UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas

TEMA:
ANÁLISIS DE AMENAZAS, RIESGOS Y VULNERABILIDADES DEL PORTAL WEB DEL COLEGIO CATÓLICO JOSÉ ENGLING MEDIANTE HACKEO ÉTICO PARA EL DISEÑO Y DESARROLLO DE UN APLICATIVO WEB DE MONITOREO DE INCIDENCIAS

AUTORES:
MARCO VINICIO BRAVO SÁNCHEZ
DAVID ALBERTO SÁNCHEZ PRIETO

TUTOR:
MANUEL RAFAEL JAYA DUCHE

Quito, julio del 2018

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Marco Vinicio Bravo Sánchez con documento de identificación N° 1722579453 y David Alberto Sánchez Prieto con documento de identificación N° 1720064870, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: ANÁLISIS DE AMENAZAS, RIESGOS Y VULNERABILIDADES DEL PORTAL WEB DEL COLEGIO CATÓLICO JOSÉ ENGLING MEDIANTE HACKEO ÉTICO PARA EL DISEÑO Y DESARROLLO DE UN APLICATIVO WEB DE MONITOREO DE INCIDENCIAS, mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

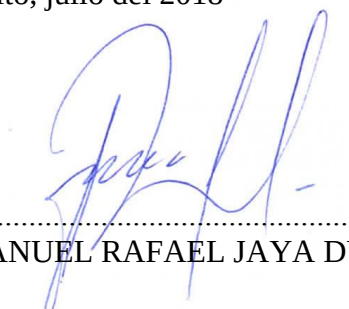
.....	.....
MARCO VINICIO	DAVID ALBERTO
BRAVO SÁNCHEZ	SÁNCHEZ PRIETO
CI: 1722579453	CI: 1720064870

Quito, julio del 2018

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: ANÁLISIS DE AMENAZAS, RIESGOS Y VULNERABILIDADES DEL PORTAL WEB DEL COLEGIO CATÓLICO JOSÉ ENGLING MEDIANTE HACKEO ÉTICO PARA EL DISEÑO Y DESARROLLO DE UN APLICATIVO WEB DE MONITOREO DE INCIDENCIAS, realizado por Marco Vinicio Bravo Sánchez y David Alberto Sánchez Prieto, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, julio del 2018



….....

MANUEL RAFAEL JAYA DUCHE

CI: 1710631035

DEDICATORIA

El presente trabajo lo dedico a Dios por haberme dado el don de la vida, y me ha llenado de sabiduría y amor.

A mis adorados padres, Ángel y Cecilia, quienes con esfuerzo y amor me han guiado por el buen camino toda la vida, enseñándome a cumplir los retos y a enfrentar las adversidades.

A mis hermanos, Christian y Diana porque con su cariño, y su incondicional apoyo se sumaron a mi meta.

A mis amigos que estuvieron conmigo en las buenas y malas, y han formado parte de mi motivación.

Marco Vinicio Bravo Sánchez

DEDICATORIA

A Dios, dueño de mi vida, verdadera fuente de amor y sabiduría.

A mis amados padres Julia y Miguel, quienes con su amor, paciencia y esfuerzo me han guiado día a día, siempre me han dado sus bendiciones, y porque me han enseñado que en el camino hacia la meta se necesita de la dulce fortaleza para aceptar las derrotas y del sutil coraje para derribar miedos.

A mi hermano Luis Miguel que siempre ha estado junto a mí brindándome su incondicional apoyo, muchas veces poniéndose en el papel de padre.

A mis adorables sobrinas, Valentina y Doménica, porque con sus sonrisas y travesuras lograron robarme tiempo de mi tesis, y jugar con ellas hasta volverme otra vez niño.

A Danny, mi inspiración a quien amo más que a mi propia vida y porque es la razón por la cual deseo siempre vivir.

A mis primos Fernando, Anita, Yadira, Andrés y Lenin, que con su carisma llenan mi vida de alegría.

David Alberto Sánchez Prieto

AGRADECIMIENTO

Agradecemos a Dios por darnos la oportunidad de vivir y la fortaleza para seguir adelante todos los días.

A la Facultad de Sistemas de la Universidad Politécnica Salesiana, por abrirnos las puertas para educarnos, y a nuestros estimados maestros, que, a lo largo de la carrera, nos brindaron sus conocimientos. A los Directores de este Proyecto de Titulación por su paciencia y dirección en la realización de este proyecto.

A nuestros compañeros y amigos de salón de clases, que nuestra amistad y recuerdos perduren para siempre.

A nuestro compañero y amigo Daniel Muñoz por su guía y apoyo incondicional en todo momento de este proyecto.

Marco Vinicio Bravo Sánchez

David Alberto Sánchez Prieto

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

El presente proyecto de titulación tiene como finalidad la aplicación de un hacking ético al portal web del Colegio Católico José Engling mediante la metodología ISSAF (Information System Security Assessment Framework), el cual permita encontrar y dar a conocer las posibles amenazas riesgos y vulnerabilidades del portal web, para identificar brechas de seguridad que afecten a la confidencialidad, integridad y disponibilidad de la información en la comunidad educativa, con el fin de presentar un plan de recomendaciones de mitigación de las incidencias encontradas al área de seguridad IT(Information Tecnología) de la institución.

Este proceso se lo realizó con la utilización de herramientas de recopilación de información, escaneo, enumeración y explotación del hacking ético en pruebas de penetración controladas y se ejecutó ataques de denegación de servicio, barrido de puertos, inyección SQL y fuerza bruta a los servicios disponibles (web, base de datos y correo electrónico).

En base a las pruebas se encontraron las siguientes amenazas a la seguridad del portal web: ataques de fuerza bruta, inyección SQL y denegación de servicio.

Adicional se desarrolló un aplicativo web de monitoreo de incidencias, el cual tiene como función garantizar la administración y gestión de los incidentes de forma ágil y adecuada a través de un proceso de registro y solución.

Abstract

The purpose of this degree project is to apply an ethical hacking to the portal web of José Engling Catholic High School, using the methodology of ISSAF (Information Systems Security Assessment System), which allows recognizing the possible threats, risks, and vulnerabilities of this portal web. Furthermore, it helps to identify the security gaps that affect the confidentiality, integrity, and availability of information in the educational community, in order to present a mitigation recommendation plan for incidents in the security area of IT (Information Technology) in this institution.

This process was carried out employing tools to collect information, scan, enumerate, and exploit ethical hacking in controlled penetration tests. It contains a denial of service attacks, port scanning, SQL injection, and brute force for available services (web, database, and email).

Based on the tests, there are the following threats to the security of the web portal: brute force attacks, SQL injection and denial of service.

In addition, a web application was developed to monitor incidents, whose main function is to guarantee the administration and management of incidents, in an agile and adequate manner through a record control process and solution.

INTRODUCCIÓN

Antecedentes

Hoy en día el Internet es una herramienta fundamental en la vida cotidiana, ya que su aparición ha revolucionado el sector de la información.

En los últimos años muchas empresas, instituciones y organizaciones manejan su información en red y utilizan el Internet como parte de su estrategia en el mercado global, obteniendo así mejores ventajas de acceso y disponibilidad a la información, publicidad, ventas directas, noticias, teletrabajo, email y mensajería.

En el mundo de la web la mayor preocupación son los delincuentes cibernéticos, personas curiosas quienes constantemente buscan la manera de romper la seguridad, comprometiendo los niveles de disponibilidad, integridad y confidencialidad de los sistemas de información de portales web de grandes organizaciones, con el fin de sacar provecho económico y personal.

> Los ciberataques o ataques informáticos suponen ya una de las principales amenazas a la seguridad de las organizaciones a nivel mundial, como el reciente ocurrido ransomware, uno de los crímenes cibernéticos más virulentos de Internet en los últimos años. Extendiendo sus tentáculos entre cientos de empresas y usuarios individuales, los hackers secuestran los archivos de las computadoras infectadas y luego extorsionan a sus víctimas exigiéndoles un rescate en bitcoins a cambio de devolverles sus archivos. La última víctima de este cibercrimen ha sido la empresa de telefonía móvil Telefónica, entre otras compañías españolas. (Gómez I. , 2017)

Problema

Debido a que el uso de portales web se encuentra en aumento, cada vez más instituciones educativas permiten a sus estudiantes, personal administrativo, docentes y sobre todo a la comunidad, acceder a sus sistemas de información. (CRIDO Santiago, 2016) Nos dice que "es fundamental saber qué recursos informáticos necesitan mayor protección para así controlar el acceso al sistema y los derechos de los usuarios en el sistema de información". Actualmente el portal web del Colegio Católico José Engling brinda servicios de notas académicas, facturación electrónica, ambientes virtuales, inscripciones en línea y correo institucional, que son de relevancia para la comunidad educativa los cuales requieren de exigentes seguridades.

La falta de medidas de seguridad en la institución es un problema, que con el transcurso del tiempo va crecimiento, de forma que aumenta la probabilidad de ser objetivo de ataques informáticos, uno de los más utilizados frecuentemente es el de fuerza bruta, el cual mediante diccionarios, genera datos de manera aleatoria (cadenas de caracteres al azar) para probar contraseñas de acceso a información personal, notas académicas o información bancaria que perjudicaría al usuario en distintas formas.

El objetivo de los hackers, es el acceso y manipulación de los datos de usuarios y la información institucional para beneficio propio, lo cual afecta en un alto grado el bienestar de la institución educativa y su comunidad.

La gestión de eventos de seguridad informática también es un punto importante en las instituciones, que muchas de las veces se dejan a un lado por falta de herramientas que ayuden a la gestión y monitoreo de incidencias, para dar una ágil atención y solución a las mismas.

Justificación

Debido al crecimiento tecnológico del Colegio Católico José Engling la seguridad informática es un tema de vital importancia que necesita de una constante evaluación y gestión, ya que la institución educativa maneja información en red, de manera que el intercambio de datos requiere de rigurosa seguridad, puesto que los mismos en manos de personas equivocadas podrían comprometer la integridad de dicha institución.

Eventualmente, medidas de seguridad como firewalls, proxys y antivirus llegan a fracasar debido a diversos motivos como configuraciones erradas, falta de actualización y principalmente el desconocimiento de la situación actual de seguridad en la red, es por eso que mediante la utilización de técnicas y metodologías de hackeo ético se busca realizar una auditoría informática web, mediante ataques controlados que permitan reforzar la protección de accesos no autorizados al portal web de la institución.

De la misma manera, la ejecución de este procedimiento de hackeo ético permite realizar un estudio más complejo sobre la accesibilidad de los usuarios a la información de dicho portal web, evitando así que el atacante obtenga información del cliente con el cual pueda realizar ataques de phishing, para robo y manipulación de información confidencial.

La ventaja del presente proyecto es que mediante el resultado de los análisis se entregará un plan de recomendaciones de mitigación y un aplicativo web de monitoreo de incidencias que ayude a reforzar los niveles de seguridad del portal web del Colegio Católico José Engling y de esta manera aumentar la defensa contra los diferentes tipos de ataques informáticos, salvaguardando la confidencialidad, integridad y disponibilidad del portal web, y su comunidad educativa pueda acceder de forma segura confiable y continua.

Objetivos

Objetivo General

Analizar las amenazas, riesgos y vulnerabilidades del portal web del Colegio Católico José Engling mediante hackeo ético para diseñar y desarrollar un aplicativo web de monitoreo de incidencias.

Objetivos Específicos

- Realizar un estudio de investigación sobre hackeo ético en portales web, para aplicar en pruebas de simulación controlada al portal web del Colegio Católico José Engling.
- Seleccionar las herramientas de pruebas de penetración óptimas y aprobadas por el área de sistemas competente para detectar las vulnerabilidades del portal web del Colegio Católico José Engling.
- Diseñar el escenario de investigación que contenga la estructura similar del ambiente que se usa actualmente en la institución, al cual se va a realizar la auditoría, sin afectar de alguna forma la disponibilidad del servicio del portal web del Colegio Católico José Engling.
- Ejecutar la auditoría al portal web del Colegio Católico José Engling, para encontrar brechas de seguridad existentes en el sistema de información que actualmente funciona en la institución.
- Obtener y analizar los resultados de la ejecución de las pruebas realizadas al portal web, con el fin de elaborar estadísticas reales que ayuden a identificar los principales problemas de seguridad informática.

- Diseñar y desarrollar un aplicativo web el cual permita monitorear los incidentes, a través de la metodología Scrum.
- Plantear un plan de recomendación de mitigación, que ayude al análisis de las alternativas de solución, con el objetivo de mantener actualizada la política de seguridad del portal Web del Colegio Católico José Engling, a efectos de velar por su vigencia y nivel de eficacia.

Alcance del Proyecto

El presente proyecto técnico pretende realizar una auditoría informática web por medio del uso del hacking ético, para encontrar posibles brechas de seguridad en el portal web del Colegio Católico José Engling, con ayuda de herramientas preseleccionadas y aprobadas por el área de sistemas de la institución, con el fin de elaborar un plan de recomendaciones de mitigación frente a los hallazgos encontrados.

Para ello se plantea realizar un estudio de investigación que permita abarcar los puntos de mayor importancia sobre hacking ético en portales web, tales como: conceptos básicos, técnicas, metodologías, leyes en el Ecuador, tipos de intrusiones, haciendo referencia a los ataques más comunes que actualmente operan y son de mayor impacto para las instituciones, para de esta manera tener un mejor enfoque al análisis de seguridad informática que se va a realizar a la página web.

Seguido de este estudio, se debe seleccionar de manera objetiva las herramientas de hacking ético que se van a usar en las pruebas de penetración, para detectar las falencias y vulnerabilidades de seguridad.

Una vez identificadas las herramientas a usarse, se va a recrear y definir un escenario de investigación, que contenga una estructura similar al ambiente de producción del portal web principal, con el fin de no afectar de alguna forma la disponibilidad del servicio del portal.

Como siguiente paso, se va a ejecutar los ataques de penetración al portal web, siguiendo el proceso completo de evaluación de seguridad de la información de la metodología ISSAF, (Ascencio Mendoza & Moreno Patiño, 2011) mencionan que consta de: “planeación y preparación, evaluación y presentación de informes”.

Una vez ejecutadas las pruebas, se obtendrán los resultados de la ejecución de los ataques al portal web, con el fin de identificar y analizar las principales vulnerabilidades que actualmente está expuesto el portal web, y comprometen la seguridad informática del Colegio Católico José Engling, lo cual puede causar riesgos referentes a la confidencialidad, integridad y disponibilidad de la información de comunidad educativa.

Con los resultados obtenidos del análisis se va a desarrollar un aplicativo web de monitoreo de incidencias (se encuentra en ANEXO 1 – Aplicativo web de monitoreo de incidencias), el cual mediante el módulo de escaneo permita ejecutar varias herramientas de hacking ético para la detención de brechas de seguridad, y a su vez tener control de ingreso de las incidencias de seguridad informática del portal web.

Con el módulo de reportes se va a brindar información de la incidencia basada en los eventos relevantes de seguridad con la ayuda de filtros según el criterio que requiera el administrador del aplicativo y conjuntamente con la gestión de solución de cada incidencia.

En el módulo de monitoreo se va a desarrollar una herramienta que realice un escaneo de puertos exclusivamente para el portal web principal del Colegio Católico José Engling, donde una vez terminado el proceso se genera un reporte de los puertos abiertos en formato XML.

Finalmente, en base a los hallazgos encontrados y las soluciones analizadas, se elaborará un plan de recomendaciones de mitigación, que ayude al análisis de las alternativas de solución, con el objetivo de mantener actualizada la política de seguridad informática del portal web del Colegio Católico José Engling, a efectos de velar por su vigencia y nivel de eficacia.

MARCO METODOLÓGICO

Para este proyecto de titulación se ha utilizado la metodología ISSAF, para el ámbito de auditoría informática al portal web del Colegio Católico José Engling y la metodología Scrum para el desarrollo del portal web de monitoreo de incidencias.

Metodología de hacking ético ISSAF

(Ascencio Mendoza & Moreno Patiño, 2011) menciona que "la metodología de pruebas de intrusión ISSAF está diseñada para evaluar la red, sistemas y aplicaciones. Esta metodología se enfoca en 3 fases: Fase I - Planeación y preparación, Fase II – Evaluación y Fase III - Presentación de Informes".

FASE I: Planeación y preparación

> Esta fase comprende los pasos para el intercambio de información inicial, planificar y prepararse para la prueba. Antes de la prueba de intrusión, se firmará un acuerdo formal entre ambas partes, (Empresa y auditor de seguridad), para proveer un mecanismo básico de protección legal. También se debe especificar el grupo de trabajo, las fechas exactas, los tiempos de la prueba, ruta de escalamiento y otras evaluaciones. Las actividades previstas en esta fase son: Identificación de contactos de parte y parte, Reuniones abiertas para confirmar el alcance, enfoque y metodología, Estar de acuerdo en los casos de prueba específicos y rutas de escalamiento. (Ascencio Mendoza & Moreno Patiño, 2011).

FASE II: Evaluación

> En esta fase es en realidad en la que se va a llevar a cabo la prueba de intrusión.
>
> Esta fase aplica un enfoque por capas, en donde cada capa representa un mayor nivel de acceso a los activos de la información. Las capas son las siguientes:

Recopilación de Información, Mapeo de la Red, Identificación de vulnerabilidades, Intrusión, Ganando acceso y escalando privilegios, Enumeración adicional, Comprometiendo usuarios/sitios remotos, Manteniendo acceso y Cubriendo rastros. (Ascencio Mendoza & Moreno Patiño, 2011).

FASE III: Presentación de informes

Informe Verbal: Según (Ascencio Mendoza & Moreno Patiño, 2011) "si en el transcurso de las pruebas de intrusión se encuentra una vulnerabilidad en el sistema, se debe informar inmediatamente a la organización para que sea consciente del problema".

Informe Final: (Ascencio Mendoza & Moreno Patiño, 2011) menciona que "tras la finalización de todos los casos de prueba definidos en el alcance del trabajo, se debe hacer un informe escrito que describa los resultados de las pruebas con las recomendaciones de mejora respectivas".

Limpiar el sistema de las pruebas de intrusión realizadas: (Ascencio Mendoza & Moreno Patiño, 2011) indique que hay que "remover todas las herramientas, archivos, software que se hayan instalado en el sistema. En el caso de no poderlos quitar, informar al cliente, para que éste tome las acciones necesarias".

Metodología de desarrollo de proyectos Scrum

Scrum es un proceso, marco de trabajo o framework, usado en equipos que trabajan en proyectos complejos; una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo, basada tres pilares: la transparencia, inspección y adaptación. (Araque, 2017)

Transparencia: (Araque, 2017) menciona que "todos los implicados tienen conocimiento de qué ocurre en el proyecto y cómo ocurre".

Inspección: (Noriega Martínez, 2017, pág. 47) indica que "los objetos de Scrum deben ser inspeccionados y el progreso ser evaluado para detectar variaciones".

Adaptación: (Noriega Martínez, 2017, pág. 47) indica que "si un inspector determina que uno o más aspectos de un proceso se desviaron fuera de los límites aceptables, y que por ello el producto resultado será inaceptable, el proceso o material en producción debe ser ajustado".

Para el presente proyecto se optó por utilizar la metodología Scrum, ya que es ágil y flexible para gestionar el desarrollo de software en entornos dinámicos, donde se necesita obtener resultados de forma rápida y cuyos requisitos son cambiantes.

Roles en Scrum:

- Product Owner: La voz del cliente y de los interesados indirectos al proyecto, define objetivos y vela por su cumplimiento.
- Scrum Master: Ayuda al equipo a mantenerse activo y productivo, proporciona soporte al Scrum Team.
- Scrum Team: Equipo que desarrolla y entrega el producto.
- Stakeholders: Grupo interesado en el producto (directores, dueños, comerciales). (Miríadax, 2017)

Evento de la metodología Scrum:

Sprints

Es el conjunto de requisitos o características que debe tener el producto. Contendrá todo lo que se considere aporta valor, aunque estará priorizado de arriba a abajo, donde arriba estarán los elementos más prioritarios y por ello, más detallados y

desgranados. En la parte inferior se tiene elementos o requisitos que todavía no están muy claros. (Miríadax, 2017)

El aplicativo web de monitoreo de incidencias desarrollado para el Colegio Católico José Engling, tiene la siguiente estructura:

PBS del aplicativo web

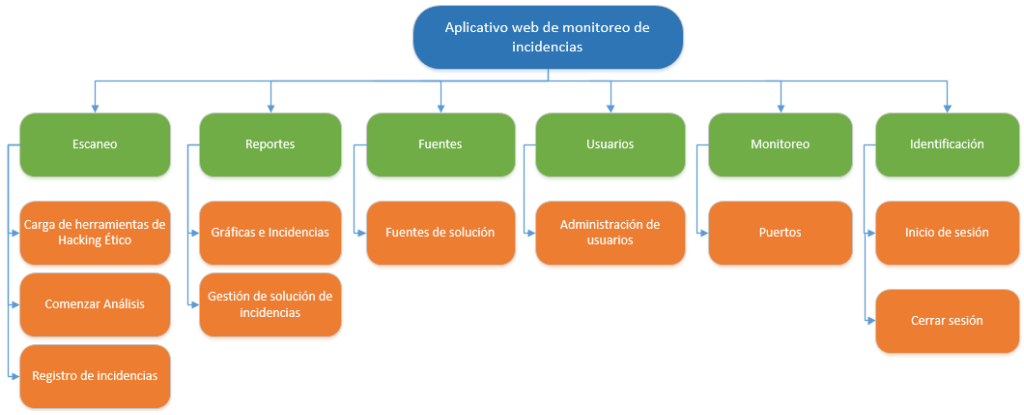


Figura 1. Modelo de los componentes del aplicativo con el diagrama PBS.
Elaborado por David Sánchez y Marco Bravo

Como se observa en la figura 1 El aplicativo web de monitoreo está compuesto de 5 módulos, Escaneo, Reportes, Fuentes, Usuario, Monitoreo e Identificación.

Módulo de Escaneo

Este módulo está constituido por Carga de herramientas de hacking ético, Comenzar análisis y Registro de incidencias.

Carga de herramientas de hacking ético. El administrador del sistema podrá registrar la ruta de instalación de las herramientas de hacking ético, para posterior llama y ejecutar.

Comenzar análisis. Permite ejecutar cada herramienta registrada, para iniciar con el análisis.

Registro de incidencias. Permite registrar las incidencias encontradas en el escaneo de cada herramienta.

Módulo de Reportes

Conformado por Gráficas e incidencias y Gestión de solución de incidencias.

Gráficas e incidencias. Mostrar graficas de incidencias filtradas por año.

Gestión de solución de incidencias. Permite registrar la solución a cada incidencia ingresada.

Módulo de Fuentes

Fuente de consulta de solución de incidencias por número y nombre de incidencia. El administrador podrá eliminar incidencias.

Módulo de Usuarios

El administrador tiene la posibilidad de registrar, eliminar y listar usuarios, además asignar un perfil a cada uno.

Módulo de Monitoreo

En este módulo se encuentra la aplicación de monitoreo de puertos del portal web del Colegio Católico José Engling, donde una vez terminado el escaneo se genera un reporte de los puertos abiertos en formato XML.

Módulo de Identificación

Está conformado por inicio de sesión y cerrar sesión.

Inicio de sesión. Los usuarios ingresan al sistema con sus credenciales, en este caso con el usuario y contraseña.

Cerrar sesión. Salir del sistema

CAPÍTULO 1

MARCO TEÓRICO

1.1. Introducción al hacking

Los medios sensacionalistas se han encargado en darle un mal significado a la palabra hacker, un significado que equipara a los hackers con criminales. Un hacker puede ser bueno o malo al igual que un abogado, un médico, un profesor o el oficio que fuera. El término hacking ético nace por tal motivo, era necesario aclarar que no es un hacking malo sino bueno, ético, pero en definitiva es hacking. (Tori, 2008, pág. 3)

1.2. Orígenes del término hacking

Originalmente, el término fue creado por TMRC (The Tech Model Railroad Club) del MIT (Massachusetts Institute of Tecnología), en los años cincuenta para acuñarlo a la persona que mediante el ingenio puede obtener una solución inteligente y efectiva llamada hack. La esencia del hack es hacerlo rápido y de forma sigilosa. (Gómez, Venegas, & Yánez, 2010)

De acuerdo con TMRC, la palabra hacker ha sido usada erróneamente de acuerdo con su significado original, para ser sinónimo de "delincuente informático". Una vez difundido, el nuevo término fue usado durante las épocas consecutivas para nombrar a los programadores expertos, aquellos que sabían aprovechar las capacidades de los dispositivos y de las tecnologías nacientes. (Gómez, Venegas, & Yánez, 2010)

Para los años noventa, Internet ya era el centro de atención para millones de usuarios de computadoras en todo el mundo y las crecientes mejoras en la red

permitían compartir información de forma más rápida y la investigación encontró un motor que aprovechaba los recursos computacionales; sin embargo y de manera inevitable sus vulnerabilidades salieron a la luz. Si bien, el conocimiento de estas fallas permitió reforzar los protocolos de comunicación existentes, también fue utilizado para explotar la intrusión de los sistemas en forma ilegal. De aquí en adelante, esta actividad quedó fuertemente ligada con el término hacker o hacking, y aunque varía de su significado original, ha trascendido hasta la actualidad. (Gómez, Venegas, & Yánez, 2010)

1.3. Hacker

Un hacker, originalmente, se describe como una persona amante de las computadoras con conocimientos altos en una o más áreas de la ciencia de la informática, especialmente en seguridad y programación. En definitiva, se trata de usuarios con conocimientos avanzados en el funcionamiento interno de las computadoras y redes informáticas. Estos usuarios suelen ser muchas veces aficionados obsesionados con la seguridad en las redes, y tratan de averiguar de qué forma se podría acceder a una red cerrada para posteriormente arreglar ese error del sistema. (ComoHacerPara, 2008)

Un hacker también puede desarrollar soluciones contra virus informáticos y programas que distribuye libremente, y ama la informática lo suficiente como para formarse día a día en esta arte, sin buscar ningún beneficio secundario u oculto que no sea la satisfacción personal. (ComoHacerPara, 2008)

1.4. Cracker

Muy al contrario de los hackers, los crackers son lo opuesto a los primeros, es decir, sujetos con conocimientos (no siempre altos) de redes e informática que persiguen objetivos ilegales, como el robo de contraseñas, destrozar la seguridad de una red doméstica o esparcir un virus informático a un gran número de computadoras. Los crackers pueden hacer todo su trabajo buscando tanto recompensas económicas (sustracción de dinero de tarjetas de crédito, estafas online, entre otras) como por el placer de creerse superiores al resto de la humanidad, o incluso por morbo. (ComoHacerPara, 2008)

1.5. Hacking ético

Es una disciplina de la seguridad informática que, brinda ayuda mediante una gran variedad de métodos para realizar sus pruebas, estos métodos incluyen tácticas de ingeniería social, uso de herramientas de hacking, uso de metasploit que explotan vulnerabilidades conocidas, en fin, son válidas todas las tácticas que conlleven a vulnerar la seguridad y entrar a las áreas críticas de las organizaciones. (Reyes Plata, 2011)

Por tanto, el objetivo fundamental del Ethical Hacking (hackeo ético) es explotar las vulnerabilidades existentes en el sistema de "interés" valiéndose de test de intrusión, que verifican y evalúan la seguridad física y lógica de los sistemas de información, redes de computadoras, aplicaciones web, bases de datos, servidores, etc. Con la intención de ganar acceso y "demostrar" que un sistema es vulnerable, esta información es de gran ayuda a las organizaciones al momento de tomar las

medidas preventivas en contra de posibles ataques malintencionados. (Reyes Plata, 2011)

1.6. Fases de un hacking ético

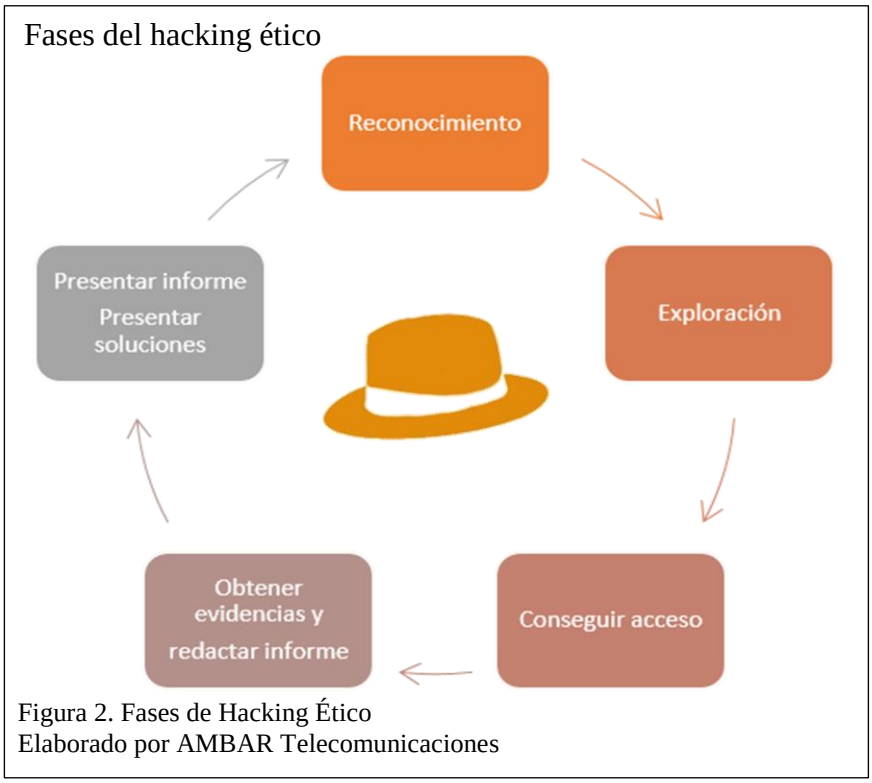


Figura 2. Fases de Hacking Ético
Elaborado por AMBAR Telecomunicaciones

La figura 2 indica las fases de hacking ético descritas a continuación:

Reconocimiento (Fase 1)

(Tapia, 2015) define que "es la fase donde el individuo o atacante necesita y busca recaudar información que le sirva de ayuda para poder llevar la operación. La fase de reconocimiento puede conseguirse de dos maneras, conocidas como pasivo y activo".

Pasivo: (Tapia, 2015)indica que "es cuando el atacante recauda la información sin la necesidad de interactuar con el objetivo, puede indagar por medio de buscadores, información por medio de los DNS (Domain Name System), dominios, etc".

Activo: (Tapia, 2015) menciona que "en este el atacante recauda información mediante la interacción con el objetivo, haciendo uso de la ingeniería social".

Exploración (Fase 2)

(AMBAR Telecomunicaciones, 2016) menciona que "en esta fase se escanea y enumeran todos los servicios, aplicaciones o elementos de red a raíz del descubrimiento en la primera fase (dispositivos, hosts, servidores, servicios)".

Conseguir acceso (Fase 3)

> En esta fase se produce el ataque en sí mismo. Para conseguir acceder al objetivo u objetivos del mismo mediante la explotación de vulnerabilidades existentes, haciendo uso ilegítimo de algún acceso; o bien, descubriendo y explotando deficiencias en el software utilizado en el objetivo. (AMBAR Telecomunicaciones, 2016)

Asegurar acceso (Fase 4)

> En esta fase se crea o modifica el sistema para permitir el acceso continuado al mismo, se intentan escalar privilegios de acceso, se establecen canales de comunicación para permitir descargar y ejecutar software malicioso, transferir información o analizar todo el tráfico que gestione el dispositivo. (AMBAR Telecomunicaciones, 2016)

Eliminación de huellas (Fase 5)

> Se intenta eliminar cualquier evidencia del proceso de intrusión en el sistema con el fin de asegurar la fase 4, y obviamente, evitar dejar cualquier información que pudiera ser utilizada legalmente en su contra, al igual que proteger el método

utilizado para romper la seguridad en su acceso. (AMBAR Telecomunicaciones, 2016)

1.7. Beneficios de un hacking ético

Al finalizar el hacking ético se entrega el resultado al cliente mediante un documento que contiene a grandes rasgos una lista detallada de las vulnerabilidades encontradas y verificables. También se provee una lista de recomendaciones para que sean aplicadas por los responsables de seguridad en la organización. Este documento se compone de un informe técnico y uno ejecutivo, para que los empleados técnicos y administrativos puedan entender y apreciar los riesgos potenciales sobre el negocio. (Reyes Plata, 2011)

Los beneficios más importantes:

- (Reyes Plata, 2011) menciona que "ofrecer un panorama acerca de las vulnerabilidades halladas en los sistemas de información, lo cual es de ayuda al momento de aplicar medidas correctivas".
- Según (Reyes Plata, 2011) "deja al descubierto configuraciones no adecuadas en las aplicaciones instaladas en los sistemas (equipos de cómputo, firewalls) que pudieran desencadenar problemas de seguridad en las organizaciones".
- (Reyes Plata, 2011) menciona que "identifica sistemas que son vulnerables a causa de la falta de actualizaciones".

1.8. Hacker ético

Un hacker ético es un experto en computadoras y redes de datos, su función es atacar los sistemas de seguridad en nombre de sus dueños, con la intención de buscar y encontrar vulnerabilidades que un hacker malicioso podría explotar. Para probar los sistemas de seguridad, los hackers éticos utilizan los mismos métodos que sus homólogos, pero se limitan únicamente a reportarlos en lugar de sacar ventaja de ellos. (Reyes Plata, 2011)

1.9. Vulnerabilidad

La vulnerabilidad es la debilidad o fallo que presenta un sistema de información. Este es capaz de poner en riesgo la seguridad de toda o parte de la información. El motivo es que este fallo o debilidad permite que el atacante comprometa la integridad, confidencialidad e incluso la disponibilidad de la información y los datos. Los orígenes de las vulnerabilidades son muy diferentes. Pueden ser debidas a fallos en el diseño del sistema, carencia de procedimientos o simples errores de configuración. (Netcloud Engineering, 2017)

1.10. Amenaza

La amenaza es la acción que se vale de una vulnerabilidad para actuar contra la seguridad del sistema de información. Estas actuaciones son siempre peligrosas, pero, obviamente, si existe una vulnerabilidad su efecto se posibilita y multiplica. La amenaza forma parte del lado contrario, no del sistema. (Netcloud Engineering, 2017)

1.11. Riesgo

(Netcloud Engineering, 2017) define que “el riesgo es una probabilidad de que se pueda producir un incidente relacionado con la ciberseguridad industrial y doméstica, y tiene como principales factores la existencia tanto de una vulnerabilidad como de una amenaza”.

1.12. Tipos de ataques más comunes a sitios web

1.12.1. Denegación de servicio distribuida (DDoS).

Un ataque de Denegación de Servicio tiene como objetivo dejar inaccesible a un determinado recurso (generalmente un servidor web). Estos ataques generalmente se llevan a cabo mediante el uso de herramientas que envían una gran cantidad de paquetes de forma automática para desbordar los recursos del servidor logrando de esta manera que el propio servicio quede inoperable. Además, se suelen coordinar ataques involucrando un gran número de personas para que inicien este tipo de ataque en simultáneo, tratándose así de un ataque de denegación de servicio distribuido, el cuál muchas veces es un poco más difícil de contener. (Catoira , 2012)

1.12.2. SQL Injection.

Una consulta SQL es una petición de algún tipo de acción sobre una base de datos. La más habitual es la petición de un nombre de usuario y una contraseña en una página web. Dado que muchos sitios web solo supervisan la introducción de nombres de usuario y contraseñas, un hacker puede utilizar los cuadros de introducción de datos para enviar sus propias peticiones, es decir, inyectar SQL en la base de datos. De esta forma, los hackers pueden crear, leer, actualizar,

modificar o eliminar los datos guardados en la base de datos, normalmente para acceder a información confidencial, como los números de la seguridad social, los datos de las tarjetas de crédito u otra información financiera. (AVAST, 2015)

Dado que un ataque de inyección SQL puede afectar a cualquier sitio o aplicación web que utilice una base de datos basada en SQL, es una de las formas de ciberataque más peligrosas y más antiguas, pero también más frecuentes. Lo que es todavía más preocupante: las inyecciones SQL están más vigentes que nunca, ya que ahora existen programas de inyección SQL automatizada, lo que significa que los hackers pueden atacar y robar con más facilidad que nunca. (AVAST, 2015)

Si bien existen una gran variedad de ataques y regularmente aparecen nuevos, uno de los más comunes y clásicos es el que intenta validar una condición de consulta como verdadera. Esto se logra a través de una consulta muy simple: **[‘OR ‘1’=’1]** (Catoira, 2013)

De esta manera, si la sección de código donde se realiza la consulta no se encuentra desarrollada con las medidas de seguridad adecuadas, esta consulta resultará en una condición verdadera y la consulta original arrojará todos los resultados. Suponiendo que se tiene una consulta como la siguiente: **SELECT * FROM usuarios WHERE usuario=’$usuario’ AND password=’$password’;** (Catoira, 2013)

En la consulta anterior, los parámetros inyectables son ‘$usuario’ y ‘$password’. Si se aplica la inyección antes especificada sobre ambos parámetros, la consulta

original se transformará en la siguiente: **SELECT * FROM usuarios WHERE usuario=' OR '1'='1' AND password=' OR '1'='1';** (Catoira, 2013)

Asimismo, la consulta arrojará todos los resultados y por consecuencia el ciberdelincuente conocerá toda la información alojada en esa tabla. Esto se debe a que "1=1" es siempre verdadero. Tal como se menciona anteriormente, esta es una de las inyecciones más clásicas y es fácilmente solucionable. (Catoira, 2013)

1.12.3. Ataques de fuerza bruta.

Estos ataques se basan en algo muy simple: de manera automatizada se van probando cada determinado período de tiempo y en determinados lugares, distintas palabras hasta "adivinar" la contraseña. Un ataque de fuerza bruta puede ayudar, por ejemplo, a recuperar el acceso a un email en caso de que se haya olvidado la contraseña y no se pueda restablecer la misma. Cuando se olvida las contraseñas de equipos o cuentas, y no se puede restablecer las mismas ni recuperarlas, utilizar software de fuerza bruta puede hacer más simple la vida. (Lozano, 2015)

Se toma un archivo de texto, se escribe allí todas las alternativas posibles, y el software se encarga de probarlas hasta dar con la correcta y recuperarla. Lo crucial en realidad son los diccionarios de ataque de fuerza bruta y los recursos del hardware con que se vaya a realizar el ataque. Un buen diccionario de ataque de fuerza bruta puede llegar a tener más de 10gb sólo en archivos de texto plano (extenso), por lo que para procesarlo y ejecutar el programa con el cual se está atacando al mismo tiempo, se va a necesitar bastante hardware. (Lozano, 2015)

1.12.4. Inspección manual del código fuente de las páginas.

Aunque parezca mentira, muchas veces es posible encontrar comentarios en los scripts de JavaScript o directamente en las páginas HTML con información tan importante como nombres de usuarios y contraseñas. Además, también es importante tener presente, cuales ficheros se suben al servidor, ya que puede ocurrir que por error, se suban ficheros fuente con información sensible cuya extensión termina en "~" por ejemplo, ".jsp~", ".asp~" o ".php~". (TheHackerWay, 2013)

1.12.5. Inspección y manipulación de cabeceras HTTP.

Como el lector ya sabe, las cabeceras HTTP permiten intercambiar información entre cliente y servidor web sobre detalles de la transacción HTTP que se lleva a cabo después de enviar una petición (cliente) y emitir una respuesta (servidor). Una práctica que se da con mucha frecuencia (desafortunadamente) es implementar filtros para temas tan importantes como la autenticación y/o autorización de los clientes utilizando como base algunas de las cabeceras HTTP intercambiadas. (TheHackerWay, 2013)

1.13. Test de penetración

(CyberSeguridad, 2015) indica que "por lo general es una acción acordada entre un pentester y una empresa o individual que desea tener sus sistemas informáticos puestos a prueba para identificar y posteriormente corregir posibles vulnerabilidades y los peligros asociados a las mismas".

Las pruebas de penetración permiten:

Según (Reyes Plata, 2011) “evaluar vulnerabilidades a través de la identificación de debilidades provocadas por una mala configuración de las aplicaciones”.

También dice (Reyes Plata, 2011) que permite “analizar y categorizar las debilidades explotables, con base al impacto potencial y la posibilidad de que la amenaza se convierta en realidad”.

Y (Reyes Plata, 2011) indica que permite “proveer recomendaciones en base a las prioridades de la organización para mitigar y eliminar las vulnerabilidades y así reducir el riesgo de ocurrencia de un evento desfavorable”.

1.14. Modelo TCP/IP

El protocolo más popular en uso hoy en día es el TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet). Actualmente, Internet y la mayoría de las intranets de las empresas utilizan TCP/IP debido a su popularidad, flexibilidad, compatibilidad y capacidad para realizar implementaciones en redes grandes y pequeñas. Aunque TCP / IP es el protocolo más utilizado, no es el más fácil de configurar e incluso de entender. El modelo TC/IP consta de cuatro capas a diferencia de las siete del modelo OSI. Las cuatro capas del modelo TCP/IP se correlacionan a las siete capas del modelo OSI, pero hay capas del modelo TCP/IP que combinan varias capas del modelo OSI. (Morales, 2017)

Capas del modelo TCP/IP

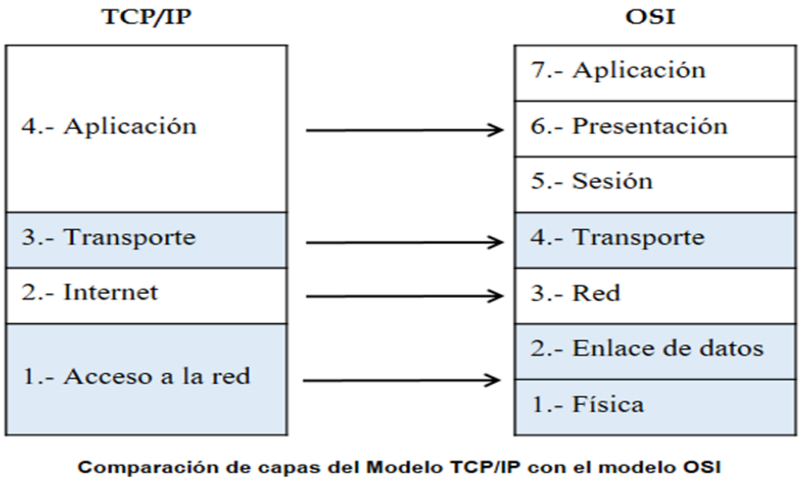


Figura 3. Comparación de capas modelo TCP/IP y modelo OSI.
Elaborado por Jonathan Morales - JMCristobalHomepage

A continuación, se explica cada una de las capas del modelo TCP/IP

Capa de aplicación

(Morales, 2017) indica que esta capa "define los protocolos de aplicación TCP/IP y cómo interactúan los programas host con los servicios de la capa de transporte para utilizar la red. Incluye todos los protocolos de nivel superior como DNS, HTTP, FTP, SMTP, DHCP, etc."

Capa de transporte

(Morales, 2017) indica que "el propósito de esta capa es permitir que los dispositivos hosts de origen y destino lleven a cabo una conversación. La capa define el nivel de servicio y el estado de la conexión utilizada para transportar los datos".

Capa de Internet

(Morales, 2017)indica que esta capa "proporciona la información de las direcciones de origen y destino (dirección IP) que se utiliza para reenviar los datos entre los hosts a través

de la red, la capa de Internet también es responsable del enrutamiento de los datagramas IP".

Capa de acceso a la red

Según (Morales, 2017) esta capa "define detalles de cómo los datos se envían físicamente a través de la red, incluyendo cómo los bits son señalados y enviados por dispositivos físicos que interactúan directamente con un medio de red, por ejemplo, un cable coaxial, fibra óptica".

1.15. Sistema operativo Kali Linux

Kali Linux es una recopilación de potentes herramientas para poder realizar un test de penetración en cualquier red. Kali Linux es una distribución Linux basada en un Debian avanzado dirigida a las pruebas de penetración y auditoría de seguridad. Kali contiene varios cientos de herramientas destinadas a diversas tareas de seguridad de la información, tales como pruebas de penetración, análisis forense y la ingeniería inversa. (seguridadroberto, 2016)

1.16. Sistema operativo Windows

(Tecnología W, 2011) indica que este sistema operativo fue "desarrollado por la empresa de software Microsoft Corporation, el cual se encuentra dotado de una interfaz gráfica de usuario basada en el prototipo de ventanas. Una ventana representa una tarea ejecutada o en ejecución, que contener su propio menú".

(Pérez & Merino, 2013) define que "es un conjunto de programas que posibilita la administración de los recursos de una computadora. Este tipo de sistemas empieza a trabajar cuando se enciende el equipo para gestionar el hardware a partir desde los niveles más básicos".

CAPÍTULO 2

RECOLECCIÓN DE LA INFORMACIÓN O FOOTPRINTING (HUELLA)

Antes de comenzar con el hacking ético, cabe recalcar que, para este capítulo y los capítulos siguientes, las pruebas, escaneos, prácticas y conceptos impartidos en los mismos, no buscan promover el uso de programas para la intrusión en sistemas informáticos, solo se lo realiza con fines educativos, por lo que cualquier uso de programas aquí mencionados por parte de los asistentes, no es responsabilidad de los autores de este proyecto de titulación o del docente tutor. Ya que esto podría llevar a problemas legales, tal como se explica en el ANEXO 2 - Código Orgánico Integral Penal.

La recolección de información o footprinting es la primera fase de ejecución en un hacking ético, consiste en recabar toda la información necesaria del portal web del Colegio Católico José Engling, para poder llevar a cabo las siguientes fases de escaneo y enumeración.

En esta fase es donde se debe dedicar mucho esfuerzo y tiempo para realizar un buen levantamiento de la información, esto se lo lleva a cabo por medio de dos técnicas, reconocimiento pasivo y activo.

Los lugares más comunes donde se tiene más cantidad información son los sitios web, anuncios publicitarios, redes sociales, entre otras.

Recolección de información

Más información recolectada = Mayor probabilidad de Éxito en el ataque

Figura 4. Ecuación de la recolección de información.
Elaborado por Curso-Profesional-de-Hacking-Ético-CPHE

En la recolección de información mientras más información recolectada se tenga, mayor probabilidad de éxito se obtendrá en el ataque tal y como lo menciona el curso Profesional de Hacking Ético en la figura 4.

2.1. Tipos de reconocimiento

2.1.1. Reconocimiento pasivo.

El atacante no interactuando directamente con el objetivo.

- Consultas de directorios en Internet.
- Búsquedas en redes sociales.
- Recuperación de información desde la papelera de reciclaje.
- Buscar en el periódico por anuncios de ofertas de empleo, en páginas web.

(Astudillo, 2013)

2.1.1.1. *Información pública del sitio web.*

El atacante adquiere información sin necesidad de interactuar directamente con el objetivo.

Computrabajo

Figura 5. Oferta de la institución en CompuTrabajo.
Elaborado por David Sánchez y Marco Bravo

(CompuTrabajo, 2018) “es la web de empleo líder en Latinoamérica. Es la bolsa de trabajo más visitada en Colombia, Perú, Argentina, Uruguay, Guatemala, Ecuador y El Salvador, y la segunda de Honduras, Venezuela, Nicaragua, Cuba y Costa Rica”.

Una de las formas de obtener información específica, como aplicaciones que usan o base de datos que maneja la institución es buscando en páginas donde se oferta trabajo, en la figura 4, se observa las ofertas laborales que requiere la institución educativa mediante la página de CompuTrabajo para reclutar personal y la experiencia en manejo de aplicaciones con las que cuenta la institución.

Adicional se puede visualizar información como la dirección de la empresa, contactos, áreas de trabajo, descripción del puesto vacante, jornadas, requerimientos, herramientas que manejan dentro de la misma, entre otras cosas.

2.1.1.2. *Redes sociales.*

Facebook

Figura 6. Redes sociales del portal web
Elaborado por David Sánchez y Marco Bravo

Las redes sociales en la actualidad se han convertido en herramientas laborales de publicidad de las instituciones, las cuales tienen gran fuente de búsqueda de información entre las más usadas, se tiene: Facebook, Instagram, Twitter, entre otras.

En la figura 6, se observa información como: números telefónicos, correos electrónicos e inclusive la dirección física de la institución

2.1.2. Reconocimiento activo.

El atacante adquiere información interactuando directamente con el objetivo mediante:

- Barridos de ping
- Conexión a un puerto de un aplicativo
- Uso de ingeniería social
- Hacer un mapeo de red

(Astudillo, 2013)

2.1.2.1. *Buscadores o motor de búsqueda.*

(Telmex, 2012) Indica que "un buscador o motor de búsqueda, es un recurso informático que permite localizar información en los servidores conectados a la red". Todo este proceso de búsqueda lo realiza mediante su spider también llamado araña web.

Buscador Google

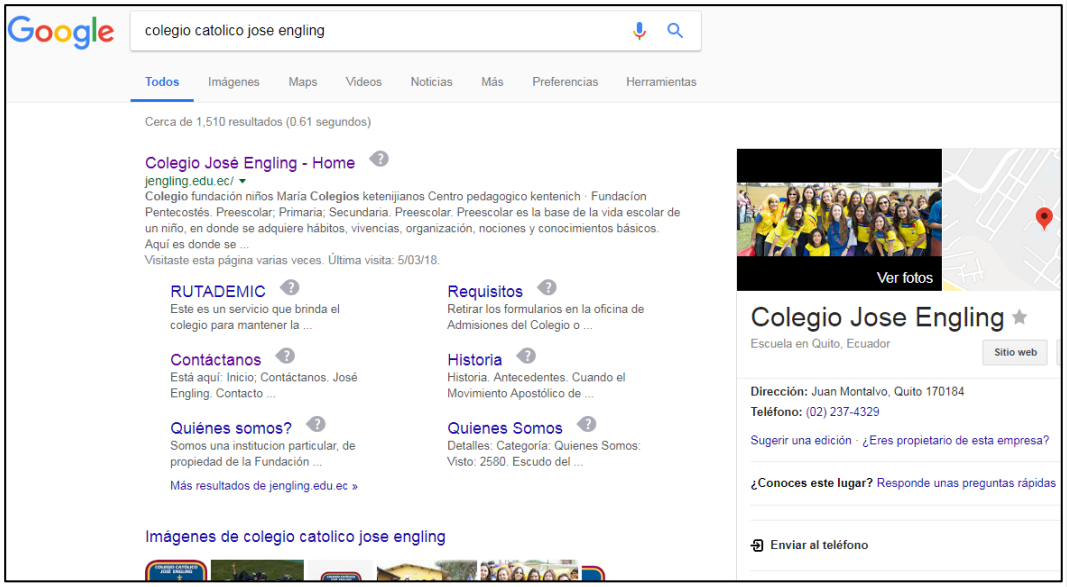


Figura 7. Uso del buscador Google
Elaborado por David Sánchez y Marco Bravo

Se obtuvo 1510 resultados en el motor de búsqueda de Google, ingresando como texto el nombre de la institución, el primer resultado de la búsqueda pertenece al portal web del colegio tal como se muestra en la figura 7 y es por donde se puede comenzar para obtener mayor información.

2.1.2.2. *Ping (Packet Internet Groper).*

(Ramírez González, 2016) menciona que el "buscador de paquetes en redes (ping) es un comando de diagnóstico, permite verificar el estado de una conexión. Por medio del dominio web este comando devuelve la dirección IP".

Ping CMD

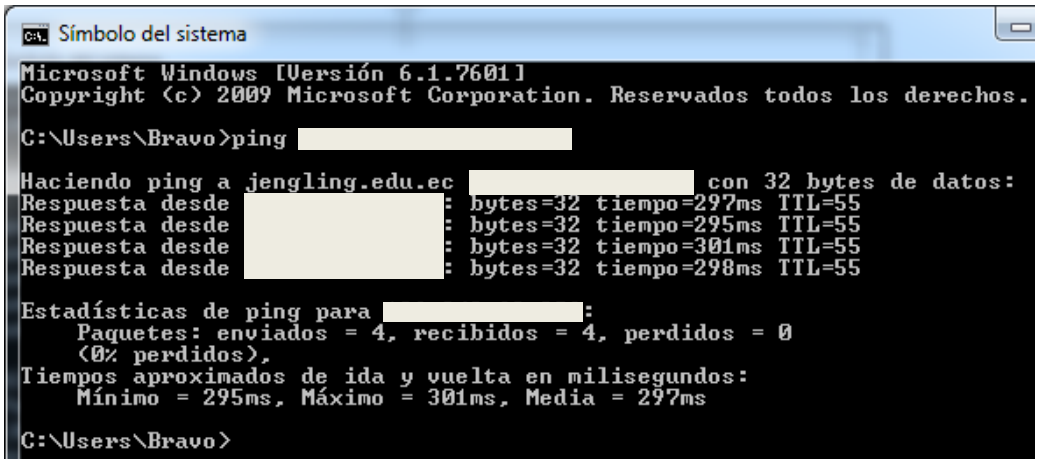


Figura 8. Ping al portal web
Elaborado por David Sánchez y Marco Bravo

En la figura 8, se puede visualizar que mediante el comando ping existe conectividad al host.

Para la ejecución de esta herramienta se abre el símbolo del sistema en Windows y se ingresa el comando **ping www.paginaweb.com** con el propósito de obtener su dirección IP.

2.1.2.3. *Resolución de nombre de dominios.*

Permite el ingreso a páginas por medio de nombres de dominio en vez de direcciones IP. Para ello se usa el programa Nslookup, según (Sánchez Patón & Prieto, 2012) “para saber si el DNS está resolviendo correctamente los nombres y las IP’s, se utiliza con el comando nslookup, que funciona tanto en Windows como en UNIX para obtener la dirección IP conociendo el nombre, y viceversa”.

Nslookup CMD

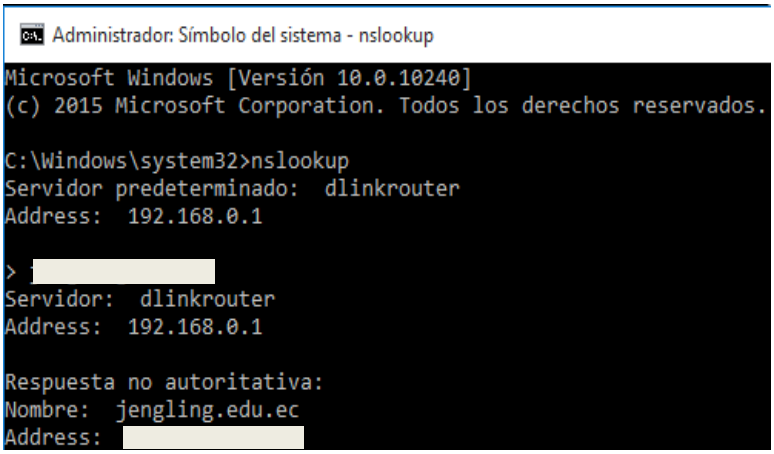


Figura 9. Comando nslookup desde cmd de Windows
Elaborado por David Sánchez y Marco Bravo

Como se observa en la figura 9, otra forma de obtener la dirección IP del objetivo es mediante el comando nslookup.

El comando a ejecutar es **nslookup** y seguido de este se coloca el dominio o dirección IP.

Nslookup comando NS

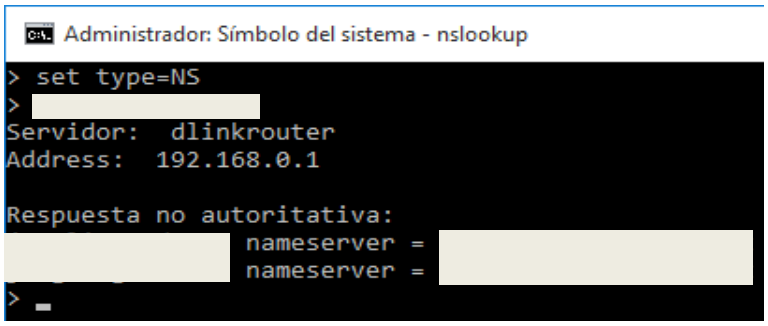


Figura 10. Comando set type=NS desde cmd de Windows
Elaborado por David Sánchez y Marco Bravo

El comando **set type=NS** muestra una lista de los servidores de nombre, tal y como se observa en la figura 10.

Nslookup comando MX

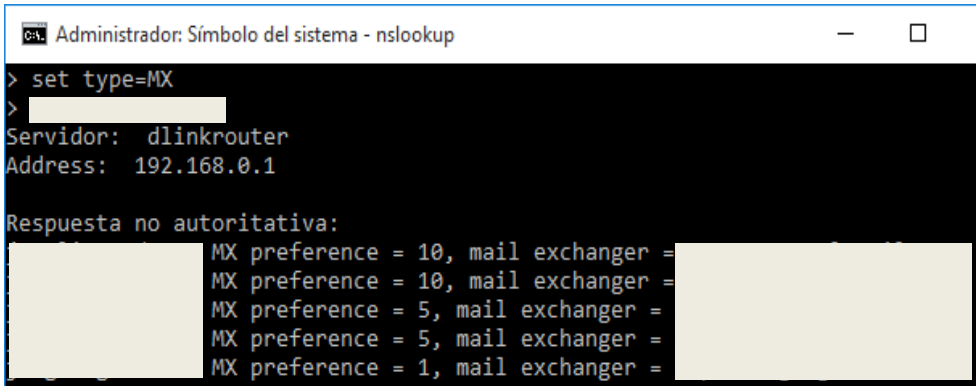


Figura 11. Comando set type=MX desde cmd de Windows
Elaborado por David Sánchez y Marco Bravo

El comando **set type=MX** muestra los servidores de correo, tal y como se observa en la figura 11.

2.1.2.4. *Directorios Who-is.*

El directorio WHOIS es una lista pública que contiene nombres de dominio y la información de contacto de las personas u organizaciones asociadas con cada nombre de dominio. Se utiliza para determinar quién es el propietario de un nombre de dominio. (Google, 2018)

Whois.net

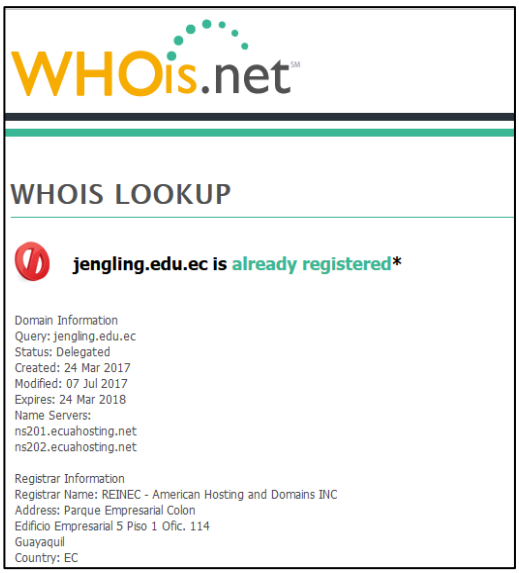


Figura 12. Directorio Whois
Elaborado por David Sánchez y Marco Bravo

Por medio de la página web que se muestra en la figura 12, bajo una petición, envía una respuesta con toda la información del dominio. En esta se puede visualizar fechas de: creación, última modificación y de expiración del sitio web, servidores de nombres, nombre del hosting donde se encuentra alojada la página, dirección física, país, números telefónicos y correos electrónicos.

2.1.2.5. *Código fuente de página web.*

El código fuente de una página web es una manera de recolección de información, esta puede estar en varios tipos de lenguaje de marcado y de programación web.

Para visualizar el código fuente se sitúa en el portal web por medio de un navegador, se ingresa a la opción **ver código fuente de la página**, esta funcionalidad la poseen todos los navegadores y se accede de la siguiente manera:

Navegador Google Chrome: botón derecho del ratón en la página y seleccionar "Ver código fuente de la página"

Navegador Mozilla Firefox: botón derecho del ratón en la página y seleccionar "Ver código fuente"

Navegador Internet Explorer: botón derecho del ratón en la página y seleccionar "Ver código fuente"

Código fuente en Mozilla Firefox

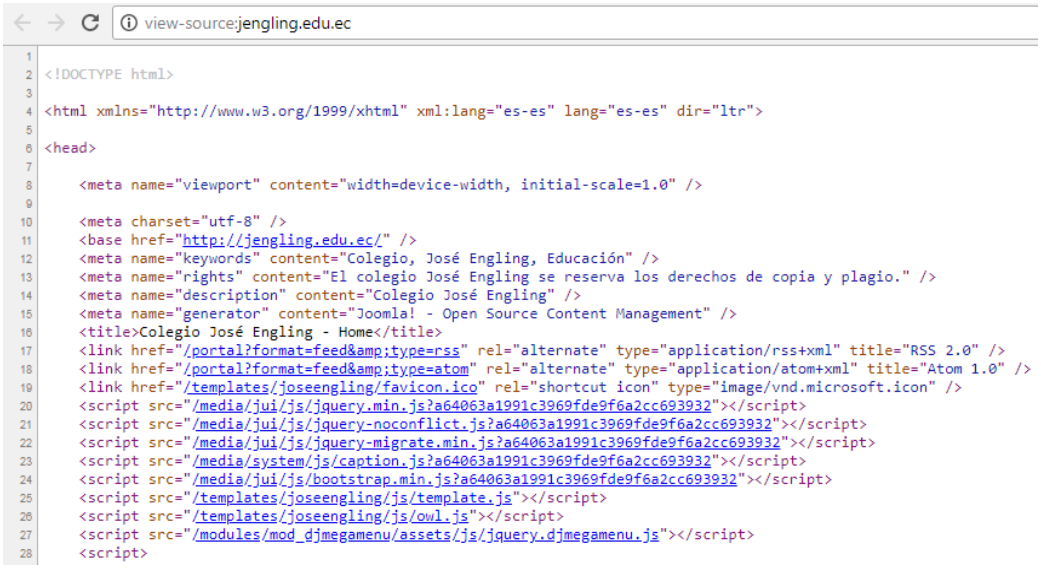


Figura 13. Código fuente, navegador Google Chrome
Elaborado por David Sánchez y Marco Bravo

En los resultados obtenidos en la página web mediante la visualización del código fuente, se encuentra la herramienta o sistema de gestión de contenidos con la cual se desarrolla la página web que en este caso es Joomla, como se observa en la figura 13.

2.1.2.6. *PassiveRecon Add-ons.*

(Brinkmann , 2008) Define que “PassiveRecon es una extensión de Firefox que consulta una multitud de bases de datos públicas y servicios de búsqueda para revelar la mayor cantidad de información posible sobre un dominio sin interactuar directamente con él”.

Esta extensión se eligió debido a que reúne varias herramientas de recolección de información y es de acceso disponible al público entre sus utilidades permite: obtener

información sobre archivos del sitio, dominio, correos, direcciones IP, sistemas operativos, DNS entre otras cosas. Para hacer uso de esta herramienta se debe instalar el Complemento de Firefox, el cual se puede descargar desde el enlace referenciado. (Mozilla Firefox, 2010).

Una vez instalado el complemento, se ingresa la página web objetivo, al pulsar clic derecho se observa una nueva opción llamada PassiveRecon con el menú de herramientas de recolección de información.

PassiveRecon

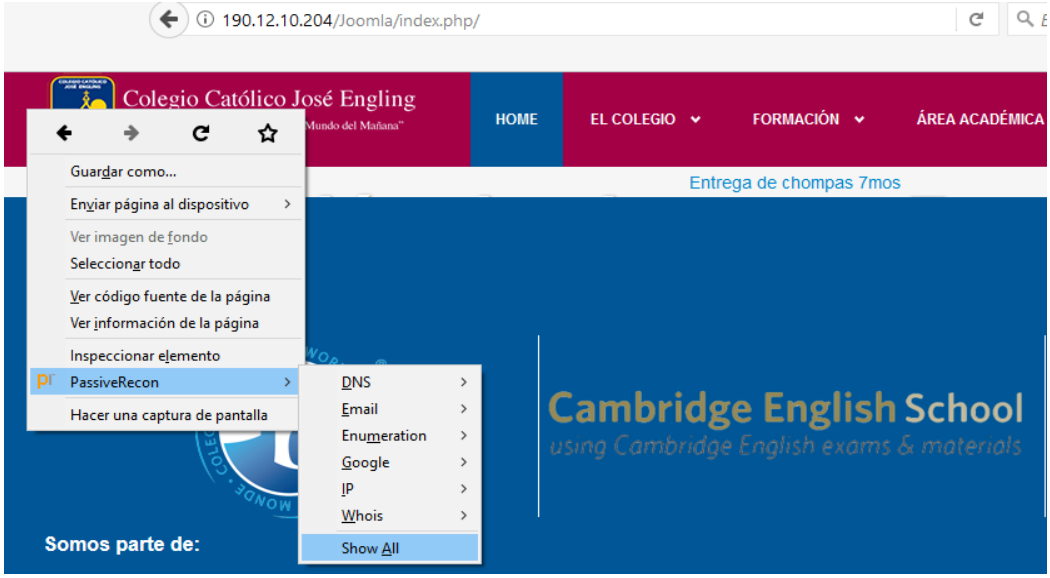
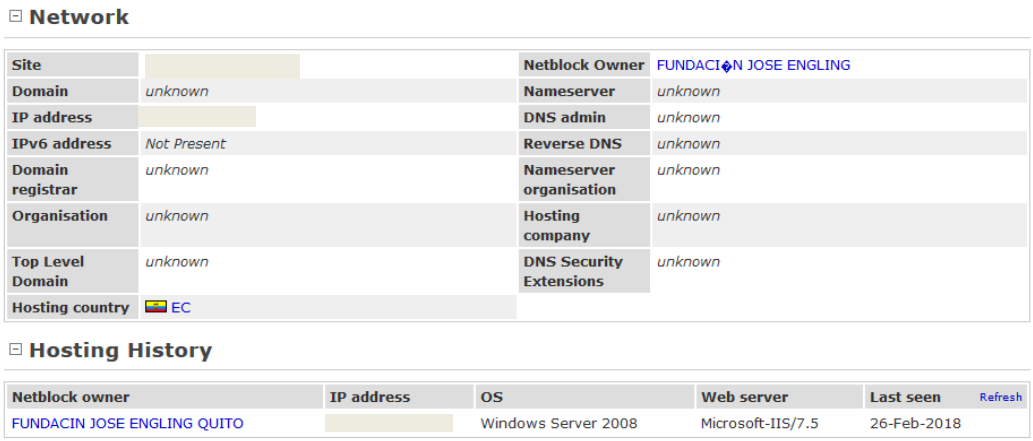


Figura 14. PassiveRecon Add-ons, complemento de Mozilla Firefox
Elaborado por David Sánchez y Marco Bravo

En la figura 14, se muestra las opciones de la herramienta de recolección de información PassiveRecon: DNS, Email, Enumeración Google Doors, IP y directorio Whois.

PassiveRecon resultado Enumeración



Network

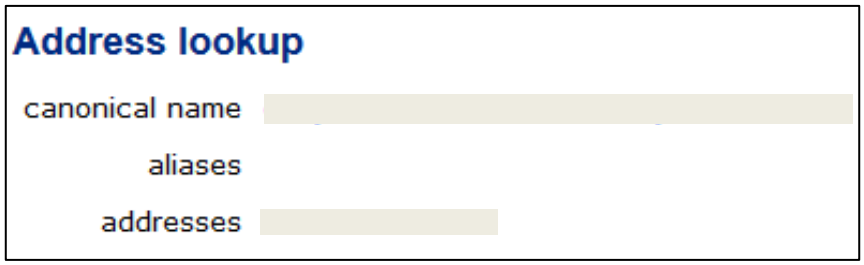
Site		**Netblock Owner**	FUNDACI◆N JOSE ENGLING
Domain	*unknown*	**Nameserver**	*unknown*
IP address		**DNS admin**	*unknown*
IPv6 address	*Not Present*	**Reverse DNS**	*unknown*
Domain registrar	*unknown*	**Nameserver organisation**	*unknown*
Organisation	*unknown*	**Hosting company**	*unknown*
Top Level Domain	*unknown*	**DNS Security Extensions**	*unknown*
Hosting country	EC		

Hosting History

Netblock owner	IP address	OS	Web server	Last seen Refresh
FUNDACIN JOSE ENGLING QUITO		Windows Server 2008	Microsoft-IIS/7.5	26-Feb-2018

Figura 15. Resultado de Enumeración, PassiveRecon
Elaborado por David Sánchez y Marco Bravo

Address lookup



Address lookup

canonical name

aliases

addresses

Figura 16. Resultado Address lookup, PassiveRecon
Elaborado por David Sánchez y Marco Bravo

Para la figura 15, se observa como respuesta el sistema operativo del servidor en el cual está montada la página, también el servidor web que se está utilizando y la información del dominio que se muestra en la figura 16.

2.1.2.7. *Traceroute.*

Traceroute es ampliamente utilizado para detectar problemas de enrutamiento end-to-end y descubrir la topología de Internet. Proporcionando una lista precisa de los sistemas autónomos (ASes) a lo largo de la ruta de reenvío harían traceroute aún más valioso para

los investigadores y operadores de red. Sin embargo, enfoques convencionales para mapeo traceroute saltos a números no son lo suficientemente exactos. (Rexford, Wang, Morley Mao, & Katz, 2003)

Open Visual Trace Route

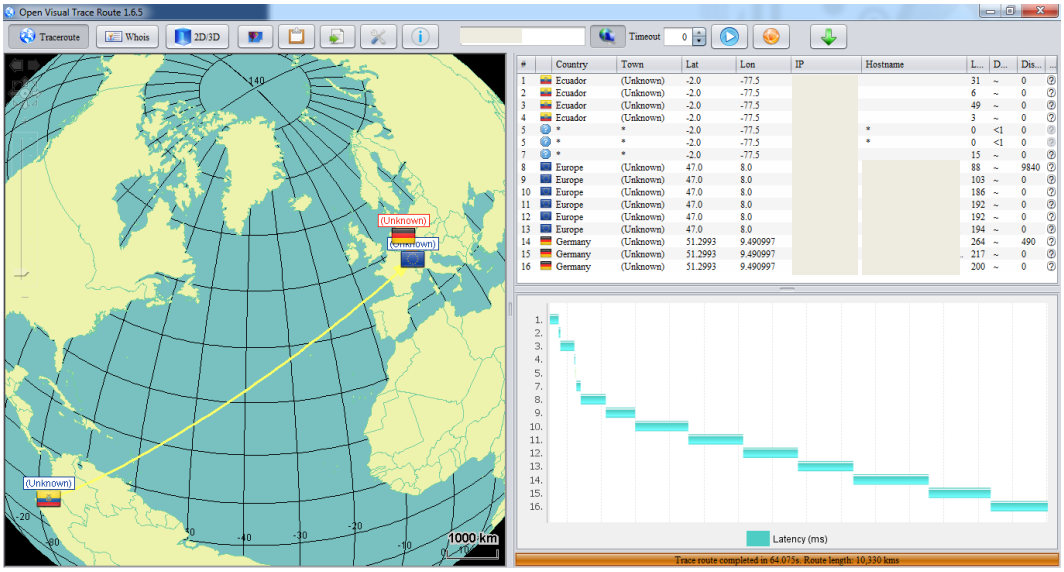


Figura 17. Herramienta Open Visual Trace Route
Elaborado por David Sánchez y Marco Bravo

En los resultados obtenidos de la herramienta Open Visual Trace Route muestra la ubicación del servidor de web, y el proveedor de Internet ISP y los saltos que realiza desde donde el cliente hace la petición, hasta el servidor destino como se puede visualizar en la figura 17.

2.1.2.8. *Mapeo de información.*

En este caso se utilizará la herramienta Maltego Community, esta permite recabar información sobre una persona, organización, infraestructura de red, documentos y sitios web a través de objetos gráficos y menús contextuales en los que se puede aplicar transformaciones. Una transformación es una operación que al aplicar sobre un objeto despliega información adicional del mismo. (Pérez I. , 2014)

Esta herramienta se eligió debido a que hace un mapeo de información recolectada y muestra qué tan expuesto se está en Internet, permite encontrar información sobre personas y empresas.

Maltego

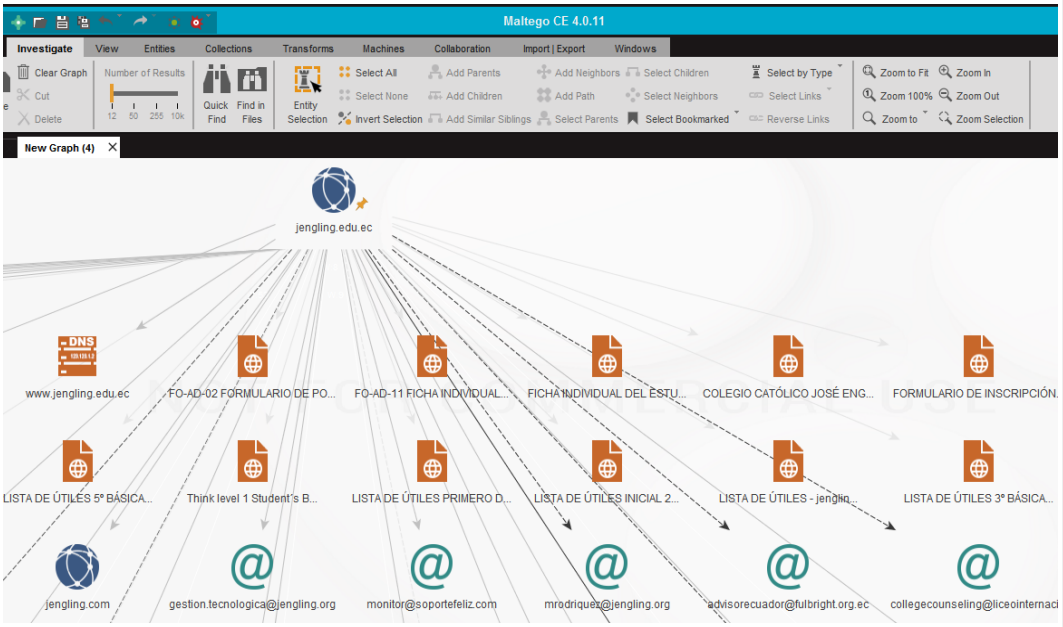


Figura 18. Transformaciones con la herramienta Maltego
Elaborado por David Sánchez y Marco Bravo

En la figura 18, se observa como resultado correos electrónicos, archivos PDF adjuntos a la página web, nombre de dominios e incluso la localización.

Maltego Transformaciones

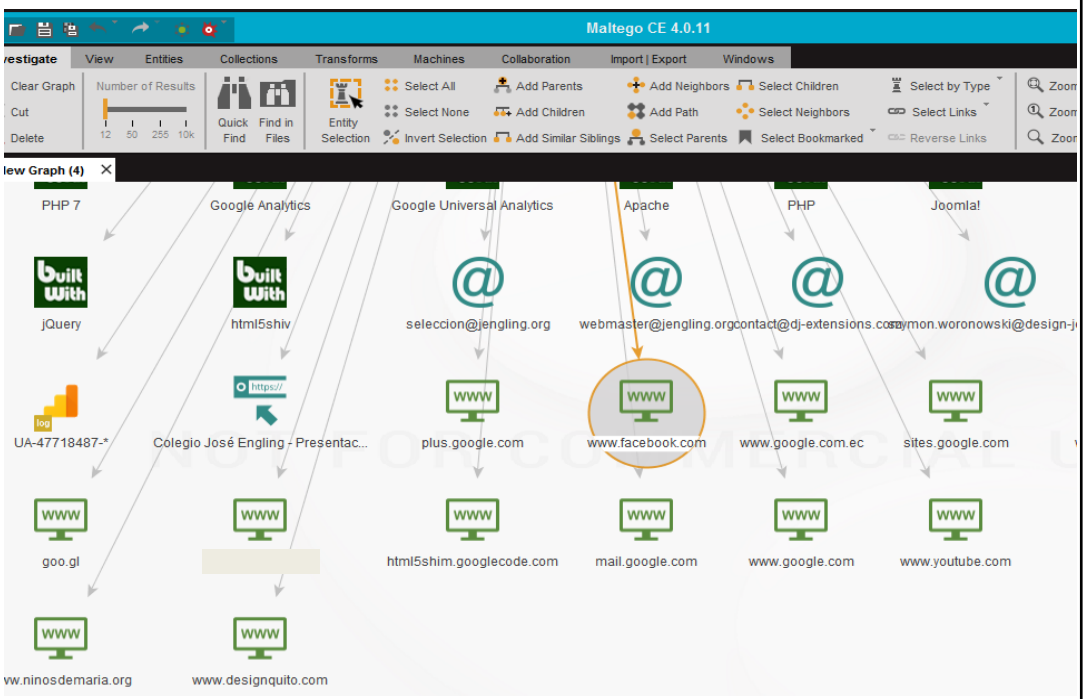


Figura 19. Páginas asociadas al dominio, Maltego
Elaborado por David Sánchez y Marco Bravo

Tambien se puede obtener como resultado páginas asociadas al dominio, como por ejemplo Facebook, Youtube, Instagram, Google+, entre otros, como se observa en la figura 19.

CAPÍTULO 3

ESCANEO

Con la información recolectada en el capítulo 2 recolección de información, se procede a la siguiente fase del hacking ético que es escaneo.

Es la segunda fase de un hacking ético y tiene como objetivo el identificar los hosts activos dentro de ciertos rangos de IP y determinar si es posible escuchar datos que están siendo transmitidos por estos medios y así conocer si los hosts de la red tienen vulnerabilidades informáticas potenciales para explotar. (Martinez & Oñate, 2017)

3.1. Correos asociados al portal web

(Pérez I. , 2015) indica que "existen direcciones de correo electrónico que no deben ser publicadas abiertamente. Esto permite a los ciberdelincuentes, de manera fácil y efectiva, enviar ataques personalizados por correo electrónico a todo el personal, aumentando la superficie de ataque y las probabilidades de éxito".

Para ello se utiliza la herramienta Harvester la cual permite conseguir información sobre emails, subdominios, hosts, nombres de empleados, etc. Desde fuentes públicas como son los motores de búsqueda, la red social LinkedIn y la base de datos de SHODAN (Buscador parecido a Google pero con la diferencia que no indexa contenido, si no que registra cualquier dispositivo conectado a Internet). (Olmedo, 2015)

El comando a usar para obtener información es:

The Harvester

Figura 20. Comando The Harvester
Elaborado por David Sánchez y Marco Bravo

Para la ejecución de la herramienta se ingresó el comando, como se visualiza en la figura 20:

theharvester -d paginaweb.com -l 100 -b google

Se describe el comando aplicado a continuación:

-d: dominio para buscar o nombre de la compañía.

-l 100: número limite de resultados a mostrar, en este ejemplo se usan 100(el buscador Bing permite 50 resultados)

-b : buscador web de internet a ser usado

The Harvester Escaneo

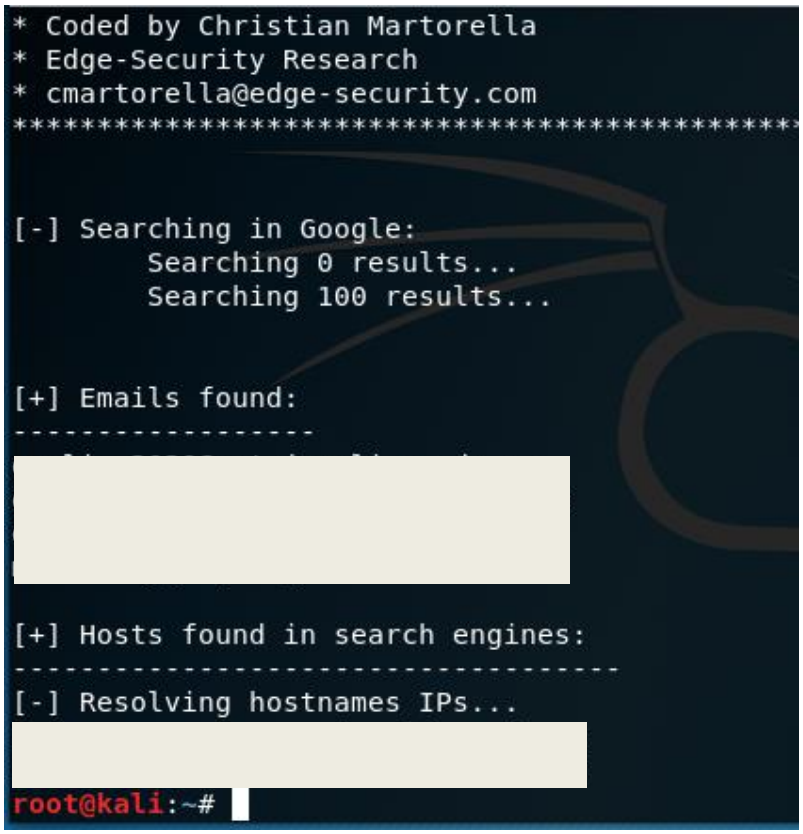


Figura 21. Búsqueda de correos con la herramienta The Harvester
Elaborado por David Sánchez y Marco Bravo

Esta herramienta permite buscar información de correos, para este caso se utiliza el buscador de Google, se hace una búsqueda para obtener los correos expuestos en el portal Web, tal como se muestra en la figura 21.

3.2. Host activos en una subred

Se usa la herramienta Net Scan, esta permite identificar cada host activo por cada IP del rango ingresado para el análisis, así mismo localiza los servicios que se encuentran en ejecución.

En este ejemplo se ingresa un rango de direcciones IP de la red que se está analizando, el cual se conoce gracias a la fase de recolección de información.

El resultado de la herramienta devuelve todos los hosts activos en el rango de direcciones IP ingresado, al igual que los servicios que se están ejecutando, tal y como se aprecia en la figura 22.

Net Scan

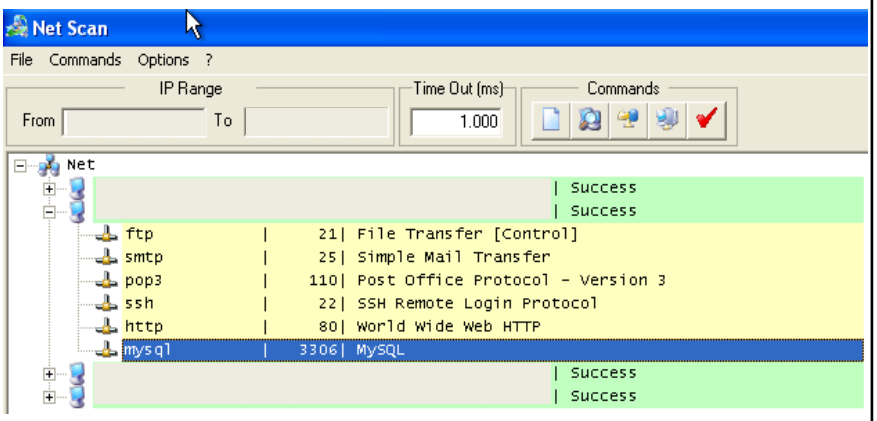


Figura 22. Uso de herramienta Net Scan
Elaborado por David Sánchez y Marco Bravo

3.3. Puertos Abiertos con Nmap

Tener puertos abierto no es malo dependiendo de qué puertos sean, la forma en que los hackers pueden obtener información de aplicaciones que están en ejecución es a través de los puertos abiertos. Por lo cual se usa la herramienta Nmap, rastrea los puertos de la máquina o máquinas en cuestión y establece si un puerto está abierto, cerrado o protegido por un cortafuego. Así, es capaz de identificar máquinas dentro de una red, determinar qué servicios utiliza dicha máquina, definir cuál es su sistema operativo e incluso devolver cierta información sobre el hardware de la máquina. (debianHackers, 2012)

Estados de puertos:

Abierto (open): (debianHackers, 2012) indica que "quiere decir que hay una aplicación aceptando conexiones TCP, datagramas UDP".

Cerrado (closed): (debianHackers, 2012) menciona que "el puerto es accesible pero no existe ninguna aplicación escuchando en él".

Filtrado (filtered): (debianHackers, 2012) define que "el paquete que se ha enviado ha sido filtrado por un firewall, reglas del router, etc y nmap no puede determinar si está abierto o no".

Sin filtrar (unfiltered): (debianHackers, 2012) indica que "quiere decir que el puerto es accesible, pero Nmap no es capaz de determinar si está abierto o cerrado".

Nmap en sistema operativo Kali Linux

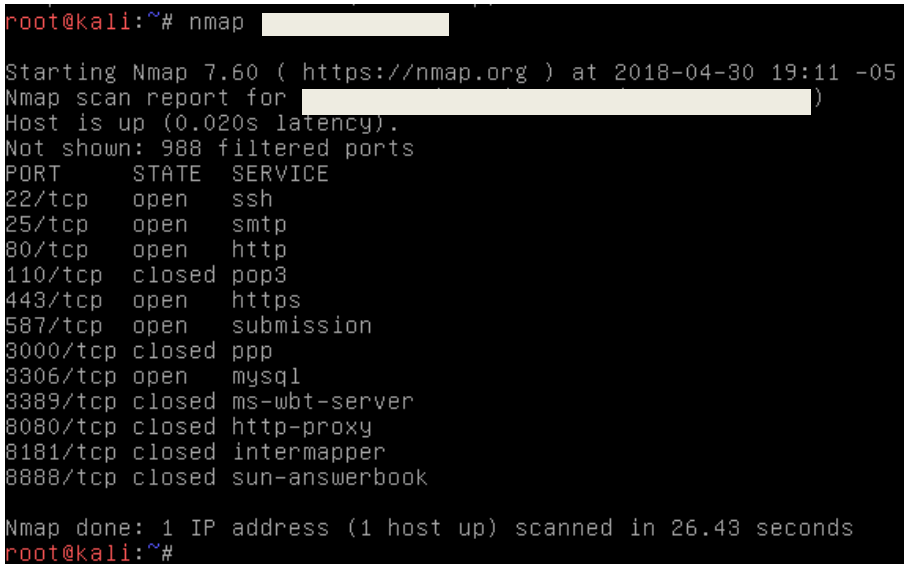


Figura 23. Puertos abiertos con la herramienta Nmap
Elaborado por David Sánchez y Marco Bravo

En este caso se usa la herramienta incorporada en el sistema operativo Kali Linux, mediante el siguiente comando: *Nmap dirección IP*

Se tiene como resultado todos los puertos abiertos y servicios en ejecución como se observa en la figura 23.

Nmap puerto 3306

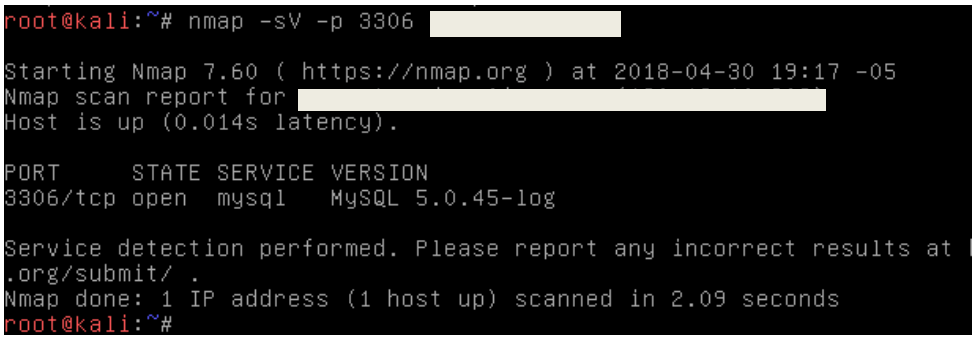


Figura 24. Descripción de versión de servicios con la herramienta Nmap
Elaborado por David Sánchez y Marco Bravo

Para conocer la versión de la aplicación en uso de los puertos encontrados, como se observa en la figura 24, se utiliza el siguiente comando: *Nmap -sV -p 3306 Dirección IP*

-sV: tipo de servicio y versión del rango de puertos abiertos

-p: puerto o rango de puertos a buscar (escanea puertos específicos)

3.3.1. Puertos abiertos con Zenmap.

Esta herramienta tiene la misma funcionalidad que Nmap, la diferencia en que es una versión interfaz gráfica que se usa en el sistema operativo de Windows y es conocida como Zenmap.

La cual se va utilizar para conocer el tiempo activo del servidor, total puertos: activos, cerrados y filtrados, el sistema operativo que utiliza, como se aprecia en la figura 25.

Zenmap de Windows

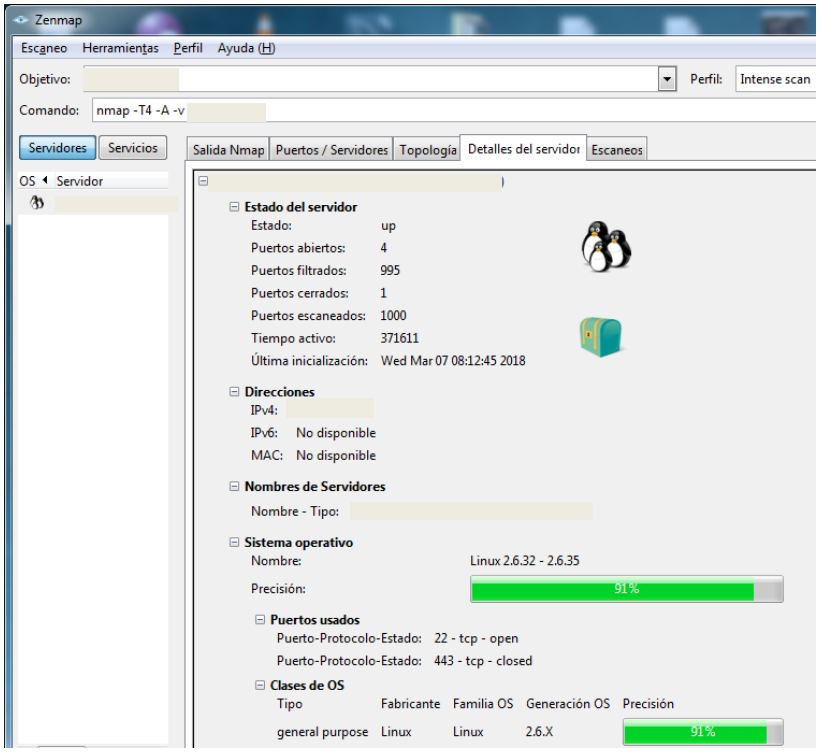


Figura 25. Interfaz gráfica Zenmap
Elaborado por David Sánchez y Marco Bravo

3.4. Evaluación de vulnerabilidades

La evaluación de vulnerabilidades permite encontrar aspectos más vulnerables en los sitios web y esto ayuda a la institución a emprender acciones oportunas y rápidas, para ello se utiliza la herramienta Nessus.

Nessus es una solución para realizar escaneo de vulnerabilidades, el cual ayuda a reducir la superficie de ataque para una organización y asegura también cumplimiento. Entre las características de Nessus se enumeran una alta velocidad para el descubrimiento de activos, auditoría de configuración, perfilamiento del objetivo, detección de malware, o incluso nubes privadas y públicas. Nessus soporta diversas tecnologías, permite escanear vulnerabilidades, amenazas y violaciones de cumplimiento en sistemas operativos, dispositivos de red, hipervisores, bases de datos, servidores web, e infraestructuras críticas. (Caballero Quezada A. E., 2016)

Nessus

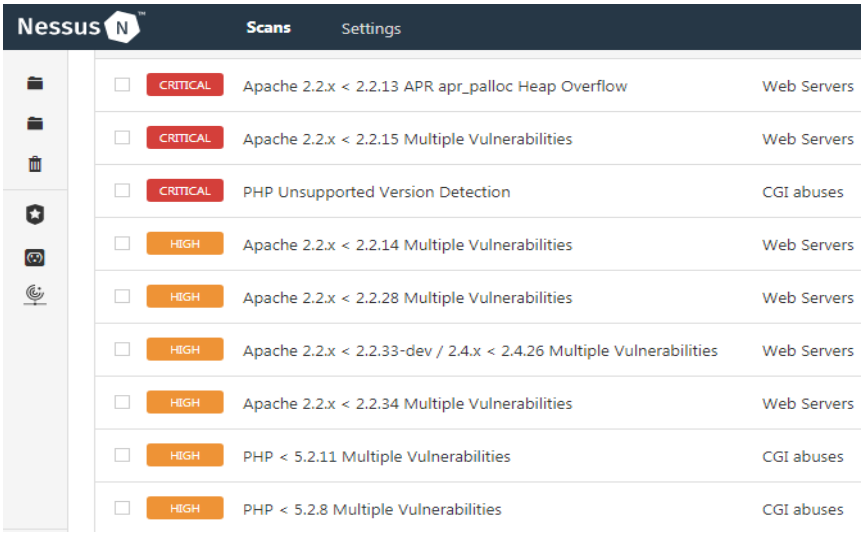


Figura 26. Vulnerabilidades herramienta Nessus
Elaborado por David Sánchez y Marco Bravo

En la figura 26, se visualiza todas las vulnerabilidades que contiene el portal web. El proceso de escaneo se realiza para todos los hosts que se encuentra asociados a la página web, con la ayuda de las herramientas antes mostradas.

Conocidas las vulnerabilidades se puede utilizar herramientas para comenzar el ataque, el cual se va a realizar en el capítulo 5 de explotación de vulnerabilidades.

3.5. Recursos no vinculados (directorios, servlets, scripts)

Los recursos no vinculados son archivos o directorio que se encuentra dentro de un sitio web, ya sea estos por olvido del mismo desarrollador de la página web, donde se puede localizar información importante como claves de acceso. Para la búsqueda de estos directorios y archivos se usa la herramienta Wfuzz que está diseñada para aplicaciones web de fuerza bruta, para encontrar recursos no vinculados, parámetros GET y POST de fuerza bruta para verificar diferentes tipos de inyecciones (SQL, XSS, LDAP, etc.), Fuzzing, etc. (Mendez, 2018)

Wfuzz

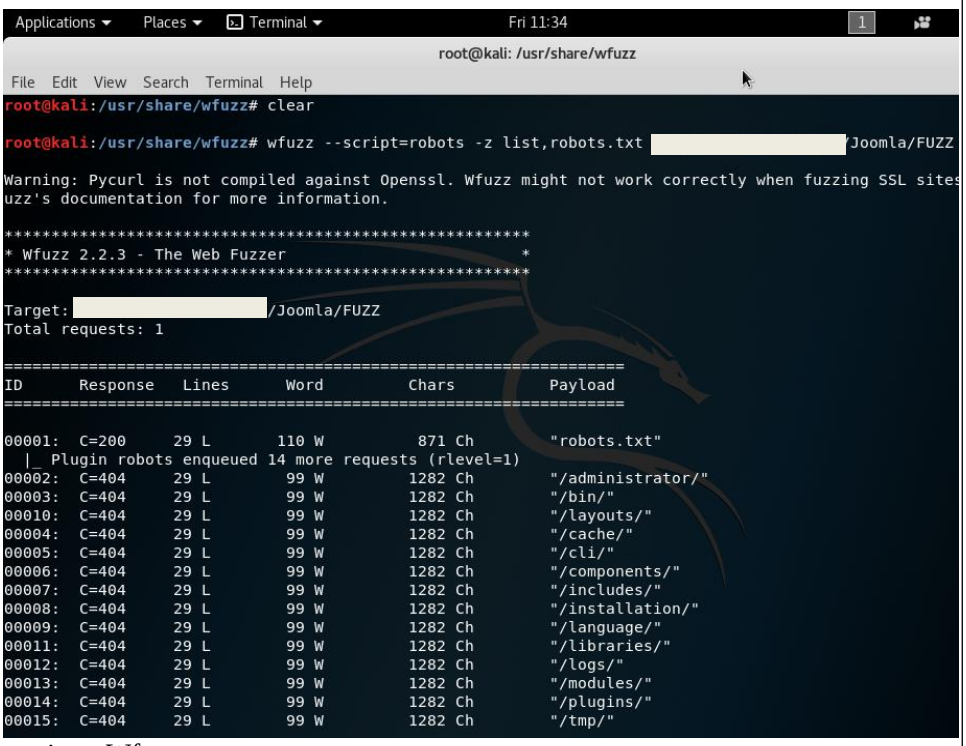


Figura 27. Herramienta Wfuuz
Elaborado por David Sánchez y Marco Bravo

Según se observa en la figura 27, la herramienta Wfuzz de fuerza bruta busca mediante diccionarios los diferentes directorios o archivos que se encuentran en el portal web, el cual es de ayuda en el capítulo 5 de explotación de vulnerabilidades realizando un análisis más profundo, donde se explica el uso de los comandos dependiendo el objetivo a escanear.

Dependiendo del diccionario utilizado se puede encontrar más directorios y métodos accesibles a la página web objetivo.

3.6. Inspección de tráfico cliente - servidor

Son las peticiones que el cliente hace al servidor y este recibe una respuesta. Para ello se usa la herramienta Paros. La función de proxy de Paros es invaluable para inspeccionar el tráfico desde y hacia un navegador. Esto permite a los desarrolladores y probadores investigar diversos aspectos de las arquitecturas de aplicaciones web, como la forma en que se establecen las cookies, las redirecciones que se envían a un navegador y las consultas enviadas desde el navegador al servidor. (Venom, 2014)

Paros

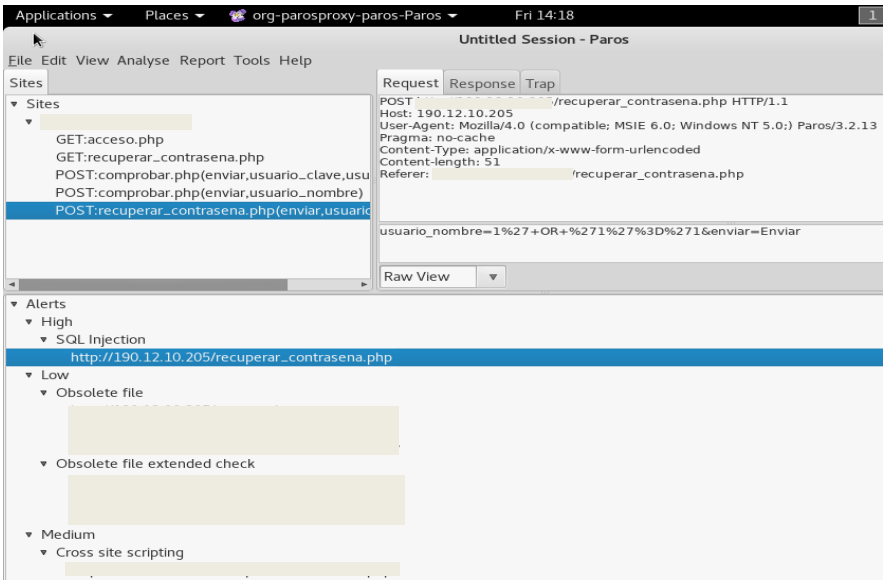


Figura 28. Herramienta Paros
Elaborado por David Sánchez y Marco Bravo

Como resultado del escaneo y análisis de la herramienta Paros se tiene: alertas prioridad vulnerabilidad de SQL Injection en el archivo recuper_contraseña.php, de prioridad baja existen archivos obsoletos, tal como se muestra en la figura 28.

Vulnerabilidades y posibles soluciones frente a brechas de seguridad.

3.7. Rastreo de correo electrónico

(Crespo, 2016) Indica que “las grandes empresas llevan a cabo esta práctica. La ubicación de un pixel en el correo electrónico (imperceptible para el usuario) permite recopilar toda la información del destinatario”.

Para ello se utiliza la herramienta o programa eMailTrackerPro, rastrea un correo electrónico mediante el análisis del encabezado, el cual contiene toda la información necesaria para rastrear de dónde proviene y dar con su verdadero punto de origen, muestra la dirección IP y la ubicación del remitente, también indica si el encabezado del correo electrónico fue manipulado. Tiene la huella de cada servidor por el que atravesó el correo electrónico, lo que en casi todos los casos llevan de vuelta al lugar donde se originó el correo electrónico. (Visualware Inc, 2014)

Cabecera de Correo

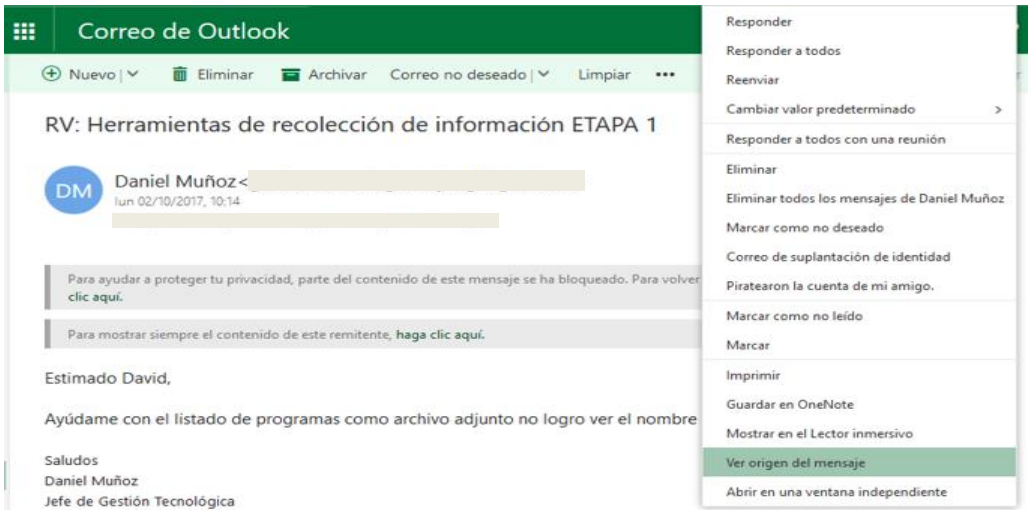


Figura 29. Cabecera de correo
Elaborado por David Sánchez y Marco Bravo

Proporciona información de contacto para el dominio origen.

Una cabecera de correo o también llamada 'encabezado de correo electrónico ' es un código fuente, el cual contiene toda la información necesaria para rastrear su origen.

En la figura 29, se puede visualizar como se obtiene la cabecera de un correo electrónico, para su posterior análisis.

Email Tracker Pro

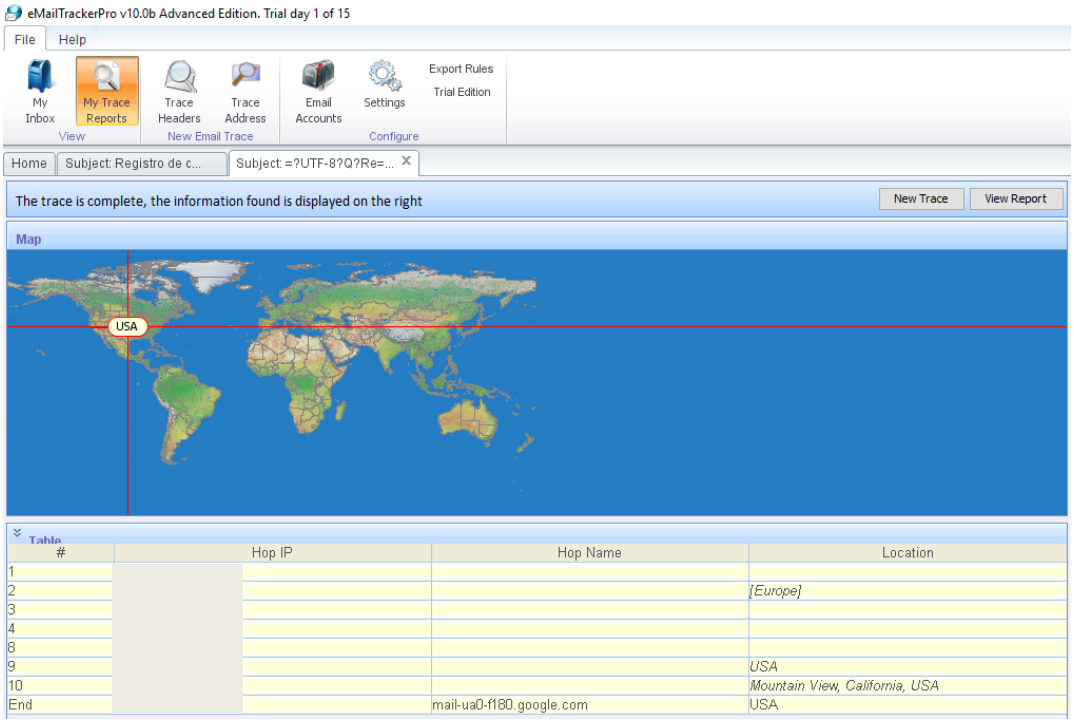


Figura 30. Herramienta eMailTrackerPro
Elaborado por David Sánchez y Marco Bravo

En la figura 30, muestra el escaneo de una cabecera de correo electrónico, en donde se verifica los saltos de servidores por los que pasó llegando hasta la dirección IP origen del servidor de correo donde se originó.

CAPÍTULO 4

ENUMERACIÓN

(Martinez & Oñate, 2017) Indica que “la fase de enumeración tiene como objetivo recolectar información relevante acerca de vulnerabilidades y hosts, aprovechando una debilidad en uno o más de los protocolos o servicios activos detectados previamente”.

4.1. Escaneo de servidores web

Para esto se usa la herramienta Nikto que ayuda al escaneo de servidores web, se encarga de efectuar diferentes tipos de actividades tales como, detección de malas configuraciones y vulnerabilidades en el servidor objetivo, detección de ficheros en instalaciones por defecto, listado de la estructura del servidor, versiones y fechas de actualizaciones de servidores, tests de vulnerabilidades XSS (Cross-Site Scripting), ataques de fuerza bruta por diccionario, reportes en formatos txt, csv, HTML, etc. (Adastra, 2011)

Nikto

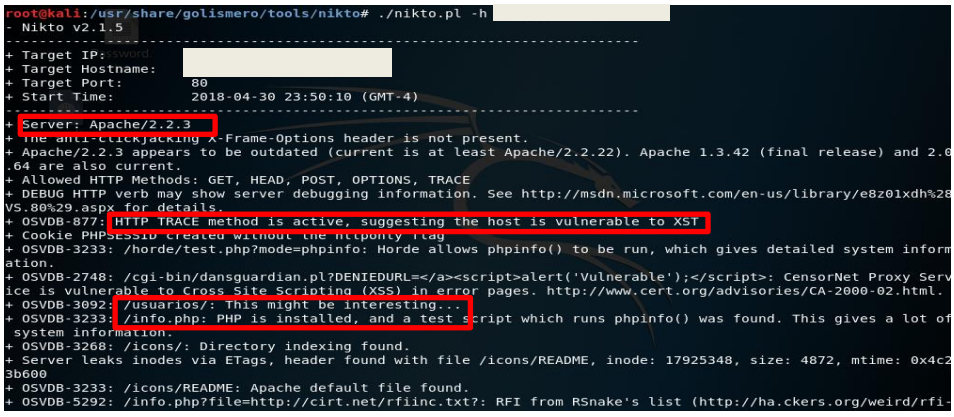


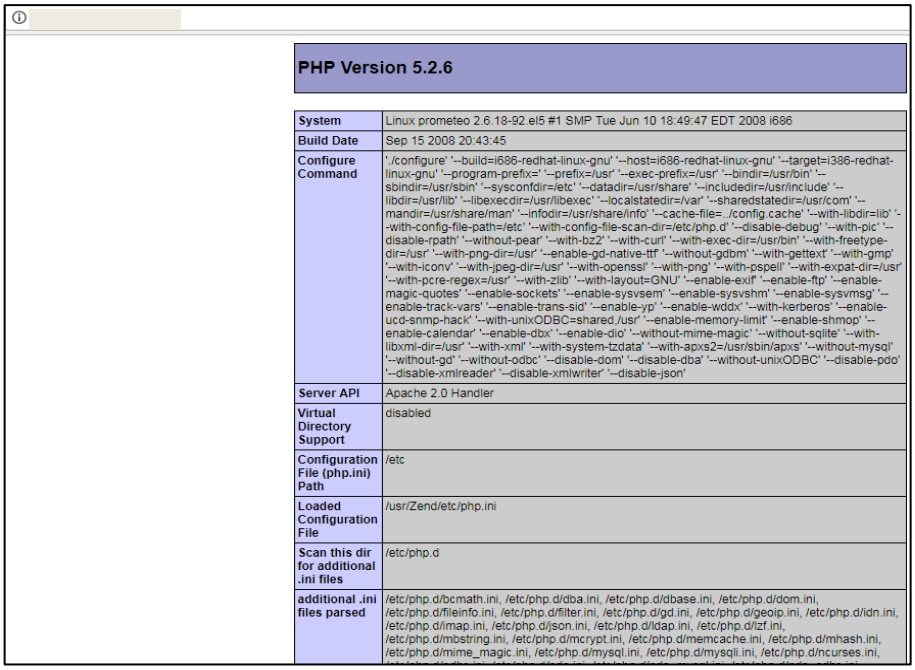
Figura 31. Resultado de la herramienta Nikto
Elaborado por David Sánchez y Marco

Con la ayuda de herramientas anteriores como Net Scan o en la página web se pudo obtener un rango de IP, que se encuentran conectadas al portal web de la institución, se procede a analizar cada una de ellas con la espera de que arroje una vulnerabilidad.

Como se puede observar en la figura 31, Nikto brinda información sobre el servidor como la versión y los distintos módulos que operan sobre el mismo.

Además, se visualiza dos directorios que pueden ser mostrados en el navegador para su análisis en busca de anormalidades.

Directorio info.php



PHP Version 5.2.6

System	Linux prometeo 2.6.18-92.el5 #1 SMP Tue Jun 10 18:49:47 EDT 2008 i686
Build Date	Sep 15 2008 20:43:45
Configure Command	'./configure' '--build=i686-redhat-linux-gnu' '--host=i686-redhat-linux-gnu' '--target=i386-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-libdir=lib' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with-bz2' '--with-curl' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-png' '--with-pspell' '--with-expat-dir=/usr' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '--enable-track-vars' '--enable-trans-sid' '--enable-yp' '--enable-wddx' '--with-kerberos' '--enable-ucd-snmp-hack' '--with-unixODBC=shared,/usr' '--enable-memory-limit' '--enable-shmop' '--enable-calendar' '--enable-dbx' '--enable-dio' '--without-mime-magic' '--without-sqlite' '--with-libxml-dir=/usr' '--with-xml' '--with-system-tzdata' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--without-odbc' '--disable-dom' '--disable-dba' '--without-unixODBC' '--disable-pdo' '--disable-xmlreader' '--disable-xmlwriter' '--disable-json'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/usr/Zend/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
additional .ini files parsed	/etc/php.d/bcmath.ini, /etc/php.d/dba.ini, /etc/php.d/dbase.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/filter.ini, /etc/php.d/gd.ini, /etc/php.d/geoip.ini, /etc/php.d/idn.ini, /etc/php.d/imap.ini, /etc/php.d/json.ini, /etc/php.d/ldap.ini, /etc/php.d/lzf.ini, /etc/php.d/mbstring.ini, /etc/php.d/mcrypt.ini, /etc/php.d/memcache.ini, /etc/php.d/mhash.ini, /etc/php.d/mime_magic.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/ncurses.ini,

Figura 32. Mostrar directorio info.php en un navegador
Elaborado por David Sánchez y Marco Bravo

El directorio info.php según (The PHP Group, 2018) "muestra información sobre el estado actual de PHP. Incluye información sobre las opciones de compilación y extensiones de PHP, versión, información del servidor, en, versión del Sistema Operativo,

rutas, valor de las opciones de configuración locales y generales, cabeceras HTTP, licencia de PHP", y el tipo de cifrado, tal como se observa en la figura 32.

Directorio /usuarios/

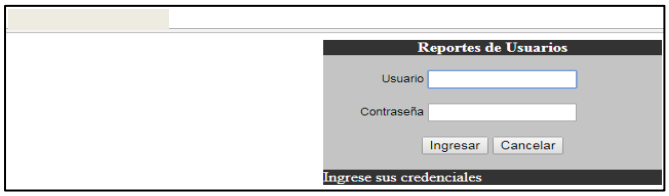


Figura 33. Mostrar directorio /usuarios/ en un navegador
Elaborado por David Sánchez y Marco Bravo

La figura 33, muestra un directorio de reportes para los usuarios, este se usará para hacer un ataque de fuerza bruta en el capítulo 5 de explotación de vulnerabilidades.

4.2. Enumeración de servidores DNS

Es el proceso de ubicación de los servidores DNS y sus correspondientes registros de una institución.

Para lo cual se va usar la herramienta DNSenum, (Caballero Quezada A. , 2014) indica que "esta permite capturar tanta información como sea posible sobre un dominio. El programa actualmente realiza las siguientes operaciones: Obtener la dirección del host (Registro A) y Obtener los servidores de nombre".

Dnsenum

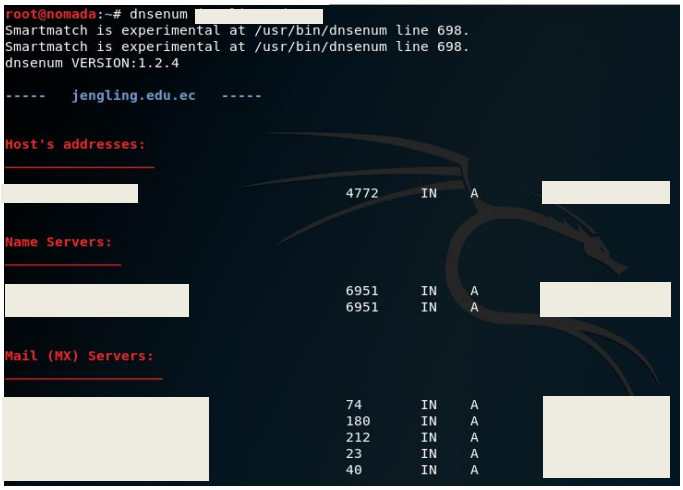


Figura 34. Herramienta dnsenum
Elaborado por David Sánchez y Marco Bravo

Esta herramienta está disponible en sistema operativo Kali Linux. El primer dato que se obtiene es la dirección del host, posteriormente se observa los servidores de nombres en el cual da a entender que proveedor de alojamiento está utilizando y por último devuelve el servidor de correo, tal como se muestra en la figura 34.

4.3. Permisos de accesos a máquinas de usuarios

En un entorno de red es importante establecer permisos de acceso y privilegios a los recursos que se comparten, para mantener un nivel de seguridad y asegurar que los recursos compartidos solo sean usados por usuarios que tengan derecho. Para ello se usa la herramienta DumpSec, (Caballero Quezada A. , 2014) menciona que "esta herramienta permite visualizar las vulnerabilidades del sistema, también está diseñada para recolectar información de usuarios y permisos aplicados en cada uno de los equipos a auditar".

Report/Dump Permisions for

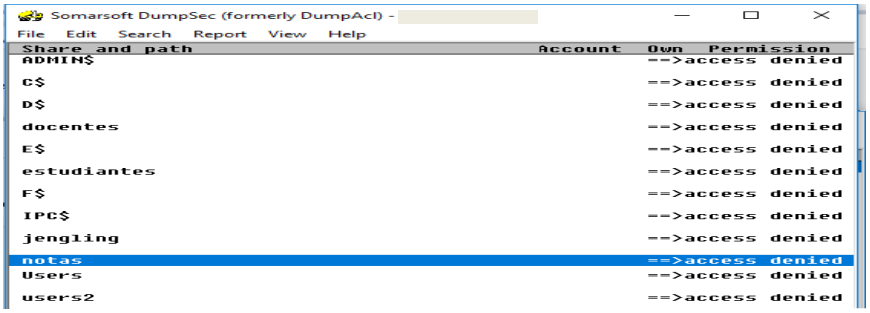


Figura 35. Carpetas compartidas mediante herramienta DumpSec
Elaborado por David Sánchez y Marco Bravo

La figura 35, muestra los recursos compartidos detallando el tipo de permisos que tiene cada directorio un directorio, el cual se usa para ver información sensible.

Report/Dump Permisions for File

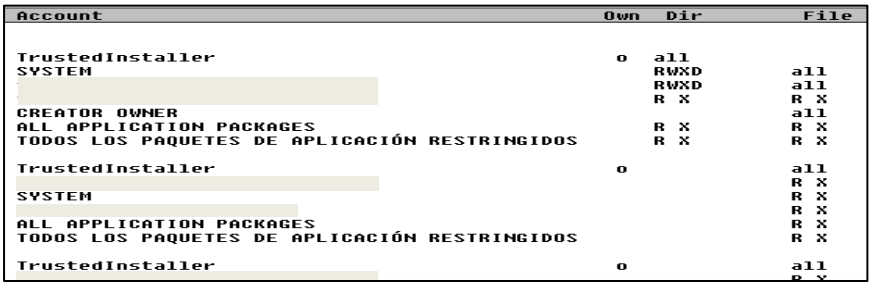


Figura 36. Permisos de directorios mediante herramienta DumpSec
Elaborado por David Sánchez y Marco Bravo

La figura 36, muestra los permisos de escritura lectura que tienen los recursos compartidos, obteniendo acceso causando una intrusión no deseada.

4.4. Recursos compartidos en una subred

Hyena reúne todas las herramientas administrativas de Windows NT, como el Administrador de usuarios, el Administrador del servidor y el Administrador de archivos / Explorador, y muchos de los componentes de MMC de Windows 2000/2003 en un único programa centralizado y fácil de usar. Hyena organiza todos los objetos del sistema, como

usuarios, servidores y grupos, en un árbol jerárquico para una administración del sistema fácil y lógica. (SDTeam, 2016)

Usuarios con herramienta Hyena

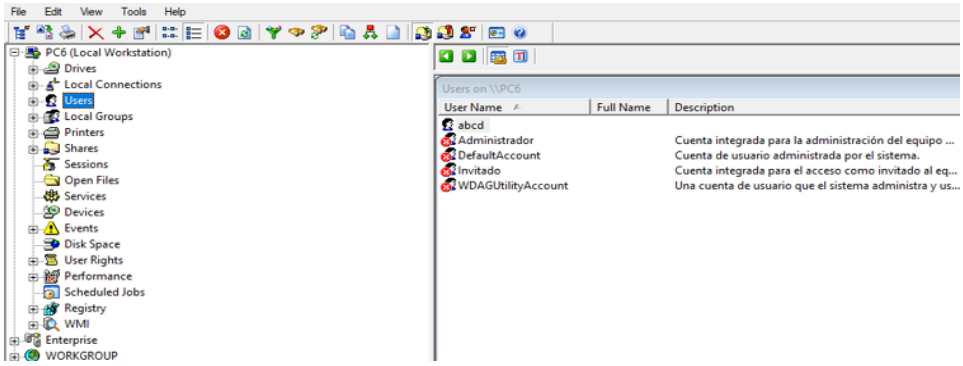


Figura 37. Usuarios mediante la herramienta Hyena
Elaborado por David Sánchez y Marco Bravo

La enumeración de usuarios es de suma importancia como se muestra en la figura 37, para entender la estructura de la organización, para de esta manera hacer un ataque de ingeniería social.

Hyena Archivos compartidos

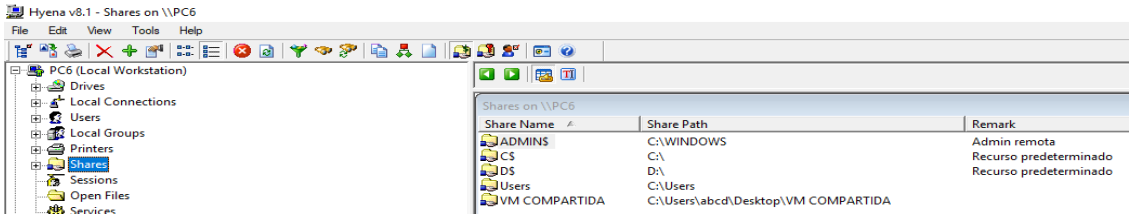


Figura 38. Archivos compartidos mediante herramienta Hyena
Elaborado por David Sánchez y Marco Bravo

Los archivos compartidos que se observan en la figura 38, muestra la ubicación en el disco duro, al igual que la forma de administración que usan para acceder, una de las mayores vulnerabilidades en la seguridad informática, es el hecho de que los usuarios dejan carpetas compartidas no por un tiempo limitado sino de forma permanente causa de un

olvido, dando como resultado la exposición de información la cual puede ser capturada por usuarios de la intranet con fines maliciosos.

4.5. Peticiones y respuestas entre cliente y servidor con OWASP ZAP

> El cliente establece una conexión con el servidor y envía un mensaje con los datos requeridos, el servidor responde un mensaje con la información solicitada, para conocer este proceso y otras funciones se usa la herramienta OWASP ZAP. Las funciones principales son: Posibilidad de comprobar todas las peticiones y respuestas entre cliente y servidor, posibilidad de localizar recursos en un servidor, análisis automáticos, análisis pasivos, posibilidad de lanzar varios ataques a la vez, capacidad para utilizar certificados SSL dinámicos, soporte para utilizar tarjetas inteligentes (DNI-e, por ejemplo) y certificados personales, análisis de sistemas de autenticación, posibilidad de actualizar la herramienta automáticamente. (Velasco, 2015)

En este ejemplo se utiliza un ataque activo, se ingresa la dirección de dominio del portal web, como se muestra en figura 39. Una vez que la herramienta comience con el ataque, este empieza hacer muchas peticiones a los diferentes archivos asociados a la página web. Como se observa en la parte izquierda la herramienta devuelve directorios, los cuales contienen archivos ocultos e incluso claves de base de datos.

Herramienta Owasp Zed Attack Proxy

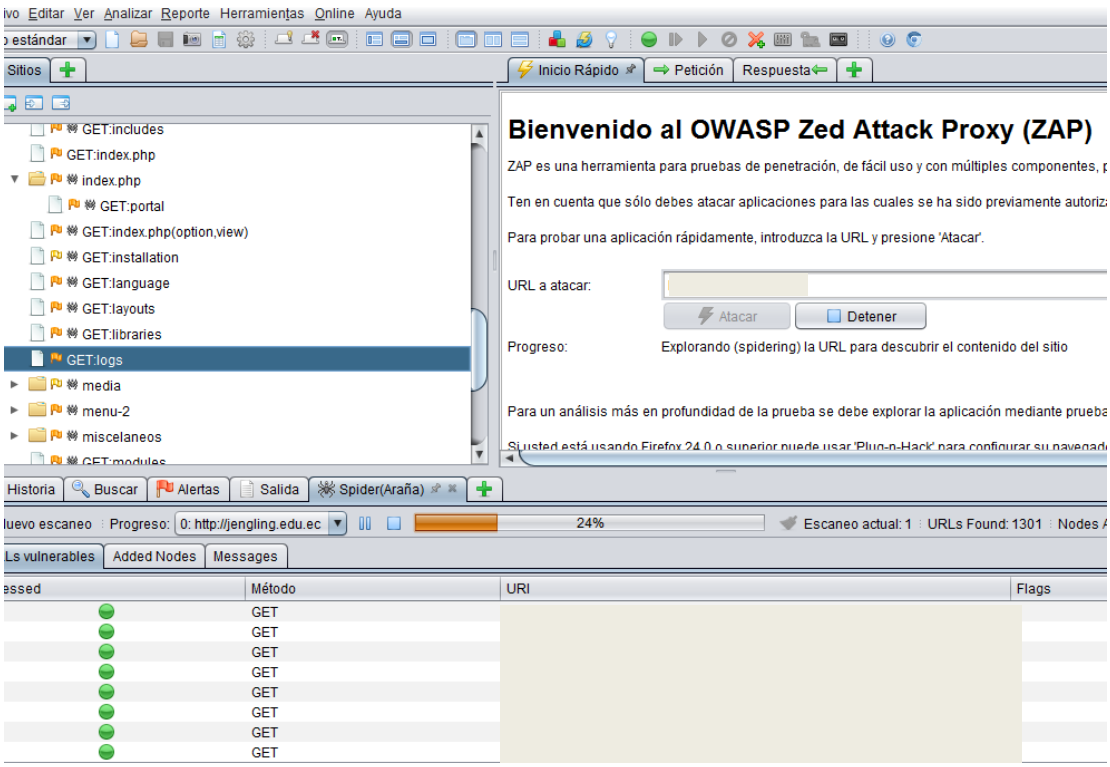


Figura 39. Escaneo activo, herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

Al final con las peticiones, se tiene como resultado 5 alertas, 1 alerta con prioridad media y 2 alertas con prioridad baja, como se observa en las figuras 40, Además con el análisis se obtuvo varias páginas de logueo, mediante esta herramienta se hace un ataque de fuerza bruta para tratar de obtener acceso a las mismas. Este ataque se realizará en el capítulo 5 de explotación de vulnerabilidades.

Herramienta Owasp Zed Attack

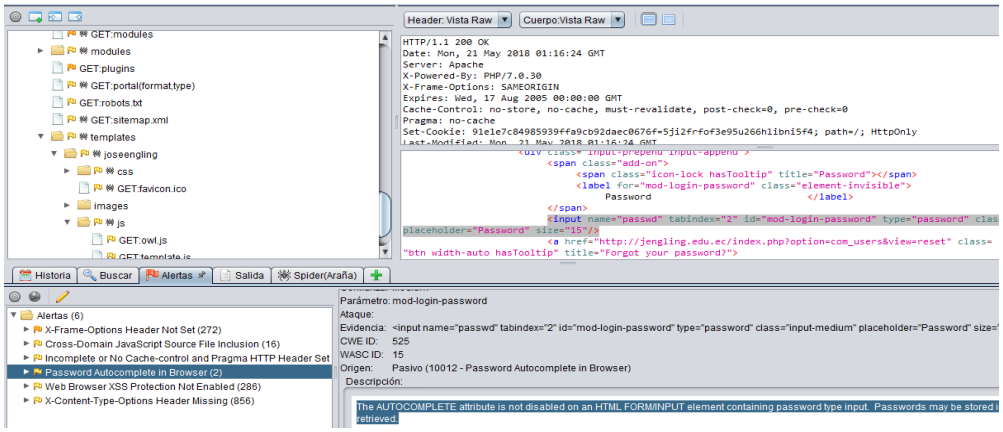


Figura 40. Resultados alertas, herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

CAPÍTULO 5

EXPLOTACIÓN DE VULNERABILIDADES (ATAQUE)

Con toda la información encontrada en los capítulos anteriores se procede a realizar el test de penetración al portal web para obtener vulnerabilidades, sin comprometer la información que se obtenga de este. Los atacantes buscan cualquier tipo de información que pueda ayudarles a perpetrar un ataque, como por ejemplo denegación de servicio, fuerza bruta o SQL Injection.

5.1. Ataque Cross Site Tracing – XST con Netcat

> La herramienta Linux Netcat a menudo se conoce como la navaja suiza de herramientas de red, y un administrador de sistemas experto podría encontrar algunos usos interesantes para esta herramienta sofisticada y versátil. En esencia, establece una conexión entre dos computadoras y permite que los datos se escriban a través de los protocolos de capa de transporte TCP y UDP. (Suri, 2017)
>
> La herramienta netcat es una de las utilidades más prácticas para el diagnóstico de redes en sistemas operativos de la familia Unix. Es capaz de leer y escribir datos a través de conexiones TCP o UDP, y al mismo tiempo posee un gran número de características que permiten crear casi cualquier tipo de conexión para depurar y explorar redes. En su uso más simple (nc host port), netcat crea una conexión TCP hacia el puerto indicado del host especificado. La entrada estándar se envía al host, y cualquier dato que llegue desde el host es volcado por salida estándar. (Linuxito, 2017)

En el capítulo 3 de SCANNING con la herramienta NMAP de la figura 22, se detectó varios puertos abiertos, en el siguiente ejemplo se utiliza el puerto 80 (http), para obtener

información del servidor web. Y con la herramienta Nikto se dio a conocer que el host víctima tiene una vulnerabilidad que son los métodos HTTP el cual se encuentran activo, esto significa que puede ser propenso a un ataque Cross Site Tracing – XST más conocido como HTTP TRACE, con este ataque se llega a mostrar las cookies que el navegador tiene para el dominio.

(Stuxnet, 2013) indica que "el método HTTP TRACE se utiliza para debugging de los servidores Web y viene activado por defecto en muchos servers. Lo único que hace es repetir toda información enviada por el cliente al server, es algo así como un echo".

Para el uso de la herramienta Netcat se utiliza el sistema operativo Kali Linux, y para establecer una conexión con un servidor HTTP, se ingresar nombre de host o IP y el puerto (en este caso el puerto 80), como se muestra en la figura 41.

La opción **-vv** se utiliza para establecer la conexión TCP/IP contra el servidor HTTP.

El resultado de la conexión es exitoso y se obtiene como resultado la resolución DNS.

Netcat

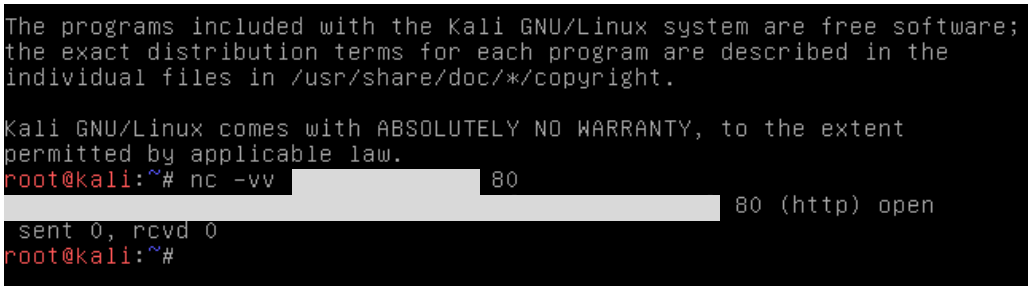


Figura 41. Establecer conexión con el host, herramienta Netcat
Elaborado por David Sánchez y Marco Bravo

Obtener el recurso raíz (/) a través del método HTTP GET:

Netcat Cabecera http

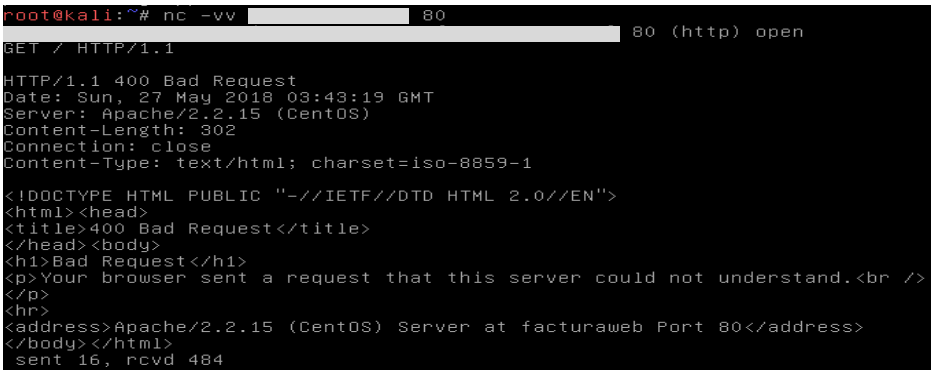


Figura 42. Obtener cabecera http
Elaborado por David Sánchez y Marco Bravo

Con los resultados obtenidos se tiene:

La primera sección del mensaje de respuesta, incluye el código de retorno del protocolo HTTP, junto con las cabeceras de respuesta.

Como se observa el servidor retorna el código 400 que se reducen a peticiones incorrectas del lado del cliente o contenidos eliminados, en otras palabras, el método HTTP TRACE regresa la petición enviada al servidor con todas las cabeceras que se le solicita al mismo, en este caso no se tuvo éxito. Si un atacante tiene éxito podría enviar código malicioso en JavaScript que el navegador puede interpretar.

En la segunda sección se tiene el contenido correspondiente al recurso solicitado, además, se conoce que se encuentra en un servidor Apache con un sistema operativo Linux, como se muestra en la figura 42.

5.2. Ataque de fuerza bruta con OWASP ZAP

En el capítulo 4 de enumeración se dio conocer el funcionamiento de esta herramienta donde se obtuvo información relevante, se encontró páginas de acceso a información con una plantilla de inicio de sesión, la cual requiere de un usuario y contraseña para el acceso de la misma.

Con el uso de esta herramienta se realiza una explotación a dicha página, el proceso se llama fuerza bruta el cual consiste en el uso de diccionarios (archivos .txt que contiene una gran cantidad de claves posibles).

Ejecutar la herramienta OWASP ZAP, ya que ZAP crea un proxy por defecto. En el navegador se configura el proxy con los mismos datos de la herramienta, para que el tráfico de navegación se lo vea reflejado en el OWASP ZAP.

Una vez configurado el navegador, se procede a cargar la página de inicio de sesión a la cual se va hacer el ataque de fuerza bruta, figura 43.

Página de inicio de sesión

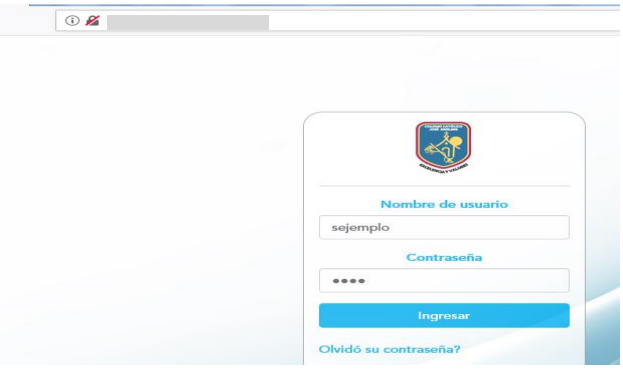


Figura 43. Página asociada de inicio de sesión
Elaborado por David Sánchez y Marco Bravo

En esta página se ingresa cualquier usuario y contraseña, lo que va a pasar es que va arrojar un mensaje de error de autenticación, pero se obtiene una petición de tipo POST del

servidor, la cual va hacer capturada por la herramienta OWASP ZAP en donde se observa la información enviada en los campos de usuario y contraseña.

En la herramienta OWASP, en la parte superior derecha de la Figura 44 se tiene la pestaña Petición, aquí muestra la cabecera y datos que se envió al servidor y en la parte inferior se observa la contraseña y usuario ingresados.

Herramienta Owasp Zap

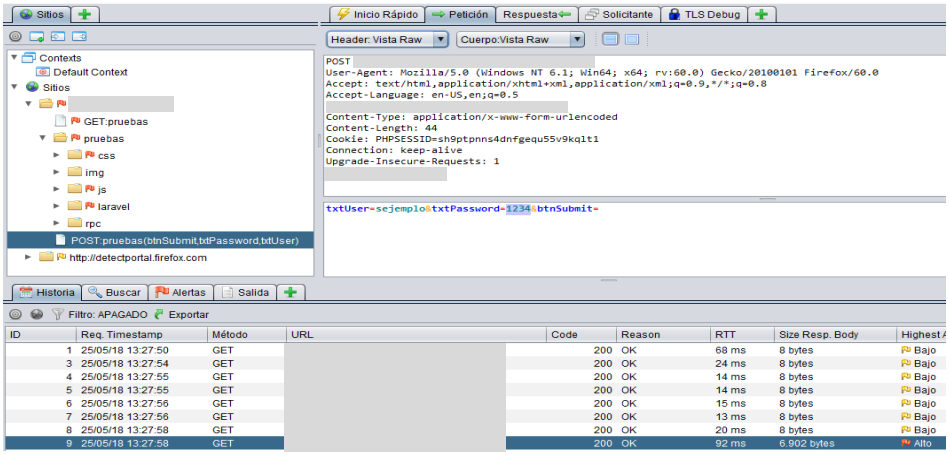


Figura 44. Petición post al servidor, mediante herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

Y si se observa la pestaña de Respuesta de la figura 45, se tiene que como resultado el servidor devuelve una página HTML (formulario) donde indica una alerta de que los datos de usuario Ingresados no son válidos.

Herramienta OWASP ZAP

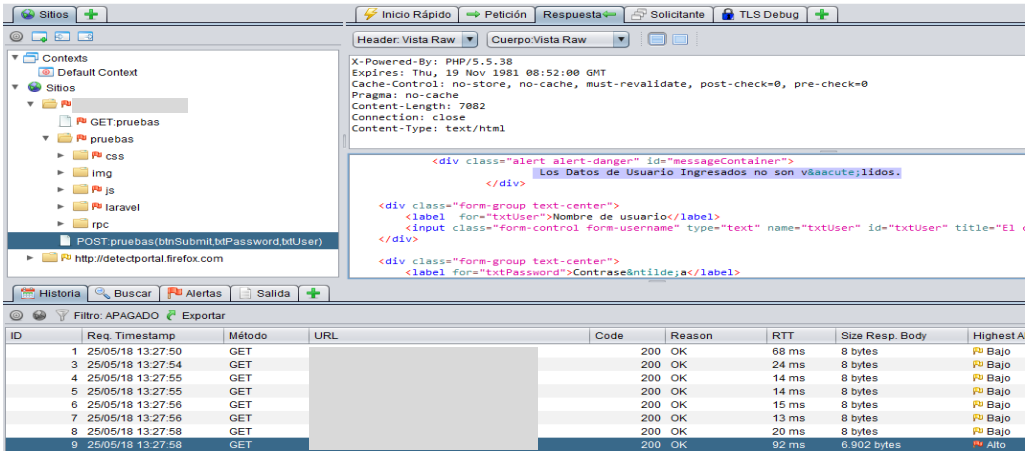


Figura 45. Respuesta del servidor a la petición post, herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

Lo que interesante en esta ventana es la pestaña de Petición, ya que es aquí donde se va a editar el campo usuario y password para enviar un archivo de texto (diccionario de claves) al servidor realizando peticiones, con la ayuda de la herramienta este proceso se realizará automáticamente, la cual va ir comparando una por una (clave y usuario) hasta hallar a la correcta, cabe recalcar que este proceso lleva tiempo ya que se necesita usar varios archivos con distintas contraseñas y usuarios y estos archivos contienen solo posibles claves y usuarios que se usan con frecuencia, es aquí donde se va a determinar cuan segura es esta página, a todo este proceso se le conoce como Fuzzing.

Paso siguiente se procede a atacar mediante la herramienta incorporada Fuzz, como se observa en la figura 46.

Herramienta Owasp Zap

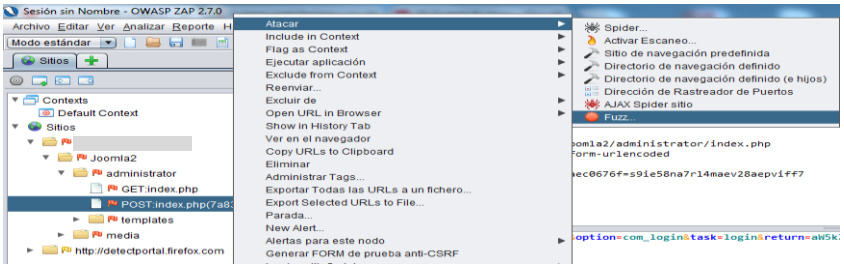


Figura 46. Ataque de fuerza bruta, mediante herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

En esta ventana se selecciona el campo a cargar el archivo, en este caso es la clave (passwd), una vez seleccionado se hace click en el botón Add y se selecciona el archivo, tal como se observa en la figura 47.

Herramienta Owasp Zap

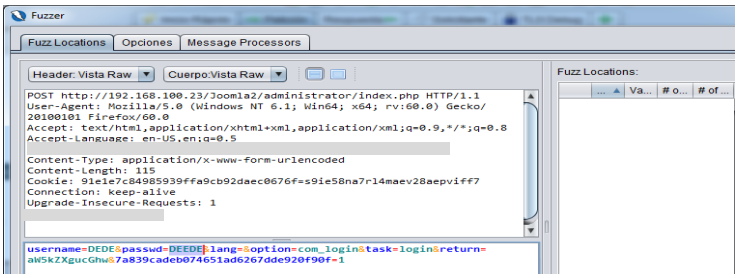


Figura 47. Fuzzer, mediante herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

Cuando los dos archivos se carguen para cada campo se tendrá un resultado de la siguiente manera, figura 48:

Herramienta Owasp Zap

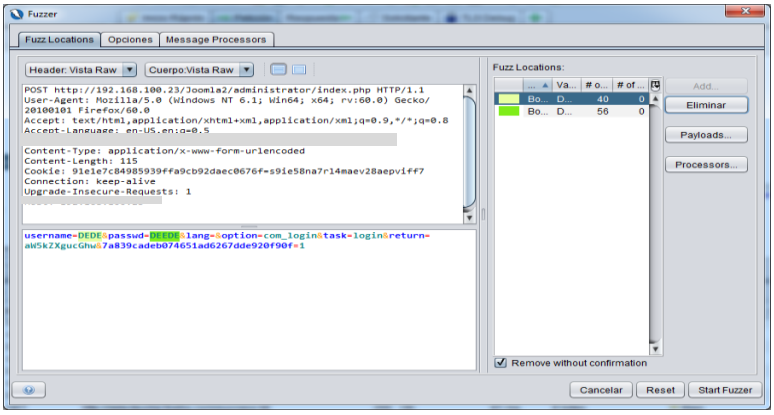


Figura 48. Carga de Diccionarios, mediante herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

Finalmente muestra el resultado final del proceso de comparación de claves con los usuarios.

¿Cómo saber cuál de las claves de la lista es la correcta?

En la figura 49, se observa el tamaño de byte para el cuerpo de cada comparación la que es distinta a las demás es la clave correcta.

Herramienta Owasp Zap

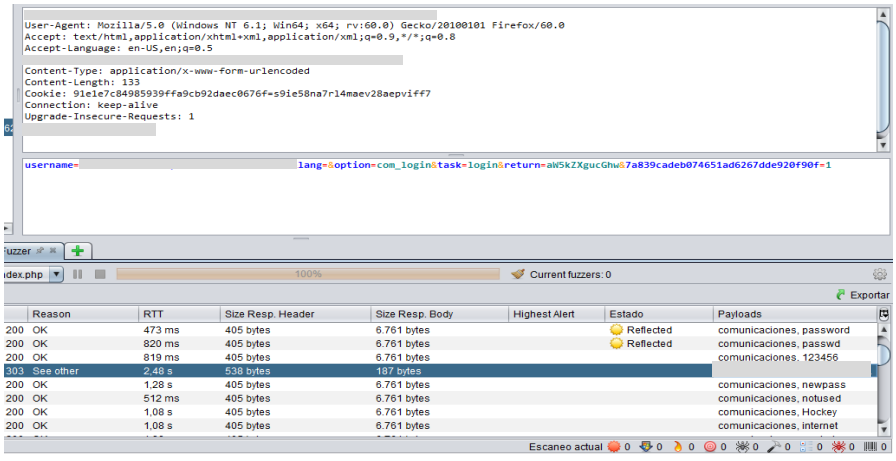


Figura 49. Fuerza Bruta, mediante herramienta Owasp Zap
Elaborado por David Sánchez y Marco Bravo

5.3. Ataque de Inyección SQL

Para que se produzca un ataque de inyección SQL, el sitio web vulnerable debe incluir directamente la entrada del usuario dentro de una declaración SQL. Un atacante puede insertar una carga útil que se incluirá como parte de la consulta SQL y se ejecutará contra el servidor de la base de datos. (Acunetix, 2013)

Existen páginas con formularios de inicio de sesión. Para realizar el ataque de SQL Injection se va a tomar una de estas e intentar hacer un bypass (forzar el ingreso en el inicio de sesión).

El formulario de inicio de sesión que se va usar para realizar el ataque, contiene 2 campos de entrada (nombre de usuario y contraseña).

Figura 50. Página de inicio de sesión que se va aplicar SQL Injection
Elaborado por David Sánchez y Marco Bravo

La funcionalidad que tiene la página de inicio de sesión es enviar una petición al servidor, el cliente ingresa su usuario y contraseña, la cual valida que exista en la base de datos y el servidor devuelve una respuesta que si existe el usuario y contraseña le da acceso a la página.

Por poner un ejemplo, se va crear una base de datos ficticia para que sea más entendible en un gestor de base de datos similar a la del host.

¿Cómo saber que gestor de base de datos maneja?

Esto se lo puede saber mediante la herramienta Nmap ya que se obtuvo el puerto abierto de la base de datos tal como se visualiza en las figuras 22 y 23.

El cliente ingresa:

Usuario: csuarez

Clave: 1234

¿Qué es lo que hace internamente la validación en la base de datos?

Recibe estos dos campos y mediante una sentencia SQL valida que los datos sean correctos.

SQL Injection

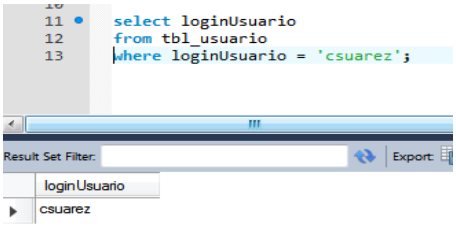


Figura 51. Sentencia SQL en base de datos MySql
Elaborado por David Sánchez y Marco Bravo

Como se observa en la figura 51, la base de datos recibe estos datos y los compara que existan en la tabla llamada tbl_usuario, se va usar solo el usuario, en la base de datos este dato es ingresado entre comillas simple ya que pertenece a un texto y esta es la forma como hace la búsqueda, como resultado devuelve un campo dando a conocer que si existe este usuario.

En el inicio de sesión de la página atacante no se conoce el usuario ni la contraseña para obtener el acceso, es por eso que en los inputs (usuario y contraseña) se va ingresar una sentencia SQL con una validación que siempre va hacer verdadera y permita acceder.

SQL Injection

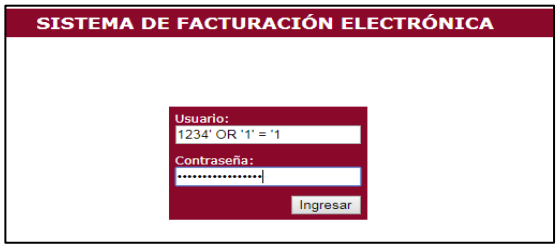


Figura 52. Sentencia SQL en los campos input de la página de inicio de sesión
Elaborado por David Sánchez y Marco Bravo

En la figura 52 se puede visualizar la sentencia SQL que va enviar al servidor de base de datos, lo que hace esta sentencia es cambien la estructura de la consulta.

SQL Injection

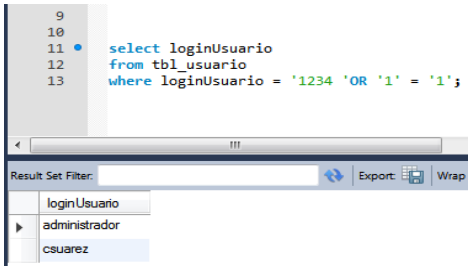


Figura 53. Sentencia SQL modificada en la base de datos
Elaborado por David Sánchez y Marco Bravo

Se regresa a la base de datos ficticia y se prueba esta sentencia, como se observa en la figura 53, la estructura cambia y se tiene dos validaciones una es por el usuario que sea igual a “1234” y por otro lado que “1” sea igual a “1”, que por lógica esta validación siempre va hacer verdadera.

Ahora con esta sentencia se va tratar de obtener acceso a la página siguiente o principal del inicio de sesión.

SQL Injection

Figura 54. Ingreso a la página principal por medio de SQL Injection
Elaborado por David Sánchez y Marco Bravo

La figura 54 muestra que la sentencia tuvo éxito y permite acceso por ataque de SQL Injection, cabe recalcar que este método no es funcional en todas las páginas que contengan acceso por inicio de sesión, ya que en algunas tienen por parte del desarrollador o ya sea por la herramienta de programación misma protección a estos ataques.

5.4. Análisis de archivos obsoletos con Paros

Una vez logrado el acceso por usuario a la página web del servidor de facturación se escanea y analiza con la herramienta Paros de Kali Linux.

Para la ejecución de esta herramienta se debe colocar el navegador web en modo escucha configurando manualmente en la opción de preferencias, proxy de red, Configuración manual del proxy: 127.0.0.0 puerto 8080, de esta manera la herramienta Paros esta lista para escanear los directorios y archivos de la página web.

Se ingresa a la página web a la cual se va a escanear, en este caso se empieza con la página de logueo, ingresando con el usuario encontrado en el SQL Injection figura 54, mientras que Paros automáticamente guarda las páginas web por donde se navega, capturando los métodos GET y POST de las plantillas web, como se visualiza en la figura 55, en esta

prueba se obtiene varias plantillas .php las cuales se las analiza mediante el siguiente proceso:

Ir a la opción de menú Analyce /Scan all.

Scan all: obtiene el código de las peticiones request, response.

Luego ir a la opción de menú Analyce /Spider

Paros escaneo y análisis

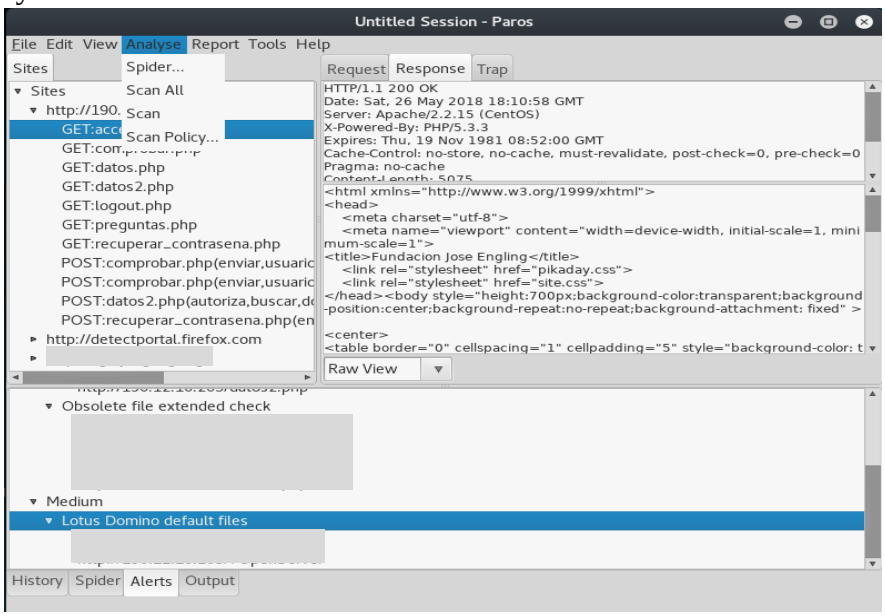


Figura 55. Escaneo y análisis a la página web por medio de la herramienta Paros
Elaborado por David Sánchez y Marco Bravo

Spider: captura las peticiones anteriormente encontradas y analiza las vulnerabilidades clasificándolas en alertas de tres tipos bajas, medias y altas.

En el escaneo y análisis de la herramienta Paros se encuentra una vulnerabilidad de archivos obsoletos, en los cuales se puede visualizar su contenido.

En este caso como se puede observar en la figura 55, muestra varios formularios.php, los cuales tienen código de programación y se puede visualizar por medio del prefijo ~ desde cualquier navegador web de forma local o pública.

El objetivo de este ataque, es analizar el código fuente para ver si existe información sensible expuesta como se puede verificar en las figuras 56, 57 y 58.

Información formulario comprobar.php:

Análisis de archivos obsoletos

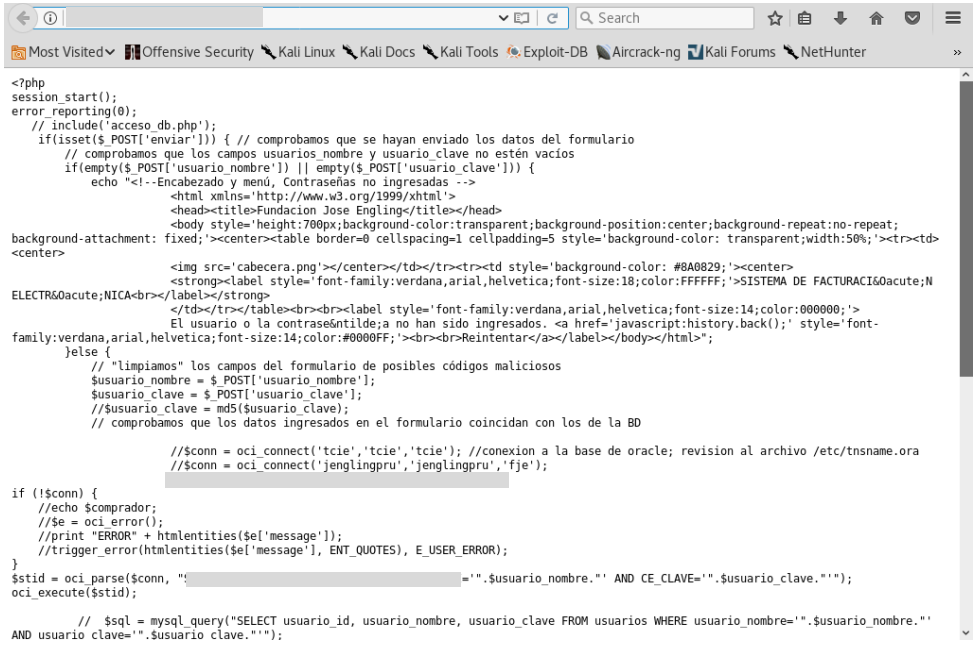


Figura 56. Análisis del código fuente de archivo obsoleto número 1
Elaborado por David Sánchez y Marco Bravo

En la figura 56 se observa que el archivo obsoleto encontrado comprobar.php contiene información sensible.

Este formulario es el que controla y permite el acceso a los usuarios para la visualización de documentos de facturación, página principal de logueo.

Información formulario datos2.php:

Análisis de archivos obsoletos

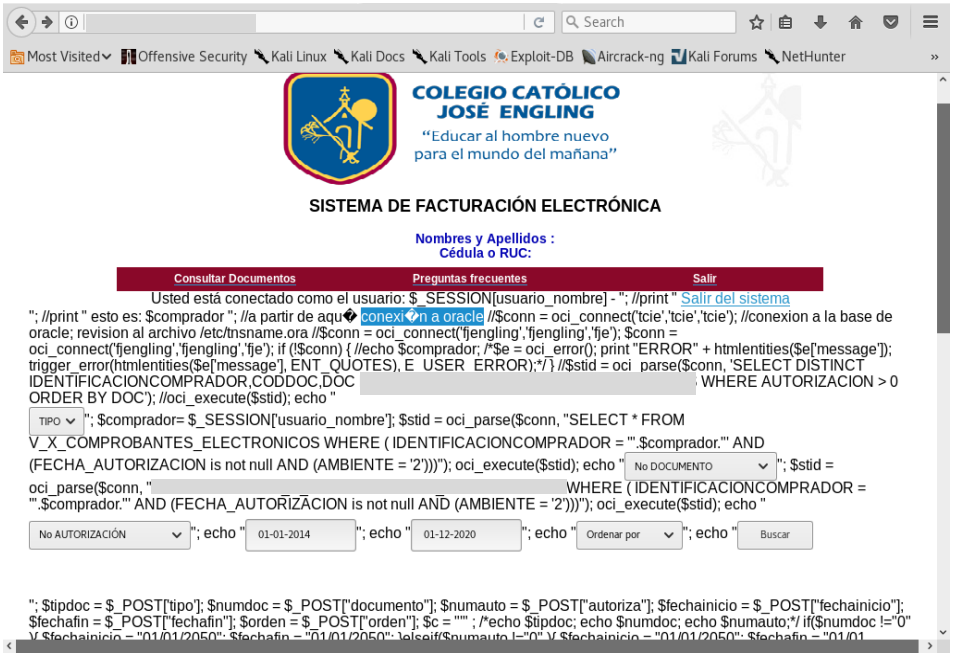


Figura 57. Análisis del código fuente de archivo obsoleto número 2
Elaborado por David Sánchez y Marco Bravo

En la figura 57 se observa que el archivo obsoleto encontrado datos2.php contiene información sensible.

Este formulario es de consulta de información a los usuarios, permite la visualización y descarga de documentos de facturación, página posterior a la autentificación del usuario obtenido en el SQL Injection de la figura 54.

Información Formulario recuperar_contrasena.php:

Análisis de archivos obsoletos

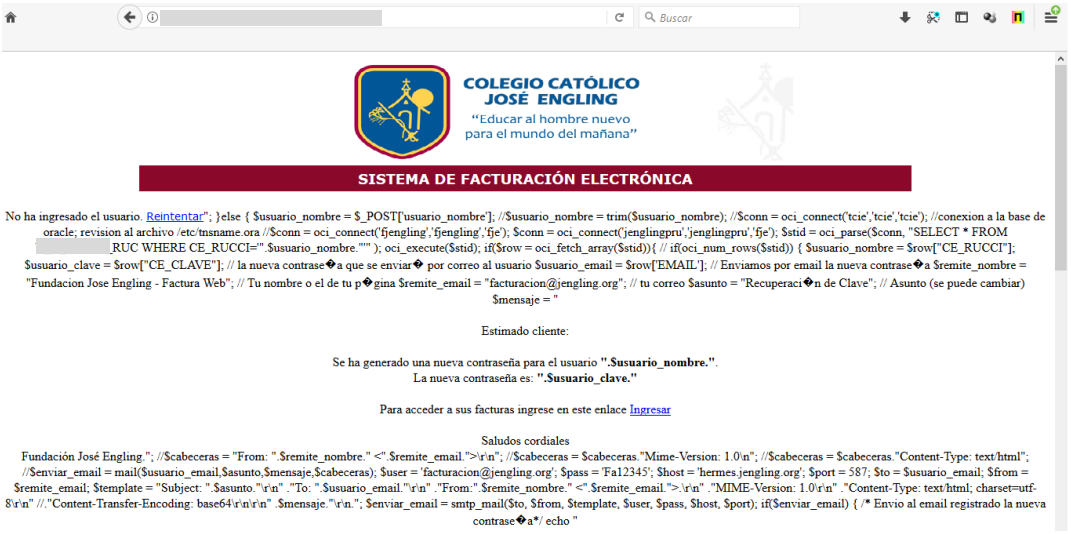


Figura 58. Análisis del código fuente de archivo obsoleto número 3
Elaborado por David Sánchez y Marco Bravo

En la figura 58 se observa que el archivo obsoleto encontrado recuperar_contrasena.php contiene información sensible.

Este formulario es de recuperación de usuarios y contraseñas como es común en toda página web institucional que tenga un sistema de autentificación, el cual permite su recuperación mediante el envío de email a la cuenta del propietario, página alterna a la del logueo del usuario.

En este caso mediante el análisis del código fuente se encuentra el método de reseteo de contraseña, el cual muestra un algoritmo que realiza de forma automática el reenvío de emails a los usuarios que quieren recuperar su usuario o clave, comprometiendo de esta manera el email de la institución con su contraseña e información sensible, como es el código del cuerpo del correo de respuesta de recuperación de cuenta.

Mediante la herramienta Email Traker en las figuras 29 y 30, se verificó el origen y la autenticación del servidor de correo institucional que se encuentra en el presente ataque de explotación.

Paros Reportes

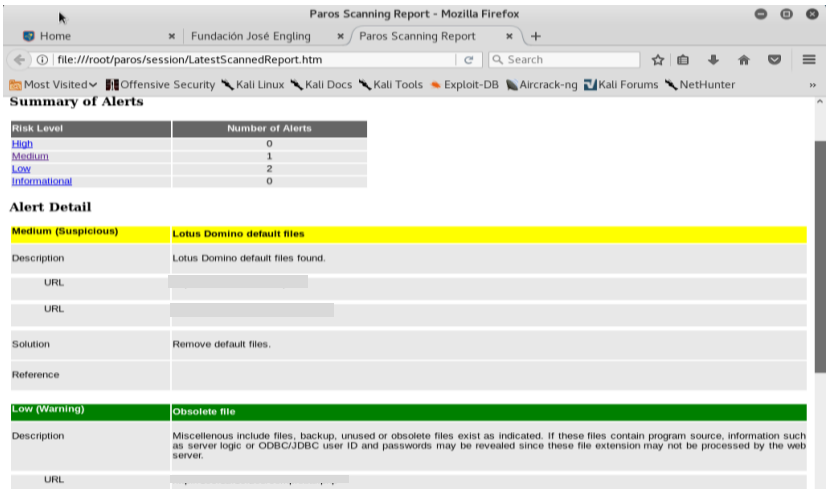


Figura 59. Reportes con la herramienta Paros
Elaborado por David Sánchez y Marco Bravo

La herramienta Paros tiene la característica de generar reportes, como se puede observar en la figura 59, los formatos de reportes que puede sacar dicha herramienta son: html, xml. Con los reportes generados se puede observar el total de alertas y el detalle de cada una.

Clasificación de vulnerabilidades de la herramienta Paros:

- Low(advertencia)
- Medium (Sospechosa)
- High (Críticas).

Adicional, en el detalle de alertas especifica una descripción, el URL o dirección donde está la vulnerabilidad, y la solución recomendada por la herramienta Paros y la referencia,

la cual es la fuente bibliográfica donde se puede consultar dicha solución de una forma más detallada.

En este caso existen vulnerabilidades de clase media, las cuales son archivos obsoletos donde se puede encontrar:

- Código comentado
- Conexiones a Base de datos
- Sentencias de consultas, métodos, funciones y procedimientos a la base de datos.
- Métodos de autentificación
- Información sensible de la institución

5.5. Ataque de DoS con Wfuzz

Mediante la herramienta Wfuzz citada en el capítulo de Scanning se procede a usar la herramienta Wfuzz en Kali Linux para enlistar archivos por defecto que se encuentren en el servidor Web:

Wfuzz escaneo de directorios y archivos

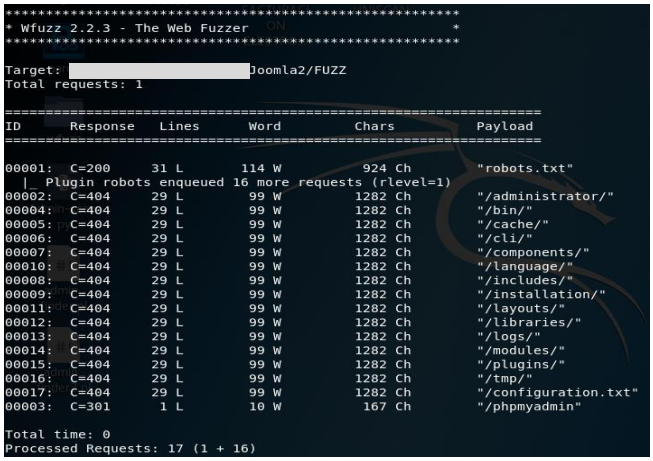


Figura 60. Escaneo de directorios y archivos mediante Wfuzz
Elaborado por David Sánchez y Marco Bravo

Para el análisis se implementó el comando:

Wfuzz --script=robots -z list, robots.txt http://1XX.XX.XX.XX4/Joomla/FUZZ

con los siguientes parámetros aplicados:

-z : carga útil: especifique la carga útil (tipo, parámetros, codificación).

list :use la lista de palabras ó un archivo de tipo listado.

/FUZZ : sección donde se desea realizar la búsqueda.

robots,txt : script ó diccionario de palabras.

Como se muestra en la figura 60, se encontró los archivos de configuración web configuration.txt y robots.txt, los cuales se puede acceder desde cualquier navegador de forma local y remota.

Archivo configuration.txt

Archivo encontrado hace referencia a la configuración de Joomla el cual muestra:

Tipo de base de datos, usuario de base de datos, clave de la base de datos, nombre de la base de datos como se visualiza en la figura 61.

Archivos de configuración

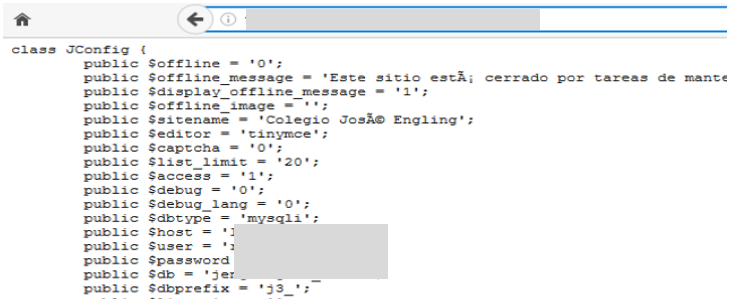


Figura 61. Análisis de archivos encontrados mediante Wfuzz
Elaborado por David Sánchez y Marco Bravo

Archivo robots.txt

Archivo donde muestra los directorios por defecto.

Archivo robots.txt

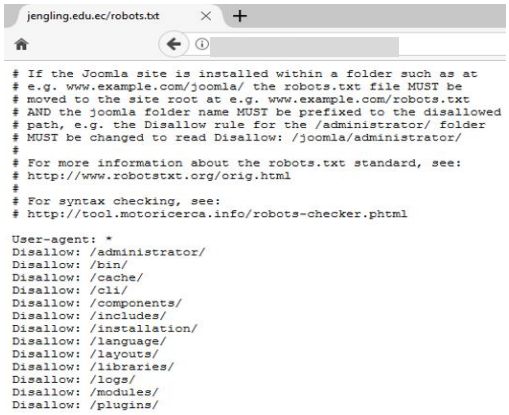


Figura 62. Análisis de directorios encontrados mediante Wfuzz
Elaborado por David Sánchez y Marco Bravo

Directorio encontrado (administrator), el cual muestra el panel de administración web de Joomla como se visualiza en la figura 62.

Con el uso de la herramienta Wfuuz se enlista directorios mediante el diccionario admin-panels donde se encuentran la gran moyoría de panales de administración web.

Directorio phpmyadmin encontrado en el escaneo Wfuzz de directorios y archivos como se visualiza en la figura 60, el cual para este caso se tiene acceso local y público, sirve para la administración de la base de datos.

El siguiente paso es el logueo a phpmyadmin(administración de la base de datos). Mediante la direccion *http://IPobjetivo/phpmyadmin/*

Accesos otorgados:

Estructura, backup, registros, permisos de escritura lectura y ejecución en la base de datos de forma remota como se puede visualizar en la figura 63.

Phpmyadmin

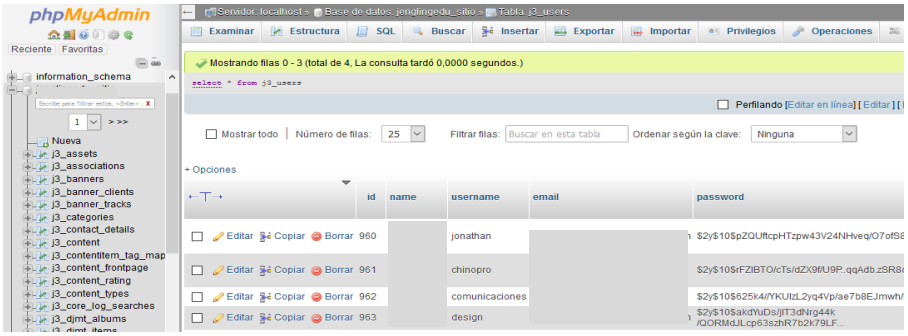


Figura 63. Administración de bases de datos por medio de phpmyadmin
Elaborado por David Sánchez y Marco Bravo

En esta figura se visualiza la consulta a la base de datos de la tabla de usuarios de Joomla, con el objetivo de editar y cambiar los campos de email y password (clave) de uno de los super usuarios para poder obtener acceso al administrador de Joomla de la pagina web.

Se prueba que Joomla cifra sus claves de usuarios en la tabla users en la base de datos, para tener mayor seguridad en el acceso a su panel de administración como se puede visualizar en la figura 61 en el campo password.

MD5 Online

Para poder saltar esta seguridad se procede a editar el campo de clave (password) colocando una clave cifrada en otro tipo de codificación, en este caso se usa cifrado MD5 mediante herramintas Online como se describe a continuación en la figura 64.

MD5 Online

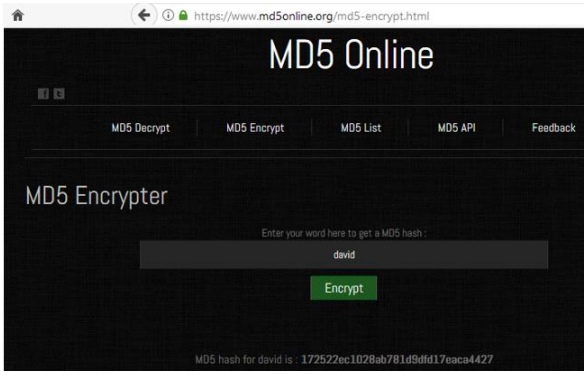


Figura 64. Herramienta Online de encriptación y desencriptación de cifrado md5
Elaborado por David Sánchez y Marco Bravo

Se cifra una clave cualquiera mediante la herramienta Online de encriptación a MD5, en esta caso se escoge la palabra david.

Encriptación a MD5:

david equivalente a **172522ec1028ab781d9dfd17eaca4427**.

En la herramienta de adminsitración de base de datos phpmyadmin se edita y reemplaza la clave actual localizada en el campo password, por la nueva clave cifrada en MD5 de la figura 64.

Phpmyadmin cambio de clave

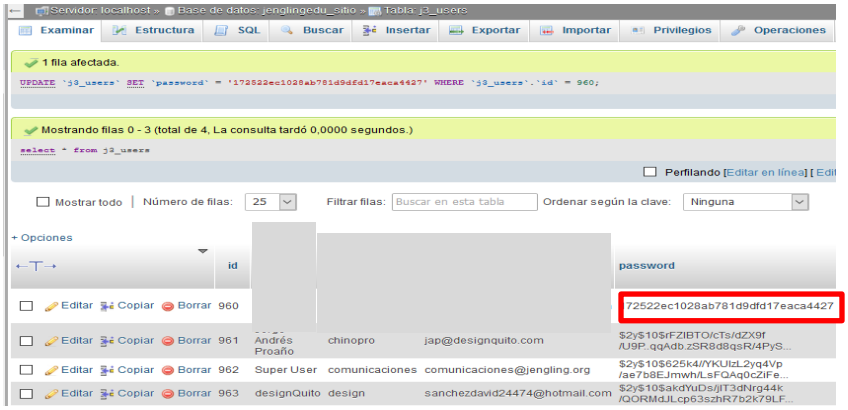


Figura 65. Actualización de campos mediante herramienta phpmyadmin
Elaborado por David Sánchez y Marco Bravo

En la figura 65, se realiza el cambio de clave a un super usuario con el objetivo de tener acceso al panel de administración de Joomla.

Se encuentra una vulnerabilidad de Joomla, que se resume en el cambio de clave por otra, en cualquier tipo de cifrado dentro de la base de datos, se debe aclarar que si se coloca en el campo password una palabra no cifrada da un error al momento de autentificarse en el panel de logueo de Joomla figura 66.

Llegando a la conclusión que este sistema de gestión de contenidos solo valida que el campo password tenga un código cifrado, causando que al momento de autenticar solo valide esa condición mas no el tipo de cifrado que tubo originalmente.

Administrador de Joomla

Figura 66. Ingreso al administrador de Joomla
Elaborado por David Sánchez y Marco Bravo

De esta forma se obtiene el usuario y la nueva clave, logrando el acceso al panel de administrador de Joomla como se puede visualizar en la figura 66, el cual permite administrar y gestionar los contenidos del sitio web.

Joomla Panel Administrador

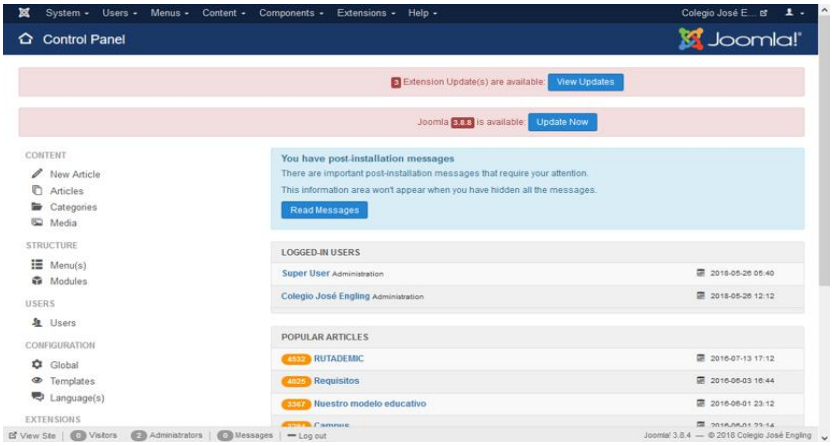


Figura 67. Panel de administración Joomla
Elaborado por David Sánchez y Marco Bravo

Como se observa en la figura 67, el acceso al panel de administrador de Joomla otorga el control total de la página Web, abriendo una brecha de seguridad que puede causar ataques como: denegación de servicio, manipulación de información o phishing.

Denegación de servicio Portal Web

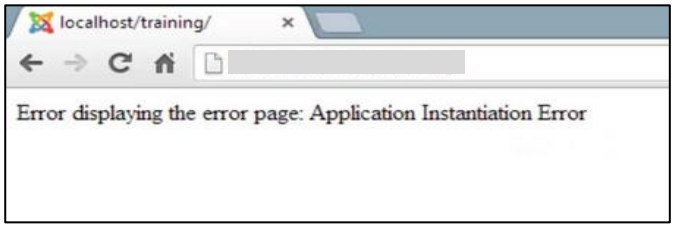


Figura 68. Error de disponibilidad del portal web
Elaborado por David Sánchez y Marco Bravo

Se porcede a cambiar la configuración global, donde se encuentra la conexión a la base de datos, los directorios fuente de memoria cache y logs y la configuración de depuración donde se habilita y deshabilita la página, causando que el portal web no despliegue su contenido y muestre un mensaje de error como se puede visualizar en la figura 68.

CAPÍTULO 6

ANÁLISIS DE RESULTADOS

Este es el último capítulo donde se encuentra la presentación analítica de lo que se obtuvo como resultado de las pruebas de penetración realizadas al portal web, en la cual se describen los hallazgos encontrados durante la investigación, análisis cualitativo y cuantitativo de la información recolectada y por último un plan de recomendación de mitigación.

6.1. Principios básicos de la seguridad de la información

6.1.1. Confidencialidad.

> Acceso a la información únicamente por los usuarios autorizados. La información se reserva exclusivamente para quien posee los permisos y privilegios correspondientes que permiten acceder a dicha información, el acceso no autorizado, clandestino, fuga de información o sustracción de la misma está protegido por las leyes vigentes en el país. (Ortiz Beltrán, 2015)

En este proyecto se analizaron los ataques que pueden comprometer la confidencialidad de la información según se expone en la tabla 2.

Tabla 1. Resultados de análisis a la Confidencialidad.

Confidencialidad	
Vector de ataque	**Resultado**
Recolección de información	1
IP accesible desde la red (Ping)	1
Obtención de Metadatos	0
Escaneo de puertos	1
Acceso a directorios	1
Acceso a la Red	0

Nota: Esta tabla contiene los vectores de ataque a la confidencialidad.

Los 4 vectores de ataque satisfactorios que se puede observar en la tabla 1 fueron:

En el vector de ataque de recolección de información ejecutado se pudo identificar información como: correos, sistemas operativos, herramientas de administración web, servidores DNS, proveedor ISP y aplicaciones institucionales entre otras para identificando la estructura informática de portal.

El vector de ataque de IP accesible desde la red, permite conocer la conectividad con los servidores objetivos, conociendo la IP y dominio asociado.

Escaneo de puertos, este vector de ataque detecta los puertos abiertos que usan los servidores del portal web, identificando motores de base de datos, servicios que son accesibles de forma remota y versiones de servicios.

El vector de ataque de acceso a directorios permitió encontrar archivos por defecto y configuraciones las cuales muestran información confidencial de uso de los administradores.

Los 2 vectores de ataque fallidos muestran que el portal web principal no es vulnerable a la obtención y análisis de metadatos, debido a un bloqueo de Joomla que no permite la descarga de archivos colgados en el portal impidiendo su posterior análisis y extracción de metadatos.

El acceso a la red de igual manera no fue factible debido a que existe un firewall configurado que impide el acceso a direcciones IP’s que no sean parte de la red interna de esta manera muestra la seguridad contra intrusos que actualmente cuenta la institución educativa.

Ataque a la Confidencialidad

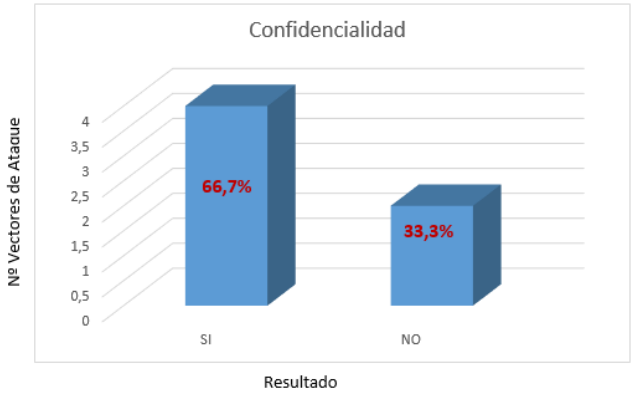


Figura 69. Estadística de ataque a la confidencialidad
Elaborado por David Sánchez y Marco Bravo

Para la elaboración de la figura 69 se tomó como referencia los 6 vectores que evalúan la confidencialidad del portal web, teniendo como resultado que los 4 ataques (Eje Y Nº Vectores de Ataque) ejecutados con éxito (Eje X Resultado) conforman el 66.7% que atenta contra este principio básico, y el 33.3% conforman los 2 ataques restantes que no lograron comprometer la confidencialidad, a los cuales el portal web está protegido.

6.1.2. Integridad.

Integridad en seguridad de la información se refiere a cómo los datos se mantienen intactos libre de modificaciones o alteraciones por terceros, cuando una violación modifica algo en la base de datos, sea por accidente o intencionado se pierde la integridad y falla el proceso. Por este motivo se debe proteger la información para que sólo sea modificada por la misma persona, evitando así que se pierda la integridad. Una manera de proteger los datos es cifrando la información mediante un método de autenticidad como una contraseña o mediante huella digital. (ISOTools, 2017)

Existen amenazas que pueden comprometer de alguna forma la integridad como se muestra en la tabla 2.

Tabla 2. Resultados de análisis a la Integridad.

Integridad	
Vector de ataque	**Resultado**
Edición de campos en la base de datos	1
Manipulación del panel de administrador de Joomla	1
Manipulación de correos electrónico	0
Edición de archivos de computadoras	0

Nota: Esta tabla contiene los vectores de ataque a la integridad

Como describe en la tabla 2, los 2 vectores de ataque que afectaron a la integridad del portal web fueron la edición de campos de campos en la base de datos, por medio del ingreso a phpmyadmin encontrado con la herramienta wfuzz el cual permite actualizar parámetros de la base de datos para escalar privilegios y dar más acceso al segundo vector de ataque de manipulación del panel de administrador del Joomla que de igual forma adquiere el control total, para editar el contenido web alterando la integridad del portal web.

Con los 2 vectores de ataques fallidos: manipulación de correos electrónico el cual permitió el análisis de la cabecera de correo receptado de la institución encontrando direcciones de IP’s y ubicación de origen de servidor de correo, no se logró una modificación o suplantación de identidad que hubiera conseguido un posible ataque de ingeniería social. El segundo fallido es la edición de archivos de computadoras realizado con la herramienta Dump Sec y Hyena no se obtuvo resultados positivos debido a la restricción de permisos tanto de escritura, lectura, y ejecución que actualmente se controla en la institución gracias a un directorio activo, el cual tiene la función de administrar permisos para cada uno de los equipos de la red interna.

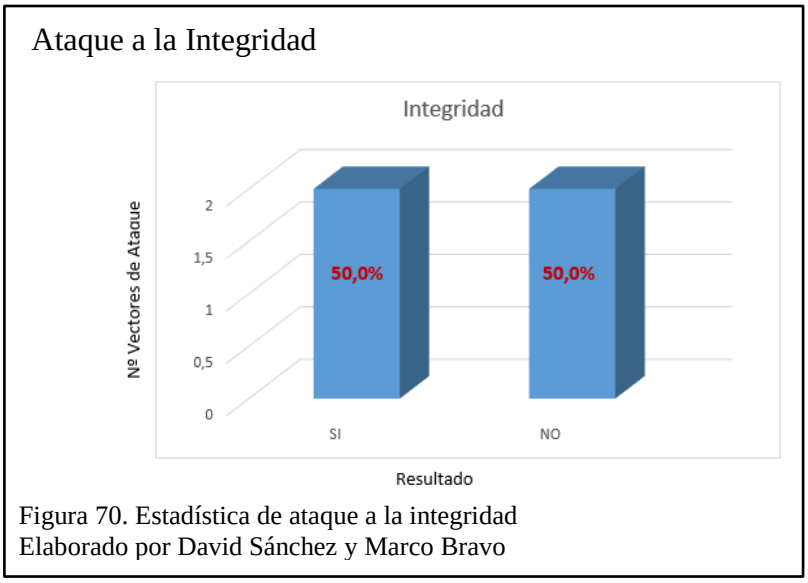


Figura 70. Estadística de ataque a la integridad
Elaborado por David Sánchez y Marco Bravo

Para la elaboración de la figura 69 se tomó como referencia los 4 vectores (Eje Y Nº Vectores de Ataque) que evalúan la integridad del portal web, teniendo como resultado que los 2 ataques ejecutados con éxito (Eje X Resultado) conforman el 50.00% que indica la manipulación de la información del sistema, y el otro 50.00% de los 2 ataques fallidos que lograron contrarrestar este tipo de ataques, debido a sus medidas de seguridad establecidas en la institución.

6.1.3. Disponibilidad.

Es un pilar fundamental de la seguridad de la información, nada se hace teniendo segura e integra la información, si no va a estar disponible cuando el usuario o sistema necesite realizar una consulta. Para cumplir con la última condición se tiene que tener claro cuál será el flujo de datos que se debe manejar, para conocer donde se debe almacenar dicha información que tipo de servicio se debe contratar, etc. (ISOTools, 2017)

Para el análisis de disponibilidad se emplearon dos ataques como se muestra a continuación en la tabla 3.

Tabla 3. Resultados de análisis a la Disponibilidad.

Disponibilidad	
Vector de ataque	**Resultado**
Denegación de Servicio	1
Autenticarse para acceso a la red interna	0

Nota: Esta tabla contiene los vectores de ataque a la disponibilidad.

Los 2 vectores de ataque que se puede observar en la tabla 3 tienen el objetivo de indisponer el acceso del portal web a la comunidad educativa y se obtuvo como resultado: El vector de Denegación de servició ejecutado mediante el ataque de fuerza bruta con wfuzz logró tener acceso al administrador de Joomla para realizar configuraciones globales bloqueando el acceso del público y haciendo caer a la página web.

El vector de ataque fallido es el intento de autentificarse para acceso a la red interna, el cual tenía como objetivo establecer una conexión remota a equipos de la red interna mediante servicios de conexión encontrados en el escaneo de puertos, no se logró debido a sesiones establecidas con un nivel alto de contraseñas y denegación de ataques de fuerza bruta para prueba de contraseñas por el proveedor de internet de la institución educativa

Ataque a la Disponibilidad

Figura 71. Estadística de ataque a la disponibilidad
Elaborado por David Sánchez y Marco Bravo

Para la elaboración de la figura 71 se tomó como referencia 2 vectores (Eje Y Nº Vectores de Ataque) que evalúan la disponibilidad del portal web, teniendo como resultado (Eje X Resultado) que el ataque exitoso que inhabilitó al portal web conforma el 50.00%, y el ataque fallido no logrado complementa el otro 50.00%, certificando que la página web debe establecer planes de contingencia de respaldos de información.

Una vez finalizado con las fases del hacking ético: reconocimiento de información, escaneo, enumeración y explotación se puede concluir que es posible burlar los principios básicos de la seguridad de la información, si no se tiene un control frecuente como una auditoría informática la cual ayuda a mejorar la política de seguridad.

6.2. Análisis cualitativo y cuantitativo

Para determinar el estado actual de la seguridad de la información por medio del hacking ético realizado al portal web del Colegio Católico José Engling se realiza un análisis cualitativo y cuantitativo basándose en los resultados obtenidos mediante los vectores de ataque descritos en las tablas 1,2 y 3.

Se obtiene una probabilidad del 66,7% de ataques que podrían poner en riesgo la confidencialidad de la información que se explica en la figura 69.

Para el análisis de la integridad existe un 50% de porcentaje que indica que es factible realizar un ataque que posibilite alterar modificar o eliminar la información como lo menciona la figura 70.

Para el último análisis de los ataques a la disponibilidad se obtiene que existe un 50% que afirma que es posible comprometer la disponibilidad del portal web tal como se describe en la figura 71.

6.3. Informe de resultados

6.3.1. Resumen ejecutivo.

El proceso del hacking ético realizado al portal web del Colegio Católico José Engling, reúne un conjunto de pruebas de intrusión, donde se encontró vulnerabilidades de seguridad informática en algunos servidores, evaluados con niveles de riesgo alto, medio y bajo.

Cabe indicar que se estableció previamente un acuerdo de confidencialidad con el colegio Católico José Engling, que indica que toda la información encontrada no puede ser divulgada, tal como se estipula en el ANEXO 3.

Las vulnerabilidades con un riesgo alto se encuentran en el servidor web principal ya que fue posible detectar la dirección IP y directorios habilitados por defecto, haciendo uso de herramientas de escaneo, causando un ataque contra la integridad de la información y modificando información de la base de datos.

Mediante un escaneo de la página se verificó varias vulnerabilidades entre las más destacadas se pudo encontrar el sistema operativo que maneja, los puertos que tiene abiertos, versiones de servicios y un caso de SQL Injection de riesgo alto, conjuntamente con la obtención de archivos obsoletos que podrían permitir a un atacante acceder a la base de datos y otros sistemas que pertenecen a la misma estructura de red.

Se obtuvo acceso al administrador web en el cual, si una persona maliciosa tiene el control total, podría utilizar el servidor para enviar spam o hacer phishing a través de él, esto podría ocasionar que las listas RBL incluyan este servidor entre las listas negras.

Así mismo otra de las vulnerabilidades encontradas fue el ataque DoS a la página web mediante la eliminación de campos mandatorios en el gestor de contenidos Joomla, lo que provocó una pérdida de la conectividad a la misma.

Como se puede visualizar en la tabla 4, se presentan las vulnerabilidades encontradas de forma general.

Tabla 4. Servidores auditados y puertos.

IP	Servicio	Sistema Operativo	Puertos abiertos
13X.24X.XXX.XX6	Web principal	Linux / Apache	21,26,53,80,110,143,443, 587,993,995,3306
19X.12.XXX.XX4	Web pruebas	Windows Server 2008 R2	25,80,110,119,143,443,465,563, 587,993,995,3306
19X.12.XXX.XX3	Notas	CentOS / Apache/2.2.15	80,443,1024,4444,8443
19X.12.XXX.XX2	Firewall, Correo	CentOS 6 /Apache/2.2.3	22,25,80,443,587,3306
19X.12.XXX.XX5	Facturación	CentOS / Apache/2.2.15	80,1024,4444,8443
2607:f8b0:4008:811::200d	Correo Educativo	Linux / GSE	80,443

Nota: Esta tabla contiene los servidores auditados con su servicio, sistema operativo y puertos.

El objetivo de esta auditoría es de analizar la información encontrada y poder identificar las amenazas, riesgos vulnerabilidades, para que se tomen las precauciones necesarias ante eventos de posibles ataques externos e internos.

6.3.2. Bitácora de actividades.

La bitácora de actividades se encuentra en el ANEXO 4.

6.3.3. Resumen de hallazgos.

- Se pudo detectar por medio de un servidor Whois registros información de contacto de la institución, emails institucionales, direcciones, teléfonos, referencias de soporte.
- Con nslookup se pudo detectar las direcciones IP de los servidores, así mismo los nombres de estos equipos y proveedor de servicio de internet (ISP).

- Con Nmap se detectaron puertos abiertos y versiones de sistemas operativos, para poder determinar el riesgo de vulnerabilidades altas, medias y bajas en los diferentes servidores encontrados.
- Con ayuda de DumpSec se pudo establecer sesiones, además con el uso de Hyena se detectó los usuarios y sus SID's que se encontraban en la red interna, archivos compartidos y sistemas operativos de los usuarios.
- En la explotación se halló que se puede realizar una denegación de servicio mediante la intrusión en el administrador de la página web por medio de la modificación de la base de datos afectando de manera crítica la disponibilidad del Portal Web.
- Se pudo encontrar una brecha de seguridad en el acceso a la página web donde se maneja información privada del cliente. Esto se logró mediante un ataque de SQL Injection.

6.3.4. Plan de recomendación de mitigación.

Las recomendaciones de mitigación de las amenazas, riegos y vulnerabilidades encontradas en el portal web del Colegio Católico José Engling se encuentran en el ANEXO 5.

CONCLUSIONES

- Se concluye que no es factible mitigar en un 100% los ataques de recolección de información o footprinting hacia el portal web, debido a la demanda obligatoria de información institucional que se publica en internet.
- Se determina que el portal web de la institución es vulnerable en un 66,7% frente a ataques a la confidencialidad de la información, un 50% a ataques contra la integridad y un 50% de probabilidad de riesgo a ataques que comprometan a la disponibilidad. Evidenciando inseguridad informática, y poniendo en peligro la información de la comunidad educativa.
- La metodología ISSAF, seleccionada para la recolección, escaneo y explotación de vulnerabilidades, ayudó a definir los pasos necesarios a seguir para el proceso de indagación de información del portal web del Colegio Católico José Engling, ya que esta metodología aplica la recopilación de información, evaluación y presentación de informes. Siguiendo estas fases se encontraron todas las posibles vulnerabilidades existentes en el portal web.
- La aplicación web de monitoreo de incidencias está desarrollada para un manejo ágil de las mismas y da como resultado un plan de recomendaciones de mitigación, que resulta una información de relevancia para el administrador de la red y así mantener la protección del portal web, la información que se presente en el aplicativo web de monitoreo, ayudará a tener un mayor enfoque de la seguridad actual de la red y favorece en una futura restructuración de la misma.

RECOMENDACIONES

- Es recomendable publicar información solo que sea estrictamente necesaria acerca de la institución, para disminuir el impacto de ataques de recolección de información o footprinting, se recomienda pagar por privacidad en los servicios de directorios Who-Is, servidores de correo y DNS.
- Se recomienda ejecutar auditorias de hacking ético de forma continua que analicen las amenazas riegos y vulnerabilidades del portal web, equipos, y servidores para la toma de decisiones y correctivos en la política de seguridad.
- Es importante actualizar el software y hardware de los productos de seguridad y aplicativos que brinden servicios a la institución, debido a que en cada versión contiene nuevos parches, se aplican correcciones a bugs e implementan nuevas funcionalidades que ayudan a detectar y evitar nuevos ataques informáticos.
- Cuando termina el tiempo de uso de un activo informático o se da de baja un aplicativo de la institución, se recomienda quitar su contenido y no dejar información sensible, muchas de las veces se dejan olvidados estos equipos o aplicativos y cuando son encontrados por atacantes, usan esta información para realizar diversos ataques. De igual manera existen archivos por defecto, se recomienda retirarlos o eliminados para evitar la visualización y análisis malicioso de los mismos.
- Se recomienda capacitar y dar charlas preventivas a los usuarios de la institución, sobre la seguridad de la información y los ataques de seguridad informática a los que se exponen, para lograr tener conocimiento de las soluciones frente a los mismos.

GLOSARIO DE TÉRMINOS

LDAP: Lightweight Directory Access Protocol o Protocolo compacto de acceso a directorios.

SOPHOS: Es una compañía británica de software y hardware de seguridad.

FORTINET: Es una empresa multinacional de Estados Unidos.

SID: Un identificador de seguridad.

RBL: Realtime Blackhole List (Listado agujero negro en tiempo real.

IP: Internet Protocol (Protocolo de Internet)

DNS: Domain Name System (servidor de nombres de dominio)

SCTP: Stream Control Transmission Protocol.

XSS: Cross-site scripting es un tipo de inseguridad informática.

TMRC: The Tech Model Railroad Club.

MIT: Massachusetts Institute of Technology.

TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet.

UDP: User Datagram Protocol.

ICMP: Protocolo de mensajes de control de Internet.

ARP: Protocolo de resolución de direcciones.

RARP: Protocolo de resolución de direcciones inversas.

IGMP: Internet Group Management Protocol.

DACLs: Discretionary Access Control List.

SACLs: System Access Control List.

ISSAF: Information Systems Security Assessment Framework.

IT: Information Technology.

LISTA DE REFERENCIAS

Acunetix. (2013). *Acunetix*. Retrieved from https://www.acunetix.com/websitesecurity/sql-injection/

Adastra. (2011, Mayo 12). *Seguridad en Sistemas y Técnicas de Hacking. TheHackerWay (THW)*. Retrieved from https://thehackerway.com/2011/05/12/conceptos-basicos-de-nikto-tecnicas-de-escaneo-de-servidores-y-aplicaciones-web/

AMBAR Telecomunicaciones. (2016, Abril 27). *Fases comunes del hacking ético*. Retrieved from https://ambar.es/soluciones/ciberseguridad/hacking-etico/#top

Araque, M. (2017, Febrero 8). *WAM*. Retrieved from https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html

Ascencio Mendoza, M., & Moreno Patiño, P. J. (2011). Desarrollo de una Propuesta Metodológica para Determinar la Seguridad en una Aplicación Web. Pereira, Colombia: Universidad Tecnológica de Pereira.

Astudillo, K. (2013). *HACKING ÉTICO 101*. Guayaquil: Registro IEPI.

AVAST. (2015). *AVAST*. Retrieved from https://www.avast.com/es-es/c-sql-injection

Brinkmann , M. (2008, Junio 5). *Firefox Passive Recon*. Retrieved from https://www.ghacks.net/2008/06/05/firefox-passive-recon/

Caballero Quezada, A. (2014, Enero 17). *ReyDes*. Retrieved from http://www.reydes.com/d/?q=DNSenum

Caballero Quezada, A. (2014, Agosto 15). *Volcar Información De Usuarios Utilizando DumpSec*. Retrieved from http://www.reydes.com/d/?q=Volcar_Informacion_de_Usuarios_utilizando_DumpSec

Caballero Quezada, A. E. (2016, Enero 21). *Instalación de Nessus en Kali Linux*. Retrieved from http://www.reydes.com/d/?q=Instalacion_de_Nessus_en_Kali_Linux

Catoira , F. (2012, Marzo 28). *welivesecurity*. Retrieved from https://www.welivesecurity.com/la-es/2012/03/28/consejos-ataque-denegacion-servicio/

Catoira, F. (2013, Abril 26). *welivesecurity*. Retrieved from https://www.welivesecurity.com/la-es/2013/04/26/funcionamiento-de-una-inyeccion-sql/

ComoHacerPara. (2008, Mayo). *Diferencias entre un Hacker y un Cracker*. Retrieved from http://comohacerpara.com/diferencias-entre-un-hacker-y-un-cracker_7792i.html

CompuTrabajo. (2018). *CompuTrabajo*. Retrieved from https://www.computrabajo.com.ec/quienessomos/

Crespo, A. (2016, Diciembre 24). *redeszone*. Retrieved from https://www.redeszone.net/2016/12/24/puedo-rastrear-correos-electronicos-enviados-desde-cuenta-personal/

CRIDO Santiago. (2016, Octubre 29). *CIBERSEGURIDAD: ¿Cómo implementar una Política de Seguridad Informática? Una propuesta.* Retrieved from Linkedin:

https://es.linkedin.com/pulse/ciberseguridad-c%C3%B3mo-implementar-una-pol%C3%ADtica-de-crido-santiago

CyberSeguridad. (2015, Agosto 23). *CYBERSEGURIDAD.NET*. Retrieved from https://cyberseguridad.net/index.php/455-las-fases-de-un-test-de-penetracion-pentest-pentesting-i

debianHackers. (2012, Octubre 8). *debianHackers*. Retrieved from https://debianhackers.net/nmap-escaner-de-puertos/

Díaz, M. (2017, Abril 5). *Auditorías Web con OWASP ZAP – Introducción y ejemplos de uso*. Retrieved from https://www.diazsecurity.com/2017/04/tutorial-y-ejemplos-de-uso-con-owasp-zap/

Gómez, I. (2017, Mayo 12). *CRIPTONOTICIAS*. Retrieved from https://www.criptonoticias.com/sucesos/ransomware-masivo-exige-300-computadora-infectada-telefonica-empresas-espanolas/

Gómez, M., Venegas, C., & Yánez, V. (2010). *Herramientas para hacking ético.* Escuela Superior de Cómputo.

Google. (2018). *Ayuda de Google Domains*. Retrieved from https://support.google.com/work/mail/answer/6233343?hl=es#

ISOTools. (2017, Julio 6). *Blog especializado en Sistemas de Gestión de Seguridad de la Información*. Retrieved from https://www.pmg-ssi.com/2017/07/cia-confidencialidad-integridad-disponibilidad-seguridad-de-la-informacion/

Linuxito. (2017, Abril 21). *Linuxito*. Retrieved from https://www.linuxito.com/gnu-linux/nivel-basico/866-como-conectarse-con-netcat-a-servidores-http

Lozano, P. (2015, Agosto 10). *Tribuna Hacker*. Retrieved from http://www.tribunahacker.com.ar/2014/05/que-son-los-ataques-de-fuerza-bruta-2/

Martinez, C., & Oñate, O. (2017). *MEJORAS EN LA SEGURIDAD DE LA RED INALÁMBRICA DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO APLICANDO HACKING ÉTICO.* Riobamba.

Mendez, X. (2018). *Edge-Security*. Retrieved from https://github.com/xmendez/wfuzz

Miríadax. (2017, Septiembre 24). Gestión de proyectos con metodologías Ágiles y enfoques Lean. *Roles en Scrum*. Madrid, España: Telefónica Educación Digital.

Morales, J. (2017, Abril 26). *JMCristobalHomepage*. Retrieved from http://jmcristobal.com.mx/2017/04/26/modelo-tcpip/

Mozilla Firefox. (2010, Septiembre 27). *Firefox Add-ons*. Retrieved from https://addons.mozilla.org/en-US/firefox/addon/passiverecon/

Netcloud Engineering. (2017, Junio 26). *Netcloud Engineering*. Retrieved from https://netcloudengineering.com/ciberseguridad-amenaza-vulnerabilidad/

Noriega Martínez, R. (2017). *El Proceso de Desarrollo de Software.* IT Campus Academy.

Olmedo, J. (2015, Marzo 9). *Hack Puntes*. Retrieved from https://hackpuntes.com/obtener-informacion-con-the-harvester/

Ortiz Beltrán, B. F. (2015). *HACKING ÉTICO PARA DETECTAR FALLAS EN LA SEGURIDAD INFORMÁTICA DE LA INTRANET DEL GOBIERNO PROVINCIAL DE IMBABURA E IMPLEMENTAR UN SISTEMA DE GESTIÓN*

DE SEGURIDAD DE LA INFORMACIÓN (SGSI), BASADO EN LA NORMA ISO/IEC 27001:2005. Ibarra: Universdiad Técnica del Norte.

Pérez, I. (2014, Febrero 19). *welivesecurity*. Retrieved from https://www.welivesecurity.com/la-es/2014/02/19/maltego-herramienta-muestra-tan-expuesto-estas-internet/

Pérez, I. (2015, Abril 8). *welivesecurity*. Retrieved from https://www.welivesecurity.com/la-es/2015/04/08/the-harvester-riesgo-nformacion-publica/

Pérez, J., & Merino, M. (2013). *Definición de Windows*. Retrieved from https://definicion.de/windows/

Ramírez González, T. (2016, Mayo 21). *¿Qué es el comando Ping y cómo funciona?* Retrieved from Computer Hoy: https://computerhoy.com/noticias/internet/que-es-comando-ping-como-funciona-42607

Rexford, J., Wang, J., Morley Mao, Z., & Katz, R. (2003, Agosto 25). *ACM Digital Library*. Retrieved from https://dl.acm.org/citation.cfm?id=863996

Reyes Plata, A. (2011). *Ethical Hacking.* Mexico: UNAM CERT.

Sánchez Patón, V., & Prieto, G. (2012, Diciembre). *Herramientas de consulta a servidores DNS: nslookup, host y dig.* Retrieved from https://vicentesanchezsri.files.wordpress.com/2012/12/herramientas-de-consulta-a-servidores-dns.pdf

SDTeam. (2016, Noviembre 21). *Security database*. Retrieved from https://www.security-database.com/toolswatch/Hyena-v8-32-bit-64-bit-released.html

seguridadroberto. (2016, Octubre 30). *Seguridad Informática de Roberto*. Retrieved from https://seguridadroberto.wordpress.com/2016/10/30/kali-linux-que-es-para-que-se-utiliza-las-diez-aplicaciones-mas-importantes-que-integra/

Stuxnet. (2013, Abril 3). *Hack x Crack*. Retrieved from https://hackxcrack.net/foro/defacing/cross-site-tracing-(xst)/

Suri, S. (2017, Diciembre 5). *The Linux Juggernaut*. Retrieved from https://www.linuxnix.com/exploring-linux-netcatnc-command-with-examples-part1/

Tapia, M. (2015, Diciembre 4). *Seguridad en Cómputo*. Retrieved from http://blogs.acatlan.unam.mx/lasc/2015/12/04/fases-del-hacking/

Tecnología W. (2011, Enero 30). *Definición de Windows*. Retrieved from http://conceptodefinicion.de/windows/

Telmex. (2012, Febrero 14). *Aula Digital TELMEX*. Retrieved from http://www.telmexeducacion.com/proyectos/DocsDobleclic/14-Doble%20clic-Buscadores%20o%20motores%20de%20busqueda.pdf

The PHP Group. (2018). *ZendCon & OpenEnterprise 2018*. Retrieved from http://php.net/manual/es/function.phpinfo.php

TheHackerWay. (2013, Enero 14). *Seguridad en Sistemas y Técnicas de Hacking. TheHackerWay (THW)*. Retrieved from https://thehackerway.com/2013/01/14/web-hacking-clasificacion-de-ataques-web-parte-xvii/

Tori, C. (2008). *Hacking Etico.* Buenos Aires: Hacking Exposed.

Velasco, R. (2015, Abril 25). *OWASP ZAP, herramienta para auditar la seguridad de una página web*. Retrieved from https://www.redeszone.net/2015/04/25/seguridad-web-owasp-zap/

Venom, S. (2014, Diciembre 22). *Using Paros for Web Application Auditing and Debugging*. Retrieved from https://null-byte.wonderhowto.com/forum/hiob-using-paros-for-web-application-auditing-and-debugging-0158950/

Visualware Inc. (2014, Marzo 23). *Visualware*. Retrieved from http://www.emailtrackerpro.com/

ANEXOS

ANEXO 1 – Aplicativo web de monitoreo de incidencias.

ANEXO 2 – Leyes para el Hacking en Ecuador

ANEXO 3 – Acuerdo de Confidencialidad y Carta de aprobación del Aplicativo

ANEXO 4 – Bitácora de actividades

ANEXO 5 – Plan de recomendación de mitigación

Los anexos pueden ser descargados del siguiente enlace:

https://1drv.ms/f/s!AsXhgOv-MCeFikvtmjdVDfHKSo3N