ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

UNIDAD DE TITULACIÓN

ESTRATEGIA PARA LA EVALUACIÓN DE VULNERABILIDADES DEL SISTEMA DE NOTAS DE INSTITUCIONES EDUCATIVAS UTILIZANDO TÉCNICAS DE HACKING ÉTICO. CASO DE ESTUDIO: INSTITUTO TECNOLÓGICO QUITO

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN SOFTWARE, MENCIÓN EN SEGURIDAD

DAVID ESTEBAN GALARZA GARCÍA
david.galarza@epn.edu.ec

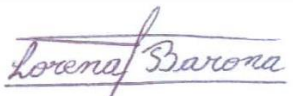
Directora: Lorena Isabel Barona López, PhD.
lorena.barona@epn.edu.ec

Codirectora: Jenny Gabriela Torres Olmedo, PhD.
jenny.torres@epn.edu.ec

2020

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación ESTRATEGÍA PARA LA EVALUACIÓN DE VULNERABILIDADES DEL SISTEMA DE NOTAS DE INSTITUCIONES EDUCATIVAS UTILIZANDO TÉCNICAS DE HACKING ÉTICO. CASO DE ESTUDIO: INSTITUTO TECNOLÓGICO QUITO desarrollado por DAVID ESTEBAN GALARZA GARCÍA, estudiante de la MAESTRÍA EN SOFTWARE, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.



Lorena Isabel Barona López

DIRECTOR

DECLARACIÓN DE AUTORÍA

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

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

David Esteban Galarza García

DEDICATORIA

Dedicado a todo aquel que mantiene su fe intacta, a todo aquel que logra sus objetivos día a día, desde el alba hasta el ocaso. Mi mayor admiración y respeto a todos ellos.

David Galarza G.

AGRADECIMIENTO

¡Gracias a la vida por dejarme ver la luz un día a la vez!

En el transcurso rutilante de la vida, he aprendido que una dosis de coraje hace posible lo imposible. Es por ello por lo que deseo entregar mi más sentido agradecimiento a las entidades y personas que aportaron de manera gradual a este proyecto.

A Gustavo Albuja, por su convicción y disposición desinteresada al momento de entregar información relevante y por apoyar el proyecto desde dentro, así como a Iván Cadena, por permitirme ejecutar el análisis en cuestión en el instituto de educación superior donde se desempeña como rector y al Instituto Tecnológico Quito por abrir sus puertas ofreciendo su sistema de notas como caso de estudio.

A Lorena Barona, por su paciencia, franqueza, confianza y apoyo durante todo este ambicioso proyecto.

A Tulia García, por ayudarme a vivir una brumosa niñez de la cual tengo recuerdos enmarcados en mi memoria y también por decidir dejarme ver la luz.

A José Galarza por enseñarme a interpretar el mundo desde su perspectiva, por su poca paciencia, por su dedicación, rectitud y convicción objetiva.

A Darío José, por brindarme un plato de comida en su mesa en mis días más oscuros y por quitarme la venda.

A Mateo Galarza, por obligarme a desaprender.

David Galarza G.

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

El presente trabajo de titulación de Maestría desarrolla una estrategia para la evaluación de vulnerabilidades del sistema de notas de instituciones educativas. Para llevar a cabo este procedimiento de análisis, se consideran las siguientes etapas de hacking ético: reconocer, escanear, explotar (obtención y mantenimiento del acceso) y cubrir o borrar huellas, lo cual permite identificar las brechas de seguridad que afectan directamente al sistema de notas. Las herramientas, guías y técnicas en las cuales se basará la evaluación son: Kali Linux y OWASP ya que establecen un ecosistema adecuado presentando resultados que sirven de insumo para ejecutar cada etapa del hacking ético.

Las instituciones educativas han visto la necesidad de salvaguardar la información generada y procesada en el sistema académico, por tal motivo, se desarrollará una guía específica en base a la información recopilada. Esta guía servirá de estrategia para evaluar las vulnerabilidades que afectan directamente al sistema de notas utilizando el criterio analítico alineado al hacking ético y las herramientas antes mencionadas.

Palabras clave: Hacking ético, vulnerabilidad, OWASP, SQL injection, clickjacking y cross site scripting.

ABSTRACT

The present Master's degree work develops a guide with the aim of executing a strategy for evaluating vulnerabilities in the grade system of educational institutions. To carry out this analysis procedure, check the following stages of ethical hacking: recognition, scanning, obtaining access, maintaining access and covering or deleting fingerprints, which allows identifying security breaches that directly affect the note system. The tools, guides and techniques on which they will be based on the evaluation are: Kali Linux and OWASP since they have an adequate ecosystem presenting results that serve as input to execute each stage of ethical hacking.

Educational institutions have seen the need to safeguard the information generated and processed in the academic system, for this reason, a specific guide will be developed based on the information collected from ethical hacking. This guide will serve as a strategy to assess vulnerabilities that directly affect the note system using analytical criteria aligned to ethical hacking and the aforementioned tools.

Keywords: Ethical hacking, vulnerability, OWASP, SQL injection, clickjacking and cross site scripting.

1. INTRODUCCIÓN

En esta sección se describe el planteamiento del problema, la justificación teórica, práctica y metodológica, así como los objetivos generales y específicos. Finalmente, la metodología que será aplicada en el desarrollo del presente trabajo de titulación.

1.1. Planteamiento del Problema

Debido al vertiginoso avance de la tecnología y el Internet, la exposición y protección de información se ve afectada directamente por criterios de seguridad. Las amenazas que atentan la seguridad de la información aumentan de manera que no se puede afirmar que existan sistemas informáticos que sean invulnerables a ataques cibernéticos [1],[2].

En la actualidad, según OWASP, los ataques más comunes contra aplicaciones web son [3]: URL del tipo semántico, Cross Site Scripting Cross Site Request Forgery, peticiones HTTP falsificadas, SQL Injection, entre otros [4]. Sin embargo, las empresas ponen mayor énfasis en que sus aplicaciones informáticas sean eficientes, rápidas, usables, accesibles e inclusivas pero muy pocas se preocupan de incorporar la seguridad necesaria.

De esta manera se ve expuesta la integridad, confidencialidad y disponibilidad que derivan en pérdidas económicas, captura y robo de información privada, retraso en la ejecución y gestión de procesos, entre otras consecuencias [5]. De acuerdo con ITRC Data Breach [6], el reporte de ataques en el 2018 está distribuido de la siguiente manera: A empresas privadas un 46%, a instituciones de salud un 29%, a instituciones financieras un 11%, a entidades gubernamentales un 8% y a instituciones educativas un 6%. Lo cual indica que las instituciones educativas pertenecen a un grupo vulnerable a ataques tecnológicos.

Además, la comunidad educativa, accede al portal web con la finalidad de visualizar y manipular información académica, donde justamente los ataques han logrado realizar alteraciones que han ocasionado que ciertos servicios, dejen de operar durante un espacio de tiempo [7].

En los sistemas de notas de instituciones educativas se han evidenciado las siguientes vulnerabilidades: SQL Injection, Exploiting SQL, Broken Authentication and Session Management, Exploiting Broken Authentication, Cross Site Scripting, Exploiting XSS y Security Misconfiguration [8],[9]. Por lo tanto, es importante establecer estrategias para la evaluación de vulnerabilidades del sistema de notas de instituciones educativas.

1.2. Justificación

El presente trabajo de titulación analiza y evalúa las vulnerabilidades en el sistema de notas de instituciones educativas que despliegan el software en navegadores web, tomando como caso de estudio al Instituto Tecnológico Quito. Con el conocimiento y la información adquirida, se generará una estrategia que sirva para el análisis y evaluación de vulnerabilidades en este tipo de organizaciones.

1.2.1. Justificación teórica

Las instituciones educativas han visto la necesidad de salvaguardar la información procesada y generada en el sistema de notas, ya que es un activo valioso para el proceso académico integral. Por tal motivo, para el análisis de vulnerabilidades se ha considerado seguir los lineamientos de hacking ético. El cual consiste en el desarrollo de pruebas a sistemas de información automatizados con la finalidad de encontrar y exponer vulnerabilidades tecnológicas.

Para garantizar que se ejecute este procedimiento, el análisis debe atravesar las siguientes etapas:

- Reconocimiento
- Escaneo
- Obtención de acceso
- Mantener el acceso
- Cubrir o Borrar huellas

Entre los principales ataques informáticos se tiene: ataques de monitorización donde se observa a la víctima y a su sistema de información, ataques de validación en el cual se suplanta al usuario mediante la obtención de sus credenciales de acceso al sistema, ataque de denegación de servicios en el que se intenta colapsar los recursos de la víctima con la finalidad de interrumpir los servicios ofrecidos por el mismo. Finalmente, ataque de modificación que tiene como objetivo manipular de manera no autorizada la información y datos de los sistemas de información de la víctima. Las herramientas de auditoria informática más utilizadas para hacking ético son Kali Linux y OWASP debido a que generan un ambiente adecuado para la analítica y el pentesting esperado.

1.2.2. Justificación metodológica

De acuerdo con la naturaleza y a las características del proyecto, la metodología que se va a utilizar es Estudio de Caso. Esta metodología consiste en conocer y comprender la particularidad de una situación mediante la búsqueda e indagación con la finalidad de

distinguir la funcionalidad de las partes y su relación con el todo. Por tanto, esta metodología permite realizar un análisis exploratorio orientado a la ejecución adecuada de hacking ético. Finalmente, se entregará una estrategia para evidenciar vulnerabilidades en los sistemas de notas de instituciones educativas, tomando como caso de estudio al Instituto Tecnológico Quito.

1.2.3.Justificación práctica

El presente proyecto de investigación corresponde a una necesidad particular de las instituciones educativas de salvaguardar su información más crítica, en este caso, información acerca de las calificaciones. Como se menciona al inicio del documento, el 6% de ataques informáticos corresponden a plataformas de instituciones educativas, lo cual indica que es un grupo vulnerable. Como se mencionó anteriormente, el 65.50% de la comunidad educativa, accede al portal web con la finalidad de visualizar y manipular información académica como tareas virtuales, sesiones en línea, y procesos que derivan en calificaciones, por lo que es importante desarrollar una estrategia para el descubrimiento y evaluación de vulnerabilidades del sistema de notas en instituciones educativas.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar una estrategia para la evaluación de vulnerabilidades del sistema de notas en instituciones educativas.

1.3.2.Objetivos específicos

- Investigar acerca de técnicas de hacking ético para la evaluación de vulnerabilidades en sistemas informáticos académico que despliegan su solución en navegadores web.
- Estudiar las vulnerabilidades que afectan directamente a los sistemas de notas de instituciones educativas con la finalidad de identificar las falencias del sistema.
- Evidenciar las vulnerabilidades detectadas y explotadas por cada técnica de hacking ético realizado al sistema informático de notas.
- Desarrollar una guía específica que permita realizar la evaluación de vulnerabilidades en los sistemas académicos informáticos.

1.4. Metodología

La metodología que se utilizará en la presente investigación es Estudio de Caso guiada por OWASP debido a que, después del análisis de vulnerabilidades en el sistema de notas, la metodología permitirá a la organización controlar las vulnerabilidades, así como también, es una metodología que está constantemente orientada a evaluar las vulnerabilidades de aplicaciones web y móviles exclusivamente.

Otra metodología que se apalanca en esta investigación es OSSTMM, debido al aporte científico que puede implementar en el presente proyecto de titulación, sin embargo, se ha tomado la decisión de ejecutar la metodología de OWASP, en su mayoría, debido a las ventajas con respecto a curva de aprendizaje, así como también, debido a la importancia académica de esta metodología como se presentada, por ejemplo, en Metasploit de Rapid7.

Se enlista a continuación, las etapas que serán ejecutadas en el presente trabajo de titulación alineados a la metodología OWASP:

a. Elaboración del ambiente de pruebas para el análisis y descubrimiento de vulnerabilidades, con la finalidad de responder las siguientes preguntas: ¿Qué puede visualizar un intruso en el sistema atacado?, ¿Qué puede hacer el intruso con la información adquirida en el ataque?, ¿Existe una entidad que registre y controle el ingreso o ataque del intruso?
b. Aplicación y ejecución de técnicas de hacking ético.
c. Análisis e interpretación de la información y los resultados obtenidos, con la finalidad de desarrollar una estrategia integral para la evaluación de vulnerabilidades en los sistemas de notas de instituciones educativas.
d. Elaboración de conclusiones y recomendaciones resultantes de la investigación.

OWASP cuenta con una serie de controles para identificar y evaluar las vulnerabilidades, a continuación, se presenta una correlación entre los controles que serán utilizados por cada etapa de hacking ético:

- En la etapa de reconocimiento se utilizarán los siguientes controles:
    - Descubrimiento de motores de búsqueda y el reconocimiento de fugas de información.
    - Análisis al servidor web para encontrar huellas digitales.
    - Búsqueda de metarchivos en el servidor web para detectar fugas de información.
- En la etapa de escaneo se utilizarán los siguientes controles:

    - Enumerar aplicaciones en el servidor web.
    - Revisar los comentarios y metadatos de la página web para detectar fugas de información.
    - Marco de aplicación web de huellas digitales.
    - Aplicación web de huellas digitales.
    - Pruebas de XSS reflejados.
    - Pruebas de XSS almacenados.
    - Prueba para inyección SQL.
    - Prueba de fugas HTML.
    - Prueba de secuencias de comandos XSS basadas en DOM.
    - Prueba de ejecución de JavaScript.
    - Prueba de inyección HTML.
    - Prueba de redireccionamiento de URL del lado del cliente.
    - Prueba de inyección CSS.
    - Prueba de clickjacking.
- En la etapa de explotación de vulnerabilidades se utilizarán los siguientes controles:
    - Clickjacking
    - Inyección SQL
- La etapa para cubrir huellas no será ejecutada de acuerdo con el alcance del proyecto.

2. MARCO TEÓRICO

En esta sección se describe la terminología utilizada en el trasncurso de este trabajo de titulación con el objetivo de fortalecer los conceptos utilizados en el desarrollo del documento. Es objetivo también, reforzar el entendimiento al momento de dar lectura al presente trabajo de titulación.

2.1. Seguridad informática

Para estudiar por completo la terminología de seguridad informática, se debe interpretar los conceptos bases con respecto a seguridad, la cual se define como la ausencia de riesgo que deriva en un estado de bienestar, puntualizado en cuatro acciones inmersas en este criterio [10]:

- Prevenir del riesgo
- Transferir el riesgo
- Mitigar el riesgo
- Aceptar el riesgo

Por tanto, la seguridad informática es la encargada de garantizar la ausencia del riesgo en todo el proceso o medio informático ya sea en la transmisión, almacenamiento o gestión de la información [10].

2.1.1. Vulnerabilidades

Se entiende por vulnerabilidad, a la debilidad de un sistema informático, del cual un atacante puede aprovecharse, con la finalidad de comprometer la seguridad de algún componente o de todo el sistema, provocando como resultado perdidas o daños [11].
El origen de las vulnerabilidades es ocasionado por diversas causas, entre las cuales se identifican fallos en el diseño arquitectónico del sistema informático, carencia de seguimiento a la automatización de los procesos, entre otras.

2.1.2. Amenazas

Según Somerville, amenaza es: "*Circunstancias que tienen potencial para causar pérdida o daño. Se puede pensar en ellas como una vulnerabilidad de sistema que está sujeta a un ataque."* [11]. Dichas amenazas sacan el mayor provecho a las vulnerabilidades propias de los sistemas informáticos.

2.1.3. Riesgos

El riesgo es la probabilidad de que ocurra un incidente o amenaza directamente relacionado con los procesos automatizados o no de una organización. Las principales características del riesgo son las siguientes [12]:

- Cuando el riesgo depende de una entidad o actividad se conoce como riesgo situacional.
- El riesgo interdependiente es aquel que tiene relación directa con recursos: tiempo, dinero o personal.
- El riesgo depende también directamente de la tolerancia a la organización, de modo que existen tres criterios de riesgo a ser evaluados:
    - Riesgos conocidos.
    - Riesgos desconocidos.
    - Riesgos transferibles.

2.2. Introducción al hacking

Es una disciplina que forma parte de la auditoria informática, en la cual se mezclan criterios técnicos y filosóficos, considerando habilidades investigativas y creativas [13]. Los medios convencionales se han encargado de tergiversar el termino hacker, equiparando su significado con acciones criminales. Es por tal motivo que se incorpora el termino ethical hacking, con la intensión de aclarar y asegurar que las acciones alineadas al hacking, se las realizan considerando criterios éticos [14].

2.2.1. Hacker

En la década de los sesenta en Estados Unidos aparece por vez primera la palabra “hacker”, utilizada por los miembros del Instituto de Tecnología de Massachusetts (MIT) para referirse al personal que utiliza sus habilidades técnicas para atacar o romper la seguridad del sistema telefónico [15].

El termino fue evolucionando en la década de los setenta haciendo referencia a los entusiastas de la computación y el internet que gozan de creatividad y distribución de conocimiento. En paralelo, se les denominaba phreakers a los primeros hackers que utilizaban su conocimiento técnico para fines maliciosos [15]. En la actualidad el termino se ha divido en una clasificación para diferenciar la intención con la que el individuo ejecuta sus conocimientos. Se dividen en hackers de sombrero blanco, gris y negro [15].

Los hackers de sombrero blanco o hackers éticos son aquellos individuos que buscan vulnerabilidades con la finalidad de dar solución y protección a los sistemas informáticos.

Su intención principal es servir de apoyo a las organizaciones ayudando en su crecimiento tecnológico e institucional [15]. Los hackers de sombrero negro son aquellos que buscan vulnerabilidades en sistemas informáticos con la finalidad de tener beneficio individual atacando directamente las brechas de seguridad encontradas en la organización exponiendo y sacando ventaja de estas debilidades [15].

Finalmente, los hackers de sombrero gris son aquellos individuos que intentan llevar a cabo los objetivos de ambas clasificaciones anteriores, es decir, buscan vulnerabilidades ayudando a la protección informática de la organización, pero sacando ventaja a su favor de estas debilidades expuestas [15].

2.2.2. Cracker

El termino cracker hace referencia a los intrusos informáticos que rompen la seguridad de un sistema, llamados también phreakers por la comunidad en respuesta a la inapropiada utilización del término “hacker” [15]. El objetivo principal de los crackers es vulnerar, destruir o alterar los sistemas informáticos para su propio beneficio. Algunos individuos se inician haciendo pequeñas búsquedas alrededor se sistemas informáticos, para más tarde realizar verdaderos actos vandálicos, borrando, alterando o vendiendo información confidencial y sensible [16].

2.2.3. Hacking ético

Se aplica la terminología de hacking ético al comportamiento y las habilidades técnicas de individuos cuya finalidad es prevenir y defender los sistemas informáticos [13]. La función principal es determinar el alcance con respecto a las actividades que puede realizar un intruso sobre los sistemas y velar por su protección [13] [17].

2.2.4. Etapas de hacking ético

El análisis de vulnerabilidades desde la perspectiva del hacking ético se lo realiza con un enfoque constructivo y con la finalidad de proteger los sistemas informáticos [18]. Es importante especificar a continuación las etapas que se llevan a cabo en el proceso integro de hacking ético [18][19]:

a. Reconocimiento: Etapa que implica el estudio previo y recopilar información acerca de una potencial victima u objetivo sin el previo conocimiento de la entidad afectada.
b. Escaneo: En esta etapa, el hacker organiza toda la información obtenida en la etapa de reconocimiento con la finalidad de realizar el análisis. Luego estudia la información de nombres de usuario, sistema operativo, software base instalado, direcciones IP, direcciones MAC, cuentas de usuario, etc. Con la finalidad de

encontrar vulnerabilidades. A continuación, se listan algunos criterios que se emplean durante esta etapa de exploración:

- Escáneres de puertos.
- Escáneres de vulnerabilidades.
- Escáneres de ICMP (Protocolo de Mensajes de Control de Internet).
- Mapeadores de red.
- Barrido de ping.
- Barrido de SNMP (Protocolo de Manejo Simple de Red).

c. Obtener acceso: En esta etapa, las vulnerabilidades encontradas en el escaneo proceden a ser explotadas con la finalidad de obtener acceso al sistema víctima. En el léxico del hacking, a esta etapa también se la conoce como “Owning the System” o simplemente “Owning”. El ataque o explotación de vulnerabilidades se realizan a través de la red local, cableada o de manera inalámbrica, por acceso local a una terminal o PC, por internet, etc.

d. Mantener acceso: Una vez que se ha logrado acceder al sistema víctima, el objetivo de esta etapa es mantener el acceso con la finalidad de ejecutar futuros ataques o explotaciones.

e. Cubrir huellas: Finalmente, después de atravesar por las etapas anteriores, el hacker debe cubrir su rastro eliminando archivos de registro (logs) o las alertas de los sistemas de detección de intrusos (IDS) para evitar ser detectado por el administrador del sistema víctima.

2.3. Ataques más comunes a sitios web

La principal herramienta para atacar sitios web, utilizando amenazas basadas en navegador, es el paquete completo de exploits y meta exploits, que ayudan a los atacantes a infectar sistemas que no cuentan con un certificado de seguridad instalado o protocolos de seguridad debidamente implementados [20].

Segun Kaspersky “*En 2012, la cantidad de ataques basados en navegador ascendió a 1 595 587 670*” [20]. A continuación, se enlista las 10 amenazas basadas en navegador más activas en Ecuador, las cuales representan el 73.23% de ataques totales en sitios web del pais:

Ataques que infectaron a sitios web del ecuador con corte marzo 2020		
Orden	**Ataque**	**Porcentaje**
1	HackTool.Win32.KMSAuto.gen	28.08%
2	HackTool.MSIL.HackKMS.a	8.28%
3	DangerousObject.Multi.Generic	6.41%
4	Trojan.WinLNK.Agent.gen	6.1%
5	HackTool.MSIL.KMSAuto.dh	5.66%
6	HackTool.MSIL.HackKMS.d	5.27%
7	HackTool.MSIL.KMSAuto.di	4.94%
8	HackTool.MSIL.HackKMS.h	3.62%
9	HackTool.MSIL.HackKMS.e	2.48%
10	HackTool.Win32.KMSAuto.ew	2.39%

Tabla 1: Ataques más comunes en sitios web del Ecuador [35]

Los sistemas web cambian de manera constante debido a la evolución de la tecnología. En la figura 1 [21], se muestra la evolución de las diez primeras amenazas que afectan a dichos sistemas según OWASP.

OWAPS TOP 10 - 2007	OWAPS TOP 10 - 2010	OWAPS TOP 10 - 2013	OWAPS TOP 10 - 2017
A1 – Secuencia de Comandos en Sitios Cruzados (XSS)	▲ 1 A1 – Inyección	= 0 A1 – Inyección	= 0 A1 – Inyección
A2 – Inyección	▼ -1 A2 – Secuencia de Comandos en Sitios Cruzados (XSS)	▲ 1 A2 – Pérdida de Autenticación y Gestión de Sesiones	= 0 A2 – Pérdida de Autenticación
A3 – Ejecución Maliciosa de Ficheros	⇧ 4 A3 – Pérdida de Autenticación y Gestión de Sesiones	▼ -1 A3 – Secuencia de Comandos en Sitios Cruzados (XSS)	⇧ 3 A3 – Exposición de Datos Sensibles
A4 – Referencia Directa Insegura a Objetos	▲ 1 A4 – Referencia Directa Insegura a Objetos	= 0 A4 – Referencia Directa Insegura a Objetos	(*) A4 - XML External Entities (XEE)
A5 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	= 0 A5 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	▲ 1 A5 – Configuración de Seguridad Incorrecta	(**) A5 - Perdida de Control de Acceso
A6 – Filtrado de Información y Manejo Inapropiado de Errores	(*) A6 – Defectuosa Configuración de Seguridad	(*) A6 – Exposición de Datos Sensibles	▼ -1 A6 – Configuración de Seguridad Incorrecta
A7 – Pérdida de Autenticación y Gestión de Sesiones	▲ 1 A7 – Almacenamiento Criptográfico Inseguro	(*) A7 – Ausencia de Control de Acceso a las Funciones	⇩ -4 A7 – Secuencia de Comandos en Sitios Cruzados (XSS)
A8 – Almacenamiento Criptográfico Inseguro	▲ 2 A8 – Falla de Restricción de Acceso a URL	⇩ -3 A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)	(*) A8 - Deserialización insegura
A9 – Comunicaciones Inseguras	(*) A9 – Protección Insuficiente en la Capa de Transporte	(*) A9 – Uso de Componentes con Vulnerabilidades Conocidas	= 0 A9 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Falla de Restricción de Acceso a URL	(*) A10 – Redirecciones y reenvíos no validados	= 0 A10 – Redirecciones y reenvíos no validados	(*) A10 - Registro y monitorización insuficiente

Figura 1: Evolución top ten amenazas OWASP [21].

El objetivo principal del Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP) es permitir que las organizaciones gestionen la seguridad basados en aplicaciones confiables.

Por lo que, de forma periódica, OWASP publica los diez principales ataques que sufren las aplicaciones web los cuales presentan a continuación [22]:

- A1:2017-Inyección
- A2:2017-Pérdida de Autenticación
- A3:2017-Exposición de Datos Sensibles
- A4:2017-Entidades Externas XML (XXE)
- A5:2017-Pérdida de Control de Acceso
- A6:2017-Configuración de Seguridad Incorrecta
- A7:2017-Cross-Site Scripting (XSS)
- A8:2017-Deserialización Insegura
- A9:2017-Uso de Componentes con Vulnerabilidades Conocidas
- A10:2017-Registro y Monitoreo Insuficientes

2.4. Pentesting

El pentesting o pruebas de penetración, consiste en una acción, previamente acordada entre el tester y la organización, para poner a prueba los sistemas informáticos, identificando vulnerabilidades y amenazas con la finalidad de identificar posibles peligros latentes en el software y corregirlos antes de que sean explotados por agentes maliciosos [23].

2.5. Clickjacking

Esta técnica hace referencia al método que emplea un atacante utilizando múltiples capas opacas o transparentes con la finalidad de engañar a un usuario para que dé clic en algún botón o enlace especifico y preprogramado. El atacante está secuestrando los clics destinados a la página oficial dirigiéndolos a otra página con funciones programadas para capturar información sensible. Utilizando un adecuado diseño en hojas de estilo, inputs, botones y cuadros de texto flotantes, se le hace creer al usuario que está ingresando sus credenciales de inicio de sesión en el sitio oficial, sin embargo, está escribiendo en etiquetas y elementos controlados por el atacante [29].

2.6. Sql Injection

Esta técnica consiste en insertar, o como su nombre lo indica, inyectar comandos sql a través de los casilleros de texto del ambiente cliente, a la aplicación informática. Si la explotación resulta exitosa, el atacante puede, no solo capturar ganar acceso al sistema informático, sino también leer información confidencial de la base de datos, modificar registros, ejecutar operaciones administrativas, recuperar y mostrar datos presentes en el

Gestor de Base de Datos (DBMS) y, en el peor de los casos, eliminar por completo los datos almacenados en el DBMS [30].

La técnica de inyección sql permite al atacante [30]:

- Falsificar identidad.
- Modificar información confidencial.
- Provocar problemas de repudio tales como anular transacciones o cambiar saldos.
- Divulgar la información completa almacenada en la base de datos.

La inyección sql es muy común en aplicaciones programadas en PHP y ASP, independientemente de su patrón arquitectónico, debido a la naturaleza programática de las interfaces [30].

2.7. Listado de vulnerabilidades según OWASP

OWASP recomienda en el listado de vulnerabilidades, criterios que muestran el impacto para el negocio, lo cual permitirá determinar el grado de criticidad presentando indicadores para categorizar el impacto y el riesgo en el sistema académico. A continuación, se muestra el impacto por cada riesgo de seguridad sugerido por OWASP con la finalidad de ser implementado en el sistema de notas [22]:

- **Inyección - A1:2017:** La vulnerabilidad produce divulgación, perdida o manipulación inadecuada de la información del sistema informático, por lo que el **impacto en el sistema de notas debe ser considerado como alto.**
- **Pérdida de Autenticación - A2:2017:** Dependiendo de los privilegios otorgados a las credenciales o cuentas vulneradas, el atacante puede realizar acciones como robar identidad y divulgar información sensible o confidencial protegida legalmente. Por lo que **el impacto en el sistema de notas debe ser considerado como alto.**
- **Exposición de Datos Sensibles - A3:2017:** Estas vulnerabilidades con frecuencia hacen referencia a datos que deberían estar protegidos tales como registros de salud, datos personales, información con respecto a tarjetas de crédito o débito, etc. En el caso de los sistemas académicos, hace referencia a datos personales de estudiantes, docentes y administrativos. No filtrar esta información confidencial y sensible significa para el sistema académico un **impacto medio.**
- **Entidades Externas XML (XXE) - A4:2017:** Explotar esta vulnerabilidad conlleva extraer datos, ejecutar solicitudes desde un servidor remoto, escanear la arquitectura de sistemas informáticos, ejecutar ataques de denegación de servicios y servir de base para realizar otro tipo de ataques. **El impacto para sistemas académicos, de acuerdo con la transacción y sensibilidad de la información procesada, es alto.**

- **Pérdida de Control de Acceso - A5:2017:** Debido a que el ataque depende de la asignación de perfiles o roles implementado al sistema académico, **el impacto es considerado como medio.**
- **Configuración de Seguridad Incorrecta - A6:2017:** La vulnerabilidad expone funciones del sistema académico, así como también permite generar accesos no autorizados al servidor web o a la aplicación directamente. **El impacto para el sistema de notas de acuerdo con lo expuesto es medio.**
- **Cross Site Scripting (XSS) - A7:2017: El impacto se considera medio o controlable en el caso de XSS reflejado y XSS en DOM, mientras que se considera alto para el caso de XSS almacenado** que permite ejecutar secuencia de comandos directamente en el navegador donde se ejecuta el sistema de notas.
- **Deserialización Insegura - A8:2017:** El atacante puede ejecutar secuencia de comandos remotos al explotar esta vulnerabilidad. Para el sistema académico utilizado por la institución educativa que figura como caso de estudio, no es necesario el proceso de deserialización cadenas de datos en los dos extremos, sin embargo, **el impacto debe ser considerado como medio.**
- **Uso de Componentes con Vulnerabilidades Conocidas - A9:2017:** El activo tecnológico que se está protegiendo transacciona información crítica y sensible, ya que además de datos administrativos, también gestiona datos académicos como calificaciones por estudiante, por lo que **el impacto es alto.**
- **Registro y Monitoreo Insuficientes - A10:2017:** Si el sistema de notas permite el escaneo exitoso de vulnerabilidades, **el impacto es medio**, ya que el atacante depende de este punto para continuar con el hacking ético.

2.8. Mantenimiento de aplicaciones seguras según OWASP

El proyecto OWASP proporciona una serie de controles técnicos orientados al mantenimiento seguro de aplicaciones web denominado Application Security Verification Standard (ASVS), este listado es también utilizado en el desarrollo de software orientado a la web [33].

El estándar de verificación de seguridad de aplicaciones (ASVS) entrega un listado de pruebas de seguridad de sistemas web que son utilizados por arquitectos de software, testers (QA), oficiales o jefes de seguridad (CISO) y por personal que adquiere herramientas informáticas con la finalidad de definir, probar y verificar aplicaciones seguras en etapas de construcción o mantenimiento [34], sin embargo, las sugerencias serán orientadas al mantenimiento del sistema de notas y mas no a su desarrollo como tal, debido a que la aplicación se encuentra en producción. Este listado de pruebas o requerimiento

orientados a aplicaciones seguras fue desarrollado por OWASP tomando en cuenta los siguientes objetivos [33]:

- **Usándolo como métrica**, de tal manera que proporcione a los desarrolladores y propietarios de aplicaciones seguras **indicadores o criterios** que garanticen la evaluación del grado de confianza que tienen los sistemas informáticos.
- **Usándolo como guía**, con el objetivo de brindar al personal de seguridad informática **controles** que satisfagan los requisitos de seguridad de la aplicación web.
- **Usándolo durante la adquisición**, con la finalidad de entregar al adquiriente una **base que especifica** los requisitos de verificación de seguridad de la aplicación detallados en los **contratos**.

ASVS define tres niveles de validación o verificación alineados a la seguridad de aplicaciones web, las cuales se listan a continuación [34]:

- **Nivel 1**: Se refiere a niveles bajos de seguridad y la evaluación por penetración es totalmente efectiva.
- **Nivel 2**: Se orienta a sistemas web que transaccionan datos confidenciales por lo que requiere protección. En este nivel se recomienda evaluar a la mayoría de las aplicaciones tecnológicas.
- **Nivel 3**: Es para aplicaciones con un nivel de criticidad alto, aplicaciones que transaccionan datos sensibles y confidenciales, es decir, aplicaciones que necesitan tener un alto nivel de confianza como sistemas médicos.

3. PLANIFICACIÓN PARA EL ANÁLISIS DE VULNERABILIDADES

El objetivo de esta etapa es generar un plan que sirva de guía para ejecutar de manera efectiva el análisis de vulnerabilidades considerando las siguientes actividades: identificar el alcance del análisis de vulnerabilidades, identificar los activos, análisis de amenazas y vulnerabilidades propias del sistema de notas, análisis del impacto y probabilidad y, finalmente, medición del riesgo.

3.1. Identificación del alcance

El análisis de vulnerabilidades en el sistema de notas de instituciones educativas tiene como objetivo generar una guía especifica que permita realizar este proceso analítico de forma secuencial, ordenada y documentada, contando con indicadores que permitan evaluar y prevenir ataques informáticos. Por tal motivo, se plantea inicialmente realizar un estudio con respecto al estado actual de los activos informáticos y de la seguridad

implementada en el caso de estudio (Instituto Tecnológico Quito). Luego, se plantea ejecutar el análisis de vulnerabilidades utilizando técnicas de hacking ético. Los dominios por evaluar son:

- Distribución lógica y física de la red interna, puertos y protocolos activos en los servidores donde se aloja el sistema de notas.
- Puertos y servicios activos en el gestor de base de datos donde se aloja la información correspondiente a las notas.
- Seguridad lógica y física de los servidores donde se encuentra el sistema de notas.
- Seguridad que emplean los usuarios del sistema de notas para su manipulación.

Posterior, se genera la guía donde se especifica el proceso a seguir para realizar el análisis de vulnerabilidades al sistema de notas considerando recursos como:

- Fechas específicas para el análisis
- Herramienta o paquetes de herramientas que sirvan para ejecutar el análisis de vulnerabilidades
- Indicadores de riesgo de acuerdo con el impacto y la probabilidad del riesgo
- Acciones posteriores al análisis.

Cabe aclarar que el análisis de vulnerabilidades en el sistema de notas de instituciones educativas no generará un SGSI (Sistema de Gestión de la Seguridad de la Información). Finalmente, para el análisis de vulnerabilidades en el sistema de notas de instituciones de educación no es necesario cubrir las huellas, ya que el objetivo es justamente evidenciar las debilidades de seguridad encontradas, con la finalidad de explotarlas, inicialmente, y luego proponer y ejecutar soluciones.

3.2. Identificación de los activos

A continuación, se describen los activos tecnológicos con los que cuenta el Instituto Tecnológico Quito. Los cuáles serán sometidos al análisis de vulnerabilidades de acuerdo con el alcance del proyecto. El Instituto cuenta con los siguientes servicios en línea, como se muestra en la figura 2 [27]:

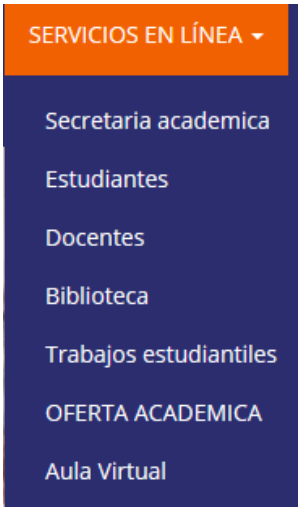


Figura 2: Servicios en línea ITQ [27].

3.2.1.Secretaría académica

El servicio en línea denominado “secretaría académica”, se encarga de los procesos para ingreso, actualizaciones, reportes, encuestas, documentos de matriculación y coordinación para la evaluación al docente, como se muestra en la figura 3 [27].

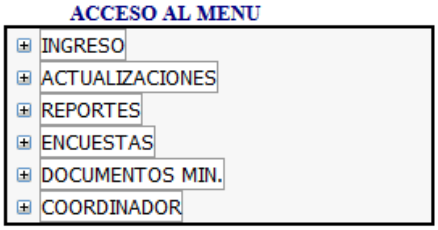


Figura 3: Procesos en línea del ambiente de Secretaría Académica [27].

La sección de ingreso tiene como objetivo: crear estudiantes, ingresar nuevos docentes; listar, ingresar, eliminar y actualizar periodos académicos; listar, Ingresar, eliminar y actualizar carreras; duplicar asignaturas; crear mallas; listar, crear, eliminar y actualizar fechas y horas para matriculación mostrando también su modalidad, como se puede visualizar en la figura 4 [27].

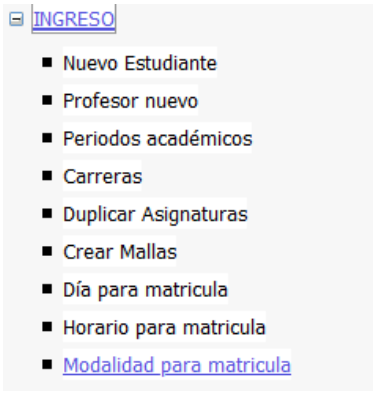


Figura 4: Sección para ingresar los actores en el proceso académico [27].

En la sección actualizaciones, como se muestra en la figura 5 [27], se procesan las matrículas por estudiante; la asignación profesores por asignatura; la actualización los datos del profesor; la eliminación de estudiantes o profesores; crear, editar, eliminar y listar asignaturas; finalmente, ingresar practicas preprofesionales. Existen algunos ambientes que no se utilizan para la gestión académica.

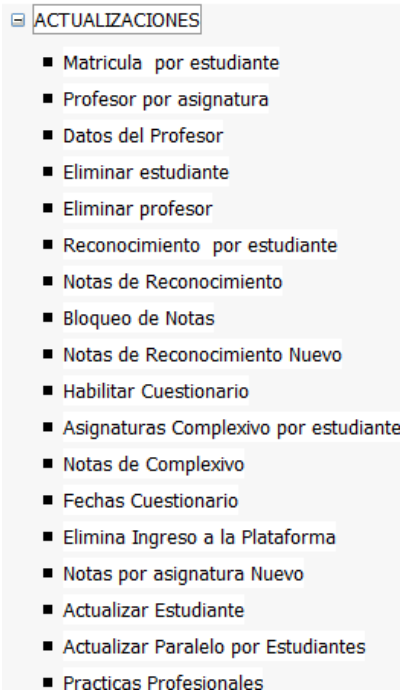


Figura 5: Sección para actualizar información académica [27].

Como se muestra en la figura 6 [27], la sección de reportes está destinada a generar datos procesados con la finalidad de comunicar información académica, entre las cuales se encuentra la información propia del sistema de notas. Los reportes que tienen más frecuencia de uso son: reporte de notas, reporte de notas por estudiante, profesor por

asignatura, lista de estudiante por carrera, ver evaluación de estudiantes, notas por asignaturas y lista de matrículas por periodos.

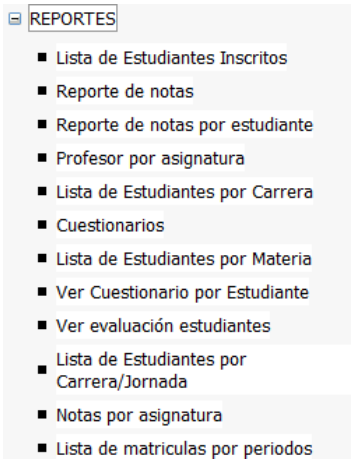


Figura 6: Sección de reportes para la gestión académica [27].

En la figura 7 [27], se visualizan las secciones que actualmente no son utilizadas ya que no generan el aporte esperado en la gestión académica.

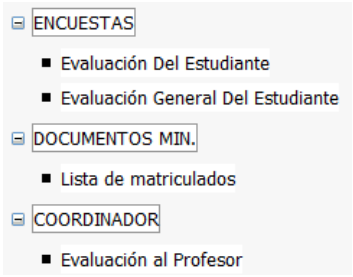


Figura 7: Secciones no utilizadas [27].

3.2.2.Estudiantes

El servicio en línea para estudiantes permite consultar las calificaciones pertenecientes a los dos últimos periodos como se indica en la figura 8 [27].

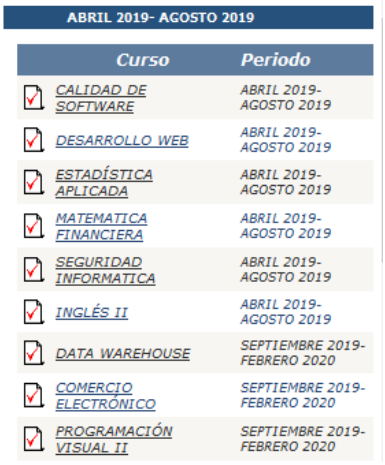


Figura 8: Calificaciones por materia y por periodo académico [27].

El estudiante puede ingresar también a servicios para consultar sus notas, visualizar sus prácticas profesionales y evaluar al docente por asignatura. La figura 9 [27], muestra los tres servicios que el estudiante puede realizar en este ambiente.

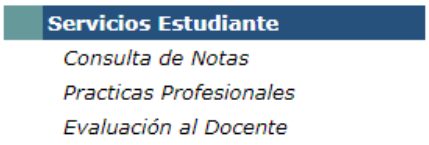


Figura 9: Servicios Estudiantes [27].

3.2.3.Docentes

Este servicio en línea permite al docente generar información académica. Esta información podrá ser creada, consultada y actualizada dependiendo del evento que se desea realizar. Los servicios que el docente tiene acceso son los que se muestran en la figura 10 [27].

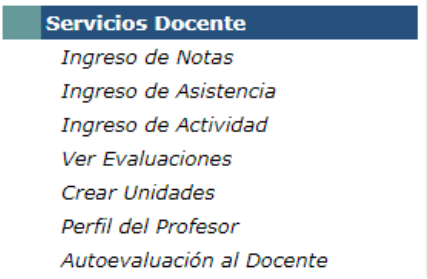


Figura 10: Servicios Docente [27].

Los servicios para el docente que se utilizan con mayor frecuencia son: ingreso de notas y autoevaluación docente. Este ambiente para el ingreso de notas es el sistema oficial donde se ingresa las calificaciones que serán procesadas para cumplir con la gestión académica interna del instituto.

De acuerdo con un periodo de tiempo específico, la aplicación para ingresar notas es habilitado y se muestra de la siguiente manera (Figura 11) [27]:

Figura 11: Ingreso de notas por asignatura y docente [27].

Mientras que, el ambiente para realizar la autoevaluación docente también depende de un periodo de tiempo y se muestra de la siguiente manera (Figura 12) [27]:

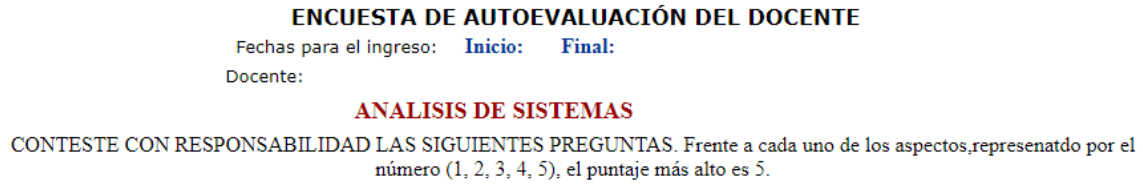


Figura 12: Encuesta de autoevaluación del docente [27].

3.2.4.Biblioteca

El servicio en línea biblioteca, muestra dos tesis de grado realizadas por estudiantes del instituto y subidas al repositorio como indica la figura 13 [27].

Figura 13: Servicio en línea biblioteca [27].

3.2.5.Trabajos estudiantiles

El servicio en línea trabajos estudiantiles, de momento no se utiliza y este enlace no se encuentra disponible.

3.2.6.Oferta académica

El servicio en línea oferta académica, como expone la figura 14 [27], ofrece los siguientes servicios: profesionalízate, especialízate, actualízate y SETEC – CERTIFICACIONES, los cuales son netamente informativos.

Figura 14: Servicio en línea oferta académica [27].

3.2.7.Aula virtual

El entorno virtual de aprendizaje (EVA), es un servicio contratado por el instituto tecnológico quito. Por tal motivo, este sistema en línea, no se encuentra alojado en los servidores locales del instituto.

El aula virtual o EVA, esta creado a partir de Moodle, por lo que cuenta con todas las funciones que esta herramienta de aprendizaje entrega para la gestión académica virtual.

A pesar de que esta herramienta permite generar contenido académico y, este a su vez, genera en calificaciones, este ambiente no es el sitio oficial donde se gestionan las notas en el instituto.

Para acceder a este servicio en línea y poder hacer uso de las herramientas propias del EVA, se necesitan credenciales únicas entregadas a cada actor del proceso académico virtual como indica la figura 15 [27.

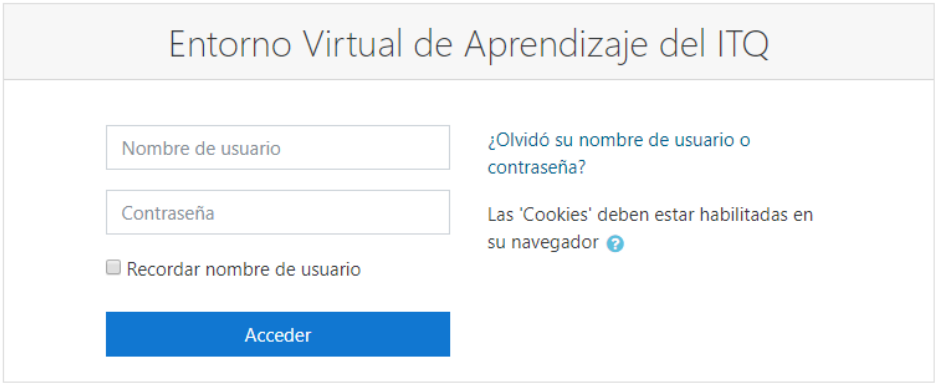


Figura 15: Entorno virtual de aprendizaje EVA [27].

3.3. Escenario de ataque

De acuerdo con lo evidenciado en el planteamiento del problema los sistemas de notas de instituciones educativas forman parte de un grupo vulnerable que son víctimas de ataques tecnológicos, por lo que a continuación se listan las vulnerabilidades más comunes:

a. SQL Injection: Consiste en realizar un ataque en el que el código SQL es agregado o insertado directamente a los parámetros de entrada que luego son enviados a un servidor de base de datos para su ejecución. Se considera vulnerable a cualquier segmento que construya sentencias SQL. La forma más común de este ataque se lo realiza en parámetros que concatenan comandos SQL para luego ser ejecutados como consultas de lectura de registros, mientras que ataques menos comunes se realizan en cadenas destinadas al almacenamiento de los registros de una tabla. Cuando un atacante logra modificar la instrucción SQL utilizando esta técnica, la ejecución se realiza con los mismos privilegios del usuario de la aplicación. [24]
b. Exploiting SQL: Realiza la explotación SQL en sitios web que muestran los resultados de la consulta directamente en el HTML. Explotar esta vulnerabilidad dependerá mucho de los privilegios del usuario que realiza las peticiones al servidor SQL, en el servidor de base de datos exacto y en que acción especifica se desea realizar, tales como, extraer datos, modificarlos o ejecutar comandos en el host remoto [24].
c. Broken Authentication and Session Management: Este tipo de ataques se realiza a través de fuerza bruta o diccionario con la finalidad de romper el hash de las contraseñas. Los atacantes utilizan diversas herramientas para poder realizar millones de combinaciones posibles logrando obtener las credenciales de usuario y contraseña de los sitios web. A demás, el atacante ejecuta esta técnica utilizando credenciales administrativas por defecto [22].

d. Cross Site Scripting: Hace referencia a un ataque informático de código malicioso destinado a sitios web. El atacante envía código malicioso a la aplicación con la finalidad de conducir al usuario a otro sitio web donde extrae información sensible. Este ataque puede derivar en negación de servicios (DDos). Generalmente se crea un sitio web similar al original, engañando al usuario para lograr obtener información específica que en ocasiones sirve como inicio de otro tipo de ataque. Las diversas variantes de este ataque se dividen en dos grupos [25]:
    i. Cross Site Scripting persistente o directo, consiste en incrustar etiquetas, como <script> o <frame>, dentro del html del sistema víctima, siempre y cuando este lo permita.
    ii. Cross Site Scripting reflejado o indirecto: Se ejecuta al modificar valores que el sitio web envía, sin emplear sesiones, de una página a otra página web. Se puede atacar también cuando la aplicación envía mensajes en la URL o en la cabecera HTTP.
e. Security Misconfigurations: Los hackers maliciosos están siempre en la búsqueda de vulnerabilidades que les permitan acceder a sitios web que no cuentan con una configuración adecuada de seguridad. La gran cantidad de estos ataques se deben a que las configuraciones sean las predeterminadas. Con la finalidad de obtener acceso no autorizado, los atacantes buscan en las aplicaciones las siguientes falencias de seguridad [26]:
    i. Fallos sin parches
    ii. Configuraciones predeterminadas, incluyendo credenciales de acceso.
    iii. Paginas o secciones de páginas sin utilizar.
    iv. Archivos y directorios publicados en carpetas del servidor web sin protección alguna.
    v. Servicios y puertos innecesarios que se encuentran activos.

El contexto del ataque que se ejecutará en el sistema de notas que sirve como caso de estudio se enfocará en Inyección SQL y clickjacking debido a que se asume que el software carece de encriptación de credenciales sensibles en el formulario de login, así como también, puede permitir envolver a todo el sistema dentro de un iframe.

4. EJECUCIÓN DEL ANÁLISIS

En el presente capítulo se realiza la ejecución del análisis utilizando la guía que proporciona OWASP para pruebas de seguridad en sistemas informáticos. OWASP muestra una lista

de controles que se recomiendan utilizar para realizar la evaluación de vulnerabilidades en sistemas informáticos, los cuales son listados a continuación [28]:

- Information Gathering: Esta prueba permite recopilar información utilizando las siguientes categorías.
    - OTG-INFO-001: Descubrimiento del motor de búsqueda y realización un reconocimiento de fugas de información.
    - OTG-INFO-002: Reconocimiento de huellas digitales en el servidor web.
    - OTG-INFO-003: Revisión de los archivos y metarchivos del servidor web con la finalidad de detectar fugas de información.
    - OTG-INFO-004: Listar aplicaciones, servicios y puertos alojados en el servidor web
    - OTG-INFO-005: Revisión de comentarios y metadatos de la aplicación web para identificar fuga de información.
    - de rutas para ejecuciones a través de la aplicación web.
    - OTG-INFO-008: Marco de aplicación de huellas digitales.
    - OTG-INFO-009: Identificar la huella digital de la aplicación web.
- Data Validation Testing: Esta prueba está orientada a la validación de información y datos para lo cual, se deben ejecutar las siguientes categorías:
    - OTG-INPVAL-001: Identificación de la secuencia de comandos XSS reflejados
    - OTG-INPVAL-002: Identificación de la secuencia de comandos XSS almacenados
    - OTG-INPVAL-005: Pruebas para inyectar SQL.
    - OTG-INPVAL-016: Prueba de división o contrabando de HTTP
- Client Side Testing: Esta prueba se ejecuta del lado del cliente para lo cual se ejecutan las categorías listadas a continuación:
    - OTG-CLIENT-001: Ejecución de pruebas orientada a secuencia de comandos de sitos cruzados basados en DOM
    - OTG-CLIENT-002: Ejecución de pruebas para la implementar javascript.
    - OTG-CLIENT-003: Ejecución de pruebas para inyección de HTML
    - OTG-CLIENT-004: Ejecución de pruebas con el objetivo de redireccionar URLs del lado del cliente
    - OTG-CLIENT-005: Prueba para inyección CSS.
    - OTG-CLIENT-009: Pruebas para ClickJacking.

Se utilizarán las pruebas y categorías listadas anteriormente con el objetivo de realizar la evaluación de vulnerabilidades alineadas a las etapas del hacking ético (descritas en la justificación teórica y en el marco teórico) que se muestran a continuación:

- Reconocimiento
- Escaneo
- Ganar acceso (explotar vulnerabilidades)
- Mantener acceso (explotar vulnerabilidades)
- Cubrir huellas

4.1 Reconocimiento

En la presente etapa de hacking ético se utilizarán herramientas de reconocimiento con el objetivo de detectar servicios de red, direcciones IP y puertos abiertos. Esta información obtenida en esta etapa servirá de insumo para la etapa de Escaneo.

4.1.1. Information Gathering

Las siguientes categorías permitirán obtener o ganar información que será utilizada como insumo en las siguientes etapas del hacking ético.

4.1.1.1 Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)

Herramienta: Sitedigger

Objetivo: El objetivo de este análisis es buscar ficheros y cadenas de texto con la finalidad de encontrar fallas en el servidor web a nivel de repositorios.

La herramienta mencionada, escanea de forma automatizada los ficheros del servidor buscando fallos que permitirán realizar la explotación de la vulnerabilidad. Se debe colocar la URL o la dirección IP en Site/Domain para luego proceder a realizar el escáner, como muestra la figura 16.

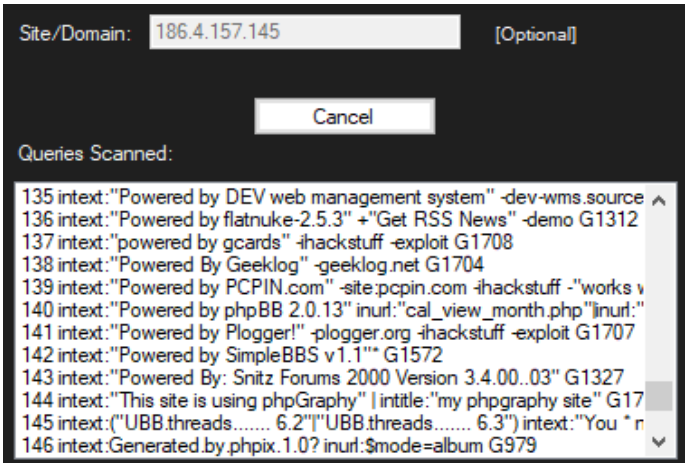


Figura 16: Ejecución de la aplicación Sitedigger. [Elaboración propia]

Resultados: El sistema de calificaciones bloquea este tipo de análisis ya que se ha configurado y programado reglas en el archivo .htaccess, también debido a que el sistema está desarrollado en .NET el proceso de compilación y publicación oculta archivos que contienen datos, información y procesos automatizados sensibles. Este criterio implementado en el sistema de notas impide encontrar ficheros de configuración del servidor donde se encuentra alojado la aplicación. El nivel de impacto para el Instituto es bajo debido a la nula obtención de información, mientras que la probabilidad de ocurrencia es baja ya que la herramienta no ha logrado acceder a los ficheros de configuración que se está buscando. De acuerdo con lo expuesto, el **riesgo se encuentra en una zona tolerable** de acuerdo con tabla 7.

4.1.1.2 Fingerprint Web Server (OTG-INFO-002).

Herramienta: NetCraft

Objetivo: El objetivo de este análisis es encontrar información del sitio como el nivel de riesgo que representa esta herramienta para el sistema de notas como se presenta en la figura 17.

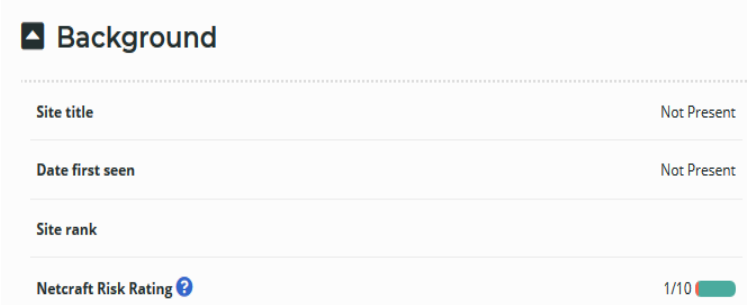
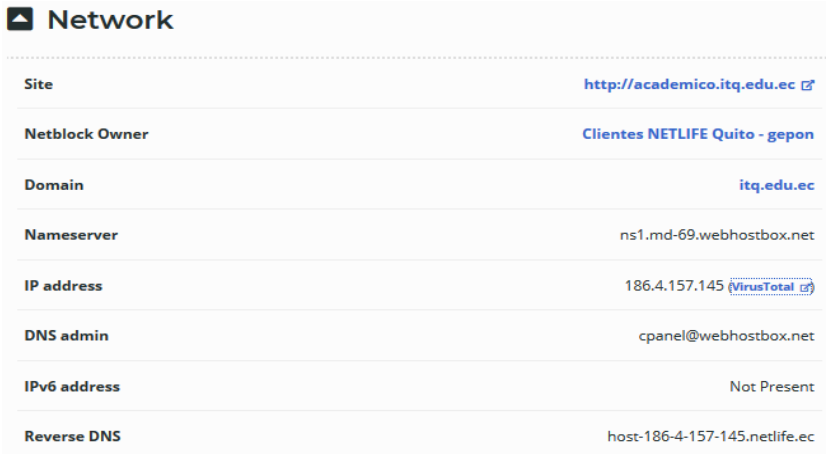


Figura 17: Ejecución de NetCraft, resulatados Backgroud. [Elaboración propia]

También presenta información referente al dominio, el nombre del proveedor, la IP pública del servidor donde se encuentra alojado el sistema de notas, entre otra información de baja relevancia como se visualiza en la figura 18.

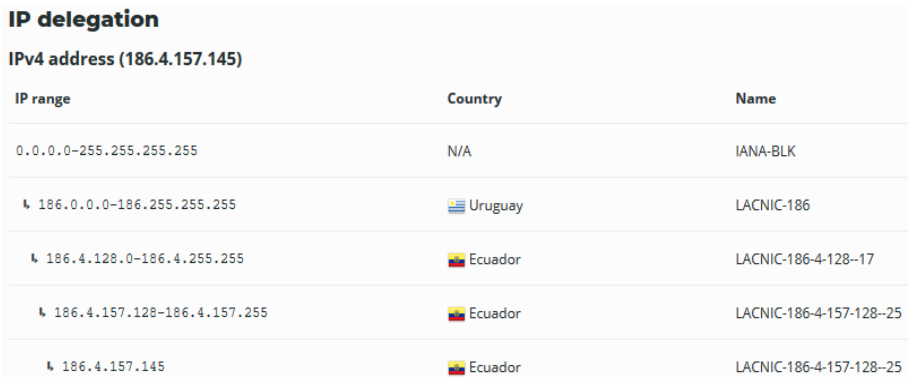


Network

Site	http://academico.itq.edu.ec
Netblock Owner	Clientes NETLIFE Quito - gepon
Domain	itq.edu.ec
Nameserver	ns1.md-69.webhostbox.net
IP address	186.4.157.145 (VirusTotal)
DNS admin	cpanel@webhostbox.net
IPv6 address	Not Present
Reverse DNS	host-186-4-157-145.netlife.ec

Figura 18: Ejecución de NetCraft, resulatados Network. [Elaboración propia]

Finalmente entrega resultados con respecto a los nodos que forman parte de la infraestructura lógica o física de la red que es necesaria atravesar para acceder al servicio de sistema de notas. Esta información parte de la IP publica como se presenta en la figura 19.



IP delegation

IPv4 address (186.4.157.145)

IP range	Country	Name
0.0.0.0-255.255.255.255	N/A	IANA-BLK
186.0.0.0-186.255.255.255	Uruguay	LACNIC-186
186.4.128.0-186.4.255.255	Ecuador	LACNIC-186-4-128--17
186.4.157.128-186.4.157.255	Ecuador	LACNIC-186-4-157-128--25
186.4.157.145	Ecuador	LACNIC-186-4-157-128--25

Figura 19: Ejecución de NetCraft, resulatados de IP delegation. [Elaboración propia]

Resultados: Al ejecutar NetCarft se logró obtener información referente a la IP pública del sitio, el dominio y el proveedor. Estos resultados son considerados como sensibles debido a que serán utilizados como insumos para explotar vulnerabilidades en etapas siguientes. El impacto de ejecutar esta herramienta es bajo ya que esta información es pública y, de cierto modo, necesaria para transaccionar en la web, mientras que la probabilidad de ocurrencia es alta debido a que la herramienta es accesible y usable mientras que los datos arrojados son altamente digeribles y no se requiere de un criterio técnico elevado para

lograr interpretarlos. Considerando estos dos criterios, el **riesgo se encuentra en una zona moderada** de acuerdo con tabla 7.

4.1.1.3 Review Web Server Metafiles for Information Leakage (OTG-INFO-003).

Herramienta: DomainTools. Para su ejecución basta con introducir el dominio en el casillero de texto y presionar el botón buscar (Search) como muestra la figura 20.

Objetivo: Encontrar o identificar la IP publica de sitio, el tipo de servidor, el dominio y el nombre de dominio, el tipo y versión del servidor y la información básica del contratante o dueño del sitio web, para lo cual se introduce la URL del sistema de notas como se muestra en la figura 20.

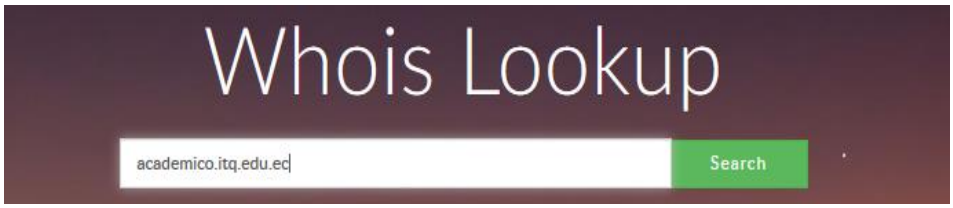
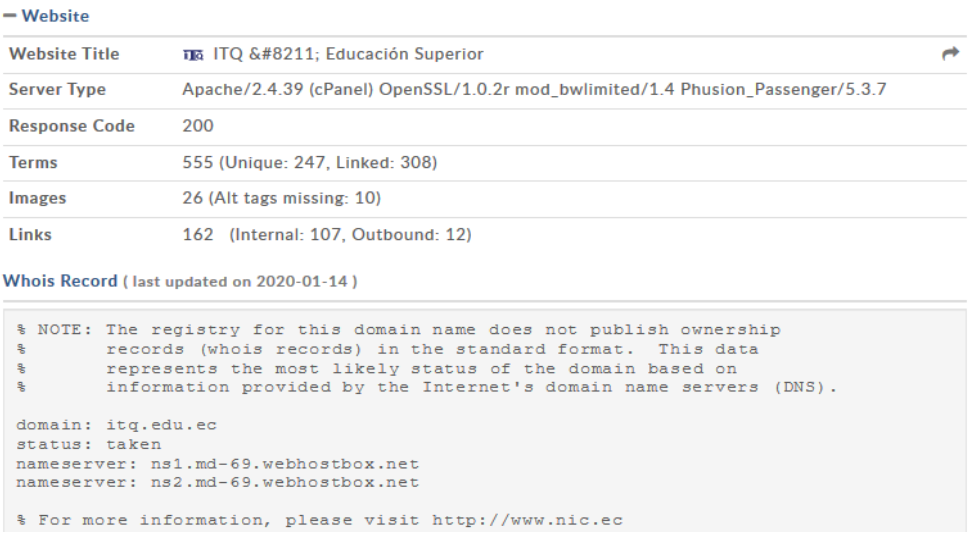


Figura 20: Ejecución del software DomainTools. [Elaboración propia]

La figura 21 muestra la información resultante posterior a la ejecución de la herramienta DomainTools. Sin embargo, no entrega los datos básicos del dueño del sitio web.



Website

Website Title	ITQ – Educación Superior
Server Type	Apache/2.4.39 (cPanel) OpenSSL/1.0.2r mod_bwlimited/1.4 Phusion_Passenger/5.3.7
Response Code	200
Terms	555 (Unique: 247, Linked: 308)
Images	26 (Alt tags missing: 10)
Links	162 (Internal: 107, Outbound: 12)

Whois Record (last updated on 2020-01-14)

```
% NOTE: The registry for this domain name does not publish ownership
%       records (whois records) in the standard format.  This data
%       represents the most likely status of the domain based on
%       information provided by the Internet's domain name servers (DNS).

domain: itq.edu.ec
status: taken
nameserver: ns1.md-69.webhostbox.net
nameserver: ns2.md-69.webhostbox.net

% For more information, please visit http://www.nic.ec
```

Figura 21: Resultados del software DomainTools. [Elaboración propia]

La información de como se ejecuta esta herramienta se encuentra en el anexo 1.

Resultados: Al finalizar la ejecución de DomainTools, la herramienta entregó como resultado el nombre de dominio, el tipo de servidor y la IP. Esta información es considerada sensible ya que será utilizada para explotar vulnerabilidades en etapas siguientes.

El impacto de ejecutar esta herramienta es bajo ya que la información obtenida es publica, mientras que la probabilidad de ocurrencia es alta debido a la exposición global de estos datos, cualquier entidad o usuario puede acceder a estos resultados. De acuerdo con lo expuesto, el **riesgo se encuentra en una zona moderada** de acuerdo con tabla 7.

4.1.2. Presentación de resultados de la etapa de reconocimiento.

La información obtenida en este capítulo servirá como entrada para realizar el escaneo de vulnerabilidades en el sistema de notas. Los resultados obtenidos fueron los siguientes:

- Information Gathering
    - Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001), utilizando la herramienta Sitedigger no entrego informacion valiosa. **Zona de riesgo es tolerable**
    - Fingerprint Web Server (OTG-INFO-002), utilizando la herramienta Netcraf entrego información referente al dominio, IP publica, la compañía que entrega el servicio de hosting y el manejador del reverse DNS. **Zona de riesgo es moderada**
    - Review Web Server Metafiles for Information Leakage (OTG-INFO-003), la herramienta DomainTools presento como resultado de su análisis información con referente al servidor web, IP publica, dominio y nombre del servidor. **Zona de riesgo es moderada**

4.2 Escaneo de vulnerabilidades

En la presente etapa de hacking ético se utilizarán herramientas para escanear vulnerabilidades en el sistema de notas considerando la información obtenida en la etapa de reconocimiento y catalogada en zonas de riesgo moderadas.

4.2.1 Information Gatering

4.2.1.1 Enumerate Applications on Webserver (OTG-INFO-004).

Herramienta: Nmap, esta aplicación ha sido ejecutada desde una máquina virtual de Kali Linux.

Objetivo: Rastrear puertos abiertos a partir del ingreso de la dirección IP del objetivo a ser evaluado.

El comando de ejecución, como se expone en la figura 22, es: nmap -T4-A-v (IP). Para lanzar el análisis se seleccionó el perfil denominado “Intense scan”.

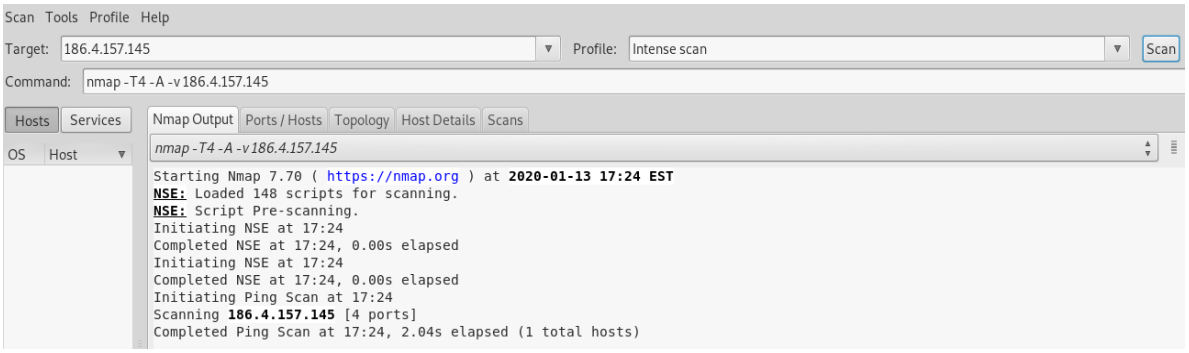
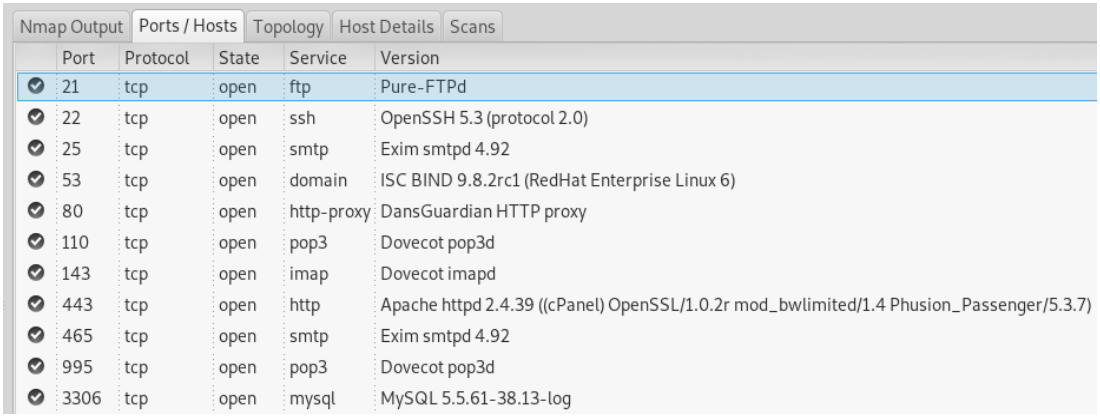


Figura 22: Ejecución Nmap en Kali Linux. [Elaboración propia]

Al terminar la ejecución, los resultados muestran un listado de puertos abiertos (figura 23), los cuales hacen referencia al tipo de servicio que se está ejecutando en el servidor web analizado a través de su IP.



Port	Protocol	State	Service	Version
21	tcp	open	ftp	Pure-FTPd
22	tcp	open	ssh	OpenSSH 5.3 (protocol 2.0)
25	tcp	open	smtp	Exim smtpd 4.92
53	tcp	open	domain	ISC BIND 9.8.2rc1 (RedHat Enterprise Linux 6)
80	tcp	open	http-proxy	DansGuardian HTTP proxy
110	tcp	open	pop3	Dovecot pop3d
143	tcp	open	imap	Dovecot imapd
443	tcp	open	http	Apache httpd 2.4.39 ((cPanel) OpenSSL/1.0.2r mod_bwlimited/1.4 Phusion_Passenger/5.3.7)
465	tcp	open	smtp	Exim smtpd 4.92
995	tcp	open	pop3	Dovecot pop3d
3306	tcp	open	mysql	MySQL 5.5.61-38.13-log

Figura 23: Resultado de puertos abiertos utilizando Nmap. [Elaboración propia]

Finalmente, en pestañas siguientes de la herramienta Nmap, se presenta como resultado de la ejecución la topología propia desde donde se origina el análisis hasta su destino como se visualiza en la figura 24.

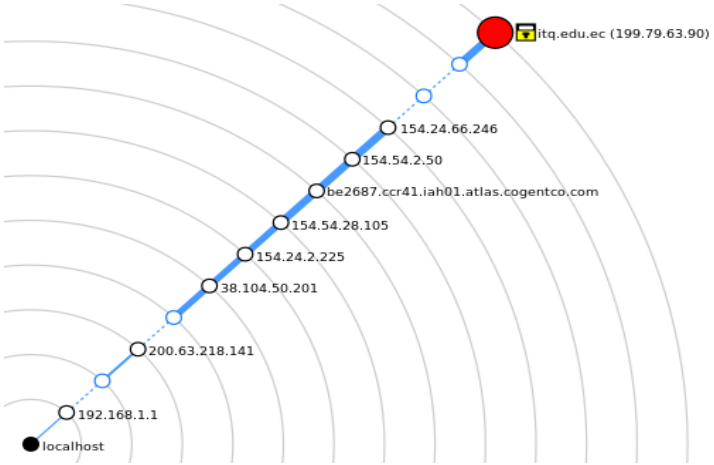


Figura 24: Topología origen - destino con Nmap. [Elaboración propia]

Resultados: Se encontraron once puertos abiertos, los cuales servirán como insumos para explotar vulnerabilidades en etapas siguientes de hacking ético.

El impacto de ejecutar esta herramienta Nmap es medio debido a que la información resultante del análisis es crítica y sensible, mientras que la probabilidad de ocurrencia es media debido a que se necesita criterios técnicos para ejecutar e interpretar el análisis. De acuerdo con lo expuesto, el **riesgo está en una zona moderada** de acuerdo con tabla 7.

4.2.1.2 Review webpage comments and metadata for information leakage (OTG-INFO-005)

Herramienta: OWASP ZAP.

Objetivo: Identificar metadatos alojados en el servidor web, en la aplicación y/o en la arquitectura del sistema de notas, los cuales estén generando filtrado o fuga no autorizada de información crítica.

En el campo de texto “URL to attack”, se ingresa el dominio del sistema de notas para luego presionar en el botón “Atacar” con la finalidad de iniciar y ejecutar ZAP como se presenta en la figura 25.

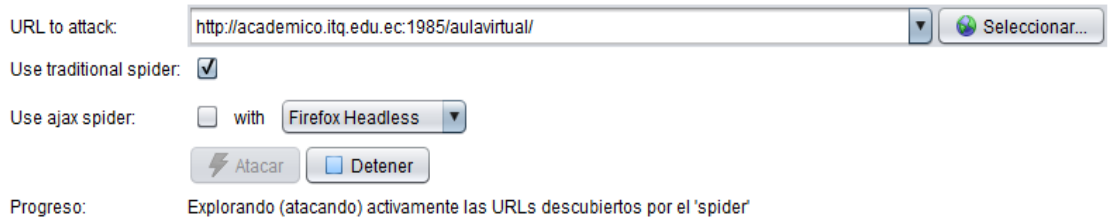


Figura 25: Ejecución de ZAP. [Elaboración propia]

El proceso completo de la ejecución, datos, información, descripción, y detalles se encuentra en el Anexo 1.

Resultados: De acuerdo con la segmentación que ZAP realiza para representar los riesgos, se tienen los siguientes resultados:

Para los riesgos altos, la ausencia del firmado o MAC en el ViewState evidencia la falta de seguridad al transaccionar información sensible en el sistema de notas.

Para los riesgos medios, evidencia que el sistema de notas es propenso a recibir ataques de ClickJacking. Finalmente, los resultados de los riesgos bajos presentas debilidades con respecto a la protección de ataques de XSS (Cross Site Scripting).

Por lo tanto, el impacto de ejecutar esta herramienta es alto debido a que la información resultante es sensible y critica, nos indica que técnicas se va a utilizar para la explotación de vulnerabilidades tal como Clickjacking y XSS. Mientras que la probabilidad de ocurrencia es media ya que para la ejecución de ZAP y la interpretación de los resultados es necesario

tener un conocimiento técnico. Por lo tanto, el **riesgo está en una zona importante** de acuerdo con tabla 7.

4.2.1.3 Fingerprint Web Application Framework (OTG-INFO-008)

Herramienta: WAPPALYZER, se instala como plugin en el navegador para ser ejecutado.
Objetivo: Identificar el lenguaje de programación, el framework, características de la base de datos y el servidor web donde está corriendo el sistema de notas.
La figura 26 evidencia la ejecución del software que realiza el análisis.

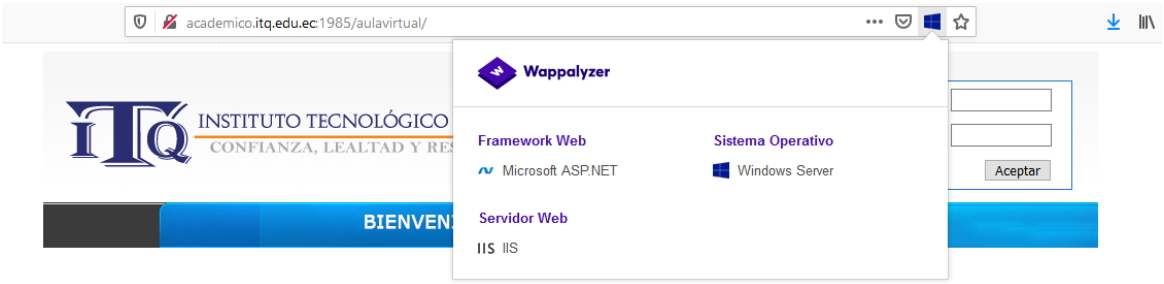


Figura 26: Ejecución de WAPPALYZER. [Elaboración propia]

Resultados: La ejecución de la aplicación WAPPALYZER muestra la información del framework (Microsoft ASP.NET), la información del sistema operativo del servidor (Windows Server) y finalmente muestra el servidor web (IIS). El impacto al ejecutar la herramienta es medio debido a que proporciona información con respecto a las características del servidor y la arquitectura de software, mientras que la probabilidad de ocurrencia es media ya que es necesario un criterio técnico para la interpretación de los resultados, por lo que **el riesgo se encuentra en una zona moderada** de acuerdo con tabla 7.

4.2.1.4 Fingerprint Web Application (OTG-INFO-009)

Herramienta: Nikto, herramienta instalada y ejecutada en una máquina virtual de Kali Linux.
Objetivo: Identificar información sensible del servidor tal como IP publica, dominio, puerto por defecto, características del servidor web, lenguaje de programación, si existe protección contra ClickJacking y ataques de XSS, IP dentro de la red local del servidor y métodos HTTP para transferencia de datos.
Para la ejecución de Nikto se utiliza el siguiente comando en una consola de Kali Linux: nikto – host (dominio o IP del sistema de notas) –output press.txt como se presenta en la figura 27.

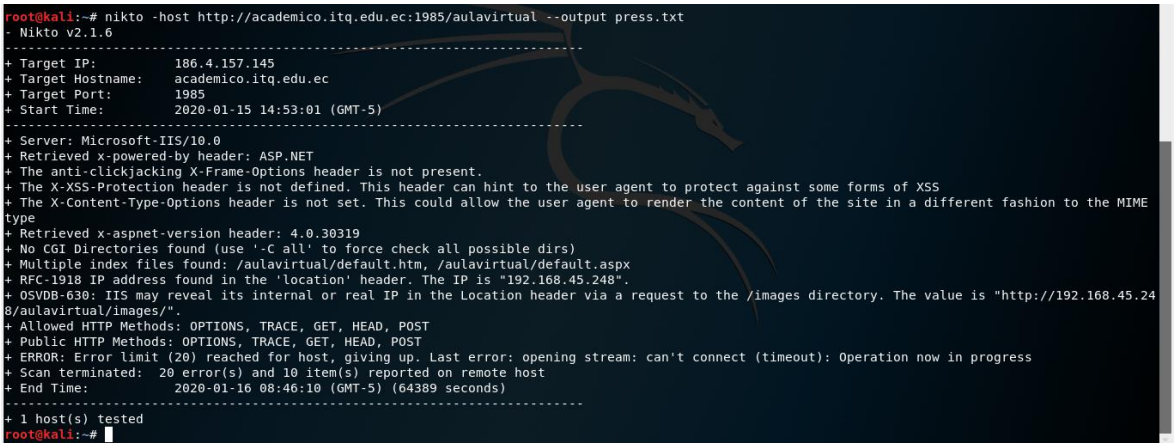


Figura 27: Ejecucion de Nikto. [Elaboración propia]

El comando mencionado almacena los resultados en un archivo denominado press.txt, el cual se muestra a continuación en la figura 28:

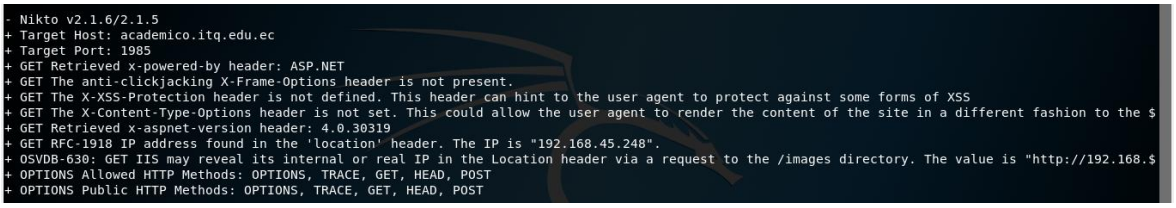


Figura 28: Archivo de resultados press.txt. [Elaboración propia]

Resultados: La ejecución de la herramienta Nikto entregó información sensible tal como la IP del servidor en la red local y evidenció que el servidor es vulnerable a Clickjacking, lo cual serán insumos para la etapa de explotación de vulnerabilidades, por lo que el impacto es alto debido a que ya presenta técnicas para el ataque y también muestra la IP del servidor dentro de su propia red. Mientras que la probabilidad de ocurrencia es media ya que se necesita un criterio técnico para la ejecución e interpretación de la herramienta y sus resultados, por lo que el **riesgo se encuentra en una zona importante** de acuerdo con tabla 7.

4.2.2 Data Validation Testing

4.2.2.1 Testing for Reflected Cross Site Scripting (OTG-INPVAL-001)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Evaluar la posibilidad de ejecutar sentencias javascript para XSS.

Para evidenciar la ejecución de javascript, se ha colocado el siguiente código a continuación de la URL del sistema de notas:

';alert(String.fromCharCode(88,83,83))//\';alert(String.fromCharCode(88,83,83))//";alert(String.fromCharCode(88,83,83))//\";alert(String.fromCharCode(88,83,83))//--></script>">'><script>alert;alert(String.fromCharCode(88,83,83))</script>

La ejecución con el código expuesto no obtuvo ningún resultado visible a nivel de cliente, sin embargo, el navegador muestra la siguiente información en la sección de la consola:

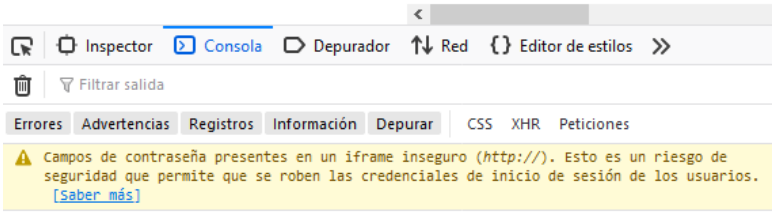


Figura 29: Información referente a riesgo de seguridad. [Elaboración propia]

Resultado: La ejecución del código en javascript evidencia la posibilidad de ejecutar sentencias para XSS y también mostró información con respecto al riesgo de seguridad latente en el sistema de notas, por lo que el impacto es alto debido a que las credenciales sensibles están altamente expuestas a ataques, mientras que la probabilidad de ocurrencia es media ya que es necesario conocimientos técnicos para ejecutar e interpretar los resultados, por los que el **riesgo se encuentra en una zona importante** de acuerdo con la tabla 7.

4.2.2.2 Testing for Stored Cross Site Scripting (OTG-INPVAL-002)

Herramienta: Navegar de internet, de preferencia Firefox.

Objetivo: Ejecutar javascript en el sistema de notas.

Para la ejecución de javascript, se utilizará el siguiente código: alert(document.cookie). con lo cual se obtiene las cookies de sesión de la aplicación como se muestra en la figura 30.

Figura 30: Ejecucion de javascript en el sistema de notas. [Elaboración propia]

Resultado: La ejecución de javascript en el sistema ha entregado las cookies de sesión del sistema de notas, las cuales serán utilizadas para explotar vulnerabilidades utilizando inicialmente ingeniería social. Debido a lo expuesto el nivel el impacto es alto, mientras que la probabilidad de ocurrencia es media ya que es imperioso el conocimiento técnico para la ejecución e interpretación de los resultados. Por tanto, el **riesgo está en una zona importante** de acuerdo con la tabla 7.

4.2.2.3 Testing for SQL Injection (OTG-INPVAL-005)

Herramientas: SQLmap, esta herramienta está instalada en una máquina virtual de Kali Linux.

Objetivo: Identificar la posibilidad de inyectar sentencias de sql en el sistema de notas.

Se utilizó SQLmap ingresando el siguiente comando en la consola de Kali Linux: sqlmap -u (IP o dominio del sistema de notas) –level (el nivel máximo es el número 5, opcional) –risk (el nivel de riesgo máximo es el número 3, opcional) como se presenta en la figura 31.

Figura 31: Ejecución SQLmap en Kali Linux. [Elaboración propia]

Resultados: Después de ejecutar la herramienta, se evidenció que el sistema de notas es vulnerable a inyección sql, esta información será importante para explotar las debilidades en etapas futuras. El impacto es alto debido a que esta técnica de hacking nos permite acceder a datos críticos y sensibles del sistema de notas tales como credenciales de usuario, entre otros. Mientras que la probabilidad de ocurrencia es media ya que es necesario conocimientos técnicos para ejecutar la herramienta, lanzar el script, e interpretar los resultados. De acuerdo con lo expuesto, el **riesgo se encuentra en una zona importante** según los criterios presentados en la tabla 7.

4.2.2.4 Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Identificar la posibilidad de obtener información del protocolo HTTP utilizado en el sistema de notas, en este caso, información con respecto al dominio. La figura 32 muestra la ejecución del comando: alert(document.domain).

Figura 32: Ejecución de javascript para obtener información del dominio. [Elaboración propia]

Resultado: La ejecución del comando en javascript muestra la información del dominio por lo que, y considerando la criticidad baja de la información obtenida, el impacto es bajo, mientras que la probabilidad de ocurrencia es media debido a que es necesario conocimientos técnicos para ejecutar el script e interpretar los resultados. Por lo tanto, el **riesgo está en una zona tolerable** de acuerdo con la tabla 7.

4.2.3 Client Side Testing

4.2.3.1 Testing for DOM based Cross Site Scripting (OTG-CLIENT-001)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Evidenciar la posibilidad de ejecutar javascript en sistemas basados en la arquitectura DOM.

Se ejecutó un comando javascript ya mencionado que muestra la cookie de sesión teniendo como resultado lo presentado en la figura 33.

Figura 33: Ejecución de comandos javascript en arquitecturas DOM. [Elaboración propia]

Resultado: Después de la ejecución del comando en javascript se evidencia la posibilidad de ejecutar dichos scripts en el sistema de notas cuya arquitectura está basada en DOM. Por tanto, el impacto es bajo debido a que estas credenciales ya fueron obtenidas con otra técnica a pesar de que se utilizó el mismo script, mientras que la probabilidad de ocurrencia es media ya que es necesario conocimientos técnicos para la ejecución del script y para la interpretación de los resultados. Por lo tanto, el **riesgo está en una zona tolerable** de acuerdo con la tabla 7.

4.2.3.2 Testing for JavaScript Execution (OTG-CLIENT-002)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Validar la posibilidad de ejecutar javascript en el sistema.

La ejecución ha sido evidenciada en la figura 46 antes presentada.

Resultado: Luego de ejecutar el comando en javascript, se garantiza la ejecución de este en el sistema de notas, por lo que, el impacto es bajo, mientras que la probabilidad de ocurrencia es media, dando como resultado, que el **riesgo se encuentra en una zona tolerable** de acuerdo con la tabla 7.

4.2.3.3 Testing for HTML Injection (OTG-CLIENT-003)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Evidenciar la posibilidad de inyectar porciones de HTML dentro del sistema de notas.

Al intentar realizar la inyección de HMTL utilizando el siguiente comando: **http://academico.itq.edu.ec:1985/aulavirtual/cabeceras.aspx?txtusuario=<img%20src='aaa'%20onerror=alert(1)>,** el servidor del sistema de notas controla el tipo de datos que se envía en la cabecera como se muestra en la imagen 34.

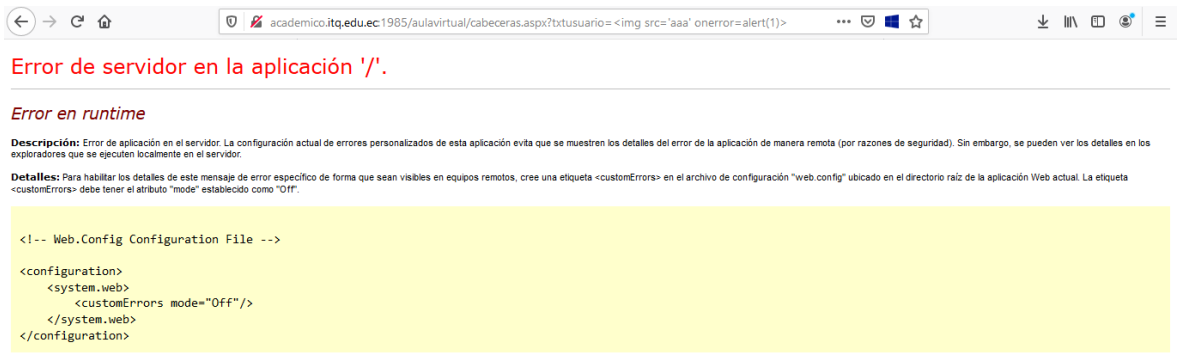


Figura 34: Ejecución de inyección HTML. [Elaboración propia]

Resultado: El servidor de sistema de notas controla este tipo de inyección HTML, por tal motivo, el impacto es bajo, mientras que la probabilidad de ocurrencia es baja, dando como resultado, que el **riesgo está en una zona tolerable** de acuerdo con la tabla 7.

4.2.3.4 Testing for Client-Side URL Redirect (OTG-CLIENT-004)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Evidenciar la posibilidad de redireccionar al usuario a sitios externos al sistema de notas introduciendo código en la URL.

Se utilizó en siguiente comando para probar la redirección en el sistema de notas: academico.itq.edu.ec:1985/aulavirtual/cabeceras.aspx?#redirect=www.itq.edu.ec.

La figura 35 muestra que el intento de redirección no fue exitoso ya que se mantiene en la misma página.

Figura 35: Redirección a sitios externos. [Elaboración propia]

Resultado: El servidor de sistema de notas controla este tipo de redireccionamientos, por tal motivo, el impacto es bajo, mientras que la probabilidad de ocurrencia es baja, dando como resultado, que el **riesgo se encuentra en una zona tolerable** de acuerdo con la tabla 7.

4.2.3.5 Testing for CSS Injection (OTG-CLIENT-005)

Herramienta: Navegador de internet, de preferencia Firefox.

Objetivo: Verificar la posibilidad de inyectar estilos CSS en el sistema de notas.

Se inserto el siguiente comando en la URL del sistema de notas validando su funcionamiento como se presenta en la figura 36: academico.itq.edu.ec:1985/#red;-o-link:'javascript: alert(1)';-o-link-source: current;

Figura 36: Inyección de CSS. [Elaboración propia]

Resultado: El servidor de sistema de notas controla las inyecciones CSS, por tal motivo, el impacto es bajo, mientras que la probabilidad de ocurrencia es baja, dando como resultado, que el **riesgo se encuentra en una zona tolerable** de acuerdo con la tabla 7.

4.2.3.6 Testing for Clickjacking (OTG-CLIENT-009)

Herramientas: Navegador de internet y un editor de texto para crear HTML de preferencia Notepad++

Objetivo: Verificar si el sistema de notas es vulnerable a Clickjacking.

Para ejecutar el análisis se realizó una plantilla en HTML en donde se especifica la URL del sistema de notas en un iframe, también se colocan los campos de texto de usuario y contraseña por sobre encima de los propios de la página de login del sistema de notas, finalmente, se crea un input del tipo botón con una función en javascript encargada de capturar los datos y mostrarlos en pantalla como se muestra en la figura 37.

```
<div style="position: relative;">
    <iframe src="http://academico.itq.edu.ec:1985/aulavirtual/" width="1000" height="500"></iframe>
</div>
<div style=" position: absolute; top: 88px; left: 899px;">
   <input type="text" id="us"/>
</div>
<br>
<div style=" position: absolute; top: 122px; left: 899px;">
   <input type="password" id="pass"/>
</div>
  <div style=" position: absolute; top: 156px; left: 931px;">
   <input type="button" value="Aceptar" onclick="capturar()"/>
</div>

<script>
function capturar(){
var us = document.getElementById('us').value;
var pass = document.getElementById('pass').value;
alert("Usuario "+us+"\nPass "+pass);
}
</script>
```

Figura 37: Diseño y programación de Clickjacking. [Elaboración propia]

La figura 38 muestra que al ejecutar el documento HTML se evidenciará que el sistema de notas es vulnerable a pruebas y ataques de clickjacking. La figura también presenta a los inputs creados bastante más grandes que los originales, esto con el fin de evidenciar que se han sobrepuesto componentes en el formulario de sesión original.

Figura 38: Ejecución de HTML evidenciando clickjacking. [Elaboración propia]

Resultado: El sistema de notas es vulnerable a clickjacking, por tanto, el impacto es alto tomando en cuenta la potencialidad que tiene esta técnica al ser ejecutada de forma adecuada. Mientras que la probabilidad de ocurrencia es media ya que es necesario conocimientos técnicos para crear HMLT y lanzar el ataque utilizando técnicas de ingeniería social. Por tanto, el **riesgo se encuentra en una zona importante** de acuerdo con la tabla 7.

4.2.4 Presentación de resultados de la etapa de escaneo.

De acuerdo con el escaneo realizado, el sistema de notas del instituto que sirve como caso de estudio, presenta los siguientes resultados:

- Information Gathering
  - Enumerate Applications on Webserver (OTG-INFO-004), la herramienta nmap presentó la topología de la red y puertos abiertos. **Zona de riesgo es moderada de acuerdo con la tabla 7.**
  - Review webpage comments and metadata for information leakage (OTG-INFO-005), la herramienta ZAP de OWASP evidenció que el sistema de notas **es vulnerable a ataques XSS** (Cross Site Scripting) o de sitios cruzados utilizando javascript **y también es vulnerable clickjacking**. **Zona de riesgo es importante de acuerdo con la tabla 7.**
  - Fingerprint Web Application Framework (OTG-INFO-008), Wappalyzer mostro información del servidor web, el framework y el Sistema operativo donde este alojado el sistema de notas. **Zona de riesgo es moderada.**
  - Fingerprint Web Application (OTG-INFO-009), La herramienta Nikto entregó como resultados información con respecto a la IP publica, IP de la red privada, el dominio, el puerto, **vulnerabilidades tales como ataques de XSS y clickjacking. Zona de riesgo es importante de acuerdo con la tabla 7.**
- Data Validation Testing:
  - Testing for Reflected Cross Site Scripting (OTG-INPVAL-001), utilizando el navegador, la consola de Firefox identificó un riesgo de seguridad en las credenciales de inicio de sesión. **Zona de riesgo es importante**
  - Testing for Stored Cross Site Scripting (OTG-INPVAL-002), al utilizar el navegador de internet se pudo comprobar que es factible manipular comandos javascript en la consola de Firefox. **Zona de riesgo es importante de acuerdo con la tabla 7.**

- Testing for SQL Injection (OTG-INPVAL-005), SQLmap evidenció que el sistema de notas **es vulnerable a inyección sql. Zona de riesgo es importante de acuerdo con la tabla 7.**
- Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016), al ejecutar el comando javascript utilizando el navegador se demostró que el sistema de notas muestra información del dominio. **Zona de riesgo es tolerable de acuerdo con la tabla 7.**

- Client Side Testing:
  - Testing for DOM based Cross Site Scripting (OTG-CLIENT-001), utilizando el navegador de internet Firefox, se evidenció la posibilidad de ejecutar comando javascript en la arquitectura DOM del sistema de notas. **Zona de riesgo es tolerable de acuerdo con la tabla 7.**
  - Testing for JavaScript Execution (OTG-CLIENT-002), utilizando la consola del navegador de internet Firefox se **validó la posibilidad de inyectar javascript** en el sistema de notas. **Zona de riesgo es tolerable de acuerdo con la tabla 7.**
  - Testing for HTML Injection (OTG-CLIENT-003), no se logró insertar HTML en el sistema de notas. **Zona de riesgo es tolerable de acuerdo con la tabla 7.**
  - Testing for Client-Side URL Redirect (OTG-CLIENT-004), no fue posible redireccionar a sitos externos desde el sistema de notas. **Zona de riesgo es tolerable de acuerdo con la tabla 7.**
  - Testing for CSS Injection (OTG-CLIENT-005), no se pudo insertar hojas de estilo CSS en el sistema de notas. **Zona de riesgo es tolerable de acuerdo con la tabla 7.**
  - Testing for Clickjacking (OTG-CLIENT-009), utilizando el navegador, criterios de diseño en HTML y programación con javascript **se demostró que el sistema de notas es vulnerable a clickjacking. Zona de riesgo es importante de acuerdo con la tabla 7.**

4.3 Explotar vulnerabilidades

En la presente etapa de hacking ético se utilizarán herramientas informáticas con la finalidad de explotar vulnerabilidades, es decir ganar y mantener el acceso, en el sistema de notas considerando la información obtenida en la etapa de escaneo y catalogada en zonas de riesgo importante.

4.3.1 Explotación de vulnerabilidades con Clickjacking

El objetico de esta técnica es lograr que el usuario de click a un elemento que el atacante ha diseñado con la finalidad de controlar los datos ingresados por dicho usuario.

Para explotar la vulnerabilidad se ha modificado el código que se presentó en las figuras 37 y 38 añadiendo style=”width : 110px” a los inputs del usuario y la clave, teniendo como resultado lo que se presenta en la figura 39 que a continuación se muestra.

```
<div style="position: relative;">
    <iframe src="http://academico.itq.edu.ec:1985/aulavirtual/" width="1050" height="500"></iframe>
</div>
<div style=" position: absolute; top: 88px; left: 899px;">
   <input type="text" id="us" style="width : 110px"/>
</div>
<br>
<div style=" position: absolute; top: 122px; left: 899px;">
   <input type="password" id="pass" style="width : 110px"/>
</div>
  <div style=" position: absolute; top: 156px; left: 931px;">
   <input type="button" value="Aceptar" onclick="capturar()" style="width : 72px"/>
</div>

<script>
function capturar(){
var us = document.getElementById('us').value;
var pass = document.getElementById('pass').value;
alert("Usuario "+us+"\nPass "+pass);
}
</script>
```

Figura 39: Nuevo diseño y programación de Clickjacking. [Elaboración propia]

Se realizó la modificación del style en los inputs del botón, usuario y contraseña con la finalidad de poner estos elementos sobre las etiquetas oficiales del sistema de notas, de tal manera la interacción visual con el usuario no se verá afectada.

A continuación, se presenta en la figura 40, como el usuario visualizará el formulario de inicio de sesión.

Figura 40: Clickjacking modificado. [Elaboración propia]

Utilizando técnicas de ingeniería social, se capturaron las credenciales del coordinador de la carrera de Análisis de Sistemas como se presenta en la figura 41, las cuales permitirán ganar acceso al sistema.

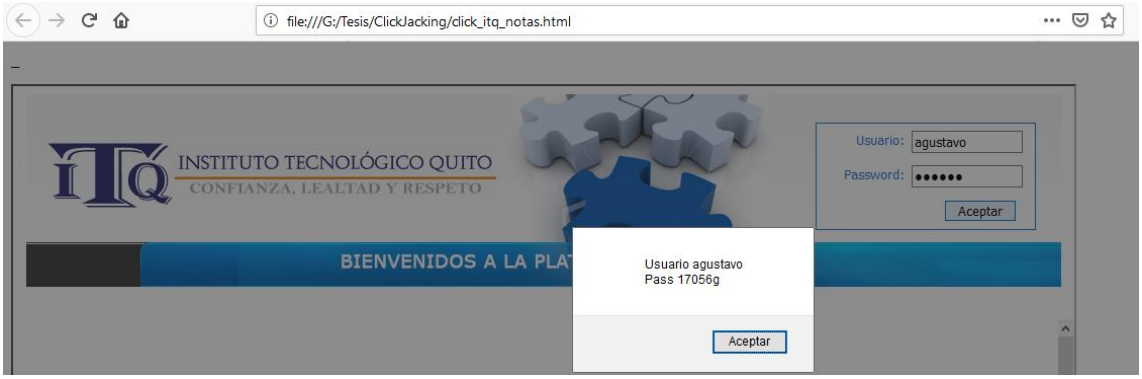


Figura 41: Captura de credenciales de usuario con perfil de administrador. [Elaboración propia]

Estas credenciales fueron probadas no solo en el sistema de notas, sino también en el sistema de secretaria académica, teniendo resultados positivos ya que es posible acceder a estos dos ambientes con dichas credenciales como se muestra en la figura 42.

Figura 42: Acceso al sistema de secretaria académica. [Elaboración propia]

Como se había explicado en la sección “3.2.1. Secretaría académica” este ambiente gestiona información sensible, tal como actualización de información de estudiantes y docentes, realización de encuestas, evaluación docente, información de coordinación de carreras, vinculación y prácticas profesionales, entre otras.

Como se presenta en la figura 43 y 44, se puede visualizar y actualizar las contraseñas de todos los colaboradores, estas credenciales permiten ganar acceso al sistema de notas, lo que deriva en una problemática grave para el instituto.

Figura 43: Visualización de credenciales sensibles. [Elaboración propia]

Figura 44: Visualización de credenciales sensibles. [Elaboración propia]

El atacante puede iniciar sesión y, entre otras acciones, ingresar o modificar calificaciones. Esta sección se encuentra en la parte inferior izquierda de la figura 45.

Figura 45: Acceso al sistema de notas con las credenciales mostradas en la figura 57. [Elaboración propia]

En el ambiente para manipular notas se presenta una serie de filtros, los cuales permitirán al usuario mostrar con precisión las calificaciones de un curso deseado, con esto, el usuario puede ingresar o modificar la información referente a notas parciales por estudiantes. A continuación, se presenta en la figura 46 el ambiente mencionado.

Figura 46: Ambiente para ingresar notas. [Elaboración propia]

4.3.1.1 Presentación de resultados obtenidos al ejecutar Clickjacking en el sistema de notas

El impacto de realizar el ataque al sistema de notas es **alto** debido a que los datos obtenidos utilizando clickjacking son sensibles, esta información pertenece a credenciales administrativas y académicas. Mientras que **probabilidad de ocurrencia es alta** debido a que el atacante que ejecuto la técnica para explotar vulnerabilidades tiene conocimientos técnicos importantes, por tanto, también está en la capacidad de interpretar y utilizar la

información obtenida. Por lo tanto, el **riesgo se encuentra en una zona importante** de acuerdo con los criterios presentados en la tabla 7.

4.3.2. Explotación de vulnerabilidades con Inyección SQL

Para realizar la inyección sql en el sistema de notas, se debe considerar la siguiente explicación: si se tiene la instrucción como se muestra en la figura 47, la consulta recibe dos parámetros, username_str y password_str, los cuales contienen un valor escrito por el usuario.

```
String sql = "SELECT * FROM tabla "
sql += "where txt_username = '+username_str+' "
sql += "and txt_password = '+password_str+' ";
```

Figura 47: Cadena sql para verificar usuario y contraseña. [Elaboración propia]

Si el usuario ingresa argumentos en las variables username_str = admin y password_str = admin1234., la sentencia sql se vería de la siguiente manera:

```
SELECT * FROM tabla
where txt_username = 'admin'
and txt_password = 'admin1234.'
```

Figura 48: Sql con parámetros para usuario y contraseña. [Elaboración propia]

El objetivo de esa instrucción es forzar la consulta a que sea verdadera debido a que la expresión ‘1’=’1’ devuelve como resultado un valor verdadero. El comando que se inyecta se presenta tiene los siguientes valores:

- username_str = [cualquier valor]
- password_str = ‘ or ‘1’=’1

El query que contiene los valores especificados en las viñetas anteriores, quedaría expresado como se presenta a continuación:

```
SELECT * FROM tabla
where txt_username = 'admin'
and txt_password = '' or '1'='1'
```

Figura 49: Inyección SQL. [Elaboración propia]

El campo username_str tiene el valor de “admin”, el campo password_str tiene un valor vacio, la inyección sql incrusta una compuerta lógica “or” y el valor ‘1’=’1’ obteniendo como respuesta una consulta es verdadera, sin importar los valores que se hayan ingresado en los parámetros de usuario y contraseña. En el sistema de notas, se utilizó la siguiente instrucción con el objetivo de inyectar sql:

- ‘ or ‘1’=’1

La figura 50 muestra la inserción de esta instrucción en el formulario de inicio de sesión.

Figura 50: Inyección sql en el formulario de inicio de sesión. [Elaboración propia]

Se logró mostrar la información que se incrusta en el campo de texto de la contraseña cambiando el tipo del input de: “password” a “text”, como se visualiza en la figura 51.

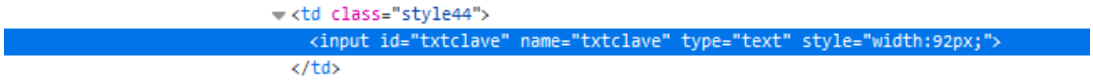


Figura 51: Cambio de tipo en el input del formulario. [Elaboración propia]

Al presionar el botón “aceptar” se evidencia en la figura 52 que se ha logrado iniciar sesión en el sistema de notas con el perfil de un usuario en específico ganado acceso al sistema de notas. Si no se modifica el criterio que tiene el sistema para mejorar la seguridad con respecto a inyección sql, se garantiza el acceso permanente utilizando el comando especificado.

Figura 52: Inicio de sesión después de ejecutar la inyección sql. [Elaboración propia]

4.3.2.1 Inyección SQL en otros portales

Se ha realizado pruebas de inyección sql en otros portales de instituciones educativas utilizando la misma instrucción que se empleó para inyectar sentencias sql en el sistema de notas, obteniendo como resultado lo evidenciado en la tabla 2.

Descripción	Resultados	Observaciones
15 instituciones educativas fueron evaluadas con la técnica de inyección sql.	Se logró acceder al sistema de notas de 3 instituciones educativas de las 15	En 1 de las 15 instituciones educativas se logró ingresar hasta el sistema de colecturía.

Tabla 2: Resultados en otras instituciones educativas. [Elaboración propia]

4.3.2.2 Presentación de resultados obtenidos al inyectar sentencias sql en el sistema de notas

Utilizando técnicas de inyección sql se pudo ganar acceso al portal de calificaciones con el perfil asignado a un docente, los datos obtenidos sirvieron para mantener ese acceso, también se expuso información confidencial del proceso académico tal como calificaciones, pasantías, vinculación, exámenes complexivos, calificaciones de periodos anteriores, etc.
Por lo tanto, el impacto es alto debido a la sensibilidad y criticidad de la información obtenida y la probabilidad de ocurrencia es alta ya que el atacante está en la capacidad de lanzar el ataque como también de interpretar resultados y utilizar esta información a su favor, lo que, al ejecutar técnicas de inyección sql en el sistema de notas posiciona al **riesgo en una zona importante** de acuerdo con los criterios presentados en la tabla 7.

5. RESULTADOS

El presente capitulo está dedicado a presentar los resultados de ejecutar técnicas de hacking ético en el sistema de notas diferenciando cada una de sus etapas: reconocimiento, escaneo, ganar y mantener acceso (explotación de vulnerabilidades) y, finalmente, cubrir huellas.
Los resultados son presentados utilizando el indicador de riesgo descrito en la tabla 7, el cual puede tener tres estados:

- Tolerable: riesgo bajo y controlable.
- Moderado: riesgo que requiere una consideración y posible mantenimiento.
- Importante: riesgo crítico que necesita atención de forma inmediata.

5.1 Resultados de la etapa de reconocimiento.

El análisis realizado al sistema de notas en la etapa de reconocimiento arrojó los resultados que se presentan a continuación en la tabla 3:

Número	Vector de ataque	Riesgo
1	Descubrimiento de motores de búsqueda y el reconocimiento de fugas de información	Tolerable
2	Análisis al servidor web para encontrar huellas digitales	Moderado
3	Búsqueda de metarchivos en el servidor web para detectar fugas de información	Moderado

Tabla 3: Resultado de análisis en la etapa de reconocimiento. [Elaboración propia]

Los dos vectores de ataque que muestran un riesgo moderado entregaron los siguientes datos:

- Información referente al dominio
- IP publica
- Información del hosting
- Información del reverse DNS

Mientras que el vector de ataque que presenta un riesgo tolerable no entregó información sensible lo que indica que el sistema de notas no es vulnerable al descubrimiento de motores de búsqueda y el reconocimiento de fugas de información.

El 67% de vectores de ataque corresponde a dos riesgos moderados, el 33% corresponde a riesgos tolerables y un 0% pertenece a riesgos importantes como se presenta en la figura 53. Esto indica que la información obtenida en la etapa de reconocimiento ubica al sistema de notas en una zona de riesgo moderada, de acuerdo con la tabla 7, permitiendo utilizar estos datos como insumo para la siguiente etapa del hacking ético.

Figura 53: Análisis de porcentual en la etapa de reconocimiento. [Elaboración propia]

5.2 Resultados de la etapa de escaneo.

El análisis realizado al sistema de notas en la etapa de escaneo entregó los resultados presentados en la tabla 4:

Número	Ataque	Riesgo
1	Enumerar aplicaciones en el servidor web	Moderada
2	Revisar los comentarios y metadatos de la página web para detectar fugas de información	Importante
3	Marco de aplicación web de huellas digitales	Moderada
4	Aplicación web de huellas digitales	Importante
5	Pruebas de XSS reflejados	Importante
6	Pruebas de XSS almacenados	Importante
7	Prueba para inyección SQL	Importante
8	Prueba de fugas HTML	Tolerable
9	Prueba de secuencias de comandos XSS basadas en DOM	Tolerable
10	Prueba de ejecución de JavaScript	Tolerable
11	Prueba de inyección HTML	Tolerable
12	Prueba de redireccionamiento de URL del lado del cliente	Tolerable
13	Prueba de inyección CSS	Tolerable
14	Prueba de clickjacking	Importante

Tabla 4: Resultado de análisis en la etapa de escaneo. [Elaboración propia]

Seis de los catorce vectores de ataque han sido considerados como tolerables, ya que no se logró inyectar HTML ni CSS, así como tampoco resulto efectivo el redireccionamiento a sitios externos del sistema de notas, sin embargo, se obtuvieron datos tales como:

- Información del dominio
- Efectividad al ejecutar javascript.

Dos de los catorce vectores de ataque fueron catalogados como riesgos moderados, de los cuales se obtuvo los siguientes datos:

- Información acerca de la topología de la red y puertos abiertos.
- Información referente al servidor web, el framework y el sistema operativo donde este alojado el sistema de notas.

Finalmente, seis de los catorce vectores de ataque fueron considerados riesgos importantes, es decir, riesgos altamente críticos de acuerdo con la tabla 7. Los datos resultantes se listan a continuación:

- Información que evidencia que el sistema de notas es vulnerable a ataques XSS (Cross Site Scripting) o de sitios cruzados utilizando javascript y también es vulnerable clickjacking.
- Información con respecto a la IP publica, IP de la red privada, el dominio, el puerto, vulnerabilidades tales como ataques de XSS y clickjacking ya antes mencionada, sin embargo, estos resultados fueron obtenidos con otra herramienta.
- Información que logró identificar un riesgo de seguridad en las credenciales de inicio de sesión.
- Información que confirmo la posibilidad de manipular comandos javascript en la consola de Firefox, navegador utilizado para el ataque.
- Información que evidenció que el sistema de notas es vulnerable a inyección sql.
- Información que demostró que el sistema de notas es vulnerable a clickjacking.

El 14% de vectores de ataque corresponde a dos riesgos moderados, el 43% en gris representa a riesgos tolerables y, finalmente un 43% pertenece a riesgos importantes, como se puede visualizar en la figura 54, lo que indica evidencia la posibilidad de explotar vulnerabilidades utilizando técnicas de clickjacking y sql inyección en el sistema de notas. Estas técnicas fueron ejecutadas en la siguiente etapa de hacking ético: mantener y ganar acceso (explotación).

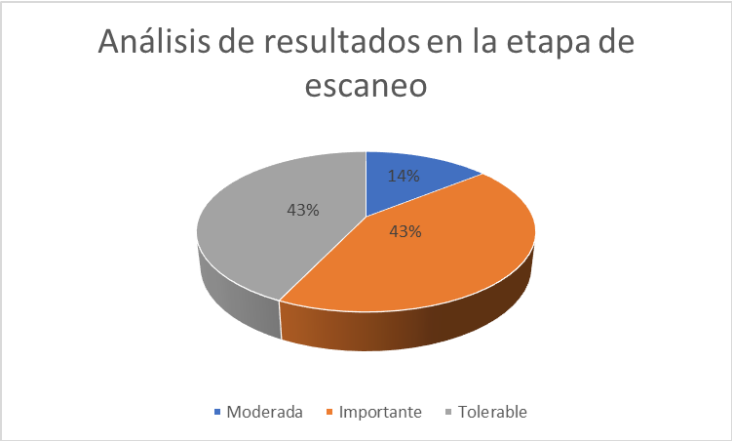


Figura 54: Análisis de porcentual en la etapa de escaneo. [Elaboración propia]

5.3 Resultados de la etapa de explotación

El análisis ejecutado al sistema de notas en la etapa de explotación aplicando técnicas de inyección SQL y clickjacking, se presenta en la tabla 5:

Número	Ataque	Resultado / Riesgo
1	Clickjacking	Importante
2	Inyección SQL	Importante

Tabla 5: Resultado de análisis en la etapa de explotación. [Elaboración propia]

El riesgo resultó ser importante al ejecutar los dos vectores de ataque aplicados en esta etapa de hacking ético debido a que permitió realizar las siguientes acciones:

- Ejecutando clickjacking se obtuvo información perteneciente a credenciales administrativas y académicas con las cuales se tuvo acceso total a datos sensibles como pares de inicio de sesión de todo el personal docente, entre otras.
- Ejecutando inyección SQL se logró acceder al portal de calificaciones con el perfil de un docente en específico (siempre con el mismo docente), con lo cual se pudo manipular información académica tal como calificaciones, pasantías, vinculación, exámenes complexivos, calificaciones de periodos anteriores, etc.

El 100% de los vectores de ataque resultaron ser riesgo importante como se puede visualizar en la figura 55, lo cual indica que la explotacion de vulnerabilidades al sistema de notas utilizando tecnicas de inyeccion SQL y clickjacking fueron exitosas en su totalidad y también evidencia las debilidades presentes en la aplicación.

Figura 55: Análisis de porcentual en la etapa de explotación. [Elaboración propia]

El sistema de notas fue vulnerado en su totalidad al ejecutar las técnicas mencionadas, por lo que es importante contar con una guía especifica para gestionar estas vulnerabilidades informáticas del sistema de notas.

6 GUÍA PARA LA EVALUACIÓN DE VULNERABILIDADES

En el presente capitulo se mostrará una guía específica alineada a la metodología OWASP descrita en el presente trabajo de titulacion que sirva como estrategia para la evaluación de vulnerabilidades del sistema de notas de instituciones educativas utilizando técnicas de hacking ético, además presenta lineamientos para mitigar y contener ataques, así como actividades posteriores al incidente.

6.1 Preparación previa al ataque informático

La preparación previa de todos los actores y entidades que forman parte del sistema de notas es fundamental para la evaluación de vulnerabilidades. Los elementos inherentes al sistema de notas que deben ser preparados se presentan a continuación:

Preparación del equipo o equipos informáticos del atacante

Para la evaluación de los sistemas académicos se recomienda utilizar una laptop con las siguientes características:

- Procesador Intel core i7 8va generación
- RAM de 16GB y Disco duro de 1TB
- Sistema operativo de 64 bits, procesador x64 Windows 10

El software recomendado se presenta a continuación:

- Firefox browser 76.0.1 de 64 bits donde utilizó la consola para lanzar comando javascript
- Wappalyzer, es un plugin para el navegador de internet para obtener información del sistema de notas
- Netcarft, herramienta en línea que presenta información referente al servidor web del sistema de notas.
- Domain tolos, herramienta en línea que proporciona información referente al -servidor web del sistema de notas.
- Virtual Box 6.0.
- Máquina virtual de Kali Linux 2018.4 x64 que proporciona un conjunto de herramientas orientadas al hacking ético.
- Nmap GUI ejecutado en la máquina virtual de Kali Linux obteniendo información de los puertos, abiertos o cerrados, del servidor de notas
- Nikto a nivel de consola ejecutado en la máquina virtual de Kali Linux que proporciono información referente al servidor web, posibilidad de ejecutar

clickjacking y sql injection y, finalmente mostro la ip de la red local donde se encuentra el servidor web del sistema de notas.

- SQLmap a nivel de consola ejecutado en la amquina virtual de Kali Lunix que indico la posibilidad de ejecutar inyección sql en el sistema de notas.
- Sitedigger instalado en Windows 10 que entrego información de datos y metadatos del sistema de notas
- OWASP ZAP instalado en Windows 10 que entrego información categorizada de las debilidades y las técnicas para ejecutar esas debilidades en el sistema de notas
- Notepad ++ instalado en Windows 10 que sirvió para crear el ambiente para clickjacking.

El equipo informático y las herramientas alojadas en el mismo permitieron evaluar con éxito las vulnerabilidades presentes en el sistema de notas del caso de estudio, por tal motivo se recomienda instalar las aplicaciones listadas en una laptop de similares características para preparar el ataque.

Identificación de activos informáticos

El siguiente criterio de preparación para el ataque informático consiste en determinar los activos tecnológicos con los que cuenta la institución educativa con la finalidad de ejecutar el análisis de vulnerabilidades en sus sistemas académicos. En base al análisis realizado en el instituto, así como la revisión de otros sistemas de notas, los activos comunes son los siguientes:

- Sistema para secretaria académica.
- Sistema para académico para docentes y estudiantes.
- Sistema de biblioteca y Repositorio documental.
- Correo institucional.

El análisis al sistema de notas del instituto evidenció que el activo tecnológico que administra las notas es el más crítico debido a que transacciona información sensible y está directamente alineado con el proceso central académico.

El dueño del sistema o del proceso es el que tiene el criterio para determinar cual de los activos tecnológicos es el más importante para cada institución educativa, así que dependiendo de esta decisión se ejecuta el análisis de vulnerabilidades utilizando técnicas de hacking ético.

Análisis de impacto, probabilidades y medición del riesgo

El ultimo criterio de preparación para el ataque informático es la determinación del impacto, las probabilidades de ocurrencia lo cual permite medir el riesgo. Estos criterios son entregados por el dueño del proceso o sistema a ser evaluado.

Se recomienda definir los criterios de seguridad CIA y los insumos que aportan para la creación de la matriz de riesgo, los cuales son: impacto y probabilidad de ocurrencia. Se listan a continuación las consideraciones sugeridas con respecto al **CIA**:

- La **confidencialidad** debe ser considerada como alta,
- La **integridad** debe ser considerada alta, y
- La **disponibilidad** debe ser considerada como media.

El **impacto y la probabilidad** son los insumos para elaborar el cuadro que mide el riesgo en el sistema de notas como se lista a continuación:

- **Impacto bajo** el cual no produce daño en el sistema de notas.
- **Impacto medio** el mismo que provoca daños considerables en el sistema de notas.
- **Impacto alto** produce daños críticos y en ciertos casos irreversibles.
- **Probabilidad baja** representa la casi inexistente ejecución de alguna técnica de hacking ético.
- **Probabilidad media** muestra la capacidad general o esperada de que ocurra un ataque informático.
- **Probabilidad alta**, finalmente, indica la facultad constante y elevada de que un atacante ejecute técnicas de hacking ético en el sistema de notas.

Estas consideraciones derivan en una matriz de riesgo como la que se ha creado y presentado en la tabla 7.

6.2 Análisis de vulnerabilidades

Con el objetivo de realizar el análisis de vulnerabilidades se recomienda utilizar herramientas y vectores de ataque por etapa de hacking ético que a continuación se listan en la tabla 6:

Número	Etapa de hacking ético	Ataque	Herramienta
1	Reconocimiento	Descubrimiento de motores de búsqueda y el reconocimiento de fugas de información	Sitedigger
2	Reconocimiento	Análisis al servidor web para encontrar huellas digitales	Netcraft

3	Reconocimiento	Búsqueda de metarchivos en el servidor web para detectar fugas de información	Domain Tools
4	Escaneo	Enumerar aplicaciones en el servidor web	Nmap
5	Escaneo	Revisar los comentarios y metadatos de la página web para detectar fugas de información	OWASP ZAP
6	Escaneo	Marco de aplicación web de huellas digitales	Wappalyzer
7	Escaneo	Aplicación web de huellas digitales	Nikto
8	Escaneo	Pruebas de XSS reflejados	Consola del navegador
9	Escaneo	Pruebas de XSS almacenados	Consola del navegador
10	Escaneo	Prueba para inyección SQL	SQL map
11	Escaneo	Prueba de fugas HTML	Consola del navegador
12	Escaneo	Prueba de secuencias de comandos XSS basadas en DOM	Consola del navegador
13	Escaneo	Prueba de ejecución de JavaScript	Consola del navegador
14	Escaneo	Prueba de inyección HTML	Consola del navegador
15	Escaneo	Prueba de redireccionamiento de URL del lado del cliente	Consola del navegador
16	Escaneo	Prueba de inyección CSS	Consola del navegador
17	Escaneo	Prueba de clickjacking	Consola del navegador y Notepad ++
18	Explotación	Clickjacking	Notepad ++, navegador y personal del instituto con credenciales sensibles.

19	Explotación	Inyección SQL	Navegador y script sql predefinido.

Tabla 6: Herramientas y vectores de ataque por etapa de hacking ético. [Elaboración propia].

Para la etapa de hacking ético en la que se debe cubrir huellas se recomienda no utilizar herramientas para realizar dicha acción, debido a que unos de los objetivos del hacking ético es evidenciar las vulnerabilidades.

Por cada etapa se obtendrá información que servirá de insumo para la siguiente etapa, el objetivo es presentar e interpretar los resultados después de analizar las vulnerabilidades. Se recomienda revisar el capítulo 5 denominado *"Resultados"* del presente trabajo de titulación.

6.3 Control de ataques informáticos y actividades de mantenimiento posteriores al incidente

El objetivo de este capítulo es ayudar a las instituciones educativas en el correcto desarrollo y mantenimiento de aplicaciones académicas seguras. Con la finalidad de cumplir el objetivo planteado, el capítulo se ha dividido en tres secciones descritas a continuación:

- Control de vulnerabilidades
- Determinación de criterios
- Verificación y validación

6.3.1. Control de vulnerabilidades

El análisis de vulnerabilidades en el sistema de notas realizado en capítulos anteriores ha evidenciado que la falta de control posibilita ataques informáticos como los ejecutados para el caso de estudio planteado, los cuales fueron: Clickjacking y SQL injection. Por tanto, es imperioso considerar un estándar que garantice el control eficiente de vulnerabilidades.

OWASP ofrece un listado de vulnerabilidades que con frecuencia se presentan en aplicaciones web, así como también presenta una serie de recomendaciones con respecto al control y prevención. A continuación, se explica la manera en la que se controlan y previenen estas vulnerabilidades y como estas son adaptadas para el sistema de notas [22]:

Inyección - A1:2017

Cualquier arquitectura informática que transaccione datos es vulnerable a inyección ya sea de parámetros, de servicios o variables de entorno y el sistema de notas no es la excepción. Las vulnerabilidades de inyección se encuentran con más frecuencia en consultas SQL y

servicios web, las cuales pueden ser identificadas con escáneres o fuzzers. Para controlar y prevenir inyecciones se requiere un tratamiento especial de los datos en el cual se procede a transaccionarlos de forma separada de los comandos, sentencias y consultas, para lo cual se recomienda:

- Utilizar una API segura que prescinda de la utilización de un intérprete en su totalidad y entregue un ambiente programático parametrizado.
- Realizar validaciones de ingreso de datos directamente en los servicios web de la aplicación o directamente en el servidor.
- Evitar transaccionar caracteres especiales con una semántica estándar.
- Utilizar LIMIT en las consultas SQL para evitar la fuga en masa de datos en el caso de inyección SQL.

Es necesario validar que el sistema de notas cumpla con estas recomendaciones para minimizar los ataques por inyección.

Pérdida de Autenticación - A2:2017

De acuerdo con lo evidenciado en el análisis de vulnerabilidades al sistema de notas, el par de credenciales, al ser de tipo numérico, posibilita ataques por diccionario o fuerza bruta. Existen herramientas que permiten realizar este tipo de ataques haciendo uso de bases de datos tales como tablas arcoíris, que contienen millones de pares de credenciales de usuarios y combinaciones para cuentas administrativas.

El atacante al ingresar con una cuenta administrativa entenderá el funcionamiento del sistema de notas, su información y procesos automatizados. Por ello, a continuación, se listan algunas formas de prevenir y controlar estas vulnerabilidades:

- Implementar en los ambientes de registro o inicio de sesión la autenticación multi-factor, con el objetivo de evitar ataques programados o automatizados.
- Evitar utilizar contraseñas por defecto, como números de cedula o fechas de nacimiento, para el acceso al sistema de notas.
- Crear contraseñas de al menos ocho caracteres, mezclar letras mayúsculas y minúsculas, mezclar también números y letras, finalmente, incluir caracteres no alfanuméricos.
- Alinear el criterio de dificultad y rotación de credenciales a las recomendaciones de la sección 5.1.1 de la guía NIST 800-63 B [31].
- Registrar cada intento fallido en el inicio de sesión de los sistemas académicos con la finalidad de socializar con el administrador del sitio web y este a su vez proceda con acciones que garanticen la confidencialidad e integridad del sistema de notas.

- Utilizar un generador aleatorio de sesiones de usuario identificados por ID.

Exposición de Datos Sensibles - A3:2017

La transacción y almacenamiento de datos en el sistema de notas no expone información crítica, sin embargo, se debe considerar que un atacante puede acceder a información confidencial utilizando técnicas como: Man in the Middle; este ataque es manual, sin embargo, el impacto es significativo; debido a que las credenciales y datos sensibles no son cifrados en los sistemas académicos. A continuación, se presentan las siguientes recomendaciones:

- Clasificar los datos que se procesan, se almacenan y se transmiten en el sistema académico.
- Evitar almacenar datos sensibles.
- Cifrar los datos al ser almacenados y transaccionados.
- Considerar algoritmos y protocolos estándares para el cifrado de los datos.
- Deshabilitar el almacenamiento de los datos sensibles en caché.
- Encriptar contraseñas utilizando funciones hashing tales como SHA1 y, finalmente, implementar criterios SALT a la clave y guardarla en tablas distintas a las de usuario. Se debe que considerar que, para implementar SALT a las contraseñas se debe definir alguna frase o cadena de carácter aleatorio individual por usuario, de tal manera que el resultado de aplicar la función HASH sobre esta fusión de SALT + clave sea único a pesar de que dos o más claves coincidan dentro del conjunto de credenciales en la tabla de usuario.
- Almacenar los datos sensibles fuera de la carpeta raíz del sistema de notas.

Entidades Externas XML (XXE) - A4:2017

El sistema de notas no presenta esta vulnerabilidad, sin embargo, es importante considerar que los atacantes pueden explotar procesadores de XML cargando contenido hostil en un documento XML. Algunos procesadores antiguos de XML permiten la especificación de un agente externo. Estos defectos son explotados con la finalidad de extraer datos, ejecutar una instrucción remota desde un servidor, escanear y estudiar sistemas internos, ejecutar ataques de DDoS o denegación de servicios, etc. Se lista a continuación la manera en la que se previene y controla esta vulnerabilidad originada desde la etapa de desarrollo o en etapas de mantenimiento:

- Utilizar formatos arquitectónicamente más complejos como JSON.
- Actualizar de forma recurrente los procesadores XML.

- Deshabilitar las entidades externas de XML
- Verificar que la carga de archivos XML valide este proceso.

Si se realiza versionamiento del sistema de notas, considerar los criterios que se han venido implementando con el objetivo de no permitir que se explote esta vulnerabilidad.

Pérdida de Control de Acceso - A5:2017

Las vulnerabilidades referentes a la pérdida de control de acceso son bastante comunes debido a la falta de pruebas funcionales por parte del personal de calidad o por parte de los mismos desarrolladores. El atacante realiza la explotación como anónimo, sin embargo, actúa como usuario que accede, actualiza, crea o elimina registros del sistema académico. El sistema de notas maneja perfiles y roles para el acceso sin embargo es preciso seguir las siguientes recomendaciones:

- Implementar el control de acceso una sola vez y reutilizarlo en todo el sistema.
- Determinar que roles de usuario pueden crear, leer, actualizar y borrar registros en el sistema informático.
- Deshabilitar el listado de directorios en el servidor.
- Garantizar que los respaldos del sistema académico no sean almacenados en carpetas públicas, y mucho menos en repositorios alojados dentro del mismo servidor.
- Incluir en el libreto de pruebas unitarias y de integración, pruebas de control de acceso.
- Registrar errores de control de acceso, con la finalidad de notificar estos incidentes al administrador.

Configuración de Seguridad Incorrecta - A6:2017

Con mucha frecuencia, los atacantes explotan vulnerabilidades que incluyen la manipulación de cuentas por defecto, páginas no utilizadas, archivos y directorios sin protección. Las configuraciones incorrectas se presentan en cualquier stack tecnológico que forman parte del sistema de notas, sean estos, los servicios de red, plataformas, servidores, frameworks, maquinar virtuales, computación en la nube, contenedores, entre otros. El sistema de notas fue diseñado utilizando .NET como framework, lo que minimiza la posibilidad de recibir ataques, sin embargo, con la finalidad de evitar la explotación de esta vulnerabilidad se listan las siguientes recomendaciones:

- Configurar los entornos de producción y almacenamiento de datos, de forma similar, pero implementando distintitas credenciales.

- Desinstalar o eliminar plugins, frameworks o cualquier otro software que sean innecesarios o no están siendo utilizados.

Cross Site Scripting (XSS) - A7:2017

El sistema de notas es vulnerable a este tipo de ataques como se evidenció en capítulos anteriores. Es importante considerar que se puede encontrar herramientas que detectan y explotan vulnerabilidades XSS sin considerar la arquitectura del sistema académico. El objetivo principal de XSS es robar credenciales y datos sensibles, secuestrar sesiones y cookies de sesión, ejecución de comandos maliciosos directamente en la aplicación, entre otros. Se recomienda seguir las siguientes indicaciones de control y prevención de vulnerabilidades en el sistema de notas:

- Utilizar frameworks, dependiendo del lenguaje de programación, que controlan el contenido XSS de manera automática.
- Implementar codificación sensitiva al contexto en elementos donde el usuario pueda ingresar datos con la finalidad de prevenir XSS.

Deserialización Insegura - A8:2017

El sistema de notas utiliza la serialización de los datos con la finalidad de que la transacción sea más ligera, sin embargo, al no cifrar dichas transacciones, el ambiente se torna inseguro. A continuación, se presentan una lista de recomendaciones para controlar y prevenir esta vulnerabilidad:

- Implementar firmas digitales en elementos considerados como sensibles tales como credenciales administrativas.
- Registrar las excepciones resultantes de errores al intentar desrealizar objetos recibidos con alteraciones.
- Monitorear y registrar actividades en la red distintas a las autorizadas para el proceso de deserialización de los datos.
- Identificar eventos en los cuales una entidad o usuario utiliza la deserialización con frecuencia.

Uso de Componentes con Vulnerabilidades Conocidas - A9:2017

Los atacantes hacen uso de exploits conocidos para lograr explotar debilidades. El sistema de notas no fue vulnerable a este tipo de ataques sin embargo es imperioso seguir las recomendaciones que a continuación se listan:

- Cambiar de ubicación o eliminar archivos, documentación, directorios, componentes y funcionalidades innecesarias y no utilizadas por el sistema de notas.
- Obtener e implementar componentes únicamente de sitios oficiales con el fin de reducir la probabilidad de utilizar versiones modificadas maliciosamente.
- Prescindir de librerías que no versionan sus componentes.

Registro y Monitoreo Insuficientes - A10:2017

Esta debilidad es la base donde se fundamentan la mayor parte de vulnerabilidades debido a que los atacantes dependen de la falta de monitoreo y respuesta oportuna para lograr cumplir sus objetivos sin ser detectados. El sistema de notas no presenta ninguna entidad que garantice el monitoreo y registro de actividades sospechosas por lo que se procede a listar las siguientes recomendaciones:

- Garantizar que todos los errores de inicio de sesión, control de acceso y validación de entrada de datos se registren para evidenciar comportamientos sospechosos.
- Se debe asegurar que la transacción de información sensible tenga controles de integridad con el objetivo principal de prevenir posibles alteraciones y detectar a tiempo posibles modificaciones.
- Establecer una socialización oportuna acerca de los errores registrados con la finalidad de tener la posibilidad de responder adecuadamente ante estos incidentes.

Recomendaciones generales con respecto al control de vulnerabilidades

- El control y gestión de las vulnerabilidades presentadas en este capítulo son actualizadas de manera continua por OWASP, por lo que se recomienda revisar el portal oficial con frecuencia para acceder a las recomendaciones planteadas por la organización.
- Es importante capacitar al personal acerca del uso y manejo de sus credenciales con el objetivo de evitar ataques que involucren ingeniería social.
- Responsabilizar a cada usuario de manejo y gestión de sus credenciales.
- Capacitar a los usuarios respecto a los posibles ataques de ingeniería social que pueden sufrir cada uno en los distintos procesos y canales tecnológicos o no tecnológicos de los institutos educativos.

6.3.2. Generalización de criterios para la identificación del impacto

La gestión, control y determinación de vulnerabilidades, forman parte de una categorización genérica que debe ser aplicado en el análisis de vulnerabilidades en el sistema de notas,

dicha determinación la realiza el dueño del sistema o del negocio, para el caso de estudio lo determinó el Coordinador de la carrera de Análisis de Sistemas como se presenta a continuación:

6.3.2.1. Análisis de impacto, probabilidad y medición del riesgo

De acuerdo con el alcance del proyecto, se ha delimitado el análisis de vulnerabilidades al sistema de notas de instituciones educativas, tomando como caso de estudio en Instituto Tecnológico Quito.

Según la seguridad informática, las aplicaciones deberían balancear los criterios del CIA (confidencialidad, integridad y disponibilidad), para garantizar la integridad de la información transaccionada por aplicaciones tecnológicas.

Teniendo en cuenta que el sistema de notas es un activo importante en el proceso académico de una institución educativa, en conversaciones verbales con el coordinador de la carrera de análisis de sistemas se ha determinado la puntuación que debería tener el aplicativo tecnológico con respecto al CIA, donde:

- El criterio de confidencialidad debe ser alto ya que transacciona datos sensibles como calificaciones, encuestas, datos personales y claves. Estas claves pueden ser mostradas sin ningún tipo de encriptación, es decir, en su texto original.
- La característica de integridad debe ser alta debido a que el sistema académico que gestiona las notas debe garantizar el principio académico en el cual se imposibilita la acción de modificar las calificaciones sin un proceso que lo supervise, y
- La disponibilidad debería ser media, considerando que la visibilidad expuesta de calificaciones finalmente se la presenta al culminar el proceso académico, sin embargo, se debe garantizar el sistema de notas este totalmente disponible al momento de ingresar información.

Coordinación de carrera socializó los criterios que se deben considerar en las etapas de hacking ético para determinar el nivel de riesgo del sistema de notas, los cuales se enlistan a continuación:

- Se considera **impacto bajo** cuando la información expuesta puede ser obtenida con facilidad o dichos resultados no pueden derivar en explotación de vulnerabilidades considerables.
- Se considera **impacto medio** cuando la información obtenida, como correos, IPs o teléfonos, pueda derivar en indicadores que el atacante pueda utilizar para explotar posibles vulnerabilidades utilizando técnicas como hacking ético.

- Se considera **impacto alto** cuando esta información obtenida o expuesta deriva en explotar vulnerabilidades que entreguen al atacante datos sensibles tales como claves, nombres de usuario, calificaciones, perfiles y roles de usuario.
- Se considera **probabilidad baja** cuando el nivel de ocurrencia del reconocimiento, escaneo o explotación es muy bajo debido a criterios de seguridad incorporados en el sistema de notas o el uso de herramientas obsoletas para obtener información.
- Se considera **probabilidad media** cuando el nivel de ocurrencia de alguna de las primeras etapas del hacking ético (reconocimiento, escaneo o explotación), es constante ya que el atacante puede acceder a esta información utilizando herramientas medianamente difundidas y que entregan justamente los datos buscados o debido a que el sistema de notas no ha protegido ciertos ambientes de la aplicación.
- Finalmente, se considera **probabilidad alta** cuando la ocurrencia es sumamente alta debido a que el atacante utiliza técnicas y herramientas altamente difundidas, también cuando el atacante tiene el perfil técnico sumamente desarrollado lo cual permite manipular herramientas o crear exploits y, posteriormente, interpretar esta información para su beneficio. En este ambiente, se asume que el sistema de notas no ha protegido su arquitectura tecnológica lo cual no garantiza la seguridad y también el personal que manipula la aplicación con perfiles y roles críticos no conocen las políticas recomendadas para no exponer sus credenciales.

De acuerdo con lo expuesto, se estructura la siguiente matriz que permitirá mediar el impacto y la probabilidad de ocurrencia en las etapas de hacking ético con la finalidad de medir el riesgo.

Probabilidad	**Alta**	Zona de riesgo moderada	Zona de riesgo importante	Zona de riesgo importante
	Media	Zona de riesgo tolerable	Zona de riesgo moderada	Zona de riesgo importante
	Baja	Zona de riesgo tolerable	Zona de riesgo tolerable	Zona de riesgo moderada
		Bajo	**Medio**	**Alto**
		Impacto		

Tabla 7: Matriz de probabilidad vs impacto [Elaboración propia]

Esta matriz tiene como resultado el nivel de riesgo al que está expuesto el sistema de cotas, el cual está dividido en: riesgo toreable, riesgo moderado y riesgo importante, siendo este el más crítico.

La tabla 8 representa un resumen del impacto por riesgo informático recomendado por OWASP contrastado con los resultados del análisis y la evaluación de vulnerabilidades en el sistema de notas en la cual se plantea al menos una recomendación aplicable a sistemas que transacciones información académica.

Riesgo	**Impacto: OWASP**	**Impacto: Sistema de notas**	**Recomendación** **Cruce de información entre lo obtenido del análisis y la recomendación de OWASP**
Inyección - A1:2017	Alto	Alto	OWASP y el análisis realizado en este trabajo coinciden en clasificar este riesgo como alto, por lo que se recomienda considerar evaluar esta característica al momento de crear una matriz de riesgo similar a la expuesta en la tabla 7.
Pérdida de Autenticación - A2:2017	Alto	Alto	OWASP y el análisis realizado en este trabajo también coinciden en clasificar este riesgo como alto. Debido a esto, se recomienda considerar esta característica al armar la matriz de riesgo.
Exposición de Datos Sensibles - A3:2017	Medio	Medio	OWASP y el análisis han coincidido en clasificar el impacto como medio para este riesgo, se recomienda tomar en cuenta este criterio de seguridad para crear con mayor criterio la matriz de riesgo similar a la de la tabla 7 de este trabajo.
Entidades Externas XML (XXE) - A4:2017	Alto	Bajo	OWASP ha considerado clasificar este riesgo como alto, mientras que el análisis al sistema de notas del caso de estudio ha evidenciado que el riesgo es bajo debido a que el framework con el que se construyó

			el software mitiga el impacto, por tanto, se recomienda prescindir de esta característica de seguridad al crear la matriz de riesgo.
Pérdida de Control de Acceso - A5:2017	Medio	Medio	OWASP y el análisis realizado en el presente trabajo consideran que al riesgo como medio, por lo que se recomienda considerarlo incorporar en la construcción de la matriz de riesgo.
Configuración de Seguridad Incorrecta - A6:2017	Medio	Bajo	Los criterios de seguridad no coinciden entre OWASP y el análisis realizado debido a que el framework que soporta el sistema de notas controla este criterio de seguridad, por lo que es recomendable descartarlo en la creación de la matriz de riesgo.
Cross-Site Scripting (XSS) - A7:2017	Medio	Alto	El criterio de seguridad evidenciado por el análisis es alto y difiere del que presenta OWASP. Se recomienda considerarlo en la construcción de la matriz de riesgo.
Deserialización Insegura - A8:2017	Medio	Bajo	Los criterios de seguridad no coinciden entre lo evidenciado por el análisis y lo presentado por OWASP y considerando que para el análisis el impacto es bajo, se recomienda no incluir esta característica en la matriz de riesgo.
Uso de Componentes con Vulnerabilidades Conocidas - A9:2017	Alto	Medio	El análisis evidencio que el riesgo y difiere de lo presentado por OWASP, sin embargo, se recomienda incorporar esta característica de seguridad en la construcción de la matriz de riesgo.
Registro y Monitoreo Insuficientes - A10:2017	Medio	Bajo	Los criterios de impacto son distintos entre los evidenciados por el análisis y presentados por OWASP. Considerando también que el impacto resultante del

			análisis al sistema de notas es bajo, se recomienda no incluir en la matriz de riesgo.

Tabla 8: Resumen impacto por riesgo de OWASP y análisis del sistema de notas. [Elaboracion propia]

Recomendaciones generales respecto a la determinación de criterios.

- Dependiendo del proceso automatizado que este gestionando el sistema académico, se debe considerar los criterios de seguridad presentados en la tabla 7 y ajustarlos a cada realidad de la academia.
- Elaborar una matriz de riesgo analizando las recomendaciones presentadas en la tabla 7, y adaptarlas a cada realidad.
- Estos criterios de vulnerabilidad son actualizados por OWASP de forma periódica.

6.3.3. Verificación y validación

Esta sección presenta sugerencias para el mantenimiento del sistema de notas a través de indicadores de verificación y validación propuestos por OWASP contrastando con los resultados obtenidos al explotar las vulnerabilidades propias del sistema. De acuerdo con la explicación de cada nivel presentada en la sección "*2.8. Mantenimiento se aplicaciones seguras según OWASP*", se presenta en la tabla 9 el criterio de aplicabilidad cruzando la información con lo evidenciado en el análisis de vulnerabilidades realizado al sistema de notas.

	Aplicabilidad genérica según OWASP	**Aplicabilidad de acuerdo con el análisis realizado en este presente trabajo**	**Aseguramiento y verificación**	
Nivel 1	Todas las aplicaciones	Aplica para el sistema de notas	Pruebas de penetración	Pruebas de seguridad de aplicaciones dinámicas
Nivel 2	Todas las aplicaciones	Aplica para el sistema de notas	Revisiones híbridas	Pruebas de seguridad de aplicaciones dinámicas
Nivel 3	Alta seguridad	No aplica para el sistema de notas	Revisiones híbridas	Pruebas de seguridad de

				aplicaciones estáticas

Tabla 9: Niveles del estándar ASVS. [Elaboración propia]

Cada nivel del estándar contiene una lista de requisitos de seguridad, y cada requisito, a su vez, asigna características y capacidades especificas orientados a seguridad que los desarrolladores de software deben incorporar en sus proyectos al implementar un sistema informático como también para ejecutar el mantenimiento de este [34].

Por cada requisito orientado a la validación y verificación de aplicaciones seguras presentado por OWASP se recomiendan a continuación acciones aplicables al mantenimiento del sistema de notas:

- V1 arquitectura, diseño y modelado de amenazas: El personal encargado de la seguridad del sistema de notas, deberá actualizarse con frecuencia respecto a las amenazas latentes en sistemas académico, tal como inyección SQL o clickjacking. Se recomienda revisar el TOP 10 de amenazas presentes en aplicaciones web que OWASP actualiza y publica con frecuencia en sus portales oficiales.
- V2 requisitos de verificación de autenticación: Este requisito es propio de la etapa de desarrollo de aplicaciones informáticas y debido a que el sistema de notas es un producto ya en producción no aplica recomendación alguna.
- V3 requisitos de verificación de gestión de sesiones: Este requisito pertenece a la etapa de desarrollo, por lo que no aplica recomendación alguna.
- V4 requisitos de verificación del control de acceso: Este requisito es propio de la etapa de desarrollo, sin embargo, se debe asegurar que las credenciales del sistema de notas están protegidas de ataques informáticos para lo cual se recomienda configurar esta verificación en el framework que se esté utilizando para el desarrollo, para el sistema de notas será .NET.
- V5 requisitos de verificación para manejo de entrada de datos maliciosos: Se recomienda validar que el sistema de notas controle los datos de entrada para evitar ataques de XSS o inyección SQL. Esta validación debe impedir inyectar javascript en casilleros de texto y también, a nivel de controlador, filtrar cadenas que contengan posibles caracteres semánticos con porciones SQL.
- V6 requisitos de verificación para la criptografía en el almacenamiento: Este requisito pertenece a la etapa de desarrollo por lo que no merece recomendación alguna para el sistema de notas.
- V7 requisitos de verificación de gestión y registro de errores: La información, referente a los errores evidenciados en el sistema de notas, deberá garantizar que

se socialice únicamente con el personal autorizado al cual se ha designado procesar dicha información. Esta información deberá estar alojada en repositorios fuera de la raíz del proyecto.

- V8 requisitos de verificación de protección de datos: Los datos son protegidos cuando se ha conseguido balancear la confidencialidad, integridad y disponibilidad (CIA) de estos. El análisis al sistema de notas evidencio que el CIA es vulnerable. Por lo que se recomienda proteger los datos de credenciales, crear una política de modificaciones periódicas de contraseñas y alojar los datos en repositorios fuera de la raíz del proyecto como ya se ha mencionado.
- V9 requisitos de verificación de seguridad de las comunicaciones: Este requisito pertenece a la etapa de desarrollo y conociendo que el sistema de notas ya se encuentra en producción, no merece recomendación alguna.
- V10 requisitos de verificación de código malicioso: Si se encuentra ambientes en el sistema de notas que se deben reprogramar, se debe considerar revisar el código ya generado y de ser posible, mejorarlo y volver a ponerlo en producción.
- V11 requisitos de verificación de lógica empresarial: Este requerimiento pertenece a la etapa de desarrollo, por lo que no se entrega sugerencia alguna.
- V12 requisitos de verificación de archivos y recursos: Los archivos del sistema de notas deberán validar su origen y si provienen de fuentes no seguras deben ser aislados en alguna ubicación distinta a la raíz.
- V13 requisitos de verificación de API y servicio web: Este requisito pertenece a la etapa de desarrollo y concomiendo que el sistema de notas se encuentra ya en producción, no merece recomendación alguna.
- V14 requisitos de verificación de configuración: Verificación de configuraciones a nivel de servicios arquitectónicos implementados en el sistema de notas. Esta verificación debe ser realizada de forma periódica.

Los detalles teóricos de estos requisitos de validación y verificación se encuentran en el Anexo 2.

6.3.4. Recomendaciones acerca de la gobernanza para la evaluación de vulnerabilidades.

- Las tres etapas, control de vulnerabilidades, determinación de criterios y, finalmente, verificación y validación garantizan la seguridad y el mantenimiento del sistema de notas.

- Debido al crecimiento de estas etapas presentadas por OWASP, mantener una política de actualización es un eje fundamental que garantiza el mantenimiento a lo largo del tiempo en el sistema de notas.
- Proteger los datos sensibles transaccionándolos en carpetas o repositorios que no se encuentren dentro del proyecto o la raíz del sistema de notas.

7 CONCLUSIONES

Las conclusiones obtenidas después de ejecutar el análisis de vulnerabilidades en el sistema de notas se presentan a continuación:

- Las técnicas de inyección SQL y clickjacking propias de hacking ético que fueron investigadas para ser aplicadas en el presente trabajo de titulación con el objetivo de ejecutar el análisis de vulnerabilidades en el sistema de notas resultaron adecuadas debido a que resultaron de gran utilidad al explotar vulnerabilidades presentes y permitieron ingresar al sistema, así como también a datos sensibles y críticos tales como historial de notas y contraseñas de docentes.
- Las vulnerabilidades que se encontraron en el sistema de notas se pudieron explotar gracias al estudio, investigación y aplicación de conocimientos técnicos que identificaron las falencias del aplicativo académico.
- La falta de aplicación de seguridad en el sistema de notas fue un factor determinante para lograr escanear y explotar vulnerabilidades con éxito, logrando obtener información crítica.
- Se determinó que el sistema de notas presenta un riesgo moderado del 67% en la etapa de reconocimiento, un riesgo importante del 43% en la etapa de escaneo y un riesgo del 100% en la etapa de explotación. Esto evidencia una deficiencia critica de seguridad poniendo en alto riesgo la información académica de la institución.
- Las técnicas de hacking ético aplicadas en la etapa de explotación de vulnerabilidades resultaron ser eficientes al 100% por lo que se concluye que el sistema de notas del caso de estudio es totalmente vulnerable.
- La guía desarrollada en el presente trabajo está diseñada para que las etapas de reconocimiento, escaneo y explotación de vulnerabilidades en los sistemas académicos sea ejecutada de manera ágil y para que la información resultante sea presentada en un lenguaje interpretable, lo que garantiza respuestas oportunas y eficientes ante debilidades de seguridad informática.

8 RECOMENDACIONES

El presente trabajo de titulación ha dejado las siguientes recomendaciones:

- Se recomienda capacitar al personal encargado de la seguridad informática en la institución educativa referente a técnicas de hacking ético con el objetivo de evaluar periódicamente el sistema de notas.
- Se recomienda al personal encargado del sistema de notas mantenerse actualizado en las nuevas tendencias y vulnerabilidades que afectan directamente al software académico.
- Es recomendable socializar las vulnerabilidades detectadas en el sistema de notas con el personal adecuado para poder implementar las medidas de seguridad necesarias y así garantizar el funcionamiento continuo del aplicativo.
- Es recomendable publicar únicamente la información institucional estrictamente necesaria, así como llegar a un acuerdo de privacidad con el proveedor tecnológico para ocultar información como la IP pública o el DNS con la finalidad de minimizar la recolección de datos en la etapa primera de hacking ético: reconocimiento.
- Se recomienda llevar a cabo de manera periódica la guía para el análisis de vulnerabilidades en el sistema de notas presentado en este trabajo de titulación con la finalidad de dar una respuesta oportuna ante riesgos y debilidades de seguridad.
- Se recomienda implementar buenas prácticas de software en los ambientes del sistema de notas de la institución con el objetivo de evitar incorporar elementos que deriven en explotación de vulnerabilidades tales como los evidenciados: inyección SQL o clickjacking.
- Se recomienda generar una política para la gestión de las credenciales del sistema de notas, así como también es recomendable cambiar las contraseñas de forma periódica y obligatoria.
- Se recomienda dar charlas de seguridad informática a todos los actores y entidades de las instituciones educativas con el objetivo de generar conciencia acerca de la criticidad y la responsabilidad compartida referente a la información que transacciona el sistema de notas.
- Es recomendable actualizar periódicamente la guía especifica que facilita el análisis de vulnerabilidades en el sistema de notas con el objetivo de garantizar la seguridad informática frente a nuevas amenazas.

REFERENCIAS BIBLIOGRÁFICAS

[1] C. Joshi and K. Singh, “Performance Evaluation of Web Application Security Scanners for More Effective Defense,” Int. J. Sci. Res. Publ., vol. 6, no. 6, p. 660, 2016.

[2] R. López de Jiménez, "Pruebas de penetración en aplicaciones web usando hackeo ético", Revista tecnológica Escuela Especializada en Ingeniería ITCA-FEPADE, no. 10, p. 7, 2017.

[3] "The Open Web Application Security Project OWASP", Owasp.org, 2019. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Accessed: 02- Sep- 2019].

[4] López de Jimenez RE, “Pentesting on web applications using ethical – hacking”. In: 2016 IEEE 36th central American and Panama convention (CONCAPAN XXXVI), pp 1–6.

[5] Joshi C, Kumar U, “Performance Evaluation of Web Application Security Scanners for More Effective Defense”, International Journal of Scientific and Research Publications, ISSN (Print) : 2250-3153, Volume-6, Issue-6, 2016.

[6] Idtheftcenter.org, 2019. [Online]. Available: https://www.idtheftcenter.org/wp-content/uploads/2019/02/ITRC_2018-End-of-Year-Aftermath_FINAL_V2_combinedWEB.pdf. [Accessed: 29- May- 2019].Idtheftcenter.org, 2019. [Online]. Available: https://www.idtheftcenter.org/wp-content/uploads/2019/02/ITRC_2018-End-of-Year-Aftermath_FINAL_V2_combinedWEB.pdf. [Accessed: 29- May- 2019].

[7] Coronel I. "Aplicar hackeo ético para detección de vulnerabilidades mediante herramientas open source en las aplicaciones web de una institución de educación superior", ESPOL, 2016. Tesis de Postgrado. [Online]. Available: https://www.dspace.espol.edu.ec/retrieve/97627/D-103391.pdf. [Accessed: 01- Oct- 2019].

[8] Sarasan S, “Detection and Prevention of Web Application Security Attacks”, International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE), ISSN (Print): 2278-8948, Volume-2, Issue-3, 2013, pp. 29- 34.

[9] El Idrissi S, Berniche N, Guerouate F, Sbihi M, “Performance Evaluation of Web Application Security Scanners for Prevention and Protection against Vulnerabilities”, International Journal of Applied Engineering Research ISS, (IJAEEE), ISSN (Print) : 0973-4562, Volume-12, Issue-21, 2017, pp. 11068- 11076.

[10] M. I. Romero Castro et al., “Introducción a la seguridad informática y el análisis de vulnerabilidades”. 2018.

[11] I. Sommerville, “Software engineering”, 9th ed. Boston: Addison-Wesley, 2011.

[12] S. M. Quiroz Zambrano and D. G. Macías Valencia, “Seguridad en informática: consideraciones, Dominio las Ciencias”, vol. 3, no. 3, pp. 676–688, 2017.

[13] D. Benchimol, “Hacking desde Cero - Conozca sus vulnerabilidades y proteja su información”, 1a ed. Buenos Aires, 2011.

[14] M. V. Bravo Sánchez and D. A. Sánchez Prieto, “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,” UPS, Tesis de Pregrado, 2018.

[15] I. S. Guzmán, F. Briones Medina, E. Cabañes Martínez, A. Miranda Díaz, J. M. Sarralde Ruiz, and G. Wolf Iszaevich, “Ética Hacker, Seguridad Y Vigilancia”, 1a ed. Universidad del Claustro de Sor Juana, 2016.

[16] E. Raymond, “The new hacker's dictionary”, 3rd ed. Cambridge, Mass.: MIT Press, 2002.

[17] E. F. Medina Rojas, “Hacking Ético: Una herramienta para la seguridad informática,” pp. 1–8, 2015.

[18] D. Mamami, “Fases de un Ataque Hacker,” pp. 1–2, 2013.

[19] R. González, "Las fases del Hacking Ético - Ethical Hack", Ethical Hack. [Online]. Available: https://ehack.info/las-fases-del-hacking-etico/. [Accessed: 20- Oct- 2019].

[20] "Amenazas web, malware de navegador de internet", Latam Kaspersky, 2016”. [Online]. Available: https://latam.kaspersky.com/resource-center/threats/web. [Accessed: 01- Nov- 2019].

[21] "OWASP publica el Top 10 – 2017 de Riesgos de Seguridad en Aplicaciones Web", INCIBE CERT, 2018. [Online]. Available: https://www.incibe-cert.es/blog/owasp-publica-el-top-10-2017-riesgos-seguridad-aplicaciones-web. [Accessed: 01- Nov- 2019].

[22] "OWASP Top 10 -2017Los diez riesgos más críticos en Aplicaciones Web", OWASP, 2019. [Online]. Available: https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf. [Accessed: 02- Nov- 2019].

[23] V. Medrano, "Las fases de un test de penetración (Pentest) (Pentesting I) - Ciberseguridad,seguridad informática,redes y programación.", Cyberseguridad, 2015. [Online]. Available: https://cyberseguridad.net/index.php/455-las-fases-de-un-test-de-penetracion-%20pentest-pentesting-idebianHackers. [Accessed: 02- Nov- 2019].

[24] J. Clarke, “*SQL injection attacks and defense”*, 2nd ed. Waltham, Mass.: Syngress, 2012.

[25] H. Gao, "*Cross-Site Scripting: The most prevalent web application risk*", Owasp, 2017. [Online]. Available:

https://www.owasp.org/images/c/c3/OWASPTop10XSSLongIsland.pdf. [Accessed: 02- Nov- 2019].

[26] "*OWASP Top 10 Vulnerabilities 2019*", SUCURI, 2019. [Online]. Available: https://info.sucuri.net/hubfs/images/owasp-ebook-2019/sucuri-ebook-OWASP-top-10.pdf. [Accessed: 03- Nov- 2019].

[27] "*ITQ – Educación Superior*", ITQ. [Online]. Available: http://itq.edu.ec/. [Accessed: 08- Nov- 2019].

[28] "*Testing Checklist - OWASP*", OWASP, 2014. [Online]. Available: https://wiki.owasp.org/index.php/Testing_Checklist. [Accessed: 01- Jan- 2020].

[29] "*Clickjacking | OWASP*", Owasp.org, 2019. [Online]. Available: https://owasp.org/www-community/Clickjacking. [Accessed: 08- Jan- 2020].

[30] "*SQL Injection | OWASP*", Owasp.org, 2019. [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection. [Accessed: 08- Jan- 2020].

[31] "*NIST Special Publication 800-63B*", Pages.nist.gov, 2017. [Online]. Available: https://pages.nist.gov/800-63-3/sp800-63b.html#memsecret. [Accessed: 17- Jan- 2020].

[32] "*Computer SecurityIncident Handling Guide*", nist.gov, 2012. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf. [Accessed: 17- Jan- 2020].

[33] "*OWASP Application Security Verification Standard",* OWASP, 2020. [Online]. Available: https://owasp.org/www-project-application-security-verification-standard/. [Accessed: 27- Jan- 2020].

[34] "OWASP/ASVS", GitHub, 2020. [Online]. Available: https://github.com/OWASP/ASVS#latest-released-version. [Accessed: 27- Jan- 2020].

[35] "ESTADÍSTICAS | Mapa en tiempo real de amenazas cibernéticas Kaspersky", MAPA | Mapa en tiempo real de amenazas cibernéticas Kaspersky, 2020. [Online]. Available: https://cybermap.kaspersky.com/es/stats/. [Accessed: 19- Mar- 2020].

ANEXOS

ANEXO I

Explicación de los controles de OWASP implementados para realizar el análisis de vulnerabilidades en el sistema de notas que necesiten una ampliación de uso.

OTG-INFO-003: Revisión de los archivos y metarchivos del servidor web con la finalidad de detectar fugas de información.

Herramienta: DomainTools

Para ejecutar el análisis es necesario seguir los siguientes pasos:

- Ir a la URL http://whois.domaintools.com/
- Ingrese el dominio o la dirección IP des sistema de notas en el casillo destinado para eso.
- Presionar el botón “Search” para ejecutar el análisis como se muestra en la siguiente figura 56:

Figura 56: Ejecución de la herramienta domain tools. [Elaboración propia]

- Finalmente se mostrarán los resultados del análisis, los cuales deberán ser procesados e interpretados.

OTG-INFO-005: Revisión de comentarios y metadatos de la aplicación web para identificar fuga de información.

Herramienta: OWASP ZAP.

Para realizar el ataque, se recomienda seguir los siguientes pasos:

- Descargar la herramienta desde la siguiente URL https://www.zaproxy.org/
- Instalar la herramienta.
- Una vez instalada, ingresar la URL del sistema de notas en el casillero de texto denominado “URL to attack” y presionar el boton “Atacar” como se presenta en la siguiente figura 57:

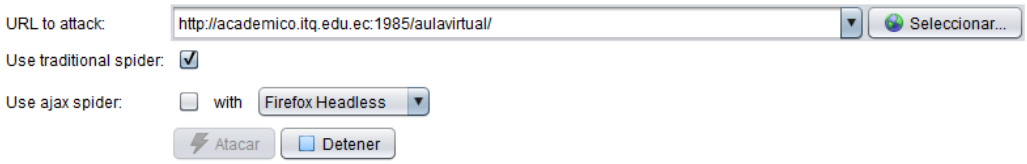


Figura 57: Ingreso de información para lanzar el ataque con ZAP. [Elaboración propia]

- El ataque al dominio inicia mostrando en la sección inferior, el avance y progreso de la ejecución en la pestaña “Escaneo Activo”. En esta sección de la aplicación encontramos más pestañas tales como “Historia”, “Buscar”, “Alertas”, “Salida”, “Spider (Araña)” como se presenta en la figura 58, la cuales, al terminar el análisis, servirán para evidenciar e interpretar información crítica, valiosa y sensible que servirá de insumo para realizar ataques en la etapa de explotación de vulnerabilidades.

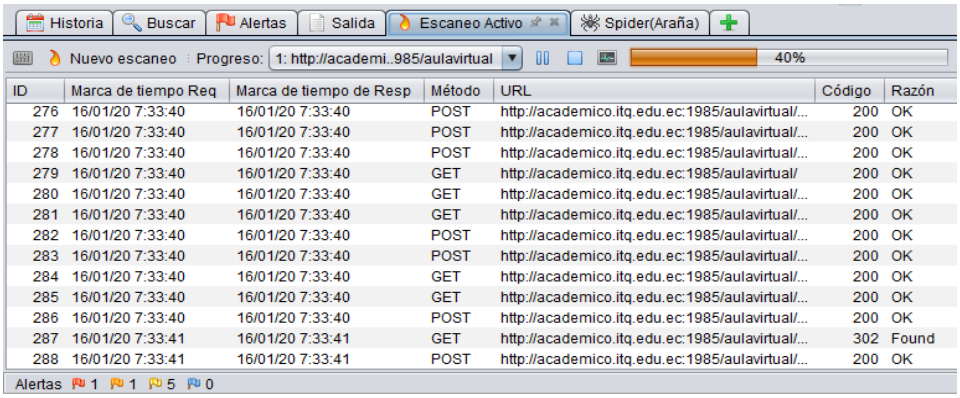


Figura 58: Avance y proceso de la ejecución de ZAP. [Elaboración propia]

- Al concluir la ejecución de la herramienta, ZAP presenta en la pestaña “Alertas” los resultados del análisis como se muestra en la figura 59. Esta sección contiene las debilidades del sistema de notas divididas en tres categorías: **riesgo alto (bandera roja), riesgo medio (bandera naranja) y riesgo bajo (bandera amarilla).**

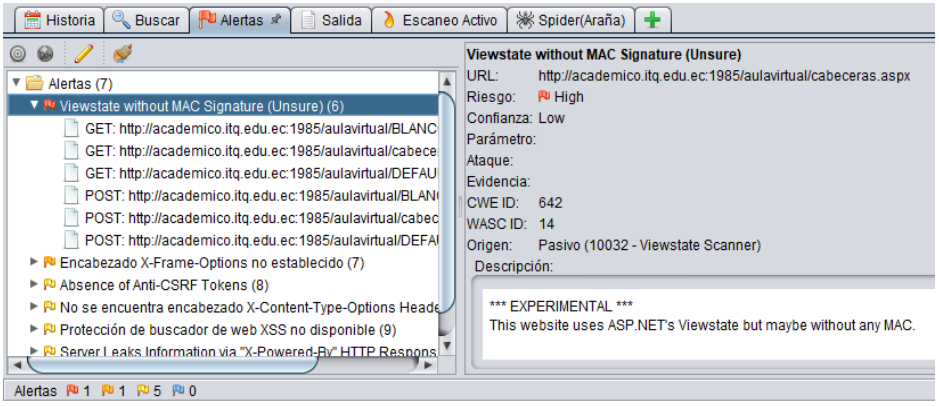


Figura 59: Alertas que muestran el criterio de riesgo. [Elaboración propia]

- En este caso de estudio, se han encontrado un riesgo alto, un riesgo medio y cinco riesgos bajos presentado en la tabla 10:

Riesgo alto	Riesgo medio	Riesgo bajo
1.- Ausencia de algún tipo de firma o MAC del Viewstate	1.- Encabezado X-Frame-Options no establecido	1.- Absence of Anti-CSRF Tokens 2.- No se encuentra el encabezado X-Content-Type-Options Header 3.- Protección de buscador de web XSS no disponible. 4.- Server leaks information 5.- Escáner de encabezado de respuesta de la versión X-AspNet

Tabla 10: Subcategorías de riesgos encontrados con ZAP. [Elaboración propia]

- El primer y único riesgo alto, presentado en la tabla 10, pertenece a la ausencia de algún tipo de firma o MAC del Viewstate de APS.NET.
- El primer y único riesgo medio, mostrado en la tabla 10, indica que el encabezado X-Frame-Options no ha sido establecido en el sistema de notas. Por lo que el sito es vulnerable ante ataques de ClickJacking como se indica en las figuras 60 y 61.

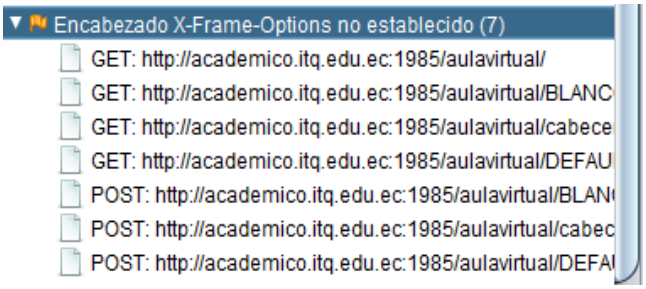


Figura 60: Encabezado X-Frame-Options no establecido. [Elaboración propia]

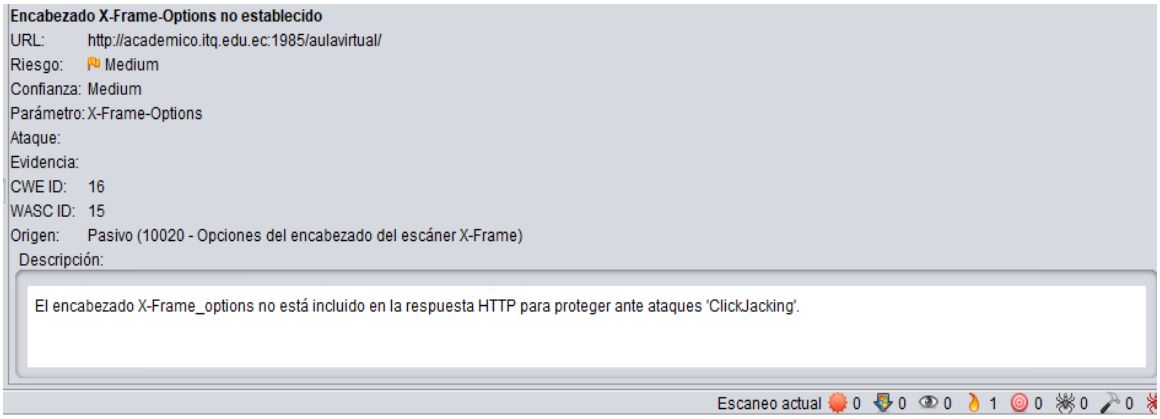


Figura 61: Evidencia para ataques de ClickJacking. [Elaboración Propia]

- El primero de los cinco riesgos bajos, mostrados en la tabla 10, presenta una ausencia de tokens Anti-CSRF como se visualiza en la figura 62**.**

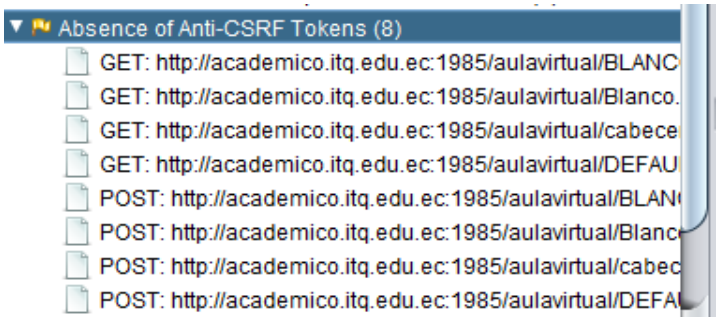


Figura 62: Ausencia de tokens Anti-CSRF. [Elaboración propia]

- La descripción resultante del análisis y perteneciente al primer riesgo bajo categorizado en la tabla 10, presenta el siguiente mensaje: “No Anti-CSRF tokens were found in a HTML submission form. Una solicitud falsa entre sitios en un ataque que compromete y obliga a una víctima a enviar su solicitud HTTP a un destino objetivo sin su conocimiento o intención para poder realizar una acción como víctima. La causa oculta es la funcionalidad de la aplicación utilizando acciones de URL/formulario que pueden ser adivinados de forma repetible. La naturaleza del ataque es que CSRG explota la confianza que un sitio web proporciona a un usuario. Por el contrario, las cadenas de comandos de los sitios cruzados (XSS) explotan la confianza que un usuario proporciona en un sitio web. Al igual que XSS, los ataques CSRG no son de forma necesaria de sitios cruzados, pero hay la posibilidad de que si pueden serlo. La falsificación de las solicitudes ente los sitios también se conoce como CSRF, XSRG, ataques con un solo clic, montaje de sesión, diputado confundido y navegación en alta mar. Los ataques de CSRG son muy efectivos en varias situaciones, que incluyen:
    - La victima tiene una sesión activa en el sitio de destino.
    - La víctima se autoriza por medio de la autenticación HTTP en el sitio de destino.
    - La víctima se encuentra en la misma red local que el sitio de destino.

  CSRF se ha utilizado especialmente para poder realizar una acción contra un sitio objetivo utilizando los privilegios de la víctima, pero se han revelado técnicas recientes para difundir información al obtener el acceso a la respuesta. El riesgo de divulgación de información aumenta de forma drástica cuando el sitio de destino se encuentra vulnerable a XSS, porque XSS se puede utilizar como una plataforma

para CSRF, lo que le permite al atacante que opere desde adentro de los límites de la misma política de origen."

- El segundo de los cinco riesgos bajos evidenciados en la tabla 10, presenta la ausencia de encabezado X-Content-Type-Option Header como se muestra en la figura 63.

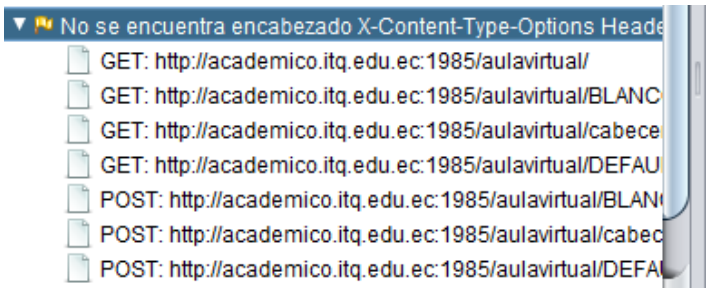


Figura 63: Escaner de respuesta de version X-AspNet. [Elaboración propia]

- Los resultados de dicho riesgo bajo se muestran a continuación en la figura 64.

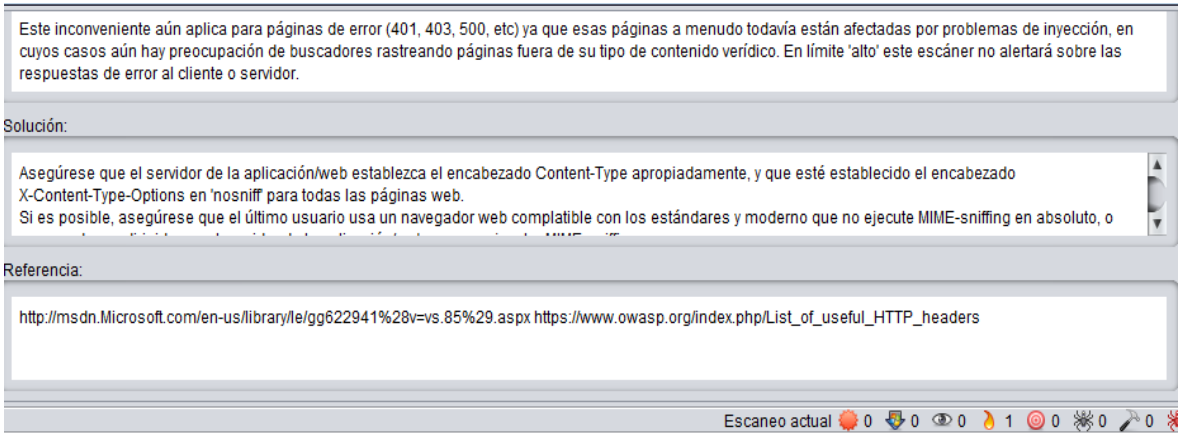


Figura 64: Descripción, solución y referencia del segundo riesgo bajo. [Elaboración propia]

- El tercero de los cinco riesgos bajos presentados en la tabla 10 evidencia la ausencia de protección de buscador de web XSS como se muestra a continuación en la figura 65.

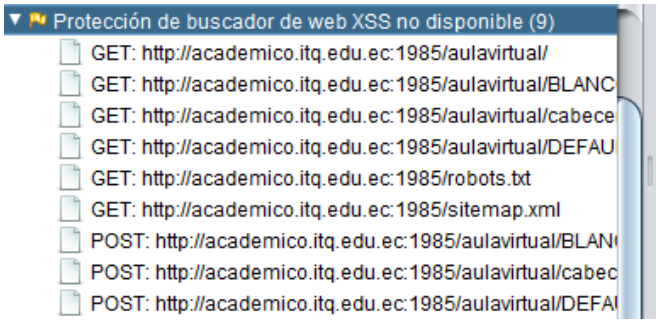


Figura 65: Protección de buscador de web XSS no disponible. [Elaboración propia]

- Los resultados de este riesgo bajo se muestran a continuación en la figura 66.

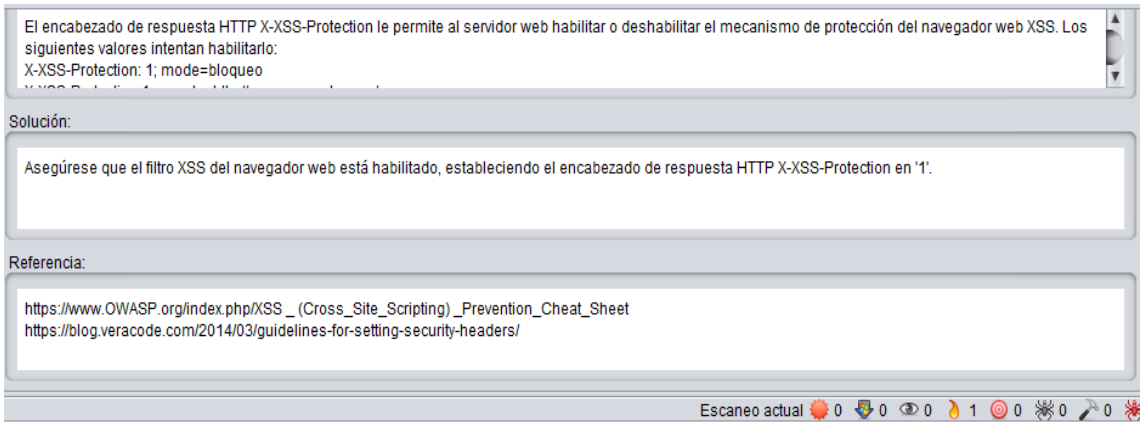


Figura 66: Descripción, solución, y referencia del tercer riesgo bajo. [Elaboración propia]

- El cuarto de los cinco riesgos bajos presentados en la tabla 10 evidencia que el servidor pierde información a través de los campos de encabezado de respuesta HTTP “X-Powered-By” como se visualiza en la figura 67.

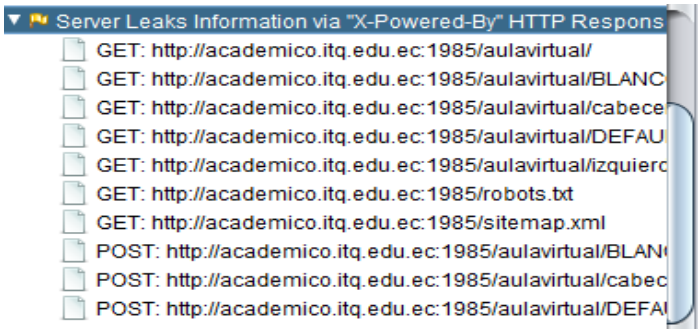


Figura 67: Perdida de información del servidor. [Elaboración propia]

- Los resultados de este riesgo bajo se muestran a continuación en la figura 68.

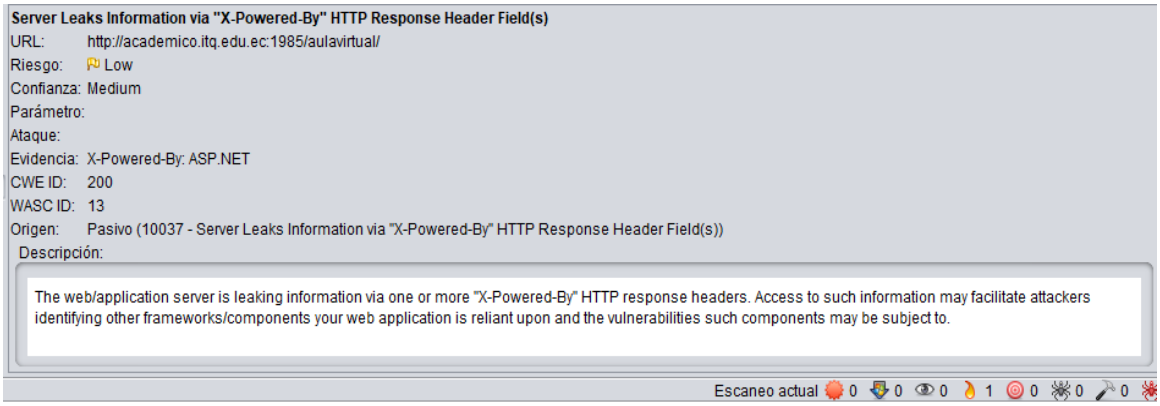


Figura 68: Descripción del cuarto riesgo bajo. [Elaboración propia]

- El quinto riesgo bajo de los cinco presentado en la tabla 10 realiza un escáner al encabezado de respuesta de la versión X-AspNet como se visualiza en la figura 69.

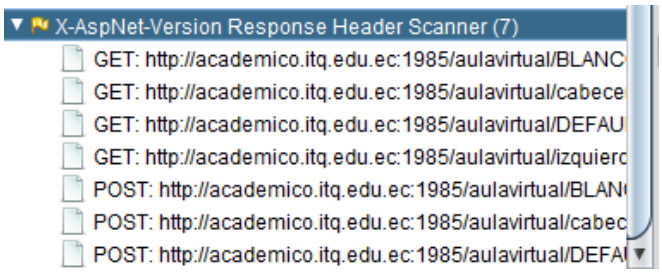


Figura 69: Escáner de encabezado de respuesta de versión X-AspNet. [Elaboración propia]

- Los resultados de este quinto riesgo bajo presentado en la tabla 10 se muestran a continuación en la figura 70.

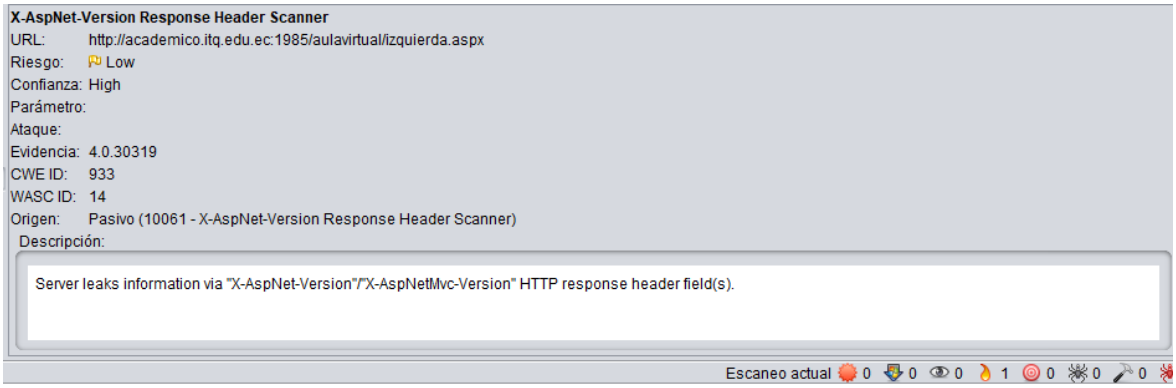


Figura 70: Resultados y descripcion del quito riesgo bajo. [Elaboración propia]

ANEXO II

Application Security Verification Standard (ASVS)

Requisitos orientados a la validación y verificación de aplicaciones seguras se listan a continuación:

- **V1: Architecture, Design and Threat Modeling Requirements:** La arquitectura de seguridad no solo es una implementación a nivel arquitectónico, sino una forma de pensar con respecto a un potencial problema que tiene una infinidad de respuestas diferentes y ninguna respuesta única marcada como correcta. El objetivo de esta primera verificacion es cumplir con los siguientes aspectos: disponibilidad, integridad, confidencialidad, privacidad, integridad del proceso y no repudio.
- **V2: Authentication Verification Requirements:** La autenticación clásica es una combinación adecuada de credenciales pertenecientes a usuario y contraseña, sin embargo, como se ha demostrado en capítulos anteriores, este proceso es vulnerable a nivel de usuario, por lo que esta verificación se enfoca en validar criterios y manejo de credenciales no solo a nivel de software sino también a nivel de personas
- **V3: Session Management Verification Requirements:** Con la finalidad de garantizar que el manejo de sesiones en aplicaciones informáticas se lleve de manera segura, se muestran las siguientes recomendaciones:
    - Las sesiones son únicas por usuario, esto impide que sean compartidas.
    - Las sesiones tienen un tiempo máximo de uso, pasado ese tiempo la sesión expira.
- **V4: Access Control Verification Requirements:** El control de acceso garantiza que usuarios con credenciales correctas ingresen a utilizar recursos de acuerdo con sus roles, para lo cual:
    - Los usuarios que acceden al sistema informático tienen credenciales para hacerlo
    - Los usuarios están asignados a un conjunto definido de roles y privilegios.
    - Los datos y metadatos de roles, privilegios y credenciales están protegidos de ataques informáticos.
- **V5: Validation, Sanitization and Encoding Verification Requirements**: La falla más recurrente en los sistemas informáticos es la validación adecuada de entradas provenientes del usuario, esta vulnerabilidad deriva en ataques de Cross Site

Scripting (XSS), inyección SQL, ataque de sistema de archivos, etc. Para garantizar este criterio de validación se debe tomar en cuenta:

  - Los datos de entrada deben estar fuertemente validados, con rango de longitud establecidos y verificados y, en el peor de los casos, desinfectados o filtrados.
  - Los datos de salida se codifican lo más cercano al controlador o al interprete.

- **V6: Stored Cryptography Verification Requirements:** Verificar que la aplicación informática cumple con las siguientes recomendaciones de alto nivel:
  - Las transacciones criptográficas fallan de manera segura y los errores generados se gestionan correctamente.
  - Se implementa un generador aleatorio de números de manera adecuad.
  - El acceso a las contraseñas se gestiona de forma controlada y segura.
- **V7: Error Handling and Logging Verification Requirements:** El objetivo principal de esta verificación es proporcionar información útil a los usuarios, administradores y al personal de seguridad informática con la finalidad de dar respuesta a incidentes. Sin embargo, al tratarse de información sensible y confidencial, el tratamiento de este proceso debe protegerse de acuerdo con las políticas internas de la privacidad de los daros, para lo cual:
  - No se debe registrar información altamente confidencial a menos que realmente se requiera.
  - Garantizar que la información almacenada se gestione de forma segura según su clasificación de datos.
  - Asegurar que la información no sea almacenada para siempre, generando políticas o criterios de eliminación. Esta vida útil debe ser lo más corta posible.
- **V8: Data Protection Verification Requirements:** Existe tres elementos que garantizan una adecuada protección de datos: confidencialidad, integridad y disponibilidad (CIA). Balancear la triada del CIA asegura que la protección de datos se aplica a un sistema confiable, como por ejemplo en un servidor que se ha reforzado de tal manera que tiene protecciones suficientes. Para lo cual, la aplicación informática debe cumplir con las siguientes recomendaciones:
  - **Confidencialidad:** Los datos deben protegerse, tanto en transacciones como en almacenamiento, de divulgaciones no autorizadas.
  - **Integridad:** Se garantiza que los datos están protegidos de creación, alteración o eliminación por parte de atacantes malintencionados que no han sido autorizados.

- **Disponibilidad:** Se asegura que los datos sean siempre accesibles por usuarios autorizados.

- **V9: Communications Verification Requirements:** Dependiendo de la confidencialidad entregada a cierta información, asegurar que:
  - El cifrado fuerte siempre se utiliza, independientemente de la sensibilidad de los datos transaccionados.
  - Las mejores prácticas más recientes son aplicadas para habilitar y organizar algoritmos y cifrados utilizados en la aplicación informática.
  - Los algoritmos débiles o que, dentro de poco serán obsoletos, son utilizados como último recurso.
  - Los algoritmos y cifrados obsoletos no son utilizados y estarán deshabilitados.
- **V10: Malicious Code Verification Requirements:** Con la finalidad de no generar código malicioso, esta verificación garantiza que el código utilizado cumpla con las siguientes recomendaciones:
  - La actividad sospechosa y maliciosa es tratada de forma independiente, de manera que no afecte al resto de la aplicación.
  - El código no tiene puertas traseras, rootkits o código desautorizado que derive en que un atacante pueda controlarlo.
  - Se debe realizar el mayor de los esfuerzos para garantizar que el código no contenga entre sus líneas secciones maliciosas inherentes o incorporadas en funcionalidades no deseadas.
- **V11: Business Logic Verification Requirements:** Validar que la aplicación informática verificada cumpla con los siguientes requerimientos:
  - El flujo de la lógica de procesos se automatiza de manera secuencial, en orden y sin omisión.
  - La lógica de negocio debe incluir controles para detectar, controlar y prevenir ataques informáticos.
- **V12: File and Resources Verification Requirements:** Esta verificación garantiza que los archivos y recursos incorporen criterios de seguridad, para lo cual:
  - Los datos que provienen de archivos que tengan un criterio de confianza bajo deben ser procesados de manera segura.
  - Los datos de archivos no confiables provenientes de fuentes inseguras deberán ser almacenadas fuera de la raíz de la aplicación web y con permisos limitados.

- **V13: API and Web Service Verification Requirements:** Asegurar que la aplicación tecnológica evaluada que utilice una capa de servicios web tenga:
    - Manejo de sesiones.
    - Autorización, bajo credenciales, de todos los servicios web solicitados.
    - Validación de las entradas de datos que pasan de un nivel de confianza bajo a uno más alto.
    - Controles de seguridad para APIs en la nube y sin servidor.
- **V14: Configuration Verification Requirements:** Asegurar que la aplicación informática verificada cuente con:
    - Gestión reforzada de la biblioteca de datos, dependencia y configuración de terceros de manera que se garantice que la aplicación no implemente componentes obsoletos o inseguros.
    - Incorporar configuraciones con criterios de seguridad internas, de manera que no incluya configuraciones por defecto.

ANEXO III

Solicitud de análisis de vulnerabilidades en el Instituto Tecnológico Quito

Quito, viernes 29 de noviembre de 2019

Ingeniero,
Iván Cadena
Rector
Instituto Tecnológico Quito
Presente. -

De acuerdo con la afirmación y autorización verbal de su parte de aplicar técnicas de hacking ético en el sistema de notas que actualmente gestiona los procesos académicos a nivel de calificaciones en el instituto, procedo a redactar esta carta con la finalidad de que exista evidencia escrita.

El Instituto Tecnológico Quito, como se explicó de forma verbal, es el caso de estudio que forma parte de la tesis creada por mi persona. Tesis titulada: "ESTRATEGÍA PARA LA EVALUACIÓN DE VULNERABILIDADES DEL SISTEMA DE NOTAS DE INSTITUCIONES EDUCATIVAS UTILIZANDO TÉCNICAS DE HACKING ÉTICO. CASO DE ESTUDIO: INSTITUTO TECNOLÓGICO QUITO".

Por la atención prestada al documento, anticipo mis agradecimientos.

Recibido
I. Cadena
29-NOV-2019

ITQ
INSTITUTO TECNOLÓGICO QUITO
RECTOR

Atentamente,

Ing. David Galarza C.
Estudiante de la Maestría en Software
Escuela Politécnica Nacional

Figura 71: Solicitud de análisis de vulnerabilidades. [elaboración propia]