UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

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

TEMA:
IDENTIFICACIÓN Y ANÁLISIS DE VULNERABILIDADES EN LOS PORTALES WEB DE LA UNIVERSIDAD POLITÉCNICA SALESIANA A TRAVÉS DE TÉCNICAS DE PENTESTING.

AUTOR:
ERIK ALEJANDRO PARRA TAPIA

TUTOR:
DANIEL GIOVANNY DÍAZ ORTIZ

Quito, febrero del 2020

CESIÓN DE DERECHOS DE AUTOR

Yo, ERIK ALEJANDRO PARRA TAPIA, con documento de identificación N° 1717154627, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: IDENTIFICACIÓN Y ANÁLISIS DE VULNERABILIDADES EN LOS PORTALES WEB DE LA UNIVERSIDAD POLITÉCNICA SALESIANA A TRAVÉS DE TÉCNICAS DE PENTESTING, mismo que ha sido desarrollado para optar por el título de INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, subscribo este documento en el momento que hago la entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.

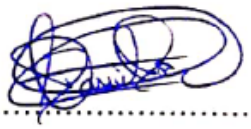
ERIK ALEJANDRO PARRA TAPIA

Cédula: 1717154627

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo, Ing. DANIEL GIOVANNY DÍAZ ORTIZ, con documento de identificación Nro. 1716975501, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, con el tema: IDENTIFICACIÓN Y ANÁLISIS DE VULNERABILIDADES EN LOS PORTALES WEB DE LA UNIVERSIDAD POLITÉCNICA SALESIANA A TRAVÉS DE TÉCNICAS DE PENTESTING, realizado por Erik Alejandro Parra Tapia, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero del 2020



.....

DANIEL GIOVANNY DÍAZ ORTIZ

CI. 1716975501

DEDICATORIA

A Dios, creador y dueño de mi ser, que me ha otorgado sabiduría suficiente para afrontar con rectitud los retos en mi carrera y en mi vida personal.

A mis amadísimos padres Lucila y Fernando, quienes son el pilar fundamental en mi vida que con su amor incondicional, esfuerzo, paciencia y comprensión han estado a mi lado día tras día guiándome, formándome para ser una excelente persona, siempre con cariño mostrándome el camino correcto para alcanzar mis metas y cumplir todos mis sueños, los amo.

A mis hermanos Jonathan y Rommel(+) que siempre han estado junto a mí brindándome su incondicional apoyo y fortaleza para afrontar buenos y malos momentos.

A mis abuelitas Angelita, Leonor y Dolores que siempre con sus sabios consejos, su esfuerzo y su infinito amor siempre han estado al pendiente de mí para que con sus bendiciones pueda culminar todas mis metas.

A mis abuelitos Luis y Rafael(+) que siempre me supieron comprender y apoyar cuando más lo necesito, siendo ellos un ejemplo de rectitud, sacrificio, esfuerzo y guías para mi vida.

A todos mis primos y primas que siempre han sido una fuente de alegría y carisma en mi vida que siempre han estado en buenos y malos momentos, a los que considero más que primos hermanos y hermanas.

A Katherine que en todo este tiempo hemos vivido hermosas experiencias y siempre ha estado junto a mí, dándome todo su amor y apoyo para cumplir mis sueños, brindándome ella y toda hermosa familia un segundo hogar.

A mis tíos Omar, Ricardo, Guillermo, Jorge(+), Luis(+), José, Vilma, Anita, María Elena, Victoria e Irina siempre han velado por mi bienestar en todos los momentos, brindándome su inmenso cariño, escucha y consejos siendo ellos un ejemplo para mí de esfuerzo, fortaleza, dedicación, optimismo y amor, gracias por hacerme de ustedes un hijo más.

A Lupita que ha sido una segunda madre para mí estando presente en todos los momentos de mi carrera, empujándome con amor, fortaleza y esfuerzo para llegar a cumplir mis metas, gracias por ser ese gran pilar en mi vida ñañita.

A Brizeira que llego a cambiarme la vida y apoyarme justo cuando más lo necesitaba.

A todos mis amigos y familiares que con su carisma están siempre para escucharme, aconsejarme, ayudarme y ser los cómplices que mi vida necesita.

Y no me quiero olvidar de nadie, esta meta se los dedico a todos, ¡muchísimas gracias!

Erik Alejandro Parra Tapia

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de vivir y la sapiencia para cursar mi vida

A la carrera de Sistemas de la Universidad Politécnica Salesiana, por abarcarme en sus aulas y hacer de mí un honrado ciudadano.

A mis queridos maestros, que, a lo largo de la carrera, me brindaron sus conocimientos, ayuda y consejo, supieron sembrar para cosechar lo mejor de mí.

A mi tutor por su paciencia y ser esa directriz que necesitaba para culminar con éxito este logro nuestro.

A mis amigos y compañeros de clases, que siempre existió esa complicidad y solidaridad para hacer de esta, la mejor etapa de mi vida.

A mi grupo de investigación ASU de seguridad informática que me ha hecho esforzar para aprender y enseñarles lo que me apasiona.

Erik Alejandro Parra Tapia

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

Este proyecto se enfoca en realizar un pentesting para el análisis de vulnerabilidades de los portales web de la Universidad Politécnica Salesiana. El proyecto sigue la metodología ISSAF que permite descubrir fallos de seguridad, amenazas y posibles filtraciones de información que incumpla las condiciones de uso del manejo adecuado de la información y expongan directa o indirectamente al usuario final.

Con dicho pentesting se pretende incrementar la seguridad al determinar la gravedad que un ataque puede llegar a tener ante los sistemas publicados y de este modo se brinde una propuesta de mitigación para el respectivo control de la seguridad en las páginas web.

Palabras clave: análisis de vulnerabilidades, pentesting, portales UPS, seguridad de la información.

Abstract

This project focuses on performing a pentesting for vulnerability analysis of web portals in Universidad Politécnica Salesiana. The project follows ISSAF methodology that enables to find out security lapses, threats and possible information filtering which do not fulfill the use conditions of the appropriate information handling and exposes the final user directly or indirectly.

This pentesting aims to increase security when determining the severity that an attack can have on published systems and thus provide a mitigation proposal for the respective safety check on web pages.

Keywords: information security, pentesting, UPS portals, vulnerability analysis.

Introducción

Antecedentes

En la actualidad el acceso al internet se ha convertido en una necesidad de la vida diaria, ya que su aparición ha hecho que muchos servicios se conecten entre sí y simplifiquen nuestra vida.

Muchas organizaciones manejan sus procesos informáticos en la red y utilizan intranet o internet para su estrategia de mercado obteniendo acceso a información, disponibilidad y velocidad en sus procesos como publicidad, email, multimedia, e-commerce, entre otros.

Entre los problemas más grandes del internet se encuentran los crackers o hackers black hat que son personas con un gran conocimiento informáticos que buscan incesantemente romper seguridades en los sistemas informáticas y comprometer la disponibilidad, confidencialidad e integridad de los sistemas, los portales web no son la excepción hay personas que se dedican a atacar exclusivamente a este tipo de portales con el fin de obtener información confidencial y obtener de ella un provecho personal y económico principalmente.

"La mayoría de organizaciones no disponen de los recursos humanos y económicos necesarios para implementar, mantener y mejorar a lo largo del tiempo un sistema de seguridad de la información eficaz, que cumpla las expectativas de la organización. Esta tarea requiere personal cualificado, herramientas apropiadas y procesos eficientes, al alcance solamente de quienes se dedican en exclusiva a ellos" (Álvarez & Pérez, 2004)

Con el aumento en el uso de portales web, muchas instituciones permiten a sus docentes, estudiantes y personal institucional, acceder a sus sistemas de información integrados a sus portales web.

Problema

Actualmente el portal web de la Universidad Politécnica Salesiana brinda servicios de biblioteca, notas académicas, facturación electrónica, ambientes virtuales, inscripciones en línea, bolsa de trabajo y correo institucional, entre otras que son de relevancia para la comunidad educativa los cuales requieren de exigentes configuraciones de seguridad.

Por la ausencia de ciertas medidas de seguridad en la institución se ha creado un problema que a largo plazo va creciendo, de forma que aumenta de muchas maneras la probabilidad de ser objetivo de atacantes informáticos.

El análisis de vulnerabilidades permite identificar las brechas de seguridad con respecto al sistema web puesto en marcha y por medio de la utilización de técnicas de pentesting y metodologías de hacking ético se desea realizar una auditoría como agente externo a los portales web de la Universidad Politécnica Salesiana, mediante ataques en ambientes controlados y sin ninguna repercusión en los sistemas puestos en marcha en los servidores web, esto con el fin crear una conciencia de prevención ante accesos no autorizados a los servicios del portal web de la institución.

"Entre los principales problemas de seguridad de los aplicativos webs podemos destacar los siguientes: Control de acceso, Autentificación de usuarios,

Gestión de la sesión, Gestión de la configuración, Visibilidad de datos sensibles, Validación de entradas de datos de usuario, Inyección en diferentes ámbitos (SQL, XSS, HTML)" (PINOS SOLANO , 2018)

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

La ventaja del presente proyecto es que mediante el resultado de los análisis se entregará un plan de recomendaciones de mitigación y un aplicativo web de monitoreo de incidencias que ayude a reforzar los niveles de seguridad del portal web del Universidad Politécnica Salesiana y de esta manera aumentar la defensa contra los diferentes tipos de ataques informáticos, salvaguardando la confidencialidad, integridad y disponibilidad del portal web.

Justificación del tema

Este trabajo está basado para dar opciones de mitigación de vulnerabilidades web encontradas en los servidores de la Universidad Politécnica Salesiana ya que al tener brechas de seguridad en sus portales web se minimice el impacto de posibles ataques los cuales puedan afectar en: el rendimiento de la red, vulnerar la integridad, confidencialidad y disponibilidad de la información que estos ofrecen.

Al realizar el estudio de riesgos y amenazas donde se identifique los factores vulnerables del sistema se puede minimizar el impacto que un ataque genere,

economizando recursos y tiempo de respuesta ante este tipo de incidentes. Esto se complementa al dar un mantenimiento constante y óptimo, al igual que utilizar protocolos y estándares de seguridad como por ejemplo la serie de los estándares ISO 27000 que fortalecen las seguridades de la información en este caso de los portales de la Universidad Politécnica Salesiana.

Esta investigación busca generar una conciencia de seguridad la cual se represente en una toma de decisiones por parte de los administradores del sistema que aporten en fortalecer las vulnerabilidades encontradas en este proyecto aumentando de esta manera la seguridad de los usuarios y la efectividad de las páginas web.

Factores muy importantes para solucionar problemas de seguridad es el hacer una programación segura en los portales para impedir futuros ataques y minimizar el alcance que puedan llegar a generar. Llevar un control de actualizaciones constantes a los sistemas siempre en busca de mejoras prácticas para la seguridad hacen de este un sistema más sólido e impenetrable. Generar historiales de logs para verificar posibles errores existentes en el sistema y en caso de ser necesario hacer un rollback del sistema para tenerlo activo.

Objetivos

Objetivo general

Determinar y mitigar las vulnerabilidades de los portales web con los que trabaja la Universidad Politécnica Salesiana mediante técnicas de pentesting.

Objetivos específicos

- Analizar el estado del arte de técnicas de pentesting sobre portales web para conocer los estudios y resultados sobre su aplicación.

Identificar las vulnerabilidades de los portales web académicos de la Universidad Politécnica Salesiana mediante el uso de herramientas open source.

Aplicar técnicas de pentesting para realizar ataques sobre las vulnerabilidades detectadas.

Analizar los resultados obtenidos y detallar los ataques con mayor eficiencia al momento de ser explotados.

Generar una propuesta para mitigar las vulnerabilidades detectadas en el proceso.

Marco metodológico

Para empezar a determinar la metodología se realiza una tabla comparativa en donde se señalan las características de cada metodología y buscamos una que cumpla con nuestras necesidades.

Tabla 1 Características importantes de las metodologías del pentesting

Metodología/Características	ISSAF	PTES	OWASP	Cyber Kill Chain	PCI
Reconocimiento del objetivo	X	X	X	X	X
Descubrir criterios a testear o evaluar	X	X	X		
Puntos a cubrir	X			X	
Recolectar requisitos o información	X	X	X		
Enfoques de estándares	X		X		
Recomendaciones de la industria			X		X
Estudio o desarrollo de herramientas				X	
Determinar cobertura del análisis	X				
Análisis de vulnerabilidades	X	X	X	X	X
Evaluación de vulnerabilidades	X			X	
Clasificación de las vulnerabilidades	X		X		X
Instalación de herramientas				X	
Explotación	X	X	X	X	X
Estimar Alcance	X				
Post-Explotación		X	X	X	
Informe de los resultados esperados	X				X
Informes de resultados	X	X	X	X	
Contramedidas y recomendaciones	X		X		
Propuesta de mitigación de vulnerabilidades	X		X		

Nota: Tabla guía para elegir la metodología a ser utilizada en el proyecto

Como se puede observar en la Tabla 1 se elige la metodología a ser utilizada en el desarrollo del proyecto basado en las características que en la justificación se propone, entonces la metodología que más se ajusta a nuestro proyecto es ISSAF.

Del inglés (Information Systems Security Assessment Framework) la metodología de evaluación de seguridad de sistemas de información permite evaluar un sistema a través de distintas fases, que permite llegar a generar un conocimiento sólido de las brechas de seguridad y vulnerabilidades que está expuesto los sistemas para una posterior mitigación de aquellos problemas de seguridad.

Este marco de trabajo permite desarrollar un pentesting ordenado y conciso pues a diferencia de PTES y OWASP son muy puntuales en los criterios que se deben cubrir para llegar a determinar los fallos de seguridad y por el contrario ISSAF organiza en criterios de evaluación de acuerdo a la red, al control o al sistema de aplicaciones que se desea determinar las brechas de seguridad, estas prácticas vienen enfocadas en sus tres fases principales que son:

Planificar.

Para esta primera etapa o fase de la metodología se realizan los pasos iniciales para desarrollar un conocimiento previo del entorno en donde se va a realizar el pentesting de igual manera de determina el alcance, los casos de pruebas, el escalamiento de privilegios, los targets a ser analizados y el enfoque que tendrá el pentesting.

Entonces para este proyecto se determinará como tipo de ataque un ataque externo independiente de la institución en donde se resguardará el rendimiento óptimo del sistema puesto en marcha, pero evidenciando los fallos que deben parchar para la posteridad.

Evaluar.

En esta fase es donde se realiza el pentesting en sí ya que es donde se realiza en la recolección de información de los targets anteriormente definidos por el hacker ético, posterior a ello se recolecta un mapeo de la red para determinar posibles causas de falsos positivos o dispositivos que impidan el correcto desarrollo del pentesting. Consecuente con la culminación del mapeo se identifican las principales vulnerabilidades y se categorizan según un criterio de prioridad ante la amenaza que exista sobre dicha vulnerabilidad. Subsiguiente se emplean todas las herramientas necesarias para realizar la penetración del sistema y por ende esta sub fase es la más polémica de todas pues se ponen a prueba las medidas de seguridad y la respuesta que tiene un sistema ante un ataque de cualquier tipo pues entre mayor tipo de ataques se realice mayor será el conocimiento de las medidas de seguridad a implementar.

Ulterior con la penetración se dispone a realizar el acceso al sistema y la escalada de privilegios que permitirá ganar accesos no privilegiados de un usuario esto se realiza descubriendo usuarios y passwords o explotando configuraciones.

Reportar, limpiar y mitigar.

En esta última etapa se presenta los reportes que en el transcurso de las pruebas se ha realizado, pero en caso de que los ataques de penetración evidencien casos críticos se debe notificar de inmediato al administrador de la red y no esperar hasta esta etapa para reportarlo, para que de este modo se tome medidas inmediatas de mitigación.

En los informes que se desglosa un pequeño resumen de la gestión, el alcance que tuvo, las herramientas, scripts, exploits u otro software o hardware que sirva como herramienta de penetración, además se debe detallar las fechas y horas que se realizó

la intrusión en el sistema y todas las salidas que las herramientas obtengan en este caso el análisis de vulnerabilidades podrán incluirse como anexo.

Posterior a ello se realiza la eliminación de todos los ficheros y registros creados en el pentesting para que no exista huella de lo realizado y de esta manera se prevenga posibles ataques.

Para finalizar se mencionan pequeñas recomendaciones de mitigación de las vulnerabilidades encontradas en sistema y toda información que se considere de relevancia adjuntarla como anexo, esto quiere decir que para futuras pruebas de penetración podrán tomar como guía para saber los sistemas que se auditaron y se comprobará si se mitigaron las vulnerabilidades o no.

Para culminar con la explicación del marco metodológico se presenta en la figura 1 en donde resume a breves rasgos los pasos que se debe seguir para realizar un pentesting con ISSAF estandarizado y de calidad.

Flujo de trabajo de ISSAF

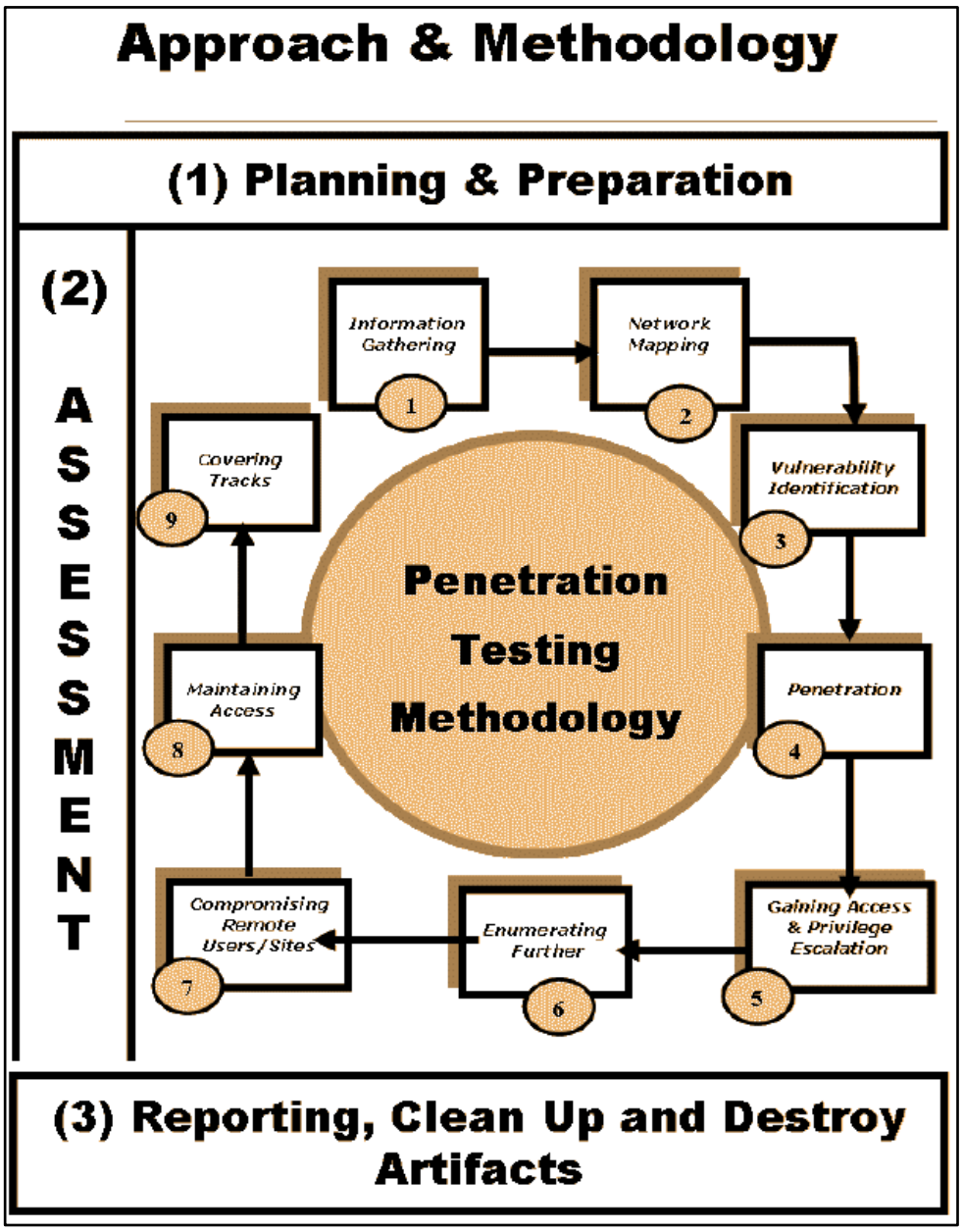


Figura 1 Flujo de metodología ISSAF
Fuente: ISSAF methodology handbook

Capítulo 1

Estado del arte

1.1 Marco referencial o institucional

La Universidad Politécnica Salesiana empieza a tener bajo su dominio un portal web en el año 2000 luego de 6 años de apertura de la universidad, en un contexto histórico llamado “Burbuja de las empresas .com” graficado en la figura 2 donde las empresas y organizaciones creaban sitios web con vagas ideas del mercadeo por internet.

Boom de los dominios .com en los años 2000

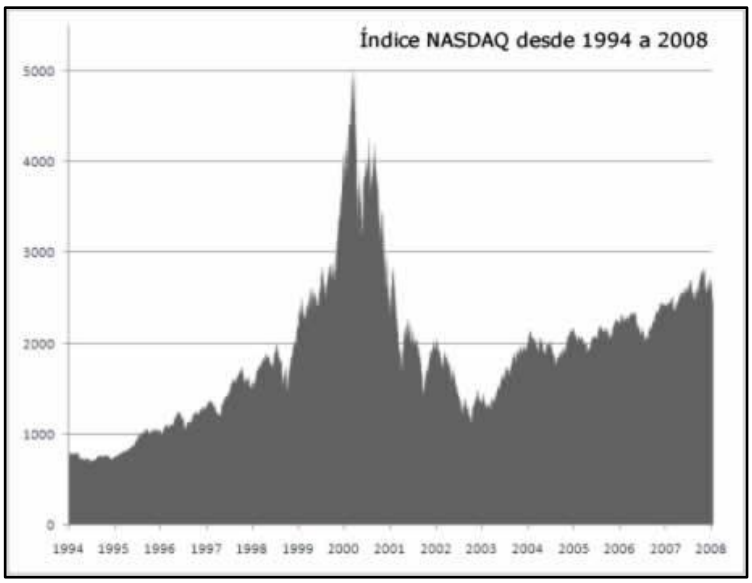


Figura 2 Indicador de acciones .com por Nasdaq
Fuente: Nasdaq.com

Entonces para la época este tipo de sitios eran muy poco explotados por el contexto histórico y cultural, pero para el equipo de desarrollo conformado por: P.Jaime Padilla sdb en la parte del diseño y Fabián Muñoz Jurado en la parte de la aplicación conformaron lo que sería el primer sitio web de la Universidad Politécnica Salesiana la cual tenía una interface como se muestra a continuación.

Primera página de la Universidad Politécnica Salesiana

Figura 3 Primera página web de la Universidad Politécnica Salesiana
Fuente: web.archive.org – ups.edu.ec

Como se puede visualizar en la figura 3 la página era muy rudimentaria, pero para aquella época significaba un gran esfuerzo económico y tecnológico, en ese entonces no se tomaba en cuenta la seguridad de la información pues se pudo recabar que la pagina no contaba con un servicio TLS o SSL el cual se hizo famoso a partir de mayo del 2000 con muchas dudas de su implementación por parte de los administradores alrededor del mundo.

Al pasar el tiempo la Universidad Politécnica Salesiana (UPS) logró suplir varias necesidades de los usuarios a través de su portal web donde actualmente se puede observar una interface amigable e intuitiva para el usuario como se muestra en la figura 4.

Página principal actual de la Universidad Politécnica Salesiana

Figura 4 Portal principal actual de la UPS 2019
Fuente: www.ups.edu.ec

1.2 Marco teórico

1.2.1 Fundamentos de seguridad informática

Seguridad Informática

La seguridad informática es una técnica que permite proteger ante daños potenciales los activos intangibles como la información, software y los activos tangibles como hardware y sistemas de red ante riesgos antrópicos y naturales.

Según Álbaro Gómez en su libro Seguridad Informática la describe como:

"Cualquier medida que impida la ejecución de operaciones no autorizadas sobre un sistema o red informática cuyos efectos pueden conllevar a daños sobre la información" (Gómez, 2010)

Hacking ético.

Un hacking ético pretende demostrar el status de seguridad que posee un sistema para que de este modo se pueda parchar vulnerabilidades encontradas en los mismos y reforzar la seguridad.

"Definimos a un hacker ético como una persona que tiene curiosidad y conocimientos informáticos que le permitan analizar un sistema para la protección del mismo." (Pérez, 2014)

El objetivo de un hacking ético es realizar una serie de pruebas de seguridad acordadas con el cliente, la empresa u organización con el fin de averiguar los fallos de seguridad de uno o varios sistemas en el ámbito que estas puedan perjudicar a la entidad y su producción.

White Hat (Hackers de sombrero blanco).

También llamados "Hackers éticos", son personas que realizan una intervención en los sistemas con el fin de descubrir amenazas y vulnerabilidades que posteriormente serán notificadas y parchadas para reforzar la seguridad.

Evitan a toda costa que terceros puedan afectar a la disponibilidad, confiabilidad e integridad del sistema y protegen la información de los usuarios internos y externos de la red.

Grey Hat (Hackers de sombrero gris).

Este tipo de personas busca su beneficio personal y admiración por algún tipo de comunidades es por este motivo que a veces realizan buenos trabajos con ética y profesionalismo, pero en ocasiones realizan actos ilícitos.

Según expertos como Shon Harris, Allen Harper creadores del libro “Gray Hat Hacking: the ethical hackers handbook” afirman que:

“El hacker gris no tiene intenciones de revelar sus secretos ni métodos sino de cobrar por reparar los fallos encontrados penetrando a los sistemas sin permiso y quizás creando nuevos fallos que a futuro le servían para cobrar de nuevo” (Harris, y otros, 2011)

Black Hat (Hackers de sombrero negro).

El black hat o llamado igualmente “Cracker” es una persona que solo busca hacer daño o perjudicar a un sistema o entidad para el beneficio propio, busca debilidades para ser explotadas y robar información confidencial además puede crear procesos falsos, encriptar dispositivos, causar desabastecimiento del servicio informático, extorsionar e incluso crear malware que puede infectar varios sistemas en red.

“Es una forma elegante para describir sus malas intenciones. Son usuarios dotados, pero no éticos que están motivados por sentimientos de poder y venganza.” (Benitez, 2016)

En la figura 5 se muestra un resumen que comprende los tipos de hackers anteriormente descritos.

Tipos de hackers

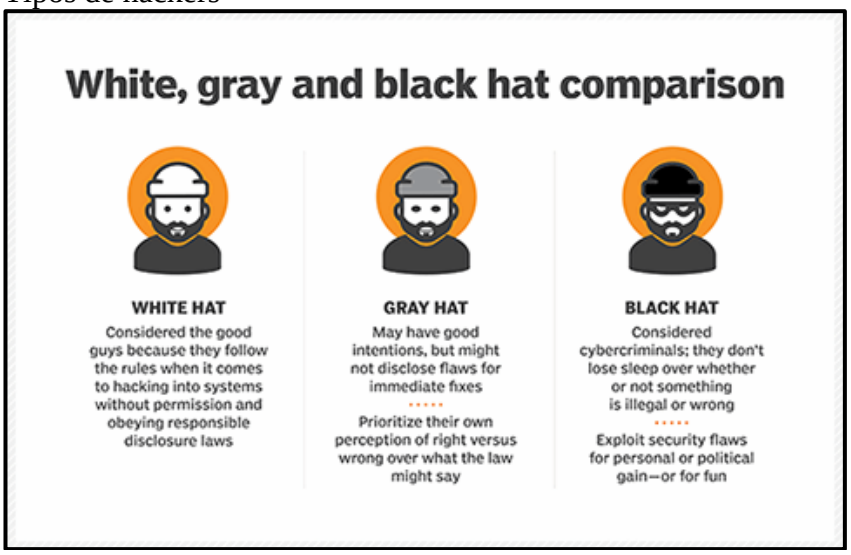


Figura 5 Tipos de hackers
Fuente: https://i.pinimg.com/originals/c6/59/ae/c659aec1792e564f5876b0c825c29b29.png

1.2.2 Tipos de ataques

Este apartado detalla los tipos de ataques más utilizados y por ende son más efectivos ante vulnerabilidades.

Ataques de reconocimiento y escaneo de sistemas.

Este tipo de ataques tiene como objetivo el delimitar y conocer lo que se va a atacar ya que al reconocer la topología de la red se puede diseñar los planes de seguridad, encontrar fallos arquitectónicos y posibles brechas que puedan conllevar a un ataque futuro.

Existen muchas herramientas para realizar este tipo de ataques una de las más reconocidas es “Nmap” con la cual se puede generar informes de: puertos abiertos, topología, resolución de DNS, sistemas operativos, detección de dispositivos de seguridad, trace del ruteo, entre otros.

En este tipo de ataques es casual ver que el atacante busca información no solamente interna sino externa como números de teléfonos, directorios de email, geolocalización de los servidores que ayudan en la comprensión del entorno al que se enfrentan.

Reconocimiento WhoIs.

Es un protocolo para realizar peticiones y respuestas a través de TCP para de esta manera determinar: a quién pertenece un dominio, ubicación geográfica del dominio, números de teléfono, direcciones, etc.

Este protocolo viene dado por la ICANN en el RFC3912, lo que más afecta a este protocolo es la falta de seguridad y además los encargados de este protocolo no piensan en arreglarlo sino migrar este a otro protocolo más complejo. (RFC3912, 2004)

Ataques de inyección SQL.

Este tipo de ataques permiten a los atacantes falsificar la identidad, anular procesos, alterar datos, causar problemas o modificar el contenido de una base de datos a través de consultas que se realizan en campos mal validados de un sistema. (OWASP, 2016)

Cuando se lleva a cabo este tipo de prácticas se puede considerar como de alto riesgo para los sistemas, pues el acceso que puede llegar a tener es absoluto causando mucho daño al momento de ingresar los distintos scripts que dan apertura a generar un CRUD sobre la base de datos del sistema si no es controlado eficientemente.

DoS y DDoS.

El ataque de denegación de servicio no es más que enviar demasiadas solicitudes al servidor para que este no pueda responderlas y se desate un colapso estos pueden ser generados con una topología, en la figura 6 muestro como se procede con dicho ataque.

Ante esta cantidad de tipos de ataques se comprende de mejor forma que los crackers o black hat pueden dañar un sistema por múltiples opciones. Cabe recalcar que la información personal o institucional es el activo más valioso pues puede usarse de diversas formas para crear conocimiento entonces este tipo de ataques busca proteger la integridad, la disponibilidad y la confidencialidad de la información y equipos (Vieites).

Ataque de DOS y DDoS

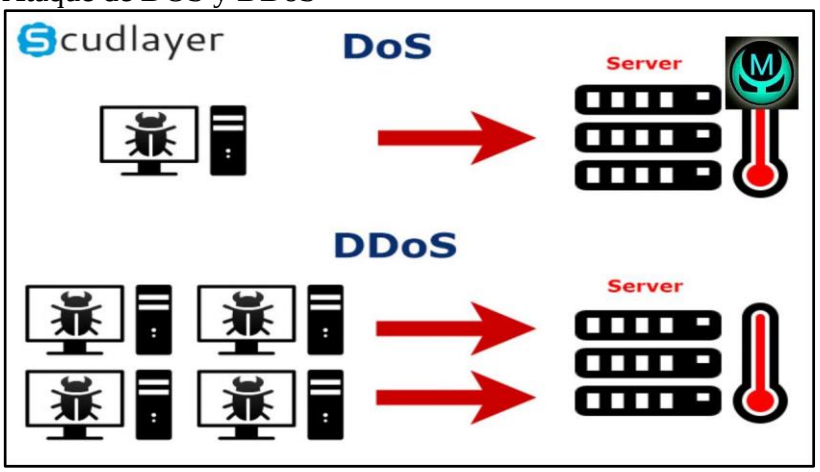


Figura 6 Operación de ataques de DOS y DDOS
Fuente: https://www.slideshare.net/johanpardosanchez/sustentacion-81832364

1.2.3 Amenaza, vulnerabilidades y riesgos

La amenaza, vulnerabilidad y riesgos son conceptos que están entre enlazados pues en el contexto de seguridad informática y de la información son una referencia muy valiosa pues actúan sobre el activo más valioso para las personas o una organización que es la información.

Vulnerabilidad informática.

Una vulnerabilidad es una probabilidad que una amenaza perpetre, entonces, se puede considerar como la capacidad de reacción ante la presencia de un factor que pueda posibilitar una amenaza o un ataque. Se es vulnerable a cualquier evento, sin importar su naturaleza interna o externa que pueda afectar los activos informáticos, los datos o

la información ante la posibilidad de la presencia de un ataque deliberado o no, por parte del personal interno o externo a la organización.

"De acuerdo a una de las clasificaciones, los tipos de vulnerabilidades que pueden presentarse a nivel informático son: Vulnerabilidad física, Vulnerabilidad natural, Vulnerabilidades del hardware, Vulnerabilidades del software, Vulnerabilidad de medios o dispositivos, Vulnerabilidad de las comunicaciones, Vulnerabilidad Humana." (Coloma Baños, 2019)

Amenazas informáticas.

Las amenazas informáticas se relacionan con la posibilidad que suceda algún evento adverso el cual se puede presentar en cualquier momento sin previo aviso, donde existe un daño material o inmaterial sobre los activos tangibles o intangibles de la información. Las amenazas son consideradas como los ataques cometidos por personas internas o externas, que pueden ocasionar daños a la infraestructura tecnológica, a los sistemas de información o a la misma información que circula en la organización. Las amenazas pueden presentarse por acciones criminales en las que intervienen seres humanos violando las normas y las leyes, o sucesos de orden físico por eventos naturales que se puede presentar, o aquellos eventos en los que el ser humano propicia las condiciones para determinar un hecho físico, o por negligencia que son las omisiones, decisiones o acciones que pueden presentar algunas personas por desconocimiento, falta de capacitación y/o abuso de autoridad. Las amenazas a los sistemas de información están latentes cada que se interactúa con los ellos, al utilizar dispositivos de almacenamiento externos, al ingresar a sitios web, por la inconformidad de empleados insatisfechos dentro de la misma organización. De

acuerdo a lo anterior las amenazas pueden ser de varios tipos, entre ellas tenemos las amenazas por intercepción, modificación, interrupción o proliferación.

(Álvarez Marañón & Pérez García, 2004)

Riesgos informáticos.

Un riesgo informático es una probabilidad de que una brecha de seguridad se ataque y esta pueda hacer daño al sistema.

"Los riesgos informáticos son problemas, que pueden afectar a los sistemas de información o a los equipos informáticos. Si no se tienen las medidas adecuadas para precautelar los datos y la información, dichos riesgos se pueden presentar por las vulnerabilidades y amenazas en un futuro, por lo tanto, los riesgos se pueden clasificar en: Riesgos de integridad, Riesgos de acceso, Riesgos de relación, Riesgos de utilidad, Riesgo de infraestructura" (Solarte, 2015)

Evaluación de riesgos

En un pentesting la evaluación de riesgos es una de las características más importantes del mismo, pues al valorar cada riesgo también se sabe el impacto que este podría tener ante la explotación del mismo.

Cuando se evalúan los riesgos de un sistema también se debe hacer una evaluación económica en caso de una explotación, ya que de esa manera se evidencia el coste que este generaría.

Se debe crear cálculos cualitativos y cuantitativos para estimar factores de impacto y con ello también calcular la probabilidad que estos ocurran.

"Conocer el riesgo a los que están sometidos los activos es imprescindible para poder gestionarlos, y por ello han surgido una multitud de guías informales, aproximaciones metódicas y herramientas de soporte las cuales buscan objetivar el análisis para saber cuán seguros (o inseguros) están dichos activos y no llamarse a engaño." (Magenit, 2006)

1.2.4 Herramientas a utilizar

Kali Linux

Sistema operativo gratuito orientado a la seguridad informática y a las pruebas de penetración, cuenta con diversas herramientas clasificadas para que el pentester tenga a su disponibilidad toda una suite en donde realizar toda clase de testeo interno y externo.

Metasploit

Programa open source que sirve para realizar pentesting con exploits y otros auxiliares que permiten explotar una vulnerabilidad remotamente.

Google Dorks.

Se conoce que un dork es una simple combinación de operadores lógicos y sintácticos que representan una búsqueda más avanzada y detallada en un buscador actuando directamente en el algoritmo de búsqueda.

Escáner Vega.

Es un programa desarrollado en Java que a través de sus módulos realza una auditoría a un dominio para determinar sus vulnerabilidades, además presenta datos vinculados para una posible mejora en el escaneo además cuenta también con un request sender.

Escáner Shodan.

Permite a través de su web escanear de forma directa IP´s y dispositivos para conocer los CVE que estos poseen y de esta manera conocer si tienen vulnerabilidades, pero este escaneo no categoriza las vulnerabilidades solo las muestra.

Escáner Nessus.

Ofrece un software de paga, fácil de utilizar ya que con su interface gráfica es muy predecible al realizar búsquedas de vulnerabilidades con diferentes módulos que su demonio analiza, en este caso se centrará en vulnerabilidades Web ofreciendo los CVE o el nombre de las vulnerabilidades obtenidas.

Escáner OpenVas.

Este escáner de vulnerabilidades ayuda de gran manera pues es gratuito, tiene una interface web amigable e intuitiva y se puede acceder a múltiples tipos de escaneos dependiendo de las necesidades del pentester, además ofrece reportes personalizados para todos los escaneos realizados.

1.3 Trabajos anteriores

El pentesting o ethical hacking no es una práctica nueva pues ya se viene realizando pruebas para reforzar la seguridad de los sistemas y equipos informáticos, ya que desde el inicio de la computación han existido personas que han vulnerado fallos de seguridad con algún fin.

En las instituciones educativas también se realizan estas buenas prácticas de seguridad ya que la información ha llevado un proceso de transformación a tal punto que ahora se realizan la mayoría de trámites administrativos o académicas a través de portales

web y estos deben estar protegidos para evitar la manipulación sensible de ellos mismos.

Por lo general hacer un hacking ético a una institución es un tema confidencial por ello se debe tomar muy en cuenta cómo serán publicados los resultados del mismo para ello se deben basar en políticas internas y externas para publicar los fallos encontrados para que no se puedan aprovechar de esta información personas externas con malas intenciones.

Como se puede observar en la tabla 2 se analiza el aporte que tiene cada proyecto elegido, pero cabe recalcar que estos no son los únicos trabajos en los que se basa este proyecto técnico por el contrario a medida que este avanza busca nuevas fuentes de información que se adapten a la metodología escogida.

Tabla 2 Trabajos referenciales para el desarrollo de este proyecto

Tipo	Autor	Tema	Objetivo del documento	Desarrollo	Conclusiones	Aporte al proyecto
Tesis	MARCO VINICIO BRAVO SÁNCHEZ DAVID ALBERTO 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	Analizar las amenazas, riesgos y vulnerabilidades del portal web del Colegio Católico José Engling	Realiza un pentesting web explotándolo con sistemas open source en su mayoría y pone a prueba un sistema desarrollado para monitorear incidencias reportadas	Se concluye que no es factible mitigar en un 100% los ataques de recolección de información o footprinting hacia el portal web, debido a la demanda obligatoria de información institucional que se publica en internet.	Genera una base de conocimiento que servirá para la utilización de algunas herramientas de footprinting
Tesis	ANDRÉS MARCELO VALLEJO ARGUELLO	ANÁLISIS DE VULNERABILIDADES EN APLICATIVOS WEB E INFRAESTRUCTURA PARA UNA INSTITUCIÓN EDUCATIVA PRIVADA DE LA CIUDAD DE QUITO	Diagnosticar las potenciales vulnerabilidades de seguridad del área tecnológica de una institución educativa privada de la ciudad de Quito y proponer las medidas correctivas que deben implementarse.	Crea un pentesting interno que utiliza varios servidores y aplicaciones en las que ataca de distintas maneras y en algunos ataques tiene éxito y en otros solo encuentra las vulnerabilidades del sistema.	Para el inicio de un ethical hacking se necesitaron los permisos de la organización, de esa manera se logró tener acceso libre a los sistemas informáticos tanto de infraestructura como de los aplicativos.	Fija tendencias para no sobrepasar pues en mi proyecto se lo realiza como pentesting externo
Tesis	JAVIER ALEJADNRO SALINAS SÁNCHEZ	DISEÑO Y CONSTRUCCIÓN DE UNA RED IP VIRTUALIZADA PARA LA APLICACIÓN DE HACKING ÉTICO	Diseñar y construir un modelo de la red IP virtualizada para la aplicación de hacking ético.	Crea un laboratorio de pruebas donde aplica varias técnicas de hacking ético y genera varios ataques de distinto tipo para algunos servidores.	La red IP virtualizada fue posible construirla y diseñara dentro del equipo anfitrión, que estuvo conformado por equipos virtuales, la cual permitió obtener un desempeño óptico en las distintas tareas que pudieron realizar hacia los servidores de prueba en servicios Web, con Apache, Correo con Postfix y bases de datos MySQL.	Modela una manera de realizar ataques a sistemas con servidor web Apache que en algunos portales de la Universidad Politécnica Salesiana lo utiliza.
Trabajo para examen complexivo	DANNY OMAR PINOS SOLANO	ANÁLISIS DE VULNERABILIDADES Y ACCIONES CORRECTIVAS SOBRE UN SISTEMA WEB	Realizar un análisis de vulnerabilidades sobre el sistema de ventas y elaborar un plan de acciones correctivas sobre las principales encontradas.	Implementar un plan de acciones correctivas sobre las vulnerabilidades encontradas y especialmente sobre aquellas que se encuentren en el top 10 de OWASP 2013 para posterior generar políticas de seguridad orientadas a un sistema web.	Es vital que las organizaciones hagan conciencia sobre la importancia de la seguridad de la información y no menospreciar el valor que este tiene, un análisis de vulnerabilidades de forma continua puede prevenir desastres informáticos y pérdidas	Crea una visión de recomendaciones para el uso de políticas en servicios web las cuales puedo usar para la propuesta de mitigación de las vulnerabilidades encontradas
Tesis	JAVIER SÁNCHEZ GONZÁLES	CIBERSEGURIDAD: MECANISMOS DE ATAQUE Y DEFENSA MÁS EXTENDIDOS	Estudiar los principales ataques a servidores web, según se recoge en los informes de las grandes marcas de antivirus y malware tales como Panda, Symantec y OWASP	Pone a prueba un entorno de pruebas llamado DVWA el cual a través de sus niveles de seguridad evidencia las fallas que estos pueden poseer.	En la aplicación DVWA, a través de sus diferentes niveles de seguridad se comprueba como un código mal estructurado y sin la suficiente validación de los datos de entrada de los usuarios puede poner en riesgo no solo el propio sitio web sino todo el sistema que está alojado.	Presenta casos prácticos a los cuales se enfrenta a través de técnicas de pentesting diseñadas exclusivamente para el laboratorio de pruebas DVWA pero en mi caso lo pondré en evidencia en sitios de la UPS

Nota: Tabla donde se evidencian los trabajos que se a tomado como referentes para el desarrollo de este proyecto técnico.

Capítulo 2

Implementación del framework

2.1 Introducción a la búsqueda de información.

A partir de este capítulo se pretende introducir ya a las técnicas enumeradas por la metodología ISSAF la cual se ha elegido para que tenga el impacto que se desea obtener con este pentesting.

En primer lugar, se hace un reconocimiento del entorno en donde se llevará a cabo la auditoría para identificar las opciones de los distintos portales que están dentro del dominio, en donde se procede a navegar como un usuario externo a la organización para detectar posibles objetivos o URL´s a auditar.

2.2 Sitios a auditar

Revisando el portal web de la Universidad Politécnica Salesiana se analiza los diferentes servicios que ofrece y los campos de entradas de los webs services y se determina por experiencia adquirida en trabajos anteriores de pentesting que son 7 portales a auditar los cuales se detallan en la tabla 3.

Tabla 3 Portales web a realizar pentesting

Target 1		
URL	https://www.ups.edu.ec/	**Expectativas**
Servicio	Dominio principal de la UPS	Al ser el dominio principal de la UPS se busca encontrar vulnerabilidades de TLS y amenazas que comprometan bajo dorks la información de sus usuarios
Target 2		
URL	https://cas.ups.edu.ec/	**Expectativas**
Servicio	Servicio de autenticación	Se busca obtener accesos mediante ataques de fuerza bruta y determinar el tratamiento que este tipo de ataques tiene al servidor
Target 3		
URL	https://servicesapp.ups.edu.ec/accounts/	**Expectativas**
Servicio	Servicio de recuperación de contraseñas	En este objetivo se busca obtener información de los usuarios finales específicamente emails personales
Target 4		
URL	https://appwfp.ups.edu.ec/ins-pub/	**Expectativas**
Servicio	Portal de inscripciones	Se busca realizar ataques de SQL injection o de dorks que nos permitan determinar la información básica de los usuarios finales
Target 5		
URL	https://appwfp.ups.edu.ec/foc-webins/index.xhtml?prog=121	**Expectativas**
Servicio	Sistema de eventos	Busca crear un ataque para llenar con datos erróneos los portales de eventos
Target 6		
URL	virtual.ups.edu.ec	**Expectativas**
Servicio	AVAC	Este target entra en juego en conjunto con el target 2 y dependiendo de este se realiza búsquedas de vulnerabilidades al motor de aula virtual.
Target 7		
URL	https:/www.ups.edu.ec/web/guest/consultas-comprobantes	**Expectativas**
Servicio	Solicitud de comprobantes	Se busca amenazas de SQL injection y ver cómo se comporta ante la respuesta masiva de datos

Nota: Tabla donde se señala la información básica de los sitios a auditar.

Como se puede observar en la tabla 3 se detalla el servicio, URL y una breve expectativa de ataque.

2.3 Planificación

En este proyecto los tiempos que se estiman para realizar el pentesting es 2 meses y estos que sirvan para buscar las vulnerabilidades, clasificarlas, explotarlas y generar un reporte final en donde se genere una mitigación posible para las vulnerabilidades.

Para culminar con la planificación se debe aclarar que los documentos e información encontrada que sean sencibles para los usuarios no se utilizará en mi beneficio ni con fines no éticos.

2.4 Métodos de búsqueda y recolección de la información

Se necesita vincular las necesidades más críticas del negocio con los sitios web investigados para el pentesting donde se ha determinado los objetivos a auditar en la tabla 3 entonces, se procede con la búsqueda de información y se realiza los WhoIs a los dominios, pero para este tipo de búsquedas se necesita empezar con la lista de IP´s que se ha determinado de los objetivos entonces se realiza bajo el comando “nslookup” como se detalla en la figura 7.

Cabe recalcar que la búsqueda “nslookup” se realiza con el servidor DNS de “Cloudflare” que pertenece a la IP 1.1.1.1, este permite observar las direcciones marcadas, que son nuestras IP objetivo.

Ejecución de nslookup en los sitios a auditar

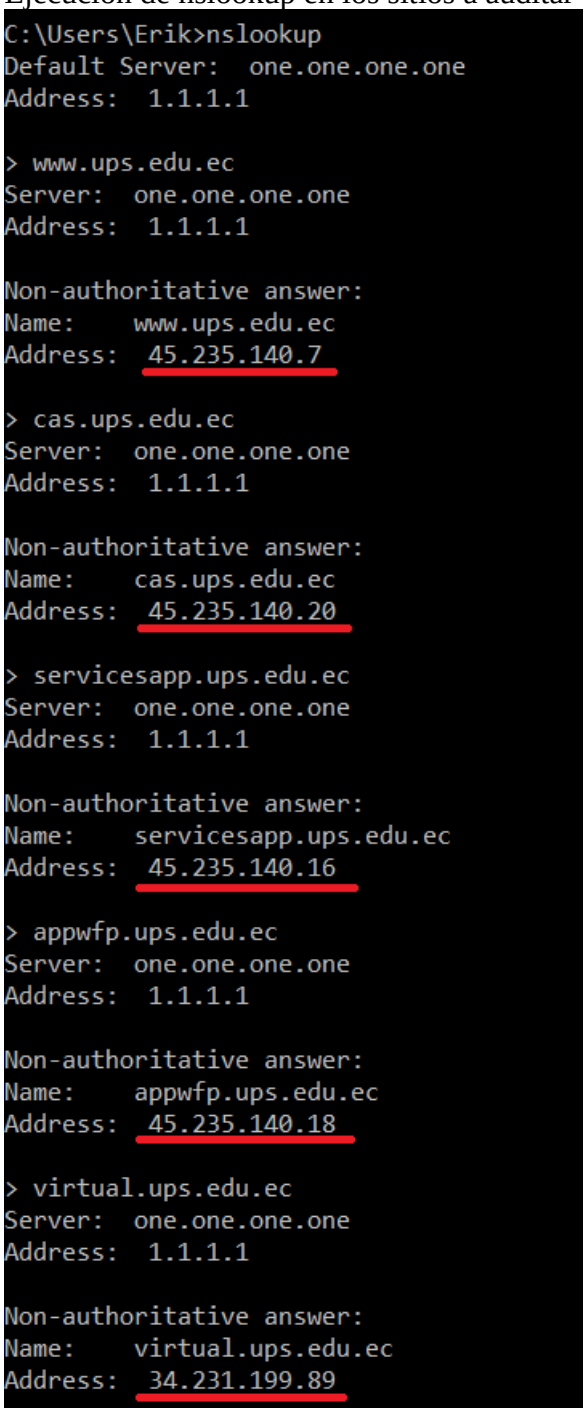


Figura 7 Comando nslookup a los dominios objetivos
Fuente: Generado por el autor en cmd

2.3.1 Búsqueda de información mediante protocolo WhoIs.

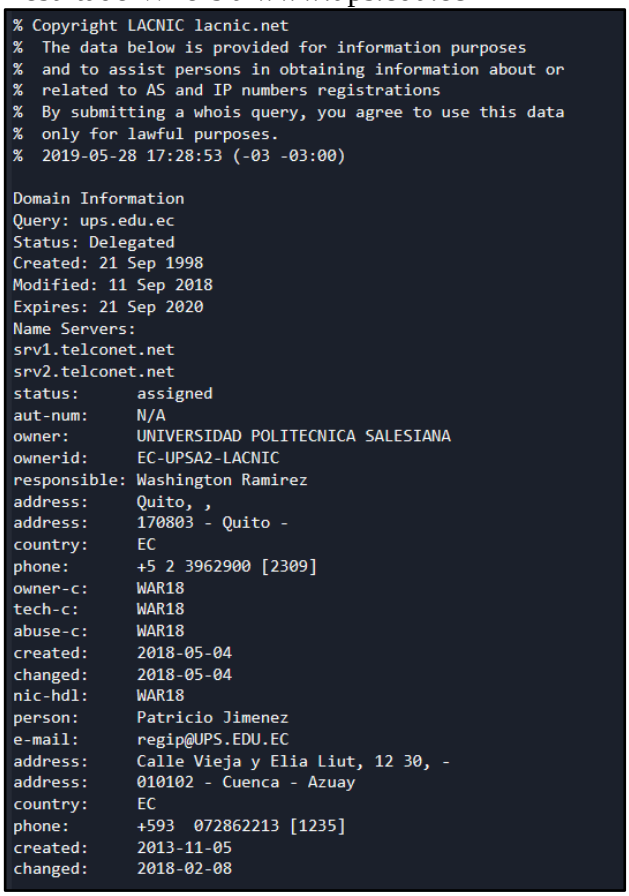
Para buscar información acerca de un dominio uno de los métodos más fáciles de emplear es la búsqueda WhoIs, entonces a continuación se realizará este tipo de búsqueda en todos los dominios mencionados en la tabla 2 en el punto 2.1.

Búsqueda al protocolo WhoIs en portal ww.ups.edu.ec

Para los dominios ups.edu.ec se necesita evidenciar a los responsables del sitio, la dirección, el correo con el cual fue registrado el portal, el número de teléfono entre otros datos que nos permitan conocer más a fondo quién y donde administra el sitio.

Como primera opción para realizar el WhoIs se usa el comando whois en el sistema operativo Kali Linux en donde ya viene pre instalada esta herramienta y permite obtener algunos detalles, pero en caso que no se haya instalado se debe ejecutar el comando “sudo apt install whois” y tras su instalación se puede realizar las consultas con el comando “whois dominio/ip” en este caso se utiliza de la siguiente forma: `$whois www.ups.edu.ec` y su resultado se muestra en la figura 8.

Resultado WhoIs a www.ups.edu.ec



```
% Copyright LACNIC lacnic.net
%  The data below is provided for information purposes
%  and to assist persons in obtaining information about or
%  related to AS and IP numbers registrations
%  By submitting a whois query, you agree to use this data
%  only for lawful purposes.
%  2019-05-28 17:28:53 (-03 -03:00)

Domain Information
Query: ups.edu.ec
Status: Delegated
Created: 21 Sep 1998
Modified: 11 Sep 2018
Expires: 21 Sep 2020
Name Servers:
srv1.telconet.net
srv2.telconet.net
status:      assigned
aut-num:     N/A
owner:       UNIVERSIDAD POLITECNICA SALESIANA
ownerid:     EC-UPSA2-LACNIC
responsible: Washington Ramirez
address:     Quito, ,
address:     170803 - Quito -
country:     EC
phone:       +5 2 3962900 [2309]
owner-c:     WAR18
tech-c:      WAR18
abuse-c:     WAR18
created:     2018-05-04
changed:     2018-05-04
nic-hdl:     WAR18
person:      Patricio Jimenez
e-mail:      regip@UPS.EDU.EC
address:     Calle Vieja y Elia Liut, 12 30, -
address:     010102 - Cuenca - Azuay
country:     EC
phone:       +593  072862213 [1235]
created:     2013-11-05
changed:     2018-02-08
```

Figura 8 WhoIs desde Kali Linux a www.ups.edu.ec
Fuente: Erik Parra – Consulta WhoIS

Haciendo una consulta en domaintools.com se pudo obtener el siguiente resultado señalado en la figura 9 en donde se destaca los nombres de los servidores, la dirección IP, el ASN de CEDIA lo que indica que la UPS trabaja en conjunto con la red CEDIA, el tipo de servidor y por último se obtiene que no tienen registrado el servidor DNS al dominio por ello si una persona introduce por primera vez en su navegador la url: ups.edu.ec el servidor DNS no resolverá efectivamente el sitio por el contrario si ingresa www.ups.edu.ec tiene la ventaja de resolverlo sin problemas porque está vinculado a DNS públicos.

Resultado ASN del WhoIS

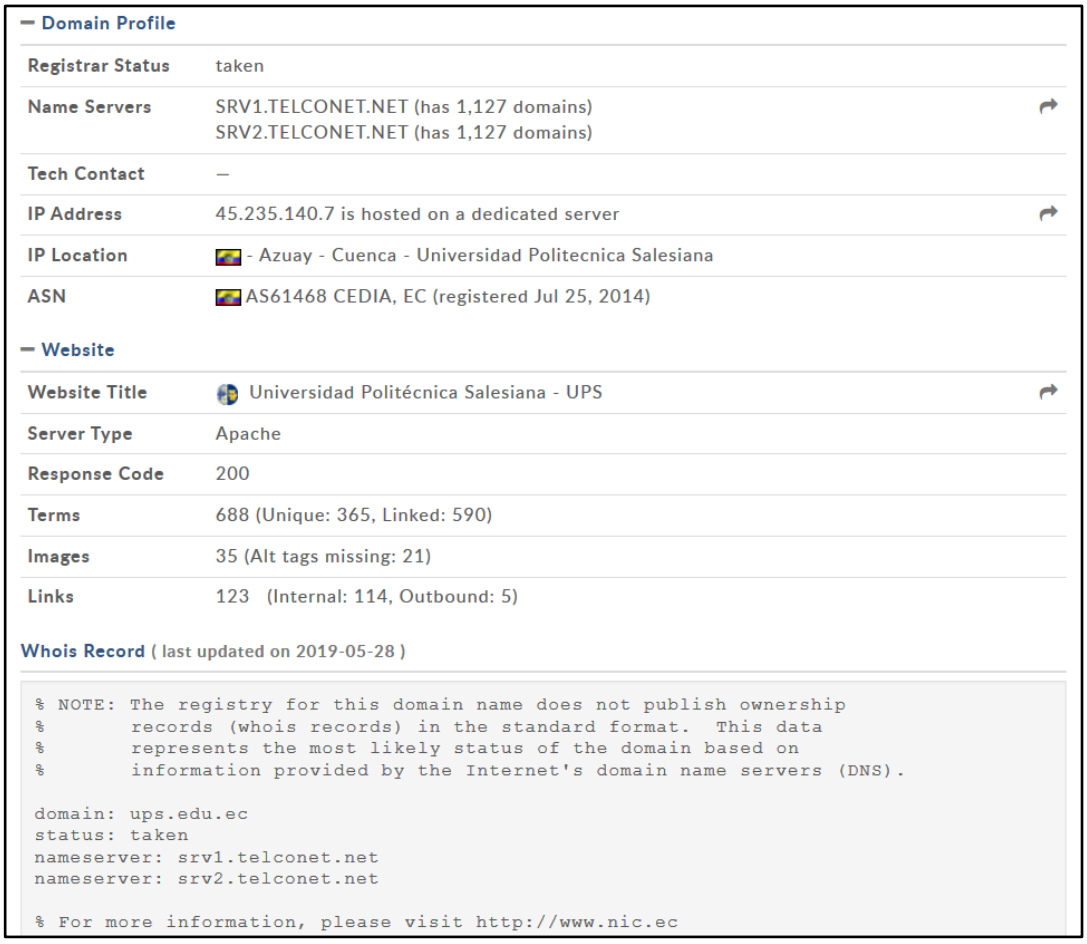


Figura 9 WhoIs domaintools.com a www.ups.edu.ec
Fuente: domaintools.com

A continuación, en la figura 10, se realiza otra prueba en la plataforma CIAME en la cual se pueden realizar las mismas búsquedas, pero se agrega la función de ubicar geo-referencialmente la ubicación del servidor donde corre el portal web www.ups.edu.ec.

Resultado CIAME de www.ups.edu.ec

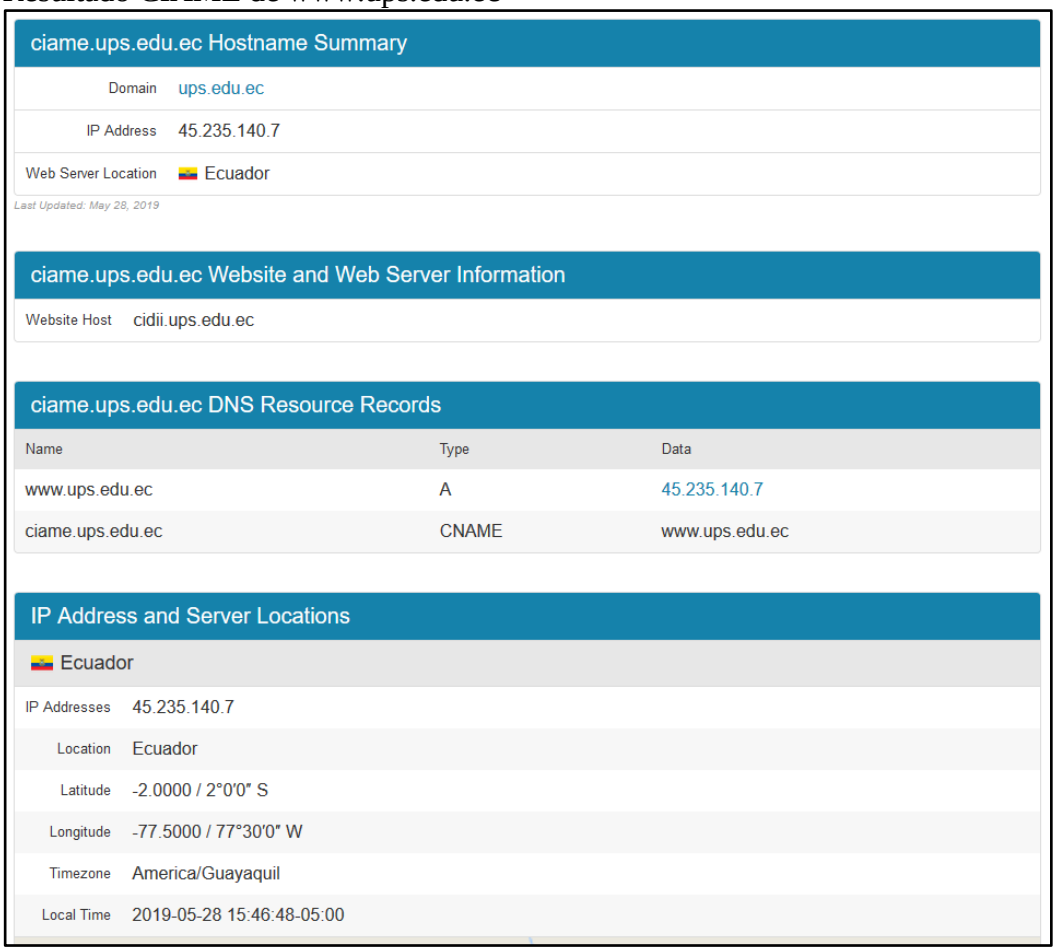


Figura 10 CIAME WhoIs a www.ups.edu.ec
Fuente: ciame.ups.edu.ec.ipaddress.com

Por consiguiente, a la información mostrada anteriormente se puede utilizar la longitud y latitud para mostrar en un mapa la ubicación del servidor el cual se muestra en la figura 11 en ciudad de Cuenca sobre la “Calle larga”.

Geo referencia aproximada del servidor del sitio www.ups.edu.ec

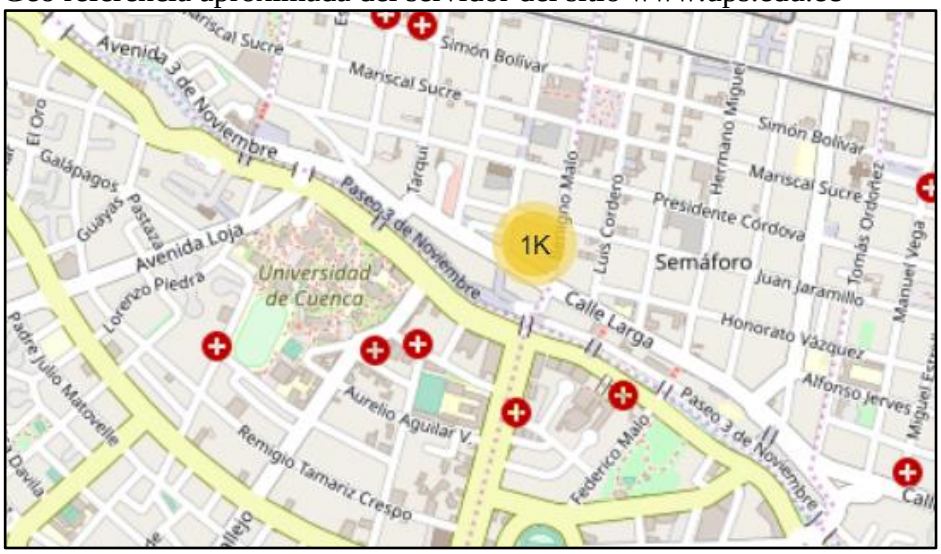


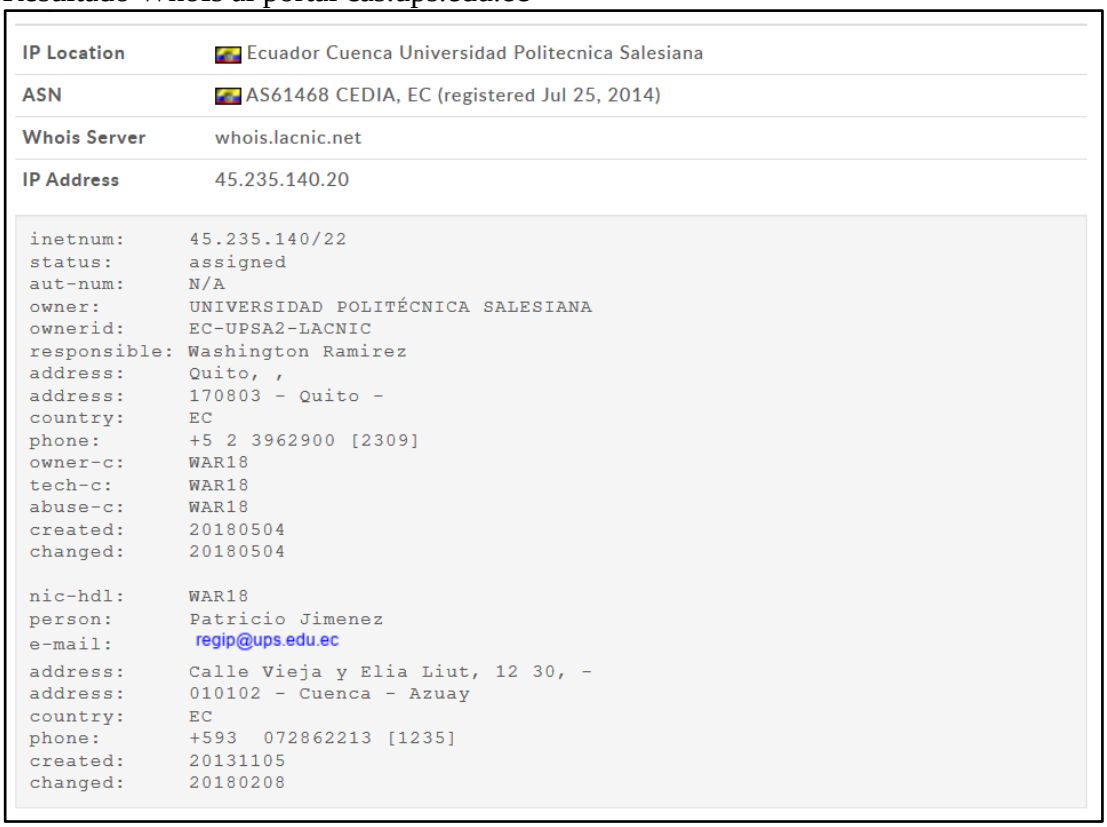
Figura 11 Geolocalización servidor www.ups.edu.ec
Fuente: ciame.ups.edu.ec.ipaddress.com

Búsqueda al protocolo WhoIs en portal cas.ups.edu.ec

La IP 45.235.140.20 pertenece al dominio cas.ups.edu.ec, que es el segundo objetivo a realizar la búsqueda de vulnerabilidades y ya desde esta IP caemos en cuenta que los servidores web estan alojados bajo la IP 45.235.140/22 entonces pueden compartir 1022 host para los diferentes portales que tiene la UPS.

Cuando se realiza con la herramienta web “domain tools” a este host se muestra que Washintong Ramirez y Patricio Jimenez que en los responsables del dominio de la UPS 2 como se muestra en la figura 12.

Resultado WhoIs al portal cas.ups.edu.ec



IP Location	Ecuador Cuenca Universidad Politecnica Salesiana
ASN	AS61468 CEDIA, EC (registered Jul 25, 2014)
Whois Server	whois.lacnic.net
IP Address	45.235.140.20

```
inetnum:      45.235.140/22
status:       assigned
aut-num:      N/A
owner:        UNIVERSIDAD POLITÉCNICA SALESIANA
ownerid:      EC-UPSA2-LACNIC
responsible:  Washington Ramirez
address:      Quito, ,
address:      170803 - Quito -
country:      EC
phone:        +5 2 3962900 [2309]
owner-c:      WAR18
tech-c:       WAR18
abuse-c:      WAR18
created:      20180504
changed:      20180504

nic-hdl:      WAR18
person:       Patricio Jimenez
e-mail:       regip@ups.edu.ec
address:      Calle Vieja y Elia Liut, 12 30, -
address:      010102 - Cuenca - Azuay
country:      EC
phone:        +593  072862213 [1235]
created:      20131105
changed:      20180208
```

Figura 12 WhoIS a cas.ups.edu.ec
Fuente: Generada por el autor en whois.domaintools.com

Se muestra precisión en la dirección y datos sensibles como extensiones telefónicas donde un atacante pueda realizar un ataque de ingeniería social, pero en todo caso los responsables del sitio están capacitados para manejar estas situaciones, pero no están exentos que puedan sorprenderlos con este tipo de ataques.

En la figura 13, se presenta los datos triangulados de la posible ubicación de este servidor web en un mapa georreferenciado el cual brinda la longitud y latitud aproximada.

Geo referencia del portal cas.ups.edu.ec

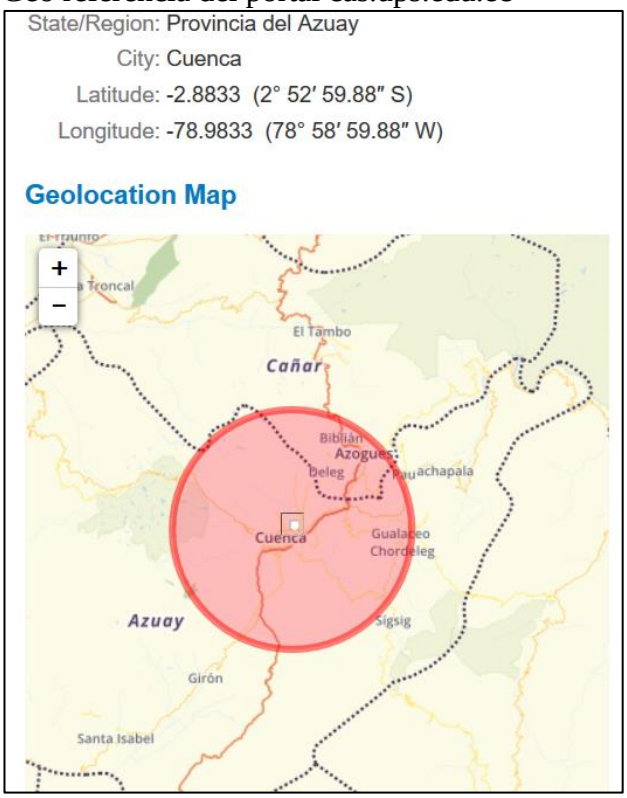


Figura 13 Geolocalización de cas.ups.edu.ec
Fuente: Generados por el autor en Open Street Map

Búsqueda al protocolo WhoIs en portal servicesapp.ups.edu.ec

En la búsqueda de información para el portal de recuperación de contraseñas se evidencia en la figura 14 que la IP está dentro de la misma infraestructura y dentro de una misma subred, los datos como nombres de los propietarios, ubicación, entre otros no varían en lo absoluto, por consecuente es una gran debilidad que tiene este portal, ya que si la IP 45.235.140/22 llagase a tener fallos todos los servicios alojados por parte de la universidad estarían en peligro de no brindar servicio a los usuarios finales, bajando de este modo el nivel de disponibilidad en esta infraestructura, por ello en conjunto con CEDIA se ha realizado respaldos de estos servicios para que estos brinden ayuda en este tipo de siniestros.

Resultado de WhoIs al portal servicesapp.ups.edu.ec

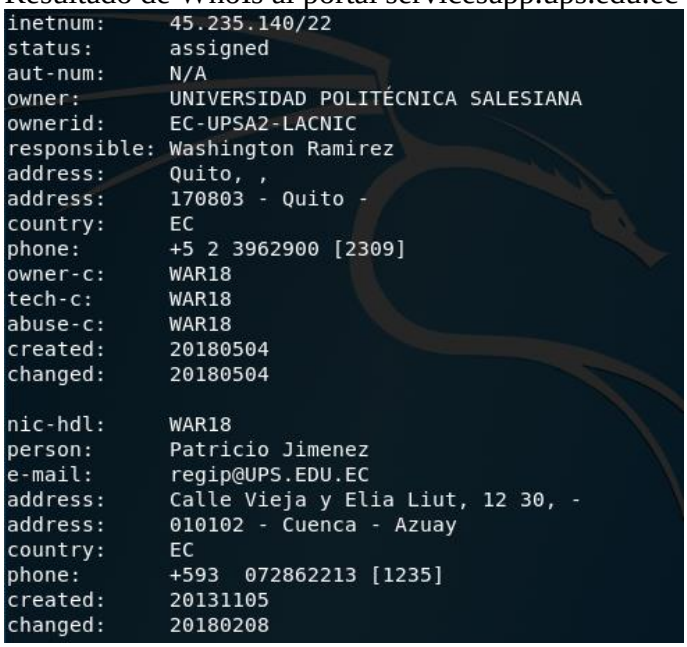
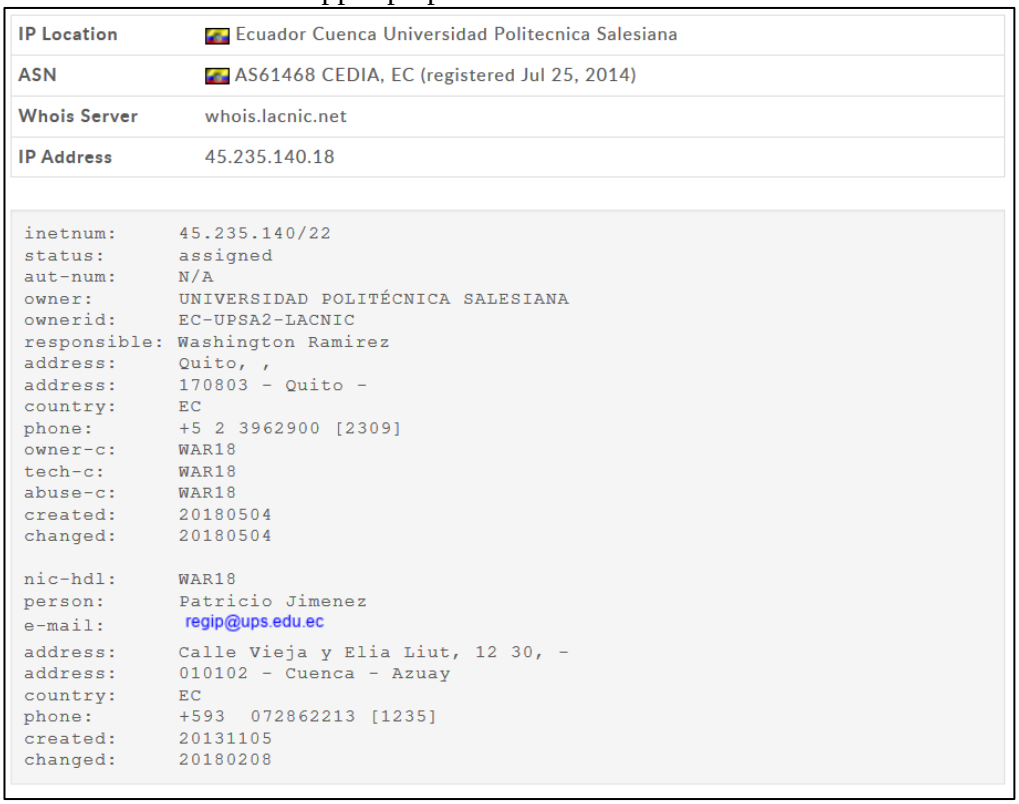


Figura 14 WhoIs a servicesapp.ups.edu.ec
Fuente: Generado por el autor whois Kali Linux

Búsqueda al protocolo WhoIs en portal appwfp.ups.edu.ec.

La búsqueda del WhoIs al servicio de aplicaciones web que se posee la IP 45.235.140.18, se encuentra los mismos datos que las anteriores búsquedas WhoIs entonces, se determina que como todo se encuentra bajo un mismo segmento de red, se recortan los tiempos de respuesta al usuario final y eso en un ambiente universitario es vital, pero cabe recalcar que este tipo de infraestructuras debe tener un sistema de prevención como respuesta a incidentes para mitigar posibles incidentes naturales y antrópicos, el resultado de esta búsqueda se refleja en la figura 15.

Resultado de WhoIs en appwfp.ups.edu.ec



IP Location	Ecuador Cuenca Universidad Politecnica Salesiana
ASN	AS61468 CEDIA, EC (registered Jul 25, 2014)
Whois Server	whois.lacnic.net
IP Address	45.235.140.18

```
inetnum:      45.235.140/22
status:       assigned
aut-num:      N/A
owner:        UNIVERSIDAD POLITÉCNICA SALESIANA
ownerid:      EC-UPSA2-LACNIC
responsible:  Washington Ramirez
address:      Quito, ,
address:      170803 - Quito -
country:      EC
phone:        +5 2 3962900 [2309]
owner-c:      WAR18
tech-c:       WAR18
abuse-c:      WAR18
created:      20180504
changed:      20180504

nic-hdl:      WAR18
person:       Patricio Jimenez
e-mail:       regip@ups.edu.ec
address:      Calle Vieja y Elia Liut, 12 30, -
address:      010102 - Cuenca - Azuay
country:      EC
phone:        +593  072862213 [1235]
created:      20131105
changed:      20180208
```

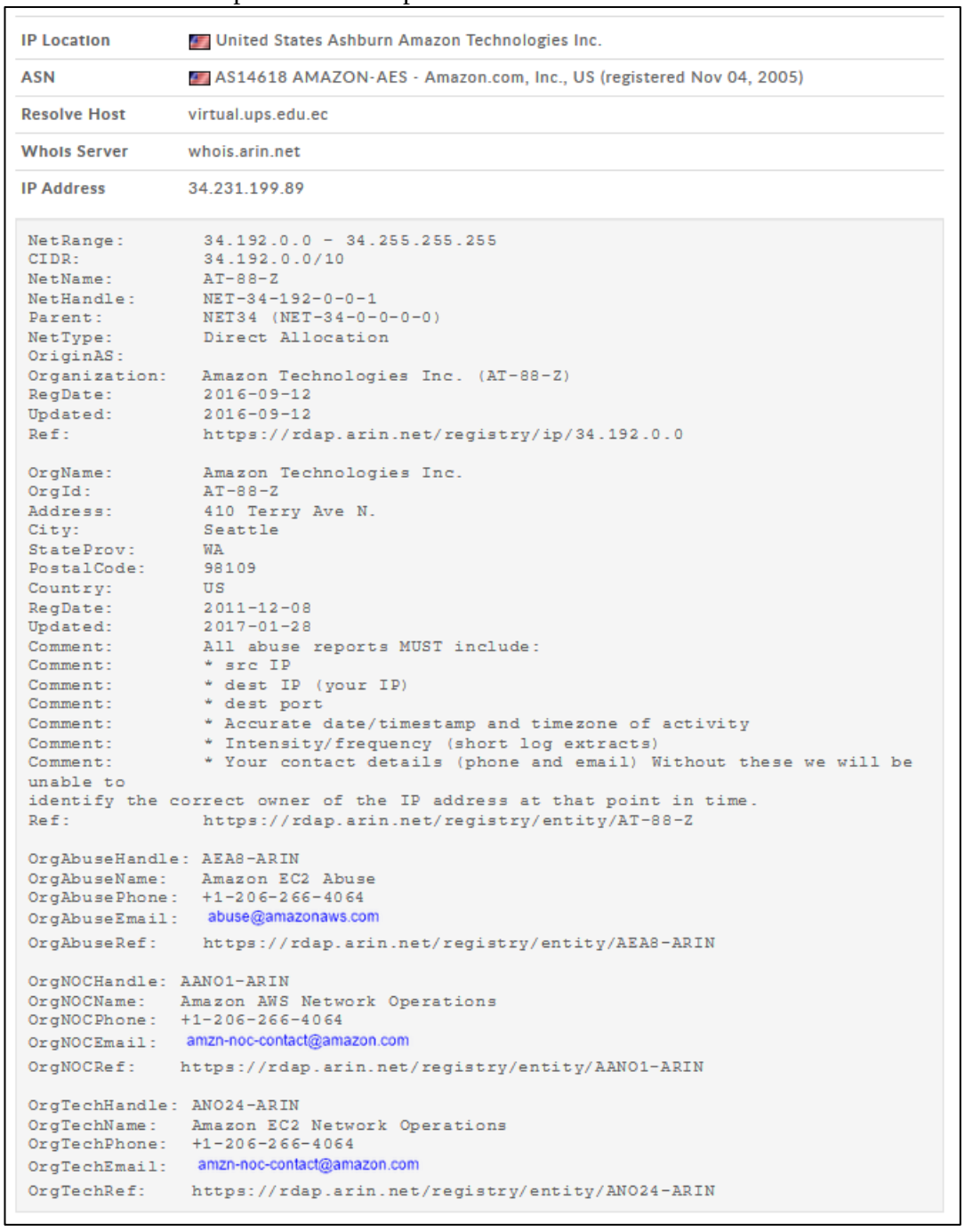
Figura 15 WhoIs a appwfp.ups.edu.ec
Fuente: Generado por el autor en domaintools.com

Búsqueda al protocolo WhoIs en el portal virtual.ups.edu.ec

Para la consulta WhoIs del aula virtual “AVAC” a través de la IP 34.231.199.89 se determina que está alojado en un servidor de Amazon Web Services, y por ende algunos datos de registro estarán ocultos, por ello en la figura 16 no se puede identificar el nombre de los dueños o algún dato que haga referencia a la UPS, pero se puede corroborar el segmento de red con el cual trabaja y que en este caso es: 34.192.0.0/10 donde puede alojar hasta 4194304 host.

Una ventaja de los servidores en cloud es que tiene réplicas, esto garantiza la disponibilidad del servicio, pero al ser utilizado con estas garantías representa gastos extras al momento de rentar un servidor en Amazon Web Services.

Resultado WhoIs al portal virtual.ups.edu.ec



IP Location	United States Ashburn Amazon Technologies Inc.
ASN	AS14618 AMAZON-AES - Amazon.com, Inc., US (registered Nov 04, 2005)
Resolve Host	virtual.ups.edu.ec
Whois Server	whois.arin.net
IP Address	34.231.199.89

```
NetRange:       34.192.0.0 - 34.255.255.255
CIDR:           34.192.0.0/10
NetName:        AT-88-Z
NetHandle:      NET-34-192-0-0-1
Parent:         NET34 (NET-34-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Amazon Technologies Inc. (AT-88-Z)
RegDate:        2016-09-12
Updated:        2016-09-12
Ref:            https://rdap.arin.net/registry/ip/34.192.0.0

OrgName:        Amazon Technologies Inc.
OrgId:          AT-88-Z
Address:        410 Terry Ave N.
City:           Seattle
StateProv:      WA
PostalCode:     98109
Country:        US
RegDate:        2011-12-08
Updated:        2017-01-28
Comment:        All abuse reports MUST include:
Comment:        * src IP
Comment:        * dest IP (your IP)
Comment:        * dest port
Comment:        * Accurate date/timestamp and timezone of activity
Comment:        * Intensity/frequency (short log extracts)
Comment:        * Your contact details (phone and email) Without these we will be
unable to
identify the correct owner of the IP address at that point in time.
Ref:            https://rdap.arin.net/registry/entity/AT-88-Z

OrgAbuseHandle: AEA8-ARIN
OrgAbuseName:   Amazon EC2 Abuse
OrgAbusePhone:  +1-206-266-4064
OrgAbuseEmail:  abuse@amazonaws.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/AEA8-ARIN

OrgNOCHandle: AANO1-ARIN
OrgNOCName:   Amazon AWS Network Operations
OrgNOCPhone:  +1-206-266-4064
OrgNOCEmail:  amzn-noc-contact@amazon.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/AANO1-ARIN

OrgTechHandle: ANO24-ARIN
OrgTechName:   Amazon EC2 Network Operations
OrgTechPhone:  +1-206-266-4064
OrgTechEmail:  amzn-noc-contact@amazon.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ANO24-ARIN
```

Figura 16 WhoIs 34.231.199.89
Fuente: Generado por el autor en domaintools.com

En la figura 17, se presenta la georreferenciación del servidor para el servicio de aula virtual AVAC el cual se encuentra en USA, Ashburn-Virginia. La latitud y longitud encontradas resultan ser muy exactas al momento de hacer el WhoIs inclusive tiene mayor exactitud que el dominio principal de la UPS

Geo referencia del portal virtual.ups.edu.ec

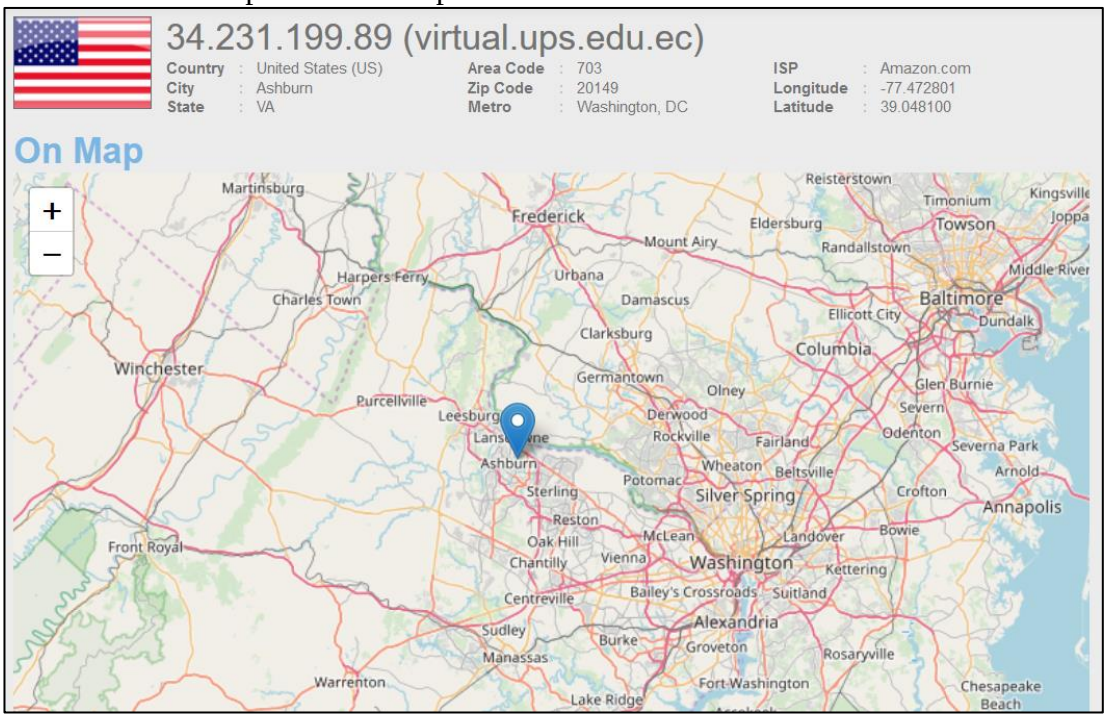


Figura 17 Georreferenciación por IP al servidor 34.231.199.89
Fuente: Generada por el autor en iplocation.net

2.3.2 Búsqueda de información mediante Google dorks

En este caso será Google, en donde se intentará crear dorks para receptar el uso de bases de datos, archivos indexados y no indexados con carga de información útil para el pentesting, vulnerabilidades de accesos, entre otros.

Como primera prueba se realiza una búsqueda con el comando `inurl:"ups.edu.ec" filetype:xml` que se refleja en la figura 18.

Apliación del dork en google

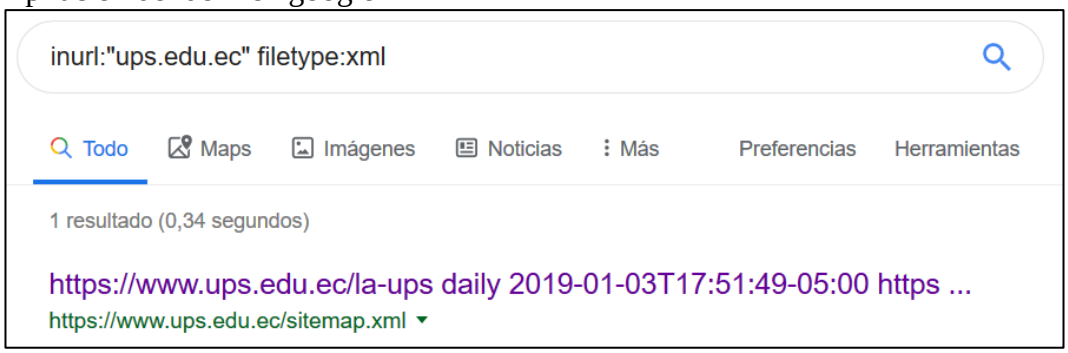


Figura 18 Dork de búsqueda XML del sitio
Fuente: Dork generado por el autor en Google.com

Muestra un mapa del sitio web completo en un XML resultante en la figura 19, el peligro con este documento radica en que al conocer todas las rutas y subdominios un atacante puede conocer los sitios donde puede hacer mayor o menor impacto.

Fragmento del mapa XML de todos los sitios de la Universidad Politécnica Salesiana

Figura 19 Dork de sitio completo en XML
Fuente: Dork generado por el autor en Google.com

Con el siguiente dork: `inurl:ups.edu.ec/documents/ intext:"Texto a buscar",` se obtiene cualquier documento que la UPS ha entregado como: resoluciones, rendición de cuentas, entre otras y lo riesgoso de este dork es que puede ser muy específico en lo que se desea obtener de él y puede ser el causante de fuga de información confidencial, en este caso se realizará una búsqueda de aprobaciones de consejo de carrera, para lo cual se ha elegido la palabra "Aprobación" y en la figura 20 podemos observar el resultado.

Búsqueda de documentos sensibles con dorks

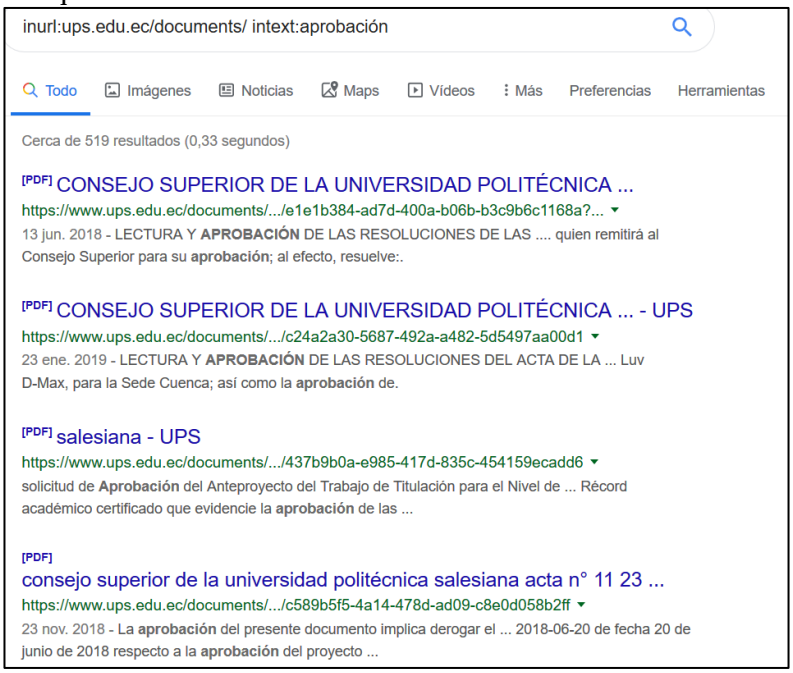


Figura 20 Dork de documentos de la UPS
Fuente: Dork generado por el autor en Google.com

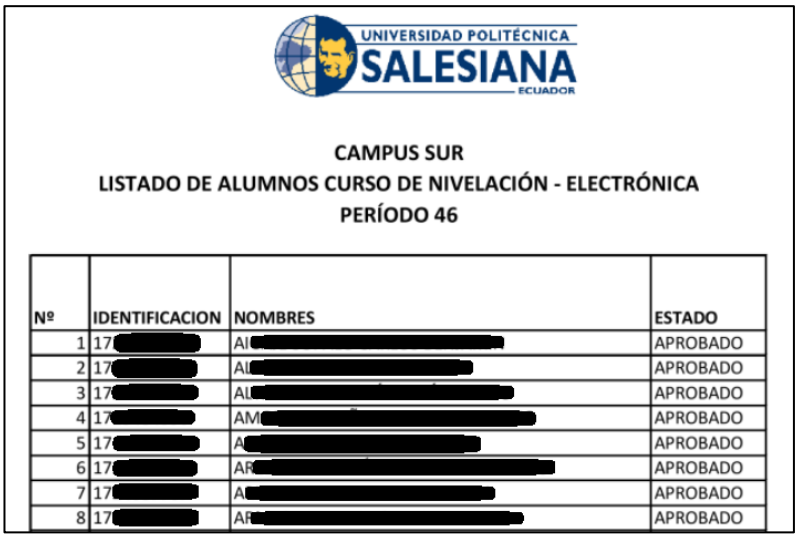
En este caso el dork: `inurl:"ups.edu.ec" intext:"Listado de aprobados",` mostrado en la figura 21, busca el listado de aprobados para saber el nombre y cédula de los estudiantes que entraron a la UPS en un determinado período y en una determinada carrera, esto podría conllevar a un intento de suplantación de identidad u otros ataques.

Dork para listar aprobados

Figura 21 Dork para buscar estudiantes aprobados
Fuente: Dork generado por el autor en Google.com

El resultado interno de la búsqueda responde con listas de los alumnos aprobados de las diferentes carreras y períodos. Los datos fueron censurados por protección en la figura 22.

Lista de alumnos obtenida



UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

CAMPUS SUR
LISTADO DE ALUMNOS CURSO DE NIVELACIÓN - ELECTRÓNICA
PERÍODO 46

Nº	IDENTIFICACION	NOMBRES	ESTADO
1	17[redacted]	AL[redacted]	APROBADO
2	17[redacted]	AL[redacted]	APROBADO
3	17[redacted]	AL[redacted]	APROBADO
4	17[redacted]	AM[redacted]	APROBADO
5	17[redacted]	A[redacted]	APROBADO
6	17[redacted]	AR[redacted]	APROBADO
7	17[redacted]	A[redacted]	APROBADO
8	17[redacted]	AR[redacted]	APROBADO

Figura 22 Resultado del dork para listar estudiantes aprobados
Fuente: Generado por el autor gracias al dork

Capítulo 3

Mapeo de la red

3.1 Escaneo de puertos abiertos e infraestructura con NMap

Para empezar, se realizará un escaneo con NMap a los objetivos de la tabla 2 anteriormente detallada para conocer valores como puertos abiertos, servicio que ocupa, sistema operativo, etc.

En este proceso se utiliza la herramienta gráfica de Nmap la cual se denomina "Zenmap" en donde a través del comando:

```
> nmap -O -p 1-65535 -T4 -A -v IP
```

busca en los 65535 puertos TCP y con la opción -O busca el sistema operativo de la IP específica.

3.1.1 Mapeo de puertos abiertos al portal www.ups.edu.ec

En la IP 45.235.140.7 perteneciente al dominio principal de la universidad se puede identificar que el servidor web actúa bajo el puerto 443 y 80 donde se verifica en la figura 23 que es un servidor Apache, posteriormente se indagará en la versión del mismo y como puede ser vulnerable ante explotaciones, también se descubre que el puerto 53 está abierto señalando una ruta para el servidor de DNS al cual realiza consultas para poder abrir los diferentes portales de esta web.

Mapeo en Nmap al portal ups.edu.ec

Puerto	Protocolo	Estado	Servicio	Versión
53	tcp	open	domain	
80	tcp	open	http	Apache httpd
443	tcp	open	http	Apache httpd

Figura 23 Escaneo de puertos TCP 45.235.140.7
Fuente: Generado en Zenmap por el autor

3.1.2 Mapeo de puertos abiertos al portal cas.ups.edu.ec

En el servicio de autenticación o también llamado "CAS" que actúa bajo la IP 45.235.140.20 se puede identificar en la figura 24 que como novedad que aparte de los puertos normales web que son el 80 y el 443 tiene abierto el puerto 8080 que sirve también como web server, además se identifica que es un servicio Tomcat/Coyote JSP engine 1.1 el cual puede ser verificado para determinar si en esta versión posee bugs o es susceptible ante ataques de exploits.

Es curioso que en este servicio el puerto 53 también esté abierto lo que representa una vulnerabilidad de DNS, ya que se pueden realizar peticiones externamente al servidor de DNS para que haga consultas.

Escaneo en cas.ups.edu.ec

Puerto	Protocolo	Estado	Servicio	Versión
53	tcp	open	domain	
80	tcp	open	http	Apache httpd
443	tcp	open	http	Apache httpd
8080	tcp	open	http	Apache Tomcat/Coyote JSP engine 1.1

Figura 24 Escaneo de puertos TCP en 45.235.140.20
Fuente: Generado en Zenmap por el autor

Consiguente al escaneo de puertos se ejecuta la IP 45.235.140.20 en navegador para encontrar que efectivamente su principal web server es Apache y está siendo ejecutado en CentOS como se muestra en el Anexo1.

3.1.1 Mapeo de puertos abiertos al portal servicesapp.ups.edu.ec

En el portal de "ServicesAPP" de la UPS luego del escaneo con NMap se llegó a una respuesta muy similar a los anteriores escaneos, con los puertos 80, 443 y 53 abiertos, pero en este servidor se pudo determinar que se está utilizando un servidor CentOS como muestra la figura 25.

Escaneo con NMap del portal servicesapp.ups.edu.ec

Puerto	Protocolo	Estado	Servicio	Versión
53	tcp	open	domain	
80	tcp	open	http	Apache httpd 2.2.15
443	tcp	open	http	Apache httpd 2.2.15 ((CentOS))

Figura 25 Escaneo de puertos TCP en 45.235.140.16
Fuente: Generado en Zenmap por el autor

Adicionalmente cuando se ejecuta la IP 45.235.140.16 en el navegador y se obtiene la página de bienvenida del servidor GlassFish con el que trabajan las apps alojadas en este servidor como se puede observar en la figura 26.

Página por default de GlassFish activada

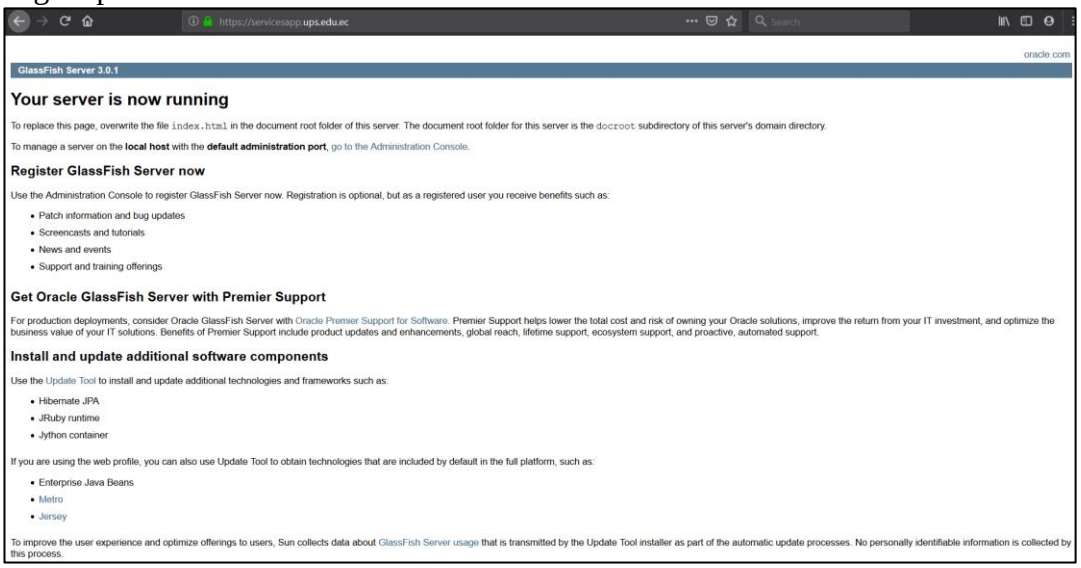


Figura 26 Ejecución de la IP 45.235.140.16 en el navegador
Fuente: Generado por el autor al ejecutar la dirección IP en el navegador

3.1.2 Mapeo de puertos abiertos al portal appwfp.ups.edu.ec

Al ejecutar el escaneo en la IP 45.235.140.18 perteneciente al servicio de aplicaciones internas se puede observar en la figura 27 que hace un barrido también por los puertos TCP 80,443 y 53 y muestra como sistema operativo un CentOS lo que hace necesaria un escaneo más profundo para indagar información por parte de este segmento de red

Escaneo con Nmap al portal appwfp.ups.edu.ec

Puerto	Protocolo	Estado	Servicio	Versión
53	tcp	open	domain	
80	tcp	open	http	Apache httpd 2.2.15
443	tcp	open	http	Apache httpd 2.2.15 ((CentOS))

Figura 27 Escaneo de puertos TCP en 45.235.140.18
Fuente: Generado en Zenmap por el autor

Es muy curioso ver en la figura 27 que introduciendo la IP en el navegador se obtiene un resultado en donde solo se logra divisar una página en blanco con el nombre de la UPS como muestra la figura 28.

Página mal dirigida en appwfp.ups.edu.ec

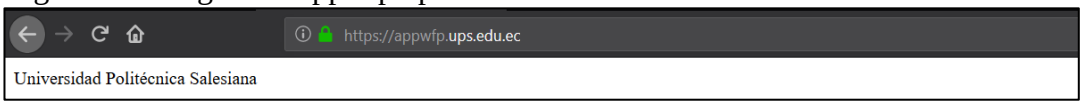


Figura 28 45.235.140.18 en el navegador
Fuente: Generado por el autor al entrar a 45.235.140.18 en un navegador web

El DNS resuelve la IP 45.235.140.18 por appwfp.ups.edu.ec y busca en el servidor web la página de inicio por lo regular llamada “index” donde su contenido solo existe “Universidad Politécnica Salesiana”.

3.1.3 Mapeo de puertos abiertos al portal virtual.ups.edu.ec

El portal del aula virtual con la IP 34.231.199.89 se caracteriza por estar fuera del segmento de red específicamente se aloja en AWS y en la figura 29 podemos observar el escaneo con NMap en dicha IP.

Escaneo Nmap al portal virtual.ups.edu.ec

Puerto	Protocolo	Estado	Servicio	Versión
443	tcp	closed	https	
80	tcp	open	http	nginx
53	tcp	open	domain	

Figura 29 Escaneo de puertos TCP en 34.231.199.89
Fuente: Generado en Zenmap por el autor

En la figura 29 observa que trabaja bajo el puerto 80 principalmente ya que el puerto 443 de este servidor se encuentra cerrado, pero comparte una propiedad muy importante el cual es el puerto 53 en donde hará consultas DNS institucional para poder acceder al sistema de aula virtual entregando datos como en la figura 30.

Mapa de flujo de conexiones al portal virtual.ups.edu.ec

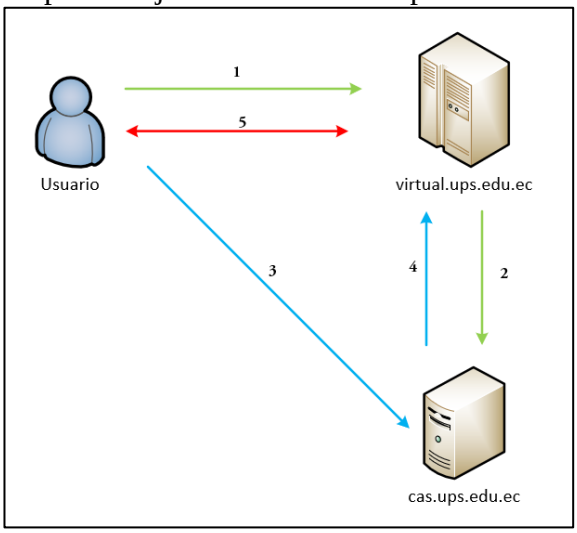


Figura 30 Interacción del usuario con el aula virtual
Fuente: Generado por el autor

Para interpretar la figura 30 se debe detallar etapa por etapa las conexiones que realiza este servicio:

1) El usuario se conecta con el servidor virtual.ups.edu.ec a través del puerto 80.
2) El servidor virtual.ups.edu.ec llama a cargar a cas.ups.edu.ec como forma de autenticación del sistema.
3) cas.ups.edu.ec se presenta ante el usuario y el usuario lo carga con username y password, cas.ups.edu.ec lo valida, esto lo realiza a través del puerto 443.
4) El sistema de autenticación envía los datos de confirmación del usuario para el servidor virtual.ups.edu.ec por el puerto 80.
5) Por último, se crea la conexión cliente – servidor entre el usuario y el aula virtual por el puerto 80.

3.1.4 Mapeo de sistema operativo

En la figura 31 se realiza la búsqueda al segmento de red 45.235.140.0 donde se obtiene como respuesta que trabaja con Linux en su versión de kernel 2.6.32 posiblemente de un CentOS 6 tomando en cuenta lo anteriormente escaneado, pero lamentablemente no proporciona una distribución exacta del sistema operativo.

Búsqueda de sistema operativo en servidores

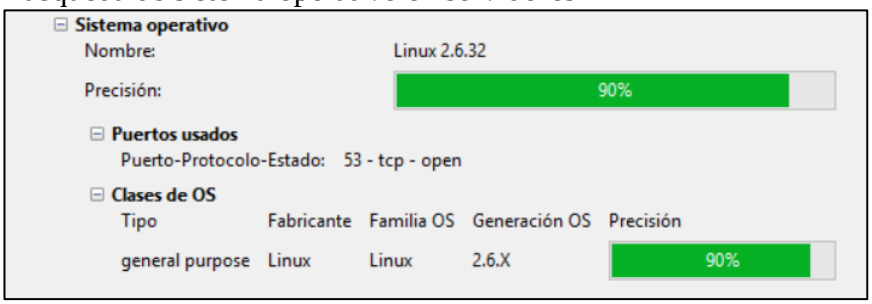


Figura 31 Sistema operativo base para subred 45.235.140.X
Fuente: Generado en Zenmap por el autor

Capítulo 4

Análisis y explotación de vulnerabilidades

En primera instancia se realizará análisis de vulnerabilidades con varias herramientas que busquen vulnerabilidades web, para este proceso el uso correcto de los escáneres web es vital pues ahorra mucho tiempo en el barrido de amenazas conocidas y realizan pruebas automáticas de procesos que permitan saber las vulnerabilidades de los sitios web.

4.1 Escaneo de vulnerabilidades TLS (Tranport layer security)

Al ingresar al dominio principal de la Universidad Politécnica Salesiana el cuál es: www.ups.edu.ec se encuentra como primera medida de seguridad observable en la figura 32 es que trabajan con el protocolo HTTPS con una seguridad TLS 1.2 que brinda una conexión encriptada en transición de 128 bits y como se puede observar la unidad certificadora es “COMODO CA” que brinda este tipo de seguridades.

Información TLS de www.ups.edu.ec

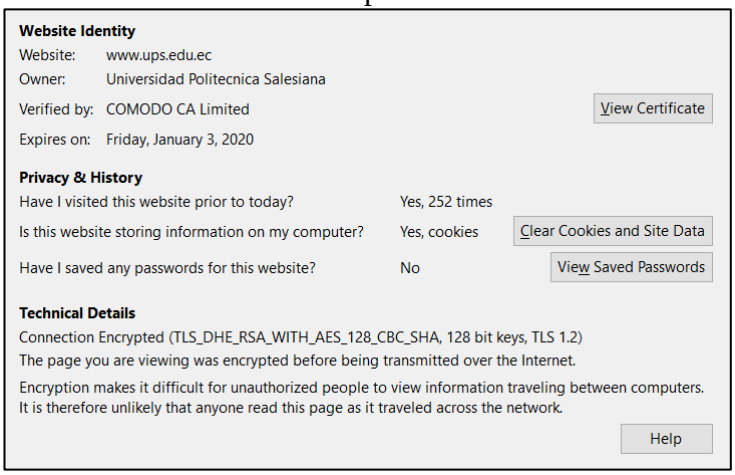


Figura 32 Información del certificado TLS al portal ups.edu.ec
Fuente: Figura obtenida por el autor

Como se puede observar en la figura 32 en el 2020 se renueva el contrato con COMODO que posee la UPS para la protección TLS, es muy importante contar en

primera instancia con este tipo de seguridades para que el tráfico entre el cliente y el servidor sea encriptado eso asegura la confidencialidad de los datos transmitidos pero, se ha dado casos que es vulnerable a ataques como ataque de vectores o recepción de claves por clonación u otro tipo de ataques que permiten al atacante leer en texto plano el tráfico generado por otro usuario.

4.1.1 Vulnerabilidades TLS en www.ups.edu.ec.

Al realizar una búsqueda de vulnerabilidades a este protocolo con la herramienta web de ssllabs.com el día 28/05/2019 a las 18:54 se encuentra lo mostrado en la figura 33.

Calificación de cifrado TLS a www.ups.edu.ec

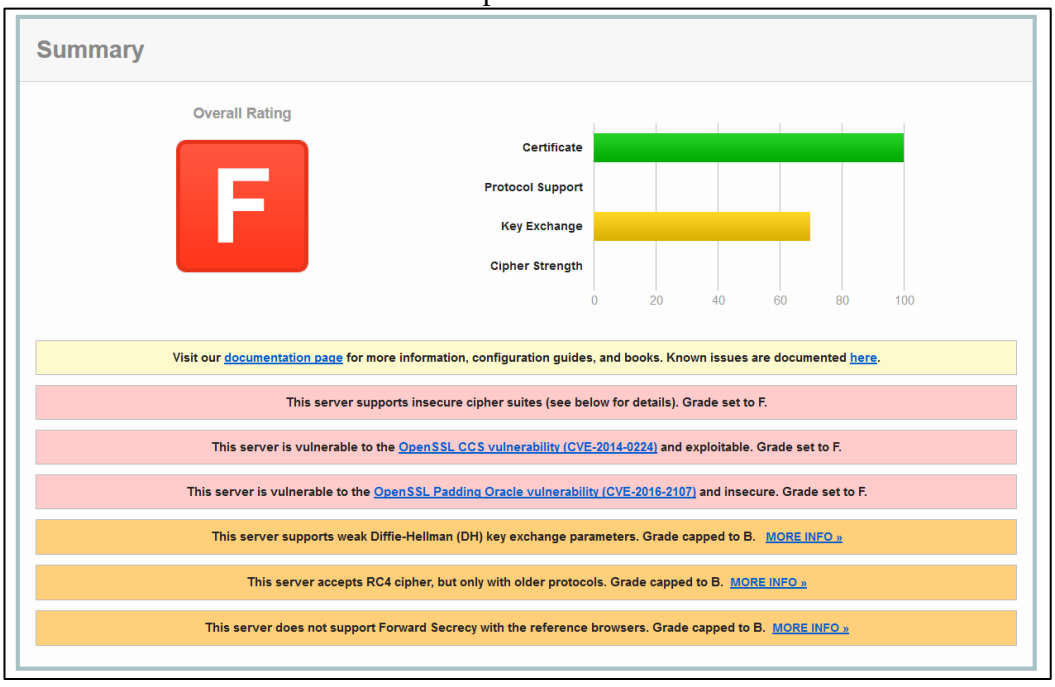


Figura 33 Calificación SSLLABS al TLS ups.edu.ec
Fuente: SSLLABS – ups.edu.ec

SSLLABS calificó con una “F” ya que encontró fallas críticas de seguridad como la vulnerabilidad CVE-2014-0224 también es vulnerable al padding Oracle que está escrito en el CVE-2016-2107 pero además esta poderosa herramienta encuentra algunas fallas en el sistema TLS y en su suite de cifrado que vuelve insegura a la información que pasa por ese canal, en la figura 34 se muestra estas vulnerabilidades.

Vulnerabilidades TLS encontradas en www.ups.edu.ec

Figura 34 Cipher Suites ups.edu.ec
Fuente: SSLLABS – ups.edu.ec

Como se puede observar en la figura 34 se obtiene tras el análisis 4 fallos críticos del portal que son:

1) TLS_RSA_WITH_DES_CBC_SHA(0x9),
2) TLS_DHE_RSA_WITH_DES_CBC_SHA(0X15),
3) TLS_RSA_WITH_RC4_128_MD5(0X4)
4) TLS_RSA_WITH_RC4_128_SHA(0X5).

Pero también se debe tomar en cuenta las suites de cifrado débiles que se marcan en naranja en la figura anterior.

Entonces bajo este protocolo se puede determinar que el protocolo RC4 del TLS es inseguro además puede ser explotado por los CVE anteriormente mencionados que puede causar perdida en la confidencialidad e integridad del servicio.

4.2 Escaneo de vulnerabilidades generado en Vega

Una curiosidad que fue tomada en consideración por parte de la búsqueda de WhoIs es que el dominio principal de la Universidad Politécnica Salesiana (UPS) es www.ups.edu.ec y no ups.edu.ec esto gana importancia al momento de realizar este escaneo. Para demostrar el uso del dominio correcto se realiza una búsqueda como la que muestra la figura 35.

Búsqueda al dominio de la Universidad Politécnica Salesiana

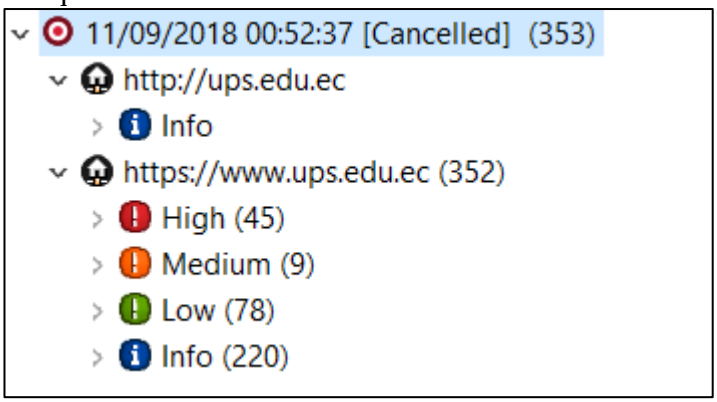


Figura 35 Búsqueda en Vega del dominio www.ups.edu.ec
Fuente: Generado por el autor en Vega

Se llega a determinar que para el dominio ups.edu.ec no existen vulnerabilidades pues el DNS no lo puede resolver, pero para el dominio www.ups.edu.ec se encuentran 45 vulnerabilidades graves, 9 vulnerabilidades medias, 78 vulnerabilidades de poca importancia y 220 recomendaciones de información.

Como resumen de este escaneo se obtiene la figura 36, donde posteriormente se detallará las vulnerabilidades más importantes.

Compilación de vulnerabilidades en Vega

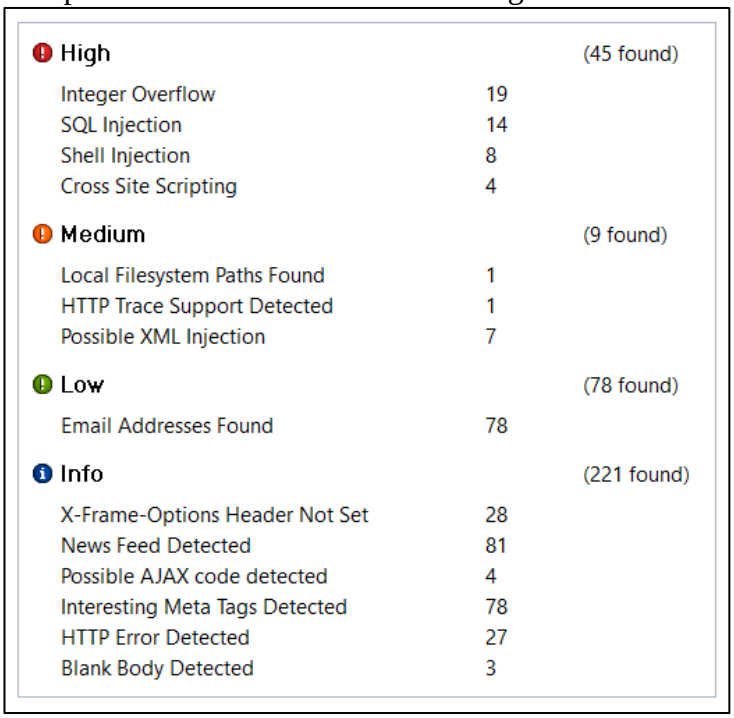


Figura 36 Resumen del escaneo en Vega a www.ups.edu.ec
Fuente: Generado por el autor en la herramienta Vega

Cabe recalcar que este escaneo detecta más vulnerabilidades proporcionalmente al tiempo de ejecución, ya que vega cuenta con un sistema recursivo para enviar datos a través de los directorios usando parámetros con el método GET y POST, por lo general este tipo de escáner puede dar falsos positivos y detectar vulnerabilidades inexistentes, entonces, no se debe confiar un 100% en este tipo de búsquedas y se recomienda el uso de otras herramientas para llegar a un consenso de vulnerabilidades y así poder identificar y categorizar adecuadamente.

4.2.1 Vulnerabilidades High encontradas por Vega.

Por parte de las vulnerabilidades High mostrados en la figura 37 se encuentra 45 vulnerabilidades divididos en 4 subcategorías, que representan los tipos de ataques realizados en el escáner.

Vulnerabilidades críticas encontradas en Vega

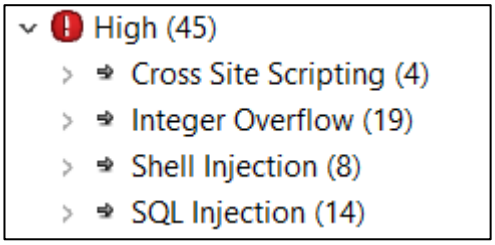


Figura 37 Vulnerabilidades high o crítical escaneadas por Vega
Fuente: Generado por el autor en la herramienta Vega

4 vulnerabilidades de XSS o Cross Site Script donde el código puede manipular todo el contenido del portal, cambiando su apariencia o realizar acciones dentro de la aplicación sin conocimiento del usuario, entonces como ejemplo de esta vulnerabilidad se encuentra el portal de /evento donde a través de una petición GET con el ingreso del evento onMouseOver se llegó a determinar el XSS como se puede ver en la figura 38.

Petición exitosa XSS

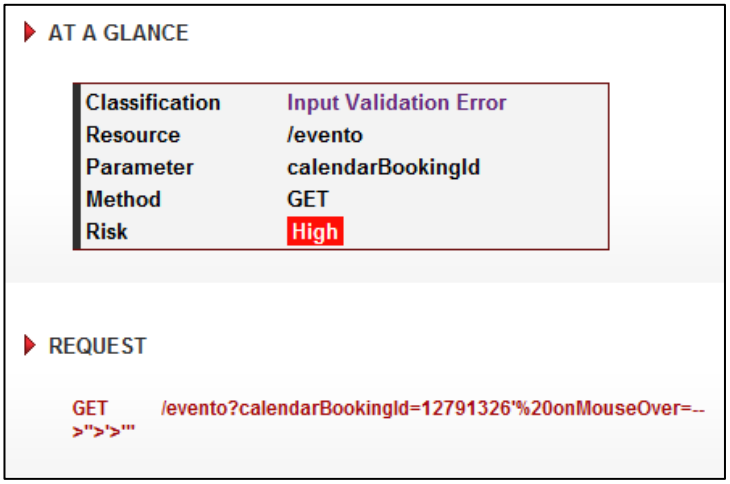


Figura 38 Ataque XSS al portal /evento
Fuente: Generado por el autor en la herramienta Vega

19 vulnerabilidades Integer Overflow esto puede proceder de campos sin control o sin validaciones donde pueden representar incorrectamente la cantidad total de datos, lo que resulta en un posible desbordamiento de búfer.

8 vulnerabilidades de Shell Injection donde los atacantes pueden ejecutar directamente comandos al servidor obteniendo una respuesta y puede conllevar a un acceso remoto no autorizado.

14 vulnerabilidades de SQL Injection que pueden conllevar a la lectura y escritura de datos a una base de datos en donde se permite crear accesos no autorizados y explotación inminente del sistema.

Estos datos deben ser posteriormente verificados en busca de determinar posibles falsos positivos o errores de escaneo.

4.2.2 Vulnerabilidades Medium encontradas por Vega.

Entre las vulnerabilidades medias encontradas por Vega se destaca configuraciones a Apache antiguas o mal realizadas, HTTP Trace, posibles ataques con XML Injection y directorios de sistema encontrados, estos pueden ser archivos del mismo Apache o no controlados por validaciones internas de URL´s como muestra la figura 39.

Vulnerabilidades medias encontradas por Vega

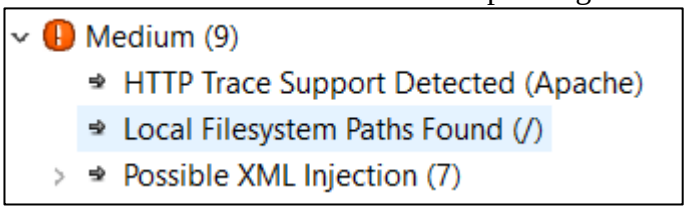


Figura 39 Vulnerabilidades Medium encontradas por Vega
Fuente: Generado por el autor en la herramienta Vega

4.2.3 Vulnerabilidades Low encontradas por Vega.

En su 100% son las 78 listas de email donde dentro de cada una de estas listas se encuentran un directorio de emails los cuales pueden ser utilizados por atacantes para realizar ataques de ransomware por email, spam y en casos ingeniería social como se muestra en la figura 40.

Vulnerabilidades bajas encontradas por Vega

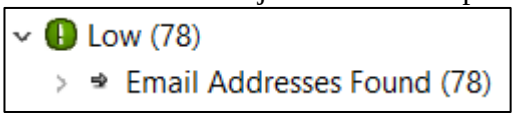


Figura 40 Vulnerabilidades Low encontradas por Vega
Fuente: Generado por el autor en la herramienta Vega

Vega hace una búsqueda en los directorios del dominio para encontrar listas de emails, pero para corroborar la información generada por Vega se crea una petición GET a través de un browser como especifica en la herramienta y se evidencia en la figura 41.

Lista de emails encontrados

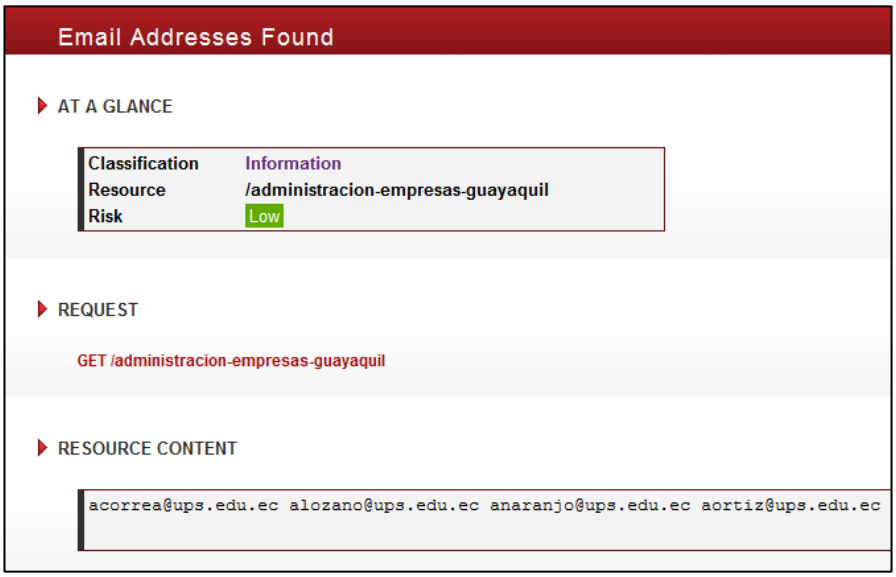


Figura 41 Petición GET para búsqueda de información por Vega
Fuente: Generado por el autor en la herramienta Vega

Entonces tras realizar la petición se obtiene el siguiente resultado en la figura 42.

Emails en lista web

Docente	Correo	Google Scholar
AGUILAR JARAMILLO RICHARD JOVANNY	raguilar@ups.edu.ec	
AGUIRRE GONZALEZ LILIANA ELIZABETH	laguirre@ups.edu.ec	https://scholar.google.es/citations?user=gPs-W1AAAAAJ&hl=es
ALAVA PINCAY CECIBEL LISBETH	calava@ups.edu.ec	https://scholar.google.com/citations?user=s5fcNFgAAAAJ&hl=es
ANDRADE MARTINEZ JHON MICHAEL	jandradem@ups.edu.ec	https://scholar.google.es/citations?user=LOIRJ98AAAAJ&hl=es
ASCENCIO BURGOS KARINA ANABELLA	kascencio@ups.edu.ec	https://scholar.google.es/citations?user=ypAc900AAAAJ&hl=es
BALAS LEON JUAN EMILIO	jbalas@ups.edu.ec	https://scholar.google.es/citations?user=iQSsEV4AAAAJ&hl=es
BASTIDAS JIMENEZ MARCELO JAVIER	mbastidas@ups.edu.ec	https://scholar.google.com.ec/citations?user=hnuHE5YAAAAJ&hl=en
BENITES MEDINA ROSA MARIA	rbenites@ups.edu.ec	https://scholar.google.com/citations?user=6O1BYF4AAAAJ&hl=es
BRIONES YELA ROBERTO JOHANN	rbrionesy@ups.edu.ec	
BUZAN RAMONA	rbuzan@ups.edu.ec	https://scholar.google.com/citations?user=zLRdMnQAAAAJ&hl=es
CABEZAS BARRAGAN WILLIAM JOSUE	wcabezas@ups.edu.ec	https://scholar.google.com/citations?user=gWnfniEAAAAJ&hl=en
CABRERA INTRIAGO MARIELA PATRICIA	mcabrera@ups.edu.ec	https://scholar.google.com/citations?user=bpt0Zk8AAAAJ&hl=es
CALLE CABEZAS RUTH ESTHER	rcalle@ups.edu.ec	https://scholar.google.es/citations?user=PamPCNEAAAAJ&hl=es
CARRERA JIMENEZ JAVIER ANTONIO	jcarreraj@ups.edu.ec	https://scholar.google.com.ec/citations?user=YLoJ_qgAAAAJ&hl=es
CONDE LORENZO EDDY	econde@ups.edu.ec	
CORREA CABRERA ANA LUISA	acorrea@ups.edu.ec	https://scholar.google.es/citations?user=XC2nf0MAAAAJ&hl=es
CUEVA ESTRADA JORGE MANUEL	jcueva@ups.edu.ec	https://scholar.google.es/citations?user=Gco2qwYAAAAJ&hl=es
DAU JARAMA GABRIELA ESTEPHANIE	gdau@ups.edu.ec	https://scholar.google.com/citations?user=8XfK7zAAAAAJ&hl=es

Figura 42 Verificación de cuentas de email encontradas
Fuente: Generado por el autor en la herramienta Vega

Se obtienen 69 cuentas de email con el nombre completo y además el perfil de Google scholar como se puede observar en la figura 42.

Entonces dentro de las 78 listas estimando 30 emails por lista, estarían involucradas alrededor de 2340 cuentas de email lo que genera un gran problema de seguridad.

4.2.4 Recomendaciones de información encontradas por Vega.

Vega también permite identificar tecnologías usadas, errores al protocolo http, recursos sin controlar, que en primera instancia no refleja una vulnerabilidad para el sistema, pero puede verse en la figura 43 se comprometen los tiempos de carga y respuesta, o posibles futuras amenazas.

Información obtenida de la página www.ups.edu.ec por Vega

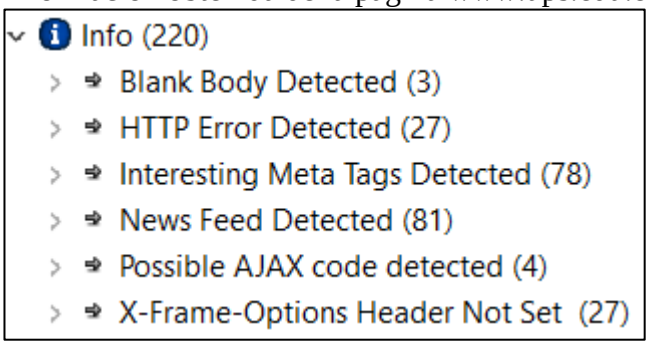


Figura 43 Muestras de información y errores detectados por Vega
Fuente: Generado por el autor en la herramienta Vega

La información otorgada en la figura 43 refleja 6 sub categorías en donde se destaca los metadatos incrustados con o sin consentimiento del árbol de directorios o de los usuarios implicados.

Los metadatos principalmente se dan en estos portales pues son fuentes bibliográficas fiables.

4.3 Escaneo de vulnerabilidades con Shodan

Cuando se realiza una búsqueda en Shodan por la IP del dominio principal 45.235.140.7, solo encuentra los datos de las tecnologías web usadas, pero no refleja

vulnerabilidades para Shodan esto se debe a que la página www.ups.edu.ec solo muestra información como se puede evidenciar en la figura 44.

Información básica obtenida de Shodan en el portal www.ups.edu.ec

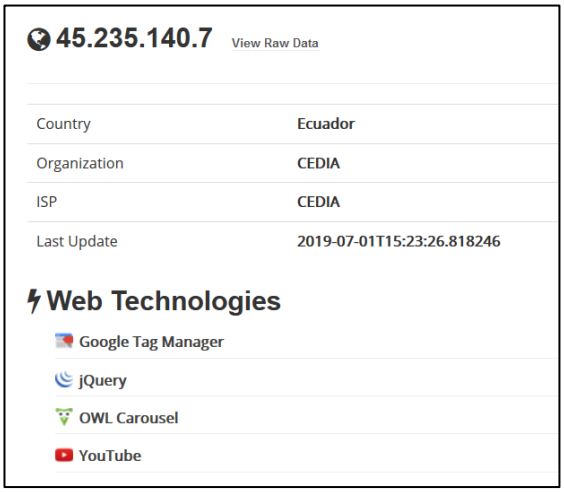


Figura 44 Búsqueda por parte de Shodan en la IP 45.235.140.7
Fuente: Generado por el autor en Shodan.io

Pero los subdominios sirven como sistemas de servicios entonces se procede a realizar búsquedas que devuelvan vulnerabilidades de nuestros objetivos.

Para el objetivo servicesapp.ups.edu.ec con IP 45.235.140.16 tras analizar los CVE se llegó a determinar que existen 5 vulnerabilidades críticas, 17 vulnerabilidades medias y 7 vulnerabilidades de bajo impacto. Las vulnerabilidades se las puede observar en el Anexo 2

Al buscar con Shodan en la dirección IP 45.235.140.18 perteneciente al dominio appwfp.ups.edu.ec se encontró que todas las vulnerabilidades de la tabla 4 son iguales, entonces esto identifica que comparten el mismo servidor por lo tanto tienen las mismas 29 brechas de seguridad.

Entre las vulnerabilidades más críticas en este análisis se encuentran los siguientes CVE que se detallará para entender el nivel de gravedad de estas amenazas.

Vulnerabilidad CVE-2017-7679.

Pertenece a un error de overread cuando el servidor web Apache en versiones 2.4 o anteriores el archivo de configuración “mod_mime” lee un byte más allá del final del bufer causando una respuesta “Content-Type” maliciosa, “National Vulnerability Database” entre otros categorizan con un 8/10 esta vulnerabilidad.

Vulnerabilidad CVE-2017-7668.

Este error pertenece a Apache en las versiones 2.2.32 hasta la versión 2.4.24 donde el demonio httpd ejecuta un método llamado “ap_find_token()” el cual hace una búsqueda extendida en la secuencia String de entrada, esto al ser explotado crea un fallo de segmentación para el método indicado anteriormente lo cual devuelve un valor incorrecto o un error.

Vulnerabilidad CVE-2011-3192.

Esta vulnerabilidad permite llevar a cabo un ataque de Denegación de servicio DOS para colapsar la RAM y CPU a través de una superposición múltiple de paquetes con gran tamaño lo cual provoca la pérdida parcial o total del servicio.

Vulnerabilidad CVE-2017-3167.

En versiones de Apache anteriores a 2.4.26 surge una fase de autenticación que puede ser alterada, en donde el ataque al método “ap_get_basic_auth_pw()” genera un bypass y omite la autenticación lo que abre una brecha de seguridad que afecta críticamente a la integridad, disponibilidad y confidencialidad del sistema.

Vulnerabilidad CVE-2013-2249.

Es una vulnerabilidad fácil de explotar, encontrada en Apache en su versión 2.4.5 la cual manipula un input a través del componente mod_session_dbd y causa una denegación del servicio.

Para finalizar con Shodan al buscar en el portal de aula virtual “AVAC” se desplegó que la página tiene 6 vulnerabilidades reportadas por CVE como se puede observar en la Tabla 4.

Entre las vulnerabilidades encontradas con Shodan a la página virtual.ups.edu.ec se identifican las siguientes, cabe recalcar que en esta figura se ha resaltado las vulnerabilidades más críticas para tomar en consideración en el momento de la exportación.

Tabla 4 Vulnerabilidades encontradas por Shodan en virtual.ups.edu.ec

CVE-2018-15919	Remotely observable behaviour in auth-gss2.c in OpenSSH through 7.8 could be used by remote attackers to detect existence of users on a target system when GSS2 is in use. NOTE: the discoverer states 'We understand that the OpenSSH developers do not want to treat such a username enumeration (or "oracle") as a vulnerability.'
CVE-2017-15906	The process_open function in sftp-server.c in OpenSSH before 7.6 does not properly prevent write operations in readonly mode, which allows attackers to create zero-length files.
CVE-2019-9641	An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_TIFF.
CVE-2019-9639	An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_MAKERNOTE because of mishandling the data_len variable.
CVE-2019-9638	An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_MAKERNOTE because of mishandling the maker_note->offset relationship to value_len.
CVE-2019-9637	An issue was discovered in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. Due to the way rename() across filesystems is implemented, it is possible that file being renamed is briefly available with wrong permissions while the rename is ongoing, thus enabling unauthorized users to access the data.

Nota: Vulnerabilidades más críticas escaneadas con Shodan

Para esta lista de vulnerabilidades se detallan las más críticas que son las siguientes: CVE-2019-9641 y CVE-2019-9639 que suscitan problemas con el componente EXIF en PHP donde existe una lectura no inicializada en exif_IFD_in_TIFF y exif_IFD_in_MAKERNOTE respectivamente esta vulnerabilidad afecta directamente a la confidencialidad de la información, además para explotar este fallo no es necesario protocolos de autenticación y su complejidad de acceso es muy bajo.

4.4 Escaneo de vulnerabilidades con Nessus

Se realiza un escaneo en conjunto de todos los targets para poder vincular vulnerabilidades con: puertos abiertos, versión de sistema operativo, versión del server web, entre otras que señala la tabla 5.

Tabla 5 Vulnerabilidades encontradas por Nessus

VULNERABILIDADES ENCONTRADAS POR NESSUS			
IP	**SERVICIO**	**VULNERABILIDAD**	**IMPACTO**
34.231.199.89	AULA VIRTUAL "AVAC"	HTTP Server Type and Version	BAJO
		HTTP Protocol in use	MEDIO
		Nessus SYN scanner	BAJO
		Remote web server screenshot	BAJO
		Web Server Directory Enumeration	BAJO
45.235.140.7	Dominio principal UPS	HTTP Server Type and Version	BAJO
		HTTP Information	BAJO
		Web Server Directory Enumeration	BAJO
		Web Server No 404 Error Code Check	BAJO
45.235.140.16	Servicios UPS	Unsupported Web Server Detection	ALTO
		Apache Banner Linux Distribution Disclosure	BAJO
		HTTP Server Type and Version	BAJO
		HTTP Information	BAJO
		Nessus SYN scanner	BAJO
		Web Server No 404 Error Code Check	BAJO
45.235.140.18	Apps UPS	Unsupported Web Server Detection	ALTO
		Apache Banner Linux Distribution Disclosure	BAJO
		HTTP Server Type and Version	BAJO
		HTTP Information	BAJO
		Nessus SYN scanner	BAJO
		Web Server No 404 Error Code Check	BAJO
45.235.140.20	Sistema de autenticación CAS	Apache Banner Linux Distribution Disclosure	BAJO
		HTTP Server Type and Version	BAJO
		HTTP Information	BAJO
		Nessus SYN scanner	BAJO
		Web Server No 404 Error Code Check	BAJO

Nota: Tabla de análisis de vulnerabilidades encontradas en Nessus

Se determina en Nessus que el sistema operativo en el cual trabajan las aplicaciones locales es Linux Kernel 2.6 on CentOS Linux release 6 y encontró 2 vulnerabilidades críticas de “Unsupported Web Server Detection”.

4.4.1 Ejemplo de resultados al escanear en Nessus a appwfp.ups.edu.ec

Al escanear el portal de aplicaciones con Nessus en primera instancia se detecta un 93% perteneciente a 48 logs de mejoras informativas vs un 7% de 1 vulnerabilidad media como muestra la figura 45.

Resultado de búsqueda de vulnerabilidades con Nessus

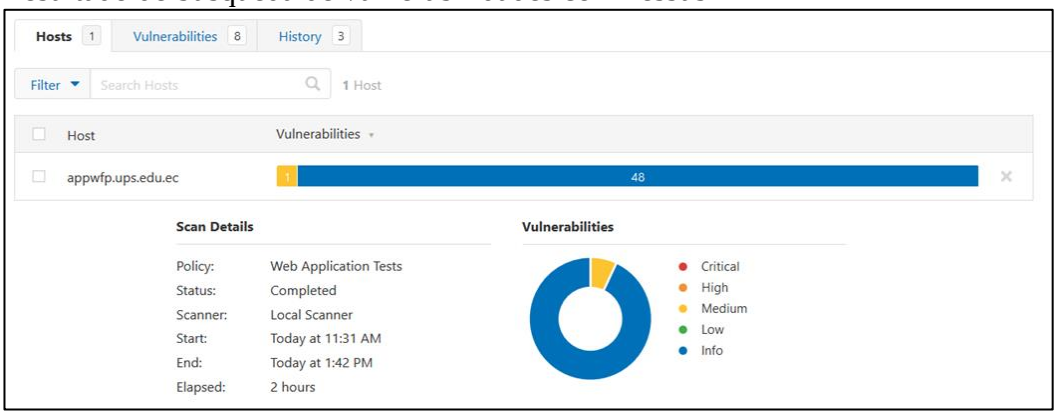


Figura 45 Escaneo con Nessus a appwfp.ups.edu.ec
Fuente: Generada por el autor en Nessus

Cabe señalar que este escaneo fue realizado por 2 horas y de igual forma como en Vega es directamente proporcional el tiempo de búsqueda a las vulnerabilidades encontradas.

En este escaneo la vulnerabilidad más grave encontrada fue la perteneciente al CVE-2003-1567, CVE-2004-2320 y CVE-2010-0386 donde el web server permite realizar peticiones TRACE y TRACK usando métodos del protocolo HTTP.

4.5 Escaneo de vulnerabilidades con OpenVas

Para el escaneo se realiza una búsqueda general con todos los targets para ahorrar tiempo de búsqueda y determinar el comportamiento que tiene OpenVas al momento de realizar escaneos en conjunto, entonces se inicia creando un objeto “Target” donde se especifican todas las direcciones IP´s a auditar como muestra la figura 46.

Todos los sitios a auditar incluidos

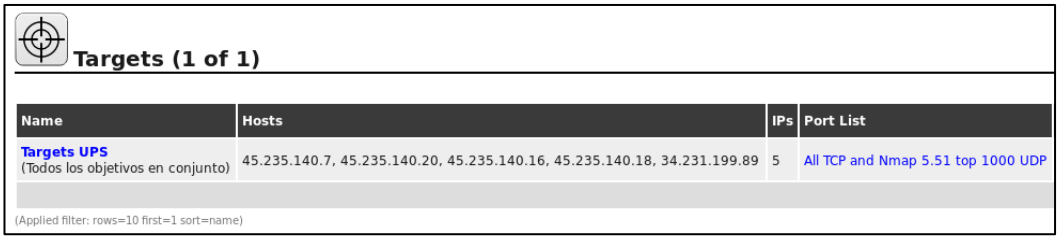


Figura 46 Target creado en OpenVas
Fuente: Generada por el autor en OpenVas

Como se observa en la figura 46 luego de crear el objeto target es necesario crear una tarea la cual será la encargada de llevar a cabo el escaneo de vulnerabilidades.

Al crear la tarea se especifica la lista de targets anteriormente creada y se determina el nivel de escaneo en este caso será “Full and very deep ultimate”, para continuar solo es necesario iniciar la acción con el botón “Start” y visualizar en reportes el progreso que tiene como en la siguiente figura.

Inicio del escaneo en OpenVas

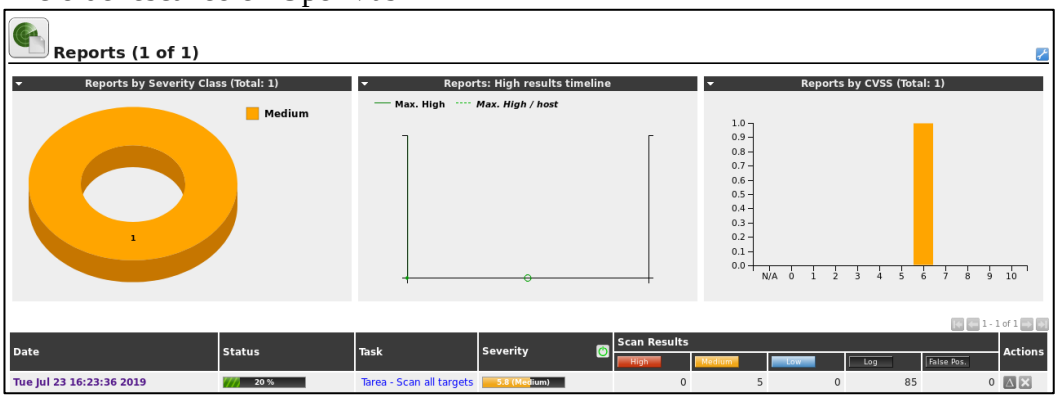


Figura 47 Escaneo de todos los targets con OpenVas
Fuente: Generada por el autor en OpenVas

Como se puede visualizar en la figura 47 el escaneo ha empezado y tiene un status del 20% en este caso en la figura 48 se muestra el escaneo al finalizar.

Culminación del escaneo con OpenVas

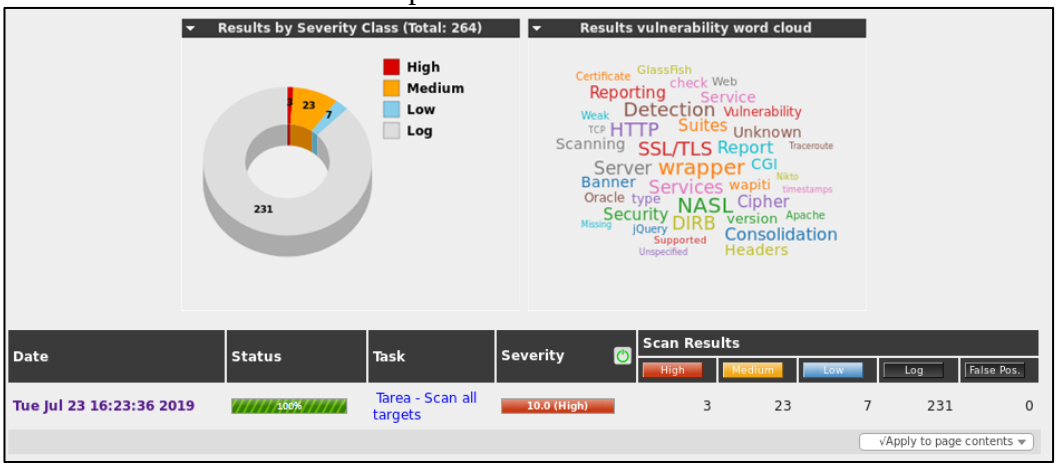


Figura 48 Escaneo completado con OpenVas
Fuente: Generada por el autor en OpenVas

Como se puede observar en la figura 48 al escanear todos los targets en conjunto, OpenVas ha obtenido 3 vulnerabilidades altas, 22 vulnerabilidades medias, 7 vulnerabilidades bajas y 231 logs que contrastan notablemente con Nessus porque este al ser de paga en su versión gratuita (donde se realizaron las pruebas) no obtiene la totalidad real de las vulnerabilidades. Lista de vulnerabilidades completa en Anexo 3.

Vulnerabilidad más crítica escaneada en OpenVas

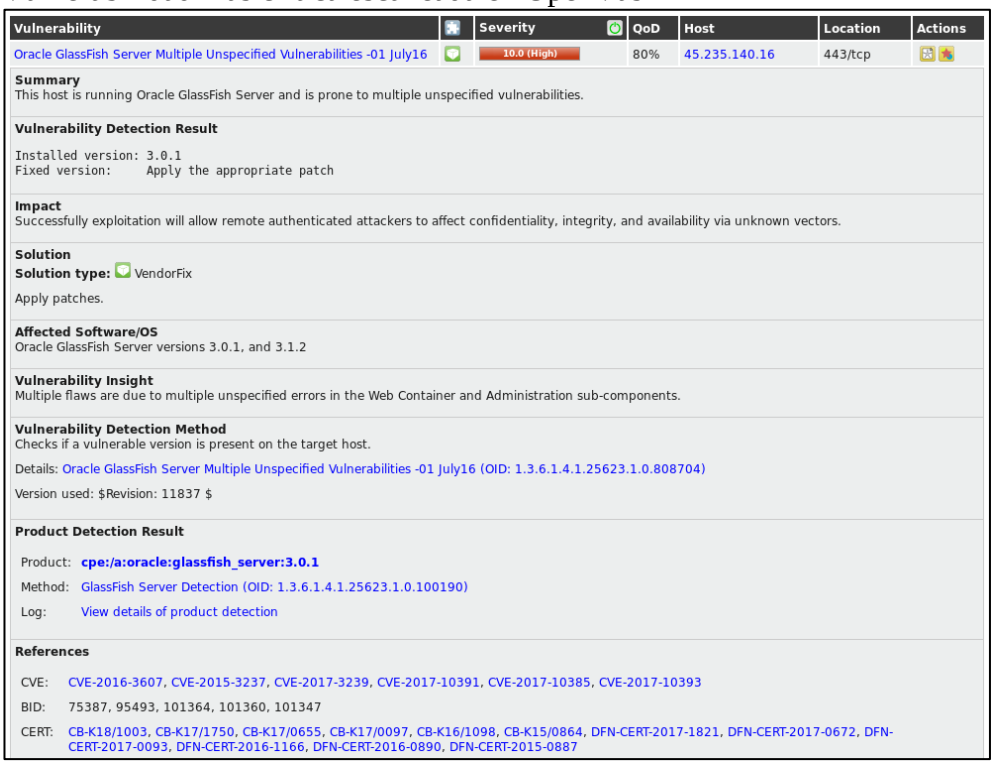


Figura 49 Vulnerabilidad más crítica encontrada por OpenVas
Fuente: Generada por el autor en OpenVas

Como se puede observar en la figura 49 se detalla el reporte con la vulnerabilidad más crítica encontrada en el escáner OpenVas.

Para finalizar con OpenVas se realiza un reporte del estado actual de los objetivos en conjunto donde califica la herramienta del 1 al 10 que tan insegura es donde se registra la figura 50.

Reporte cuantificado por OpenVas

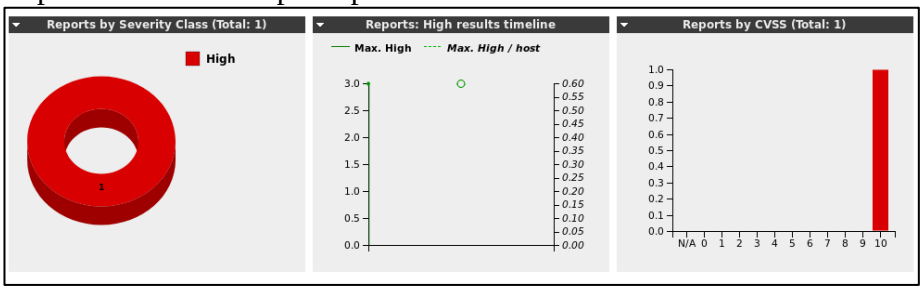


Figura 50 Calificación de riesgos por OpenVas
Fuente: Generada por el autor en OpenVas

4.6 Explotación de vulnerabilidades

En esta fase se necesita listar las vulnerabilidades con respecto a los puntos a auditar para posibles ataques y comenzar en la búsqueda de exploits, bugs u otros métodos de intrusión al sistema expuesto.

Se genera una tabla compilada de vulnerabilidades críticas para su posible explotación como muestra la tabla 6.

Tabla 6 Vulnerabilidades críticas o altas

VULNERABILIDADES HIGH O CRÍTICAS
TLS
CVE-2014-0224
CVE-2016-2107
TLS_RSA_WITH_DES_CBC_SHA (0x9)
TLS_DHE_RSA_WITH_DES_CBC_SHA(0X15)
TLS_RSA_WITH_RC4_128_MD5(0X4)
TLS_RSA_WITH_RC4_128_SHA(0X5)
VEGA
XSS
Integer Overflow
SQL Injection
Shell Injection
Shodan
CVE-2017-7679
CVE-2017-3167
CVE-2013-2249
CVE-2017-7668
CVE-2011-3192
CVE-2019-9641
CVE-2019-9639
Nessus
CVE-2003-1567
CVE-2004-2320
CVE-2010-0386
Unsupported Web Server Detection
OpenVas
Oracle GlassFish Multimple Unspecified Vulerabilities -01 July16
CVE-2016-3607
CVE-2015-3237
CVE-2017-3239
CVE-2017-10391
CVE-2017-10385
CVE-2017-10393
Oracle GlassFish/System Application Server Web Container DOS Vulnerability
CVE-2011-3559
Oracle GlassFish Server Multimple Unspecified Vulnerabilities-02 Oct16
CVE-2016-5519
CVE-2016-5528
CVE-2017-3250
CVE-2012-3249
CVE-2011-3247

Nota: Tabla compilatoria de las vulnerabilidades críticas y altas

Luego de enumerar las vulnerabilidades de la tabla 6, se realiza la búsqueda de exploits o bugs, para su explotación, primero se debe buscar los CVE en Google, entender cómo funciona y tratar de encontrar su explotación. Una página muy recomendada para esto es: www.exploit-db.com donde se puede encontrar una gran cantidad de exploits para vulnerar algún fallo de seguridad.

Este buscador de fallos también viene integrado en Kali Linux con una herramienta llamada searchsploit donde se puede descargar los exploits que se encuentran en esta página.

También una famosa y muy buena herramienta que permite realizar explotaciones mediante exploits es Metasploit Framework que dentro de el con el comando:

```
msf> search "Vulnerabilidad"
```

Busca en todo el directorio de exploits y otros algún bug que pueda realizar la explotación.

4.6.1 Explotación de vulnerabilidades del TLS

Explotación de vulnerabilidad CVE-2014-0224

Con esta vulnerabilidad se comprobará si se puede realizar un ataque de hombre en el medio o MITM con una inyección Change Cipher Spec o CCS.

Este ataque consiste en poder cambiar en la etapa de handshake el tipo de cifrado de la conexión.

La intromisión vendrá dada desde un script en Python donde detectará si ha sido vulnerado o no.

Para descargar el script en Kali Linux se usa una clonación al repositorio de GitHub con el siguiente comando:

```
git clone https://github.com/Tripwire/OpenSSL-CCS-Inject-Test.git
```

Posterior creará una carpeta en donde contiene el script llamado "OSSL_CSS_InectTest.py", entonces se debe otorgar permisos con el comando:

```
chmod 777 OSSL_CSS_InectTest.py
```

ahora se tiene que ejecutar de la siguiente manera:

```
#python OSSL_CCS_InjectTest.py www.ups.edu.ec
```

Al ejecutar este comando se realiza la comprobación de la figura 51.

Escenario para ataque de hombre en el medio

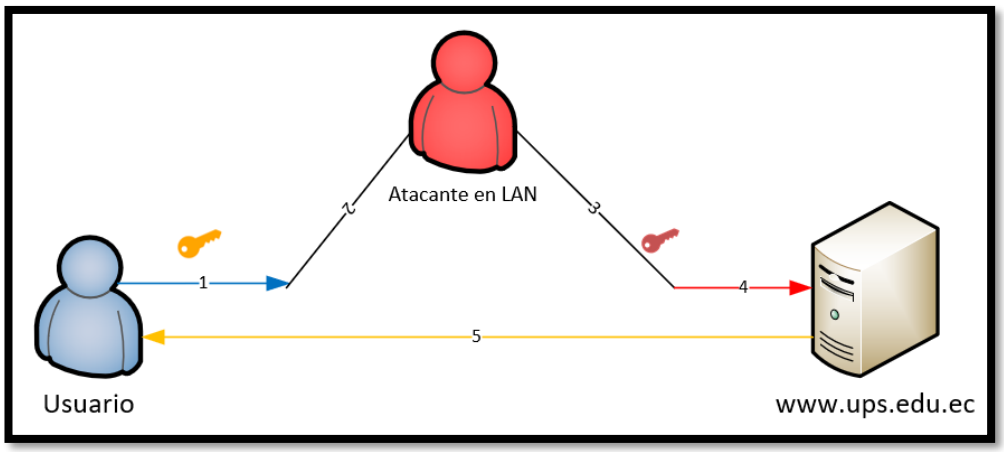


Figura 51 Ataque de MITM a la vulnerabilidad CVE-2014-0224
Fuente: Generado por el autor

1. El usuario envía un mensaje de client hello al servidor proponiendo el handshake en TLS
2. El atacante intercepta y cambia la clave con código CCS,
3. Envía la clave al servidor
4. Se hace pasar como conexión del usuario
5. El servidor responde con la clave de conexión que puede ser desencriptada por el atacante.

Y los resultados son los que muestra en la figura 52

Resultado de vulnerabilidades CCS

```
Brought to you by Tripwire VERT (@TripwireVERT)
[TLSv1.2] www.ups.edu.ec:443 may allow early CCS
[TLSv1.1] www.ups.edu.ec:443 may allow early CCS
[TLSv1] www.ups.edu.ec:443 may allow early CCS
[SSLv3] www.ups.edu.ec:443 Invalid handshake.
***This System Exhibits Potentially Vulnerable Behavior***
If this system is using OpenSSL, it should be upgraded.
```

Figura 52 TLS vulnerables de CCS
Fuente: Generado por el autor en Kali Linux

Como resultado se visualiza que se puede llevar a cabo este tipo de ataques a este dominio con la versión de TLS 1.2, 1.1 y 1. Se puede llegar a explotar más esta vulnerabilidad complementándola con el auxiliar de metasploit

“auxiliary/scanner/ssl/openssl_heartbleed” donde se logra interceptar esta comunicación, pero este pentesting como tiene fines educativos solo se lo mencionará.

Explotación de vulnerabilidad CVE-2016-2107

Para esta vulnerabilidad de SSL se descarga el exploit 39768 desde exploit-db.com

Para proseguir con la explotación en una máquina virtual Kali Linux se ejecutan los siguientes comandos para importar exploits desde Exploit-DB a Metaexploit y de esta manera tener mayor amplitud de exploits contra las vulnerabilidades detectadas.

```
service postgresql start
```

Inicia el servicio de postgresql

```
msfconsole
```

Inicia Metasploit framework, para conocer los datos que posee en primera instancia es necesario obtener una imagen del estado actual que muestra la figura 53.

Situación inicial pre importación de exploits

Figura 53 Número de exploits en Metasploit framework
Fuente: Generado por el autor en Metasploit

Donde cabe señalar que encontró 1909 exploits que ya vienen integrados con Metasploit para los cuales se agrega algunos de Linux, Windows, JSON y los que necesitemos puntualmente.

```
searchsploit -u
```

Actualiza listas de exploits desde dependencias de Kali Linux, dentro de root se encuentra el directorio llamado .msf4/modules donde se crea una carpeta llamada exploits con el siguiente comando

```
mkdir exploits
```

Dentro de la carpeta Exploits se mueven los archivos del directorio usr/share/exploitdb/ que se necesiten, una vez movidos se ejecuta el comando `updatebd` y se vuelve a ejecutar el `Metasploit` con `msfconsole` y como se puede observar en la figura 54 ya ha crecido la base de datos de exploits.

Situación final de Metasploit

Figura 54 22 Exploits añadidos en Metasploit
Fuente: Generado por el autor en Metasploit

Se necesita descartar que no sea un falso positivo esta vulnerabilidad, para eso se utiliza la herramienta online Filippo.io que verifica si una amenaza está presente en los servidores especificados, en este caso será la amenaza CVE-2016-2107 que se determinará si es vulnerable o no, y muestra el resultado la figura 55.

Escaneo de vulnerabilidad CVE-2016-2107

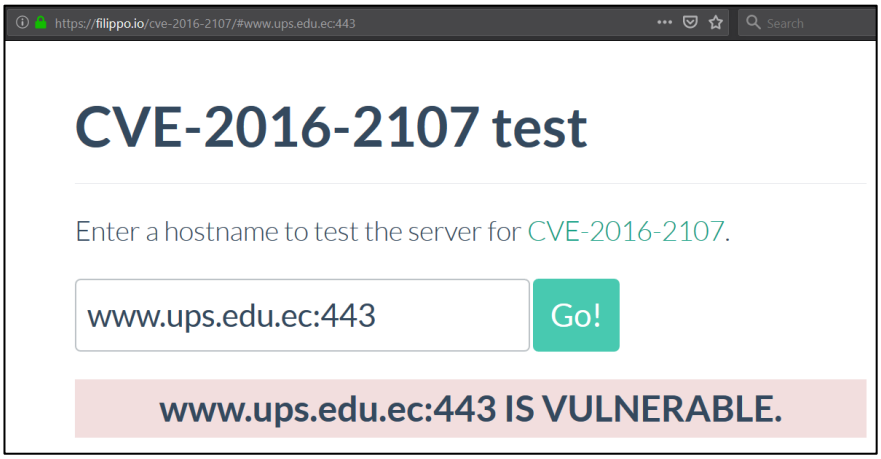


Figura 55 Comprobación CVE-2016-2107 en filippo.io
Fuente: Generado por el autor en Filippo.io

Una vez indicado el dominio, el portal ejecuta una serie de pruebas para determinar si es vulnerable o no y en este caso se indica que efectivamente es vulnerable.

Para proseguir con la explotación se comprueba internamente que el exploit 39768 representa un gran riesgo de seguridad y se cambia por un auxiliar de Metasploit llamado “dos/ssl/openssl_aesni” para generar un ataque DOS igualmente, ya que esta vulnerabilidad genera la clave AES la cual lleva a un DOS interno al realizar peticiones, en esta ocasión se configuro con 4000 paquetes de procesamiento enviado y al host www.ups.edu.ec con el puerto 443.

Entonces al configurar los parámetros se debe obtener un resultado como en la figura 56.

Resultado de las opciones configuradas

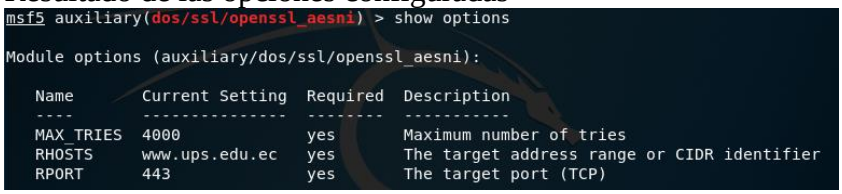


Figura 56 Auxiliar openssl_aesni configurado correctamente
Fuente: Generado por el autor en Metasploit

Para comprobar el ataque se lleva a ejecución un detector de tiempo de carga el cual medirá el tiempo al inicio y al final del DOS para conocer si la página puede colapsar.

Se realiza una prueba con 4000 paquetes que es un número bajo de paquetes para que el servidor no llegue a colapsar, pero si se ralentizará el servicio como se muestra en la figura 57.

Ataque DoS realizado

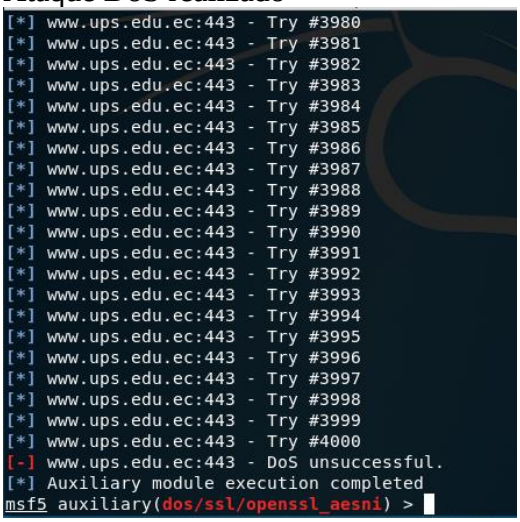


Figura 57 Ataque controlado de DoS
Fuente: Generado por el autor en Kali Linux

Como se puede observar en la figura 57 el propio Metasploit informa del estado del ataque, en este caso aparece como DoS unsuccessful porque la página no llegó a caer, este envió de paquetes duró 2 minutos y estos son los resultados.

Comparación del tiempo de tardanza

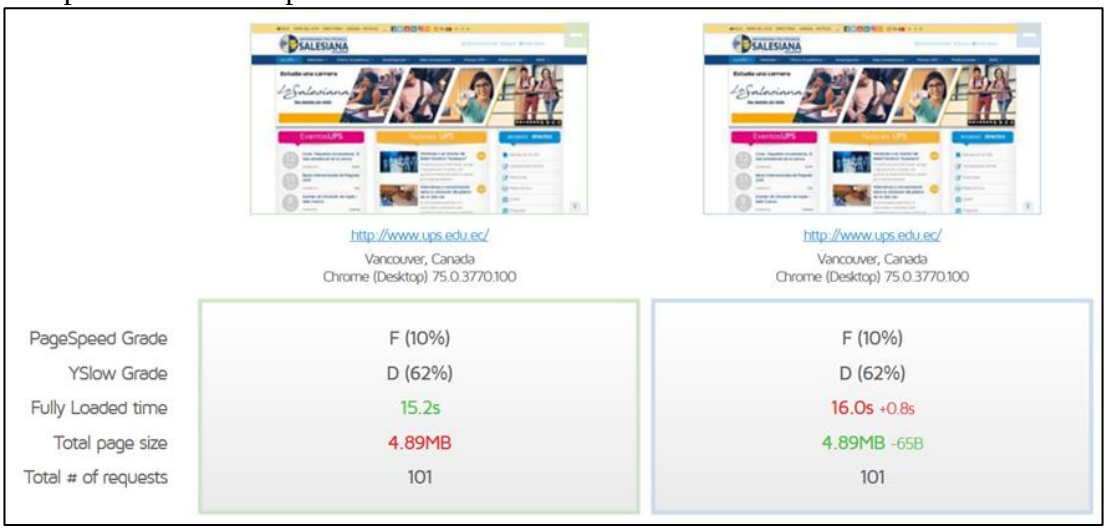


Figura 58 Tiempo de carga inicial y final en GTmetrix.com
Fuente: Generado por el autor en gtmetrix.com

Como muestra la figura 58, se obtiene que ralentizo el servidor 1 segundo en 2 minutos de explotación entonces el servidor puede sufrir de colapsos gracias a este fallo de seguridad.

4.6.2 Explotación de vulnerabilidades de Shodan

Explotación de vulnerabilidades CVE-2017-7679, CVE-2017-3167, CVE-2013-2249.

Para explotar estas vulnerabilidades se debe atacar internamente y se ocupará el exploit respectivo de este repositorio de GitHub

```
https://github.com/gottburgm/Exploits.git
```

Donde se lo instala en Kali Linux clonando el repositorio, buscando el exploit y se ejecuta de la siguiente manera:

```
./CVE-XXXX-XXXX.pl <URL:PORT> <Apache_Install_Path>
```

Debido a que en este proyecto solo se harán pruebas de vulnerabilidades externas, los administradores necesitan tomar cartas sobre estas una de las mejores formas es actualizar el Apache que están utilizando a una versión reciente.

Explotación de vulnerabilidad CVE-2011-3192.

Para este ataque se copiará un script en Perl que pertenece a Kingcope en la plataforma exploit-db.com donde crea una ejecución remota de memoria provocando esto en un DOS al sistema.

El script se lo encuentra en el siguiente link:

```
https://www.exploit-db.com/exploits/17696
```

Este se lo descargará y copiará a un archivo nuevo en Kali Linux donde se ejecutará y se realizará el DoS.

Ataque DoS para el CVE-2011-3192

```
root@kali:~/Desktop# nano dos_cve20113192.pl
root@kali:~/Desktop# chmod 775 dos_cve20113192.pl
```

Figura 59 Creación del script y generación de permisos
Fuente: Generado por el autor en Kali Linux

Al visualizar la figura 59 en la primera ejecución se crea el archivo de Perl donde se pega el script y en la segunda ejecución se le asigna permisos de lectura y ejecución al archivo.

Para que el ataque tenga efecto se deben ejecutar en un entorno interno a la red con varias terminales, para que el DoS cree mayor tráfico y pueda ralentizar el sistema e inclusive si el ataque dura mucho tiempo y es un ataque distribuido (DDoS) puede llegar a ser víctima de una saturación total de la memoria y colapsar la página web, para este caso se ejecutará 3 terminales paralelos como se puede observar en la figura 60.

Esenario de Flooding DoS

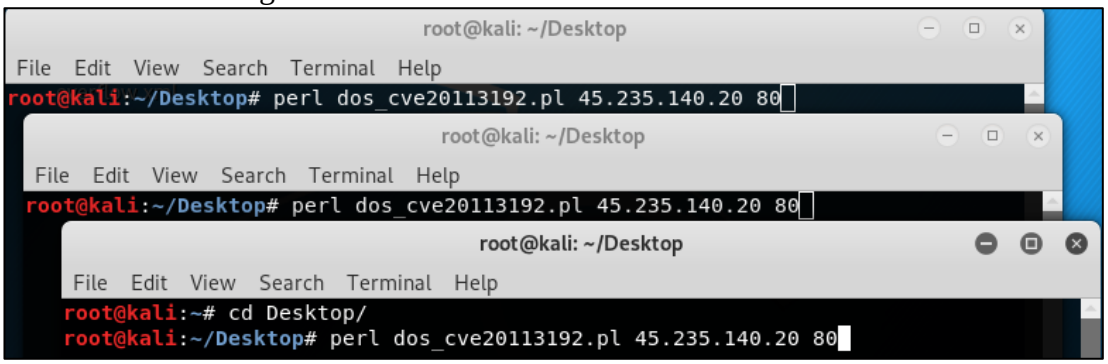


Figura 60 Preparación de ataque a CVE-2011-3192
Fuente: Generado por el autor en Kali Linux

Al momento de ejecutar en el portal cas.ups.edu.ec se ralentizo el sistema y el sistema proporcionó un 403 error para denegar las conexiones entrantes de la IP atacante, lo curioso es que se llevaron ajustes de seguridad a este tipo de flooding que es muy recomendable, pero se debe parchar el Apache para que no exista ya el riesgo de tener esa vulnerabilidad abierta.

Explotación de vulnerabilidad CVE-2019-9641 y CVE-2019-9639

Para estas vulnerabilidades no se encontró un exploit esto se puede deber a que es una vulnerabilidad relativamente nueva, en cualquier momento se puede desarrollar un

“zero day exploit” o hacer público un exploit, por ello debe ser parchado el PHP a su última versión para que ese riesgo sea mitigado y de esta manera aumentar la seguridad de esta vulnerabilidad crítica.

4.6.3 Explotación de vulnerabilidades encontradas en Vega

En este escáner de vulnerabilidades se encontró diversos tipos de ataques que deben ser verificados, pues Vega tiene reputación de entregar falsos positivos, entonces se tiene que indagar en sus resoluciones.

Explotación de vulnerabilidades XSS

Para este tipo de ataque se tiene en cuenta las URL detectadas y se utiliza un repetidor de peticiones para conocer si los resultados entregados por Vega son correctos o incorrectos.

En la búsqueda de vulnerabilidades VEGA encontró 4 vulnerabilidades XSS entonces, con ayuda de un repetidor de peticiones como Fiddler, Burpt Suite entre otros se realiza las peticiones necesarias, para la comprobación de los eventos reportados.

Para iniciar en portal /evento, se encuentra la siguiente petición realizada:

```
GET /evento?calendarBookingId=12791326'%20onMouseOver=-->">'>'"
```

Donde al generar el request se obtiene el resultado de la figura 61:

Request para XSS

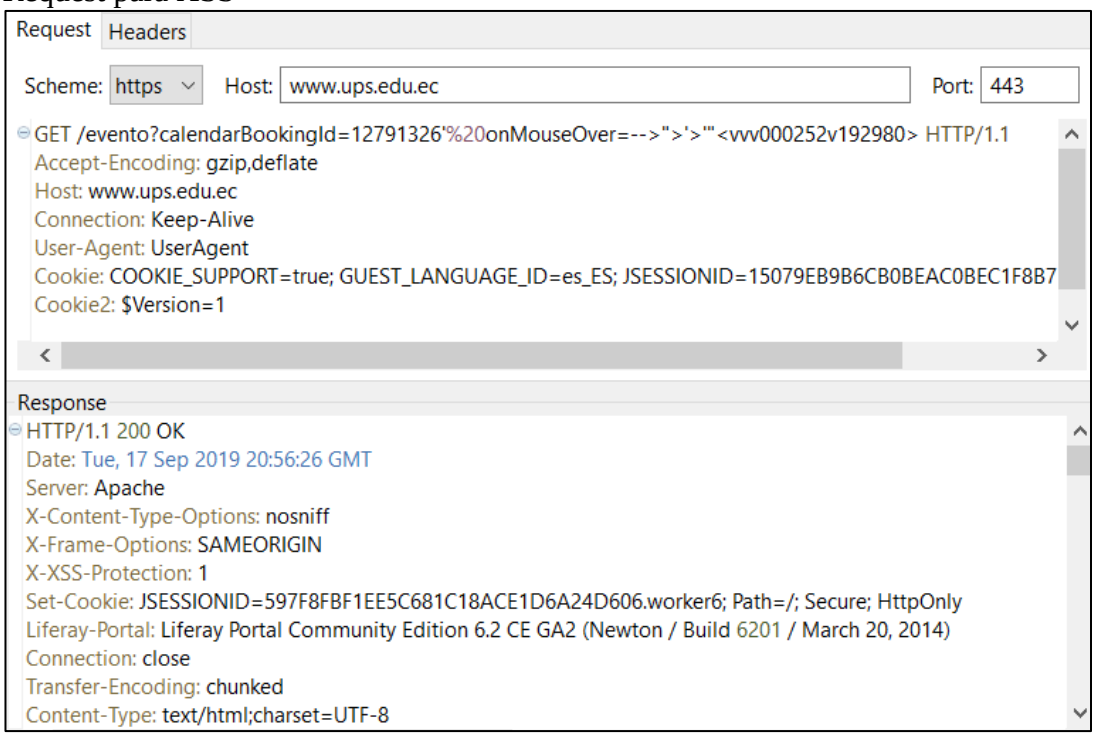


Figura 61 Respuesta ante XSS en /evento
Fuente: Generado por el autor en la herramienta Request Editor de VEGA

Como se puede observar en la figura 61 la reproducción de la petición hacia el servidor dio una respuesta positiva ante esta vulnerabilidad esto es comprobable al momento de revisar el código fuente de la página en el retorno, como muestra en la figura 62.

XSS incrustado en la página web

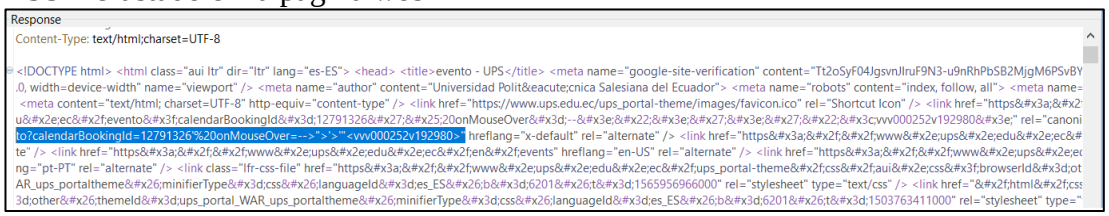


Figura 62 Comprobación de código fuente ante la petición GET
Fuente: Generado por el autor en la herramienta Request Editor de VEGA

Se observa en azul de la figura 62 la petición GET que realizó el “Request Editor” y como este afecta al formato del estilo HTML propio de la página al devolver su resultado. Un punto a favor es que el sistema reconoce automáticamente como ataque XSS, aunque solo envía un valor de 1 en una cabecera llamada “X-XSS-Protection” pero no evita que se realice la petición con éxito, este tipo de defensas solo sirven para

alertar de posibles XSS. Los demás intentos explotación con XSS se encuentran en el Anexo 4.

Cabe recalcar que no se modificaron los flags de peticiones por lo que una persona con experticia puede llegar a comprometer estos sitios modificando la petición.

Explotación de vulnerabilidades interger overflow.

En este tipo de peticiones se realiza un monitoreo del valor máximo que una variable puede tener y si se ingresa un byte más puede causar un volcamiento de memoria si no es controlado eficientemente.

Entonces para realizar esta validación se puede ejecutar el siguiente request.

```
GET /ups-portal-comprobantes-publico-portlet/css/main.css?browserId=-
2147483648&themeId=ups_portal_WAR_ups_portaltheme&minifierType=css&languageI
d=es_ES&b=6201&t=1540218623000
```

Resultado de request para overflow

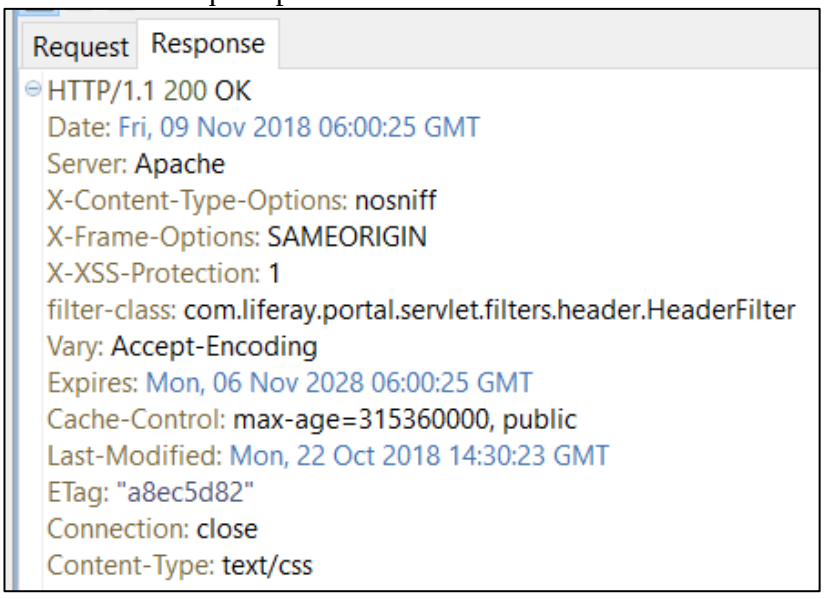


Figura 63 Integer Overflow vulnerado
Fuente: Generado por el autor en la herramienta Request Editor de VEGA

En la figura 63 se obtiene una respuesta favorable, ya que se controla el máximo rango que puede tener la variable “max-age” que es 315360000 y se obtiene también que la variable public esto quiere decir que puede ser llamado desde otras clases, la petición

también muestra cuándo fue la última modificación que tuvo en donde resalta la fecha y hora exacta “22 de Octubre 2018 14:30”.

Los demás resultados de vulnerabilidades encontradas con SQL Injection y Shell Injection en Vega fueron falsos positivos que se terminaron por descartarlos.

4.6.4 Explotación de vulnerabilidades encontradas en Nessus

CVE-2003-1567, CVE-2004-2320 y CVE-2010-0386.

Esta vulnerabilidad puede ser afectada sin autenticación por la configuración del método HTTP Handler en los procesos de TRACE/TRACK y para comprobar que realmente exista esta vulnerabilidad y no sea un falso positivo se ejecuta en Kali Linux el siguiente comando:

```
curl -k -X TRACE https://HOST
```

Donde sí es positiva la vulnerabilidad, debe devolver un mensaje que comprueba que efectivamente está aceptando las peticiones TRACE que se envía, en un caso de necesitar utilizarlo con el método TRACK solo se intercambia las palabras en el comando, entonces en nuestro caso se obtuvo el resultado de la figura 64.

Validación de Trace

Figura 64 Validación del CVE-2003-1567
Fuente: Generado por el autor en Kali Linux

Se observa en la figura 64 la aceptación de todos los paquetes TRACE enviados.

4.6.5 Explotación de vulnerabilidades encontradas en OpenVas

Explotación de vulnerabilidad CVE-2016-3607.

Esta vulnerabilidad puede ser explotada mediante un ataque DOS, el ataque será exitoso tras enviar/recibir mediante el método GET más de 4Gb de información. El

exploit de repetición se lo puede encontrar en https://www.exploit-db.com/exploits/41769 pero el portal de la universidad ya posee una mitigación contra los ataques de flooding, que, tras un cierto número de peticiones, el servidor automáticamente bloquea los intentos de conexión, hay que tener mucha precaución ya que esta mitigación puede ser vulnerada en cajas de texto mal validadas que permitan el ingreso de información sin control en donde se pueden replicar este tipo de peticiones.

Explotación de vulnerabilidad CVE-2015-3237.

Para esta vulnerabilidad no se encontraron bugs ni exploits, pero para prevenir ataques de "zero day exploit" se recomienda parchar esta vulnerabilidad.

CVE-2017-10391, CVE-2017-10385, CVE-2017-10393, CVE-2017-3239, CVE-2011-3559: No se encontró un bug o exploit para estas vulnerabilidades, pero se necesita actualizar el sistema a la versión más reciente para parchar estas vulnerabilidades y ante posibles ataques.

Explotación de vulnerabilidad CVE-2017-3250

Con esta vulnerabilidad se basa en un ataque XSS que responde a peticiones por parte de scripts del atacante en este caso para comprobar la existencia de esta vulnerabilidad se aplicará en el portal de aplicaciones donde se encuentra el portal de eventos donde se ejecutará un error, esto se lleva a cabo con el bug encontrado en https://www.exploit-db.com/exploits/34834 que detalla el ataque contra "Oracle Fusion Middleware 10.1.2/10.1.3 - BPEL Console Cross-Site Scripting", tras haber establecido el script en el navegador se obtiene el resultado de la figura 65.

XSS Oracle Fusion Middleware

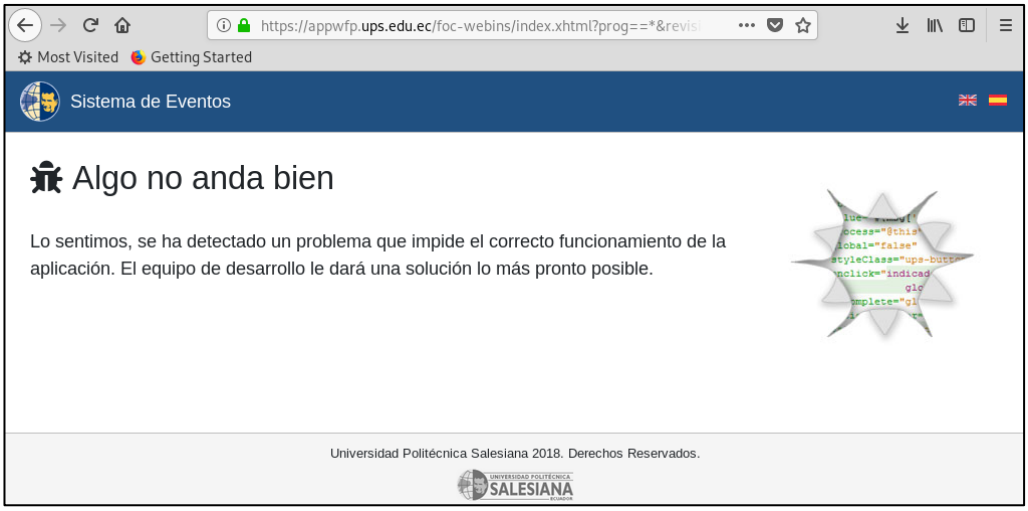


Figura 65 XSS contra portal de eventos
Fuente: Generada por el autor en appwfp.ups.edu.ec

Luego de analizar el resultado de la figura 65 se obtiene que el portal tiene un nivel de seguridad para controlar las peticiones XSS, pero al modificar la petición del script en el navegador se obtiene un sitio no regulado en el cual puede ser vulnerado por XSS.

XSS no controlado en appwfp.ups.edu.ec

Figura 66 XSS no controlado en portal de eventos
Fuente: Generada por el autor en appwfp.ups.edu.ec

Como se puede observar en la figura 66 se modifica la URL para encontrar un sitio no restringido por el control del portal y este puede ser explotado con ataques XSS.

Para mitigar esta vulnerabilidad se necesita controlar todos los subdirectorios con la prohibición de acceso.

Explotación de vulnerabilidades CVE-2012-3249, CVE-2011-3247.

Se determina que estas vulnerabilidades son falsos positivos tras haber realizado una prueba de servicios con los resultados de nmap y verificar que no cuenta con vulnerabilidades de HP Fortify Software Security Center externamente y Apple QuickTime correspondientemente, entonces se eliminan estas vulnerabilidades del pentesting tras ser reconocidos como vulnerabilidades falsas.

4.6.6 Explotación a nivel de usuario

Este es un tipo de intrusión que se realiza directamente con la interacción del pentester y el navegador, realizando peticiones y validaciones para obtener un resultado que permita obtener información sensible, crear un comportamiento no deseado en la web app o crear fallos de seguridad que afecten los 3 pilares de la seguridad de la información como es la integridad, confidencialidad e integridad de la información.

Cabe recalcar que este debe ser el último paso de la explotación pues aquí se utilizan algunos de los recursos anteriormente detectados como: dorks, vulnerabilidades de validación, entre otras.

A continuación, se describe la explotación a nivel de usuario los sitios auditados indicados en la tabla 2 punto 2.1.

Explotación al sistema de recuperación de contraseñas de la UPS

Como primera prueba se realiza al portal https://servicesapp.ups.edu.ec/accounts/ que es el portal de recuperación de contraseñas olvidadas, donde se identifica que el proceso para la recuperación es:

1. Ingresar el correo institucional.

2. Deslizar un elemento slider y pulsar en un botón para continuar.

3. El servidor devuelve el correo electrónico personal en un campo label y se envía un mensaje al correo personal.

Entonces para atacar este proceso se lo puede realizar de diversas formas, pero una muy utilizada para este tipo de ataques es crear un script a la medida, en donde a través de procedimientos programados se pueda obtener el campo del correo personal, este tipo de información es muy sensible para el usuario pues se pueden aplicar varios tipos de ataques después de la extracción del correo personal.

Para programar este tipo de scripts es fácil realizarlo con .NET donde a través de la librería “System.Windows.Forms.WebBrowser” se puede acceder como browser a la URL, entonces se toma la libreta de direcciones de correo que se obtuvieron con las pruebas realizadas anteriormente donde se pudo obtener correos de docentes y estudiantes, esta lista se carga al textbox de la página web llamado “email” como muestra en la figura 67.

Input de texto para recordar contraseña

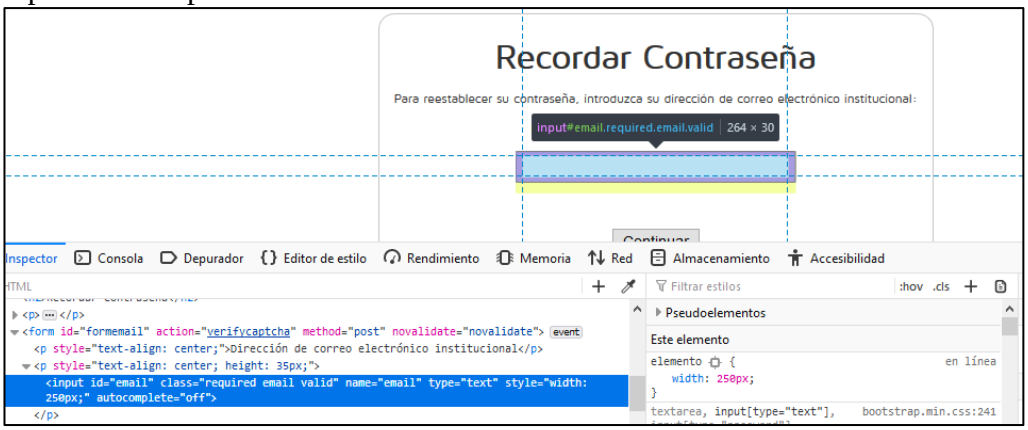


Figura 67 Caja de texto "email" del servicio de recuperación de contraseñas
Fuente: Recuperación de contraseñas UPS https://servicesapp.ups.edu.ec/accounts/

Igualmente se programa el ejecutar el botón “Continuar” para una vez lleno el campo de texto “email” se realice un clic y prosiga con el proceso

Una vez validado, el sistema continúa con el procedimiento por parte del cliente donde se debe deslizar un elemento slider hasta que se apruebe, pero esto se puede automatizar modificando directamente el elemento como muestra en la figura 68.

Slider modificado automaticamente

Figura 68 Estado inicial / final de Slider de protección
Fuente: Recuperación de contraseñas UPS https://servicesapp.ups.edu.ec/accounts/

Una vez cambiado este estado se activa el botón “Continuar” y el script ejecuta el evento de dar clic sobre él, posteriormente el servidor devuelve toda la información de recuperación en donde el script debe recuperar el label llamado “chkopcion2” que es aquel que tiene el correo personal como muestra la figura 69.

Label con correo personal vulnerado

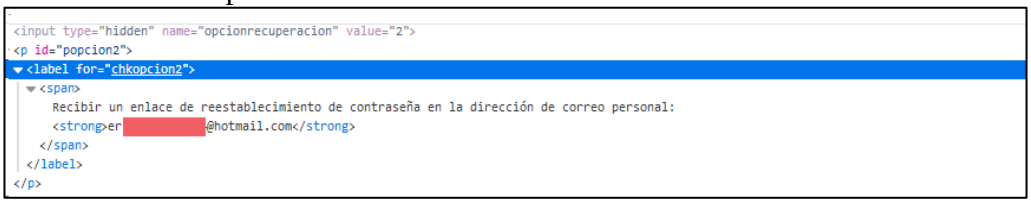


Figura 69 Email personal obtenido
Fuente: Recuperación de contraseñas UPS https://servicesapp.ups.edu.ec/accounts/

Para terminar este proceso se puede realizar de forma recursiva sin impedimentos y los resultados pueden ser exportados en archivos de texto, por motivos de seguridad no se publica código fuente del script ya que puede ser replicado y causar robo de información personal.

Explotación al portal de eventos de la UPS

```
https://appwfp.ups.edu.ec/foc-webins/index.xhtml?prog=#deEvento
```

La página normalmente se comporta de esta manera al momento de ingresar información errónea en la URL como indica en la figura 70.

Control de URL en app.wfp.ups.edu.ec

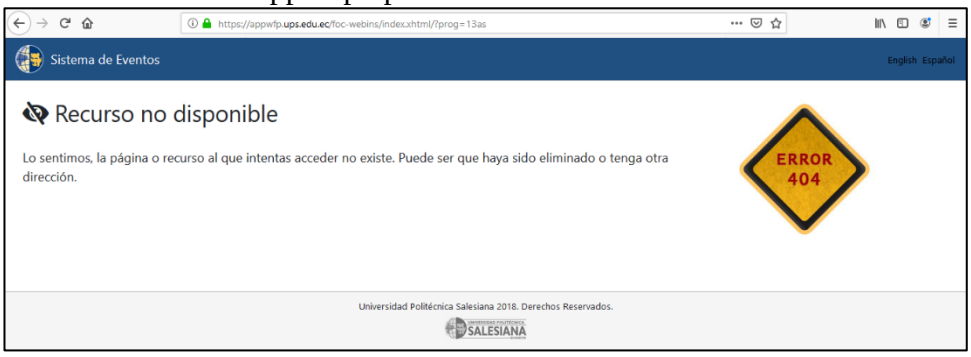


Figura 70 Error 404 en sistema de eventos
Fuente: Generado por el autor en sistema de eventos de la UPS

En el siguiente ataque XSS se muestra cómo se envía un mensaje de alerta ante el ingreso del script, pero esto puede causarse también a las validaciones de campos URL que indican que después de un igual lo demás es texto e invalidar el XSS creando un falso positivo.

Pero al identificar este sistema de prevención se puede generar un error de validación con el siguiente código `<script>alert("Owned")<script>` que se traduce en :<script>alert("Owned")</script> y muestra la figura 71 en su resultado.

XSS Exitoso

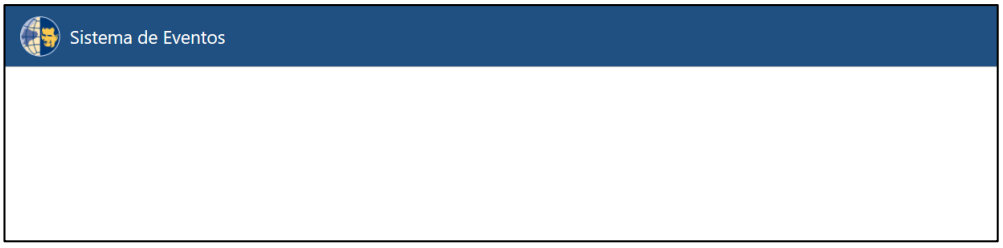


Figura 71 XSS exitoso en sistema de eventos
Fuente: Generado por el autor en sistema de eventos de la UPS

Como se puede observar en la figura 71 ya no muestra el mensaje, pero tampoco envía el error lo que se identifica con un Sender Request como un mensaje enviado, procesado y al no encontrar la página no lo muestra, pero si ejecuta dentro del servidor el código XSS, entonces esto indica que si es vulnerable a este tipo de ataques.

Esto se tiene que solucionar con un validador de caracteres HTML.

En este sitio la vulnerabilidad más grande que puede identificar es el de mostrar información personal sin validaciones de autenticación. Esto convierte al sitio en un foco de filtración de información confidencial.

Esta vulnerabilidad viene en conjunto con una carga de datos inservible que se guardan en el servidor y eso en masa puede llevar a una sobrecarga de información.

La URL en la que se va a probar este tipo de vulnerabilidades es:

```
https://appwfp.ups.edu.ec/foc-webins/index.xhtml?prog=140
```

Donde en primera instancia se muestra un formulario para llenar con datos personales y de esta manera registrarse para este taller como en la figura 72.

Taller del sistema de eventos

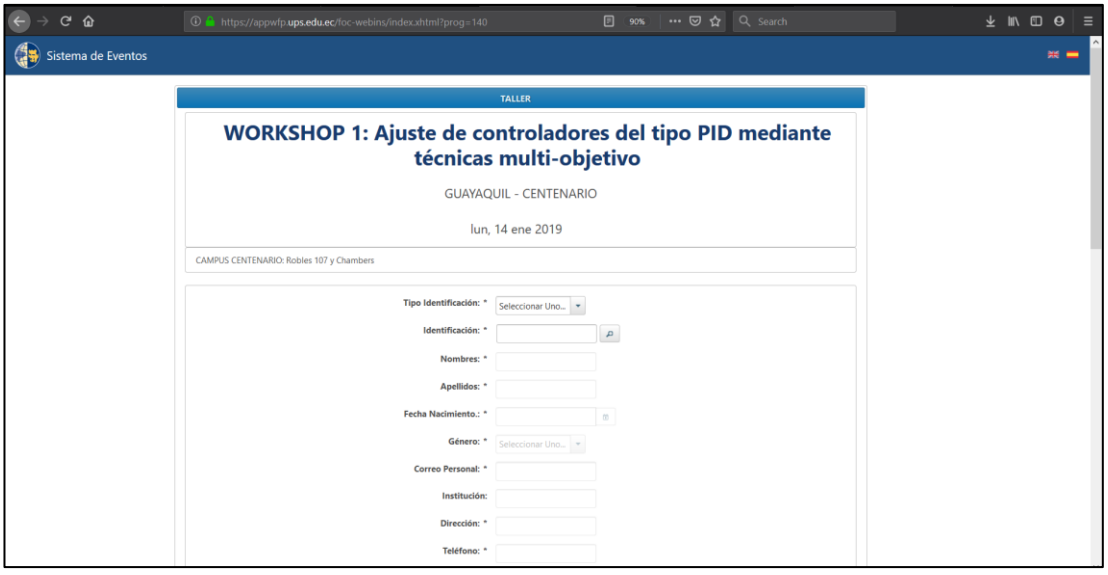


Figura 72 Evento vulnerable
Fuente: https://appwfp.ups.edu.ec/foc-webins/index.xhtml?prog=140

Entonces se procede a identificar que entre los componentes se encuentra un botón buscar en número de identificación el cual hace una comprobación a la base de datos y si encuentra un registro rellena los campos: nombres, apellidos, género, dirección y correo electrónico como se puede observar en la figura 73.

Datos obtenidos automaticamente desde el portal

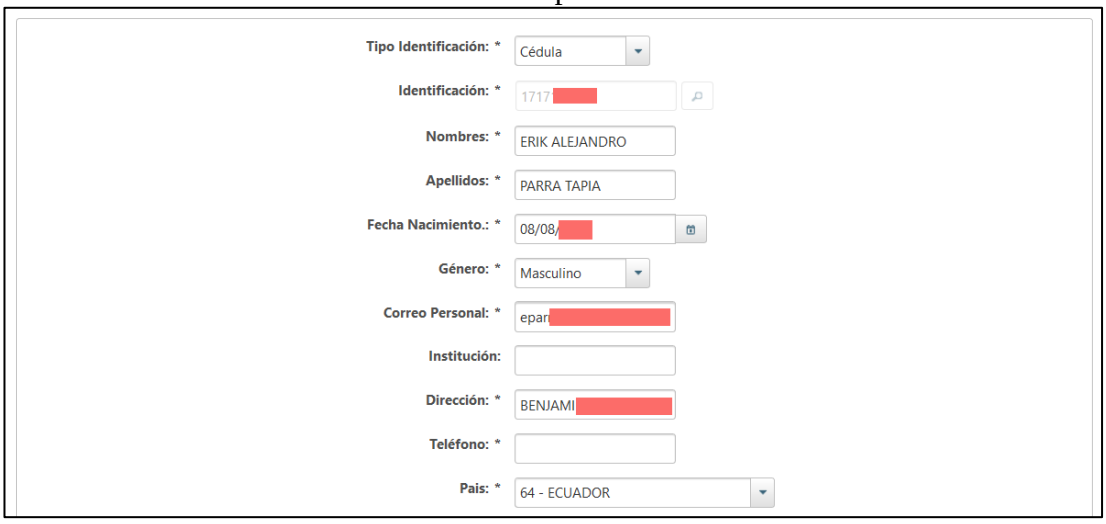


Figura 73 Datos automáticamente llenos
Fuente: https://appwfp.ups.edu.ec/foc-webins/index.xhtml?prog=140

Se pueden extraer datos medianamente sensibles, pero en otros eventos se rellenan datos como teléfono, nivel de estudios y correo personal que son mucho más sensibles.

Aparte de ello los campos de entradas no están validados correctamente por longitud y permite que todo el documento en total pese 1.28 Mb, lo que significa que en un ataque de DDoS por un script de repetición de envío del formulario en 1000 peticiones ya completará 1 Gb de datos subidos al servidor.

Pero este problema no solo es que sirve para los alumnos, al momento de probar con profesores también funciona.

Mucha de la información para este tipo de explotación se puede obtener de los Google dorks que se analizaron anteriormente entonces, se puede descargar toda la información recursivamente a través de los campos llenados automáticamente.

Explotación al portal de inscripciones en línea

En este portal se puede realizar inscripciones en carreras que ofrece la Universidad Politécnica Salesiana en las modalidades presencial y en línea.

El problema con este portal es que muestra información privada que puede ser tomada por terceros sin conocimiento ni consentimiento de la persona implicada e inclusive puede generar una inscripción en alguna carrera.

Para empezar el portal de inscripciones presenta un formulario en el cual se puede ingresar cédula o pasaporte como muestra en la figura 74, este filtro se puede saltar usando los datos obtenidos por los dorks de Google como en la figura 22 del punto 2.3.2.

Portal de inscripciones en línea

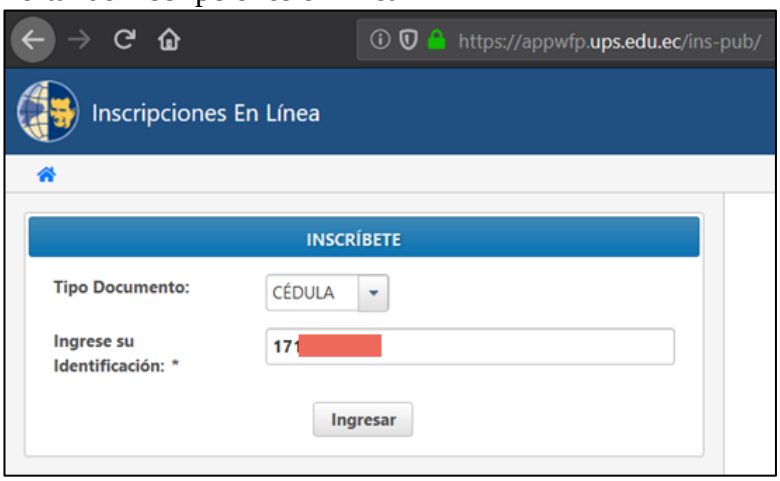


Figura 74 Ingreso del portal de inscripciones
Fuente: https://appwfp.ups.edu.ec/ins-pub/ Portal de inscripciones UPS

Una vez dentro del portal se presenta la información de la figura 75, que es sensible para todos los usuarios.

Información personal obtenida desde el portal de inscripciones

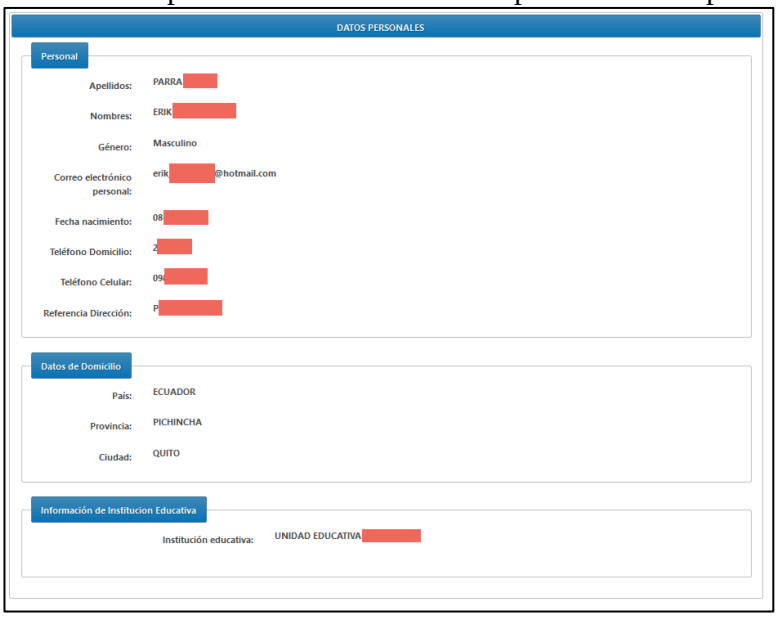


Figura 75 Datos personales en portal de inscripciones
Fuente: https://appwfp.ups.edu.ec/ins-pub/ Portal de inscripciones UPS

Se presentan en la figura 75 todos los datos personales del usuario el cual está expuesto a varios tipos de ataques como: Ingeniería social, extorción, estafa, entre otros.

El mayor error de este sistema es que perite al usuario inscribirse en una carrera sin presentar algún control como se observa en la figura 76.

Formularo para matricularse en línea

Figura 76 Formulario para inscripción directa en una carrera
Fuente: https://appwfp.ups.edu.ec/ins-pub/ Portal de inscripciones UPS

Esto expone al portal con posibles datos erróneos ingresados por el usuario o por terceros que deseen perjudicar al sistema, entonces posteriormente cuando los campos del formulario se llenan, el sistema genera una prefectura la cual necesita ser pagada.

Este tipo de ataques evidencia las brechas de seguridad que menciona la UPS en la política de tratamiento y uso de datos personales en donde en el apartado de “Deberes de la universidad como responsable del tratamiento de datos personales” numeral 4 y 5 presenta la figura 77.

Inciso para prohibir la consulta de información personal

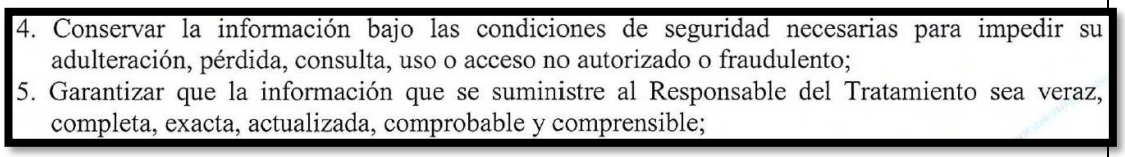


Figura 77 Política de tratamiento y uso de datos personales, y clasificación de la información
Fuente: https://www.ups.edu.ec/group/guest/politica-de-tratamiento-y-uso-de-datos

Entonces como se puede observar en la figura 77 se determina que las brechas de seguridad para exponer los datos de los usuarios en los sistemas web no están siendo eficientemente administrados para que respondan ante la política de tratamiento y uso de datos personales.

Capítulo 5

Reportes y plan de mitigación

Se muestran resultados cuantitativos y cualitativos para generar un plan de mitigación para reparar o minimizar el impacto en caso de sucederse un ataque en los sitios auditados.

5.1 Resultados de vulnerabilidades atacadas

El primer ataque se llevó a cabo fue la explotación TLS con la vulnerabilidad CVE-2014-0224 donde tras 3 horas de ataque se obtuvo un resultado exitoso en el ataque de hombre en el medio mencionado en la figura 55 con inyección CCS donde la herramienta utilizada para realizar el ataque fue Metasploit con el auxiliar "Heartbleed" con el que se pudo interceptar sesiones y filtrar todo el tráfico que se comunicaba a través del mismo.

En la búsqueda de vulnerabilidades con Shodan se encontraron 4 vulnerabilidades críticas las cuales 4 fueron explotadas y en 1 vulnerabilidad se tuvo éxito en su explotación esto se interpreta de la siguiente manera:

Calidad del ataque

$$Calidad = \left| \frac{Explotaciones\ exitosas - Explotaciones\ realisadas}{Número\ de\ vulnerabilidades\ explotadas} \right| * 100$$

$$Calidad = \left| \frac{1-4}{4} \right| * 100 => \frac{3}{4} * 100 => 60\%$$

Efectividad del ataque

$$Efectividad = \frac{Explotaciones\ exitosas}{Explotaciones\ totales} * 100$$

$$Efectividad = \frac{1}{4} * 100 => 25\%$$

Esto demuestra que Shodan tiene gran efectividad al momento de mostrar vulnerabilidades externas, a continuación, se detalla los resultados obtenidos.

De las vulnerabilidades CVE-2017-7679, CVE-2017-3167 y CVE-2013-2249 se encontró un exploit dentro de github el cual puede ser atacado por medio de la red interna de la organización.

Para la vulnerabilidad CVE-2011-3192 se logró encontrar un script donde se lleva a cabo un exitoso ataque DoS el cual puede ser replicado y generar un DDoS, en primera instancia tras observar el aumento de tiempo de respuesta del servidor se dejó de atacar pues el propósito del pentesting no es dar de baja a los servicios. Posteriormente tras analizar el ataque se evidencia que el servidor tiene un sistema de respuesta ante este tipo de ataques en donde genera un error 403 y bloquea temporalmente la IP del atacante protegiendo de cierta manera el servidor.

En este ataque al contar con datos lineales se puede graficar los paquetes enviados en función del tiempo y esto se grafica en la figura 78.

Dados los 4000 paquetes enviados en 2 minutos se generó una latencia de respuesta de 0.8 s

Tiempo de respuesta ante ataque DoS

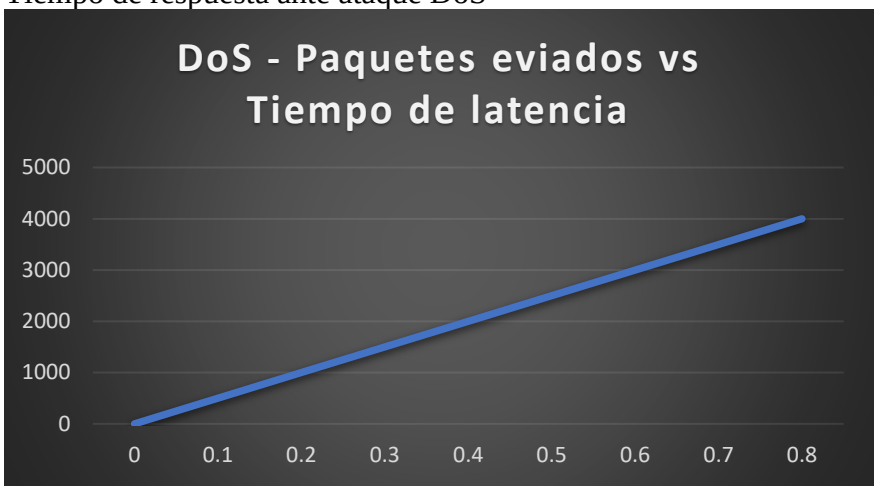


Figura 78 Ataque DoS
Fuente: Generado por el autor en base de datos obtenidos del ataque DoS

Entonces se calcula la ecuación de la recta para calcular la razón de aumento de latencia y poder de esta manera predecir una suspensión del servicio.

Pendiente de la recta

$$m = \frac{0.8}{4000} = 0.0002$$

Ecuación de nuestra recta:

$$y - y_1 = m(x - x_1)$$

$$y = 0.0002x$$

Esto quiere decir que en un caso ideal a los 100 minutos de ataque el servidor tendrá un retraso de 40 segundos y se recibirá 200000 paquetes DoS esto generaría muchas molestias para el usuario final.

En las vulnerabilidades CVE-2019-9641 y CVE-2019-9639 no se encontró un exploit ni script el cual pueda explotar externamente estas vulnerabilidades.

Tras la explotación de vulnerabilidades críticas encontradas por VEGA se determina que los ataques XSS e Interger Overflow son los que tuvieron real impacto en el sistema al momento de hacer el pentesting, por el contrario, los ataques de Shell Injection pueden tomar lugar en un ambiente interno y finalmente las vulnerabilidades SQL Injection que aparecieron en el scanner se consideran falsos positivos pues al hacer el reply request no se obtienen una respuesta positiva.

Al momento de testear las vulnerabilidades XSS se obtiene que, se puede ingresar peticiones a los portales https:/www.ups.edu.ec/web y https:/www.ups.edu.ec/evento? exitosamente.

Por igual el pentesting a una de las vulnerabilidades de interger overflow con una replicación de petición GET al portal /ups-portal-comprobantes-publico-portlet/ comprueba que se puede desbordar un buffer de una variable interna.

De las 3 vulnerabilidades “CVE-2003-1567, CVE-2004-2320 y CVE-2010-0386” críticas obtenidas por Nessus se comprobó a través de Curl que pueden ser explotados los procesos de Trace/Track para enviar métodos GET, POST, PUT, Head y DELETE.

Cabe recalcar que para el uso del método DELETE se tiene un control con usuario y contraseña del administrador para controlar una posible eliminación no deseada.

En la vulnerabilidad de Apache CVE-2016-3607, se pudo generar el ataque bajo el exploit “Apache < 2.0.64 / < 2.2.21 mod_setenvif - Integer Overflow” un ataque DoS combinado con Interger Overflow, pero tuvo una pequeña respuesta por parte del servidor rechazando y bloqueando temporalmente el ataque, pero no se controla las

cajas de texto que implican un ingreso de datos desmedido que puede llenar con datos inservibles los registros.

La vulnerabilidad CVE-2017-3250 se logró ejecutar con un exploit llamado “Oracle Fusion Middleware” el cual realiza ataques XSS dirigidos en donde al verificar su funcionamiento se tuvo que rectificar algunos parámetros de ingreso para que el ataque se muestre efectivo.

El pentesting a nivel de usuario fue una de las partes más importantes para el pentesting pues una vez explotados los portales a nivel de usuario se evidencian las fallas más importantes que estos sitios web poseen.

Para iniciar se analiza el sistema de recuperación de contraseñas en donde tras validar con datos de dorks, y generando un script propio en visual basic el cual llena automáticamente los datos del email institucional se puede llegar a obtener el correo personal del usuario de forma masiva o individual señalando como positiva la prueba de vulnerabilidad, generando el siguiente índice de efectividad.

Efectividad del ataque

$$Efectividad = \frac{Explotaciones\ exitosas}{Explotaciones\ totales} * 100$$

$$Efectividad = \frac{100}{100} * 100 => 100\%$$

Este nivel de efectividad es crítico al conocer que siempre se va a tener una respuesta por parte del servidor que comprometa la confidencialidad del usuario final.

Como siguiente target se tiene al sistema de eventos en donde se realiza en primera instancia un ataque XSS teniendo éxito tras utilizar etiquetas poco convencionales como: `&lt` para señalar menor que y de esta manera ingresar el script de prueba.

Prosiguiendo con el sistema de eventos se evidenció que algunos eventos poseen formularios que se llenan automáticamente con los datos de la base de datos de los usuarios de la UPS, en una de las pruebas de instrucción se detectó un evento en donde al llenar la cédula se autogeneraba los nombres, apellidos y la dirección. En otros eventos se generaba hasta los números de teléfonos lo que comprueba la facilidad con la que un agente externo puede tomar datos privados de un usuario.

Para finalizar al realizar las pruebas en el sistema de inscripciones se puede acceder a toda la información confidencial de los estudiantes como dirección, teléfono celular, teléfono fijo, formación académica, fecha de nacimiento, nombres completos y lo peor de todo es que puede generar una orden de pago por inscribirse en un curso o carrera el cual en un ataque puede ser llenado con datos erróneos los cuales pueden generar una saturación de datos falsos.

5.2 Propuesta de mitigaciones para las vulnerabilidades explotadas

La vulnerabilidad CVE-2014-0224 afecta a los sitios web www.ups.edu.ec y virtual.ups.edu.ec donde se plantea lo siguiente para mitigar esta vulnerabilidad.

Actualizar el OpenSSL, para este procedimiento se debe ejecutar como super usuario los siguientes comandos:

```
$ sudo openssl version -a
```

```
# yum -y install openssl
```

```
# openssl version -a

# shutdown -r now
```

De este modo prevenimos el ataque a la vulnerabilidad CVE-2014-0224, pero se necesita también gestionar con Comodo C.A para el óptimo desempeño de su cifrado TLS de la siguiente manera:

1. Ingresar a https://www.comodoca.com/ssl-certificate-comparison
2. Elegir un plan con Validación Extendida (EV)
3. Implementar la licencia comprada en todos los dominios y subdominios.

Una opción también es comprobar la configuración que ofrecen los browsers para hacer las conexiones más seguras un ejemplo es la siguiente página web: https://ssl-config.mozilla.org y para comprobar las seguridades existen diferentes páginas que ofrecen este servicio gratuitamente una de ellas es https://www.hardenize.com

Para las vulnerabilidades CVE-2017-7679, CVE-2017-3167 y CVE-2013-2249 encontradas en el sitio servicesapp.ups.edu.ec se necesita actualizar la versión de Apache, esto se realiza con los siguientes comandos como super usuario:

```
# httpd -v
# yum install -y epel-release
# yum -y install httpd
```

Una vez culminado este proceso se puede verificar la versión instalada y es necesario reiniciar el servicio de Apache con el siguiente comando:

```
# systemctl restart httpd
```

Para la vulnerabilidad CVE-2011-3192 que afecta al sitio servicesapp.ups.edu.ec se necesita usar herramientas externas como CloudFlare que permitan permitir una barrera de tiempo de conexión entre petición y petición al servidor y poder retener de esta manera los ataques.

La siguiente vulnerabilidad XSS encontrada en VEGA para el portal www.ups.edu.ec/web/guest tubo muchos problemas para realizarse pero se evidenció que cuentan con un control de XSS y ataques de Interger Overflow donde una variable llamada X-XSS-Protection filtra las peticiones por defecto en el navegador, esta configuración se la detecta en las configuraciones internas del servidor web y se las puede realizar como muestra la figura 79.

Filtro XSS

PHP

```
header("X-XSS-Protection: 1; mode=block");
```

Apache (.htaccess)

```
<IfModule mod_headers.c>
  Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

Figura 79 Habilitar filtro XSS
Fuente: https://developer.mozilla.org/es/docs/Web/HTTP/Headers/X-XSS-Protection

Para la vulnerabilidad de Interger Overflow del portal www.ups.edu.ec/ups-portal-comprobantes-publico-portlet se necesita controlar desbordamiento de buffer, entonces requiere limitar el lenguaje de programación a las excepciones propias como por ejemplo en C# utiliza “System.OverflowException” para controlar la memoria disponible del servidor y no sobrepasar el buffer.

Con las vulnerabilidades CVE-2003-1567, CVE-2004-2320 y CVE-2010-0386 que afectan a los sitios: servicesapp.ups.edu.ec y appwfp.ups.edu.ec se debe mitigar realizando estas simples acciones:

1. Abrir el archivo de configuración del servidor Apache “httpd.conf”, puede utilizar el siguiente comando:

```
# nano /etc/httpd/conf/httpd.conf
```

2. Escribir: `TraceEnable off`
3. Guardar los cambios y salir
4. Reiniciar el servicio de apache con el comando:

```
# systemctl restart httpd
```

Con estos simples pasos mitigamos estas vulnerabilidades de trace.

Para la vulnerabilidad CVE-2015-3237 que poseen los sitios: servicesapp.ups.edu.ec y appwfp.ups.edu.ec que no pudieron ser explotadas, pero se obtuvo una mitigación ante posibles futuros zero day exploits y se lo puede realizar con el documento base del repositorio de GitHub configurándolo en nuestro servidor web:

```
https://github.com/curl/curl-www/blob/master/CVE-2015-3237.patch
```

En la vulnerabilidad CVE-2017-3250 que influye en los sitios web: servicesapp.ups.edu.ec y appwfp.ups.edu.ec se evidenció que un exploit puede ejecutar ataques XSS y también es posible a nivel de usuario ingresar a subdirectorios del repositorio por ello se debe configurar el archivo htaccess.conf de Apache para controlar la prohibición de acceso y filtrar los datos de entrada ingresados por el usuario en la URL de los sitios.

Finalmente se debe mitigar las vulnerabilidades a nivel de usuario las cuales son sumamente críticas pues en todas ellas se evidencia la filtración de información, incumpliendo las políticas propuestas por la Universidad Politécnica Salesiana.

En primera instancia se mitiga el sistema de recuperación de contraseñas en el sitio https://servicesapp.ups.edu.ec/accounts/ donde tras analizar su funcionamiento y su explotación se determinan tres puntos clave para asegurar la confidencialidad de la información para el usuario final.

1. Prohibir recursividad de consultas.

   Con esto impide la petición masiva por parte de un script repetitivo.

2. Agregar un captcha para que el proceso no pueda ser automatizado.

   Es muy común el encontrar páginas web donde se verifique la intervención humana para ello existe recaptcha, donde a través de implementar el API de Google donde se genera una línea de código con la key generada del API de la siguiente manera:

   ```
   <div class="g-recaptcha" data-sitekey="6LcePAATsbV5WaW"></div>
   ```

3. No presentar la dirección de correo personal completa.

   Se necesita implementar una función que devuelva el correo acortado y oculto que reciba por ejemplo jose.perez@hotmail.com y devuelva jo*****@hotmail.com.

Esto se lo puede realizar con validaciones en Arrays donde el String ingresado se transforme en array y del nombre principal tome 2 caracteres y rellene con 5 asteriscos hasta que encuentre el símbolo “@”.

Para el portal https://appwfp.ups.edu.ec/foc-webins/index.xhtml perteneciente al sistema de eventos debe mitigar la vulnerabilidad XSS validando los caracteres especiales de HTML en el campo de número de evento de la URL.

Además, se necesita controlar la información personal y privada que está filtrando al rellenar automáticamente datos personales en los eventos específicamente en el registro a los eventos, entonces es necesario eliminar la opción de rellenar los datos personales automáticamente.

Para culminar tenemos al sistema de inscripciones en el sitio: https://appwfp.ups.edu.ec/ins-pub/ se necesita controlar con un login para el ingreso al sistema y un captcha con un formulario restrictivo de la carrera que esta incito en primera instancia. Con respecto a la información mostrada debe bastar con el nombre completo para conocer el usuario ingresado y no mostrar toda la información personal.

Conclusiones

- Tras analizar más de 100 vulnerabilidades se las categorizó en críticas, medias y bajas donde se tomó las vulnerabilidades críticas, que conforman un total de 23, donde tras su explotación se obtiene que 13 fueron exitosamente explotadas, 5 son vulnerabilidades que solo pueden ser explotadas internamente, 3 de ellas no se encontraron exploits con los que se pueda explotar dichas vulnerabilidades y finalmente 2 resultaron ser falsos positivos lo que demuestra en base a la efectividad de ataques que, el pentesting al encontrar 18 de 23 vulnerabilidades críticas tiene un 78.26% de efectividad.

- Se realizó pruebas de penetración con diferentes herramientas que permitieron comprobar las vulnerabilidades en los portales auditados, obtener falsos positivos y conocer a breves rasgos lo que podría causar un hacking en estos portales, pero no se comprometió la disponibilidad de los sistemas ni se realizaron acciones que podrían haber causado un daño permanente en la ejecución de cada servicio.

- Al realizar los ataques a nivel de usuario se evidenció notoriamente que en el 100% de sitios auditados tienen una falta de control web que ofrece datos confidenciales a terceros que pueden perjudicar en un ataque de ingeniería social o en otro acto fraudulento a los usuarios.

- Se estudió que el ataque más importante al que se encuentra expuesta la Universidad Politécnica Salesiana es en la filtración de datos personales y registro a cursos y carreras del sistema de inscripciones online, ya que perjudica principalmente la confidencialidad de la información.

- Se generó planes de mitigación acordes a las vulnerabilidades explotadas en donde se propone más 8 medidas y procedimientos que protejan la confidencialidad, integridad y disponibilidad de la información y servicios que los portales web ofrecen a los usuarios, así también proponen el control de seguridad informática de los mismos.

Recomendaciones

- Tras la explotación se obtuvo que la mayoría de estas vulnerabilidades se pueden parchar actualizando los servidores web, actualizando el sistema operativo y teniendo un mayor control ante recursividad de peticiones en los portales auditados por lo que es probable que tengan un efecto “bola de nieve” a lo largo del tiempo.

- Se recomienda implementar un equipo de respuesta ante emergencias informáticas para tener bajo control la infraestructura web que ofrece la Universidad Politécnica Salesiana y de esta manera actuar eficazmente ante riesgos informáticos minimizando el impacto de ataque.

- Se recomienda seguir las mitigaciones de este proyecto para mejorar las brechas de seguridad encontradas, ya que de esta manera se fortalecerá la seguridad inmediatamente y se controlará futuros ataques que puedan concatenar las vulnerabilidades encontradas generando un ahorro económico por pérdidas de ataques maliciosos.

- Trabajar en la implementación semestral del pentesting como política interna para mejorar la calidad, confidencialidad, seguridad, integridad y disponibilidad del servicio web e infraestructura

Lista de referencias

Álvarez Marañón, G., & Pérez García, P. P. (2004). *Introducción a la seguridad de la información.* Madrid: McGraw-Hill.

Álvarez, M., & Pérez, G. (2004). *Seguridad informática para empresas y particulares.* Retrieved from https://bibliotecas.ups.edu.ec:2708.

Apache. (1 de 02 de 2019). *Ecured*. Obtenido de https://www.ecured.cu/Servidor_Web#Protocolos_del_Servidor_Web

Benitez, C. (22 de 11 de 2016). *Chriss Benitez.* Obtenido de ¿Qué son los Hackers Black Hat y Hackers White Hat?: http://chrissbenitez.com/que-son-los-hackers-black-hat-y-hackers-white-hat/

Coloma Baños, N. C. (2019). La seguridad informática para la toma de decisiones en el distrito de educación 12d03 Mocache-Quevedo.

Gómez Montoya, C. E. (2013). Seguridad en la configuración del servidor web apache. *REDICUC*, 1.

Gómez, Á. (2010). *Seguridad Informática.* Madrid: STARBOOK EDITORIAL.

Gonzáles Pérez, P., Sánchez Garcés, G., & Soriano de la Cámara, J. M. (2015). *Pentesting con Kali 2.0.* Madrid: 0xWord.

Harris, S., Harper, A., Ness, J., Eagle, C., Lenkey, G., & Williams, T. (2011). *Gray Hat Hacking: the ethical hackers handbook.* New York: Mcgraw-Hill.

ISO. (2013). *ISO 27000.* Obtenido de http://www.iso27000.es/sgsi.html

Magenit. (2006). Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Secretaría del Consejo Superior de Administración Electrónica. En S. Group. Madrid.

OWASP. (1 de 10 de 2016). *SQL Injection* . Obtenido de OWASP: https://www.owasp.org/index.php/SQL_Injection

Pérez, P. G. (2014). *Ethical Hacking Teoría y práctica para la realización de un pentesting.* Madrid: 0xWORLD.

PINOS SOLANO , D. O. (2018). Análisis de vulnerabilidades y acciones correctivas sobre un sistema web. *dspace.espol.edu.ec/handle/123456789/43624*, 1-37.

Pulgarin, C. L. (19 de Octubre de 2017). *LinkedIn.* Obtenido de https://es.slideshare.net/calube55/construccion-de-instrumentos-para-recoleccion-de-informacion-en-investigacion

RFC3912. (Septiembre de 2004). *WHOIS Protocol Specification.* Obtenido de Request For Comments: https://tools.ietf.org/html/rfc3912

Serrano, A., García , L., León , I., Gil , B., & Ríos, L. (2010). *MÉTODOS DE INVESTIGACIÓN DE ENFOQUE EXPERIMENTA.*

Solarte, F. N. (2015). Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001. *Revista Tecnológica ESPOL- RTE*.

Vieites, Á. G. (s.f.). TIPOS DE ATAQUES E INTRUSOS EN LAS REDES INFORMÁTICAS. *Edisa*, 1-5.

Yanez, D. (11 de 01 de 2019). *lifeder*. Obtenido de Método descriptivo: características, etapas y ejemplos: https://www.lifeder.com/metodo-descriptivo/

ANEXOS

ANEXO 1 Página por default Apache en CentOS

ANEXO 2 Vulnerabilidades encontrada en el portal de servicios de la UPS.

ANEXO 3 Vulnerabilidades encontradas por OpenVas

ANEXO 4 Request Get de Vega para verificación de vulnerabilidades.

Los anexos pueden ser descargados del siguiente enlace:

https://1drv.ms/w/s!AmwF6lhrPILbjxBsF5mkQ3h48g9S?e=dggzpE