdótica: En las redes inalámbricas, los puntos de acceso con mayor señal son los que los dispositivos de usuario confían incluso si son maliciosos.

Solución Antipatrón

Internet no fue diseñado pensando en la seguridad; ni muchas tecnologías inalámbricas. Por ejemplo, tanto los portátiles habilitados para Wi-Fi como los teléfonos móviles de Global System for Mobile Communications (GSM) aceptan cualquier estación base que conozca los respectivos protocolos. Hay una herramienta de seguridad gratuita llamada Karma que puede convertir cualquier computadora portátil con Wi-Fi en un punto de acceso inalámbrico impostor. Descrito como Internet en una caja, Karma engaña a otros ordenadores portátiles a compartir sus cookies para los principales sitios web y otros propósitos.

Muchos de los problemas de seguridad en Internet se deben a que el software asume que todos los demás están jugando según las reglas; por ejemplo, suponiendo que otros programas siguen todas las especificaciones de las Normas de Solicitudes de Internet y siempre intercambian parámetros razonables. El código de explotación y el malware explotan estas suposiciones de diseño al infringir deliberadamente las reglas

y capturar la tecnología de forma desprevenida, haciendo que el software de destino realice operaciones que no estaba diseñado para hacer y en nombre del atacante.

Causas, Síntomas y Consecuencias

Las causas y síntomas de este antipatrón incluyen:

- Pérdida de autenticación del servidor.
- Pérdida de autenticación del cliente.
- No hay monitoreo de redes por paquetes y protocolos mal formados.

Solución Refactorizada y Ejemplos

Hay muchas debilidades inherentes en las tecnologías de Internet que no se pueden mitigar. Lo que puede hacer es utilizar las mejores prácticas de seguridad cibernética para hacer de sus sistemas objetivos duros. Por ejemplo, endurezca las configuraciones del sistema de acuerdo con las directrices de las mejores prácticas. Utilice las soluciones más avanzadas y actualizadas para antivirus, anti-spyware, IDS, IPS y sistema de seguridad basado en host (HBSS). Configure sistemas como Wi-Fi portátiles para requerir autenticación de host. Ingeniería de seguridad en el sistema desde el inicio del ciclo de vida de desarrollo.

Soluciones Relacionadas

Algunas autoridades han defendido un replanteamiento fundamental de Internet con un apoyo mucho mayor a la delegación de confianza y a la atribución de acciones de los usuarios.

A4. Fuerte en el Exterior, débil en el medio.

Nombre Antipatrón: Fuerte en el Exterior, débil en el medio.

También conocido como: Defensa en profundidad, seguridad de perímetro, protegerse de todas las amenazas.

Nombres de soluciones refactorizadas: Enjaulamiento de Red, Sistemas de seguridad basados en host.

Fuerzas primarias desequilibradas: Administración de la confidencialidad.

Evidencia anecdótica: Cada navegador de usuario está enviando miles de spyware cada día.

Solución Antipatrón

Las arquitecturas de red tradicionales incluyen tres dominios principales: el límite de Internet (o DMZ), la red de área de almacenamiento (SAN) del centro de datos y el resto de la red (intranet). Entre la DMZ y la intranet, hay dispositivos de seguridad de red, incluyendo un firewall y posiblemente un IDS / IPS. La seguridad de la red se concentra en el firewall y se supone que los firewalls protegen a toda la red.

En teoría, los firewalls protegen la red ocultando las direcciones IP internas a través de la traducción de direcciones de red (NAT) y bloqueando el tráfico de paquetes entrantes en los números de puerto denegados. En la práctica, la mayor parte del tráfico de paquetes se concentra en muy pocos puertos salientes, principalmente: 53, 80 y 443. Corresponden a los protocolos de Domain Name System, HTTP y HTTPS, que son los protocolos centrales de la World Wide Web. Estos puertos salientes están abiertos en prácticamente todos los firewalls. Los escritores de malware y spyware son muy conscientes de este hecho, y elaboran su código para aprovechar estos puertos abiertos. Malware de Botnet y software espía basado en navegador envían paquetes de beacon de máquinas infectadas dentro del firewall al puerto 80 en servidores de control externo. Para el cortafuego, estos paquetes parecen ser tráfico web normal.

En la unidad de software malicioso, los atacantes aprovechan el hecho de que la mayoría de los navegadores de Internet están configurados para ejecutar código de script por defecto. Si su navegador encuentra un sitio infectado por software malicioso,

el código del atacante se ejecuta en su sistema. Los sitios de propagación de malware están muy extendidos en Internet. Por ejemplo, hay más de 9.000 sitios que distribuyen la protección antivirus libre, que es realmente un malware disfrazado. Una variedad de malware, Ransomware, bloquea el sistema y exige el pago.

Dentro del cortafuego, hay pocas protecciones internas en las intranets. Sin embargo, la mayor amenaza de todos, la amenaza interna, está dentro del firewall. Las amenazas internas son más peligrosas porque tienen credenciales de red legítimas, y conocen la información más valiosa.

Las amenazas externas penetran en las redes y entran en el firewall, a menudo a través de medios furtivos. En un escenario común de APT, los empleados específicos son estudiados usando su información en línea, como páginas de Facebook, LinkedIn pro les, y otros datos públicos. En los ataques de phishing, los correos electrónicos específicos se crean con archivos adjuntos de malware. Orar sobre la credulidad y la curiosidad de la gente, el malware se abre y el sistema infectado, por ejemplo, un registrador clave. El registrador de claves devuelve los datos de pulsación de teclado al atacante en el puerto 80, obteniendo rápidamente las credenciales de inicio de sesión del usuario y eventualmente las credenciales del administrador del sistema cuando se conectan para realizar el mantenimiento. Con credenciales administrativas, el malware se puede propagar a muchas otras máquinas dentro del firewall. En efecto, toda la intranet es propiedad del atacante.

Causas, Síntomas y Consecuencias

Las causas y síntomas de este antipatrón incluyen:

- Redes no protegidas dentro del firewall en la intranet.
- No utilizar HBSS.
- No realizar monitoreo de las configuraciones.

- Otros ciberpatrones como no leer los logs.

Excepciones Conocidas

Para redes pequeñas, quizás menos de 50 usuarios, una arquitectura de red tradicional podría ser viable. Sin embargo, se deben implementar medidas adicionales, como el endurecimiento del sistema y el HBSS.

Solución Refactorizada y Ejemplos

Para redes más grandes, con amplios activos de información, la seguridad de la intranet debe ser cuidadosamente diseñada. ¿Cuáles son los activos de información más importantes en la empresa? Éstos merecen una protección adicional, como un enclave de red con el cortafuegos separado, con monitorización de red IDS / IPS. La seguridad debe centrarse en los activos que más merecen salvaguardias adicionales.

Las soluciones de seguridad de última generación incluyen monitoreo continuo de la configuración. Existen herramientas (como Tripwire) que supervisan los cambios en las claves del sistema (como el kernel y las bibliotecas de vínculos dinámicos). Otras herramientas encapsulan tanto los cambios de sistema como las llamadas de la Interfaz de Programa de Aplicación (API), evitando acciones malintencionadas, como el McAfee HBSS. Algunas herramientas realizan pruebas periódicas de vulnerabilidad y configuración de la seguridad, como Retina de eEye, Nessus de Tenable y NeXpose de Rapid7.

A5. Todo por WEB.

Nombre Antipatrón: Todo por WEB.

También conocido como: Cross Site Scripting, solicitud de falsificación Cross Site, Red de Energía de Estados Unidos en Internet, Sistemas Financieros Globales en el Internet.

Nombres de soluciones refactorizadas: Separación Física, Separación Fuera de Banda.

Fuerzas primarias desequilibradas: Administración de integridad y disponibilidad.

Evidencia anecdótica: ¿Porqué rayos colocan la red de energía eléctrica en el Internet?

Solución Antipatrón

La mentalidad de "webificar todo" es de sentido común cuando prolifera interfaces web para infraestructura crítica. ¿Tiene sentido la proliferación de interfaces remotas de fácil mantenimiento y replicabilidad masiva para controlar plantas de energía eléctrica y dispositivos de red de núcleo? La llamada "Smart Grid" webifica sus dispositivos de control y pantallas, y los principales proveedores de dispositivos de red webifican sus interfaces de control.

El problema se ve agravado por la técnica común de malware llamada cross-site scripting (XSS). Lo que hacen los navegadores de Internet es ejecutar código remoto en forma de HTML, JavaScript y otras anotaciones de scripting estáticas y dinámicas. HyperText Markup Language (HTML) ya no puede considerarse una notación benigna estática. La introducción de HTML 5.0 agrava los problemas de seguridad al agregar instalaciones para la ejecución remota de código y acceso de lectura y escritura a los discos cliente del navegador local.

Cuando el código remoto se ejecuta en un navegador de Internet, tiene acceso completo a todas las ventanas abiertas del navegador, todos sus datos y todas sus autoridades implícitas. Los ataques XSS combinados con el control de la

infraestructura remota web es una receta para el desastre. Siempre que las interfaces administrativas remotas estén conectadas y el usuario se dirija a otros sitios de Internet, existe una clara posibilidad de que los ataques XSS puedan obtener el control de objetivos de infraestructura altamente valorados.

Causas, Síntomas y Consecuencias

Las causas y síntomas de este antipatrón incluyen:

- Los navegadores web son una plataforma de interfaz de usuario para aplicaciones, denominadas thin clients. Los clientes delgados se utilizan ubicuos y son convenientes para los administradores del sistema porque no hay ninguna instalación del software del cliente o actualizaciones del software del cliente.
- Los usuarios tienen el hábito de abrir varias pestañas de navegador y conectarse con varios sitios web. Los sitios web con contenido malicioso son una amenaza significativa y frecuente. El contenido malintencionado (como los scripts de malware) se puede incrustar en el sitio o distribuirse mediante anuncios proporcionados por terceros.

Solución Refactorizada y Ejemplos

Las redes privadas virtuales (VPN) proporcionan una separación fuera de banda de las comunicaciones a través de las redes públicas. Esto significa que la intercepción de los paquetes que usan tecnologías como sniffers de la red se evita esencialmente. VPNs son una tecnología ampliamente desplegada; uno se pregunta por qué las VPN no se utilizan universalmente. Por ejemplo, los proveedores que envían mensajes de correo electrónico sin cifrar a través de Internet están invitando a la explotación de los datos y posibles ataques futuros.

A6. No hay Tiempo Para la Seguridad

Nombre Antipatrón: No hay Timempo para la seguridad.

También conocido como: Agregar al último la seguridad, Culpar a la seguridad por los retrasos.

Nombres de soluciones refactorizadas: Los requerimientos de seguridad son requerimientos reales, Administración del Riesgo Cibernético.

Fuerzas primarias desequilibradas: Administración de confidencialidad, integridad y disponibilidad.

Evidencia anecdótica: "Espera hasta que se termine de probar, y luego preocupate de la seguridad"

Solución Antipatrón

Los desarrolladores de proyectos de software y ahora también desarrolladores de widgets, a menudo esperan hasta el final del ciclo de vida de desarrollo para abordar la seguridad. Cerca de la fecha en que el proceso de lanzamiento de la empresa probará las vulnerabilidades de seguridad, los administradores y los desarrolladores iniciarán un proceso de encubrimiento para enmascarar las prácticas inherentemente inseguras de software, cuentas de usuario y configuración. Cuando se enfrentan, los desarrolladores pueden alegar ignorancia; no son expertos en seguridad después de todo.

Causas, Síntomas y Consecuencias

Las causas, síntomas y consecuencias que incluye este anti patrón son:

- La seguridad nunca fue parte del requerimiento
- Ahorrar en costos de desarrollo y el tiempo de lo caro de la seguridad.
- El proyecto no esta a tiempo.
- Compartir cuentas de administrador.
- No entrenar a los desarrolladores para estar prevenidos sobre seguridad.

Excepciones conocidas

Si el software está fuera de la caja, ya está cerca del final del ciclo de desarrollo y puede configurarse con seguridad justo antes del despliegue. Sin embargo, usted está tomando asumiendo que los desarrolladores de software originales contaban con la seguridad y fue construido con configuración apropiadas.

Solución Refactorizada y Ejemplos

Los riesgos y requisitos de seguridad deben ser analizados al principio del ciclo de desarrollo al mismo tiempo que los requisitos funcionales. Esto no es tan difícil o caro como suena. Las partes interesadas del negocio deben categorizar el sistema, tales como: con dencialidad alta, medio de integridad y medio de disponibilidad

Anexo B. Caso práctico de la metodología ESIFE.

Debido a la cantidad de información confidencial que contiene el anexo correspondiente esta información no se adjunta al presente documento y se entrega únicamente a la institución correspondiente para su análisis de los resultados obtenidos.